Shock Analysis Toolkit (IPSKIT)

Documentation

Erkka Lumme, Timo Mäkelä

August 30, 2024

1 Introduction

The Shock Analysis Toolkit (IPSKIT) enables the user to analyse fast interplanetary shock waves detected in spacecraft data. The toolkit provides all the necessary automatised parts for the analysis. It produces an output data file as well as PS and PNG plots for the analysed shocks. These outputs can be uploaded to the Database of Heliospheric Shock Waves (`https://ipshocks.helsinki.fi`). The basics of the analysis methods are described at `https://ipshocks.helsinki.fi/documentation`.

This toolkit was originally developed by Erkka Lumme in 2017 and was written in IDL. In 2024 it was translated to Python by Timo Mäkelä. During the translation, the Parker Solar Probe (PSP) and Solar Orbiter (SolO) spacecraft were added, and the code was simplified to better suit the current workflow where shocks are found using the automated machine learning code IPSVM (`https://bitbucket.org/raysofspace/ipsvm`).

2 Requirements

All required dependencies are listed in `requirements.txt`.

2.1 Downloading the software

The package is available at `https://github.com/matimove/IPSKIT.git`.

2.2 Shock analysis

The shock analysis program is contained in `shock_analysis.py`.

2.2.1 Input

The file `shocks.dat` contains all the necessary input data for the analysis. The file contains the times of the shocks (see example file in the package directory). The times of the shocks are given to the precision of seconds. The program reads the times from this file, downloads the data around the given time, and completes the analysis.

In the options of the input file `shocks.dat` (see example file) the user should define the following:

1. **The spacecraft that the data is downloaded from.** Spacecraft IDs: 0 = ACE, 1 = Wind, 2 = STEREO-A, 3 = STEREO-B, 4 = Helios-A, 5 = Helios-B, 6 = Ulysses, 7 = Cluster 3, 8 = Cluster 1, 9 = Cluster 4, 10 = OMNI, 11 = Voyager 1, 12 = Voyager 2, 13 = DSCOVR, 14 = PSP, 15 = SolO.

2. **Whether the plot window is shown.** Recommended to be turned off for the first run as many events given by the automatic shock detection code are not shocks. Should be turned back on for subsequent runs.

3. **The use of a data spike filter.**

   - 0 = No filtering is applied to the input data.
   - 1 = Data spikes are filtered using default median filter. For each data point the median filtered value is calculated (the default filter length is 5). When calculating the medians, missing points are substituted by their respective median values. If the actual value P and the median filtered value $\langle P \rangle$ fill the condition $|P - \langle P \rangle| > \mathrm{tol} * \langle P \rangle$, where tol is the filter length, the data point is classified as a spike and its value is set to NaN. The variable tol depends on the input filter parameter given in `shocks.dat`. The default values defined in `shocks.dat` are $[\mathrm{tol}, N_\mathrm{p}, T_\mathrm{p}, V] = [5, 0.75, 1.5, 0.2]$. If the velocity magnitude, V, has a spike, the corresponding velocity vector components, V_x, V_y, and V_z, are also set to NaN.
   - 4-element float array = User-defined filter. If the user gives a 4-element array of values (e.g., $[7, 0.5, 1.0, 0.1]$), the data is filtered in the same way as for the default filter case, but instead of the default values, the specific tolerances defined by the user for the different time series as well as the size of the median window are used.

   Notes about using the filter

   - **The filter is a sensitive tool and not applicable for all spacecraft data.** Magnetic field data is not filtered at all since it only has a few spikes, and its good resolution mitigates the effect of individual spikes.
   - **The filter is too sensitive for ACE and Ulysses plasma data** and should therefore be used only in cases where there is a clear spike that affects the analysis. The user may also have to modify the filter settings to get rid of the spike without deleting actual data points.
   - **OMNI data has already been despiked by the team.** However, if some spikes have passed the filter, the user should try applying the filter carefully, first using the default settings. Then, if the default settings do not work, the user should try modifying the parameters.

- **For Helios, STEREO, and Wind spacecraft, the filter has been tested and can be used for all events using the default settings.** Still, there may be cases for which the user has to modify the filter parameters to get rid of some spikes.
- **The filter has not been tested in any way for Voyager or DSCOVR data.** It is recommended that if a spike is detected, the user should test the default filter first and modify its parameters if necessary.
- **For SolO and PSP, the filter testing is still ongoing.** SolO seems to despike well using the filter parameters $[5, 0.75, 0.1, 0.2]$, but some PSP events seem to have so many spikes that it is difficult to get rid of all of them without losing a lot of actual data in the process.

2.2.2 Output

The analysis program produces two output files as well as plots of the shocks. The outputs are found in:

```
clear_shock_parameters.csv
shock_parameters.dat
shock_out.dat
clear_shock_plots/
```

The file `clear_shock_parameters.csv` includes the results of the analysis in the same format and order that are used in the IPShocks database. The file `shock_parameters.dat` has better formatting, if the user wants to take a look at the results of the analysis.

The file `shock_out.dat` has the same formatting as the input file. If the shock times in the input file are definitive, this file is a perfect copy of the input file. If the the shock times in the input file are only preliminary times of 30-second resolution, this file contains the new, more accurate shock times determined automatically by the program. This file can be used to replace the input file for the later analysis runs.

The directory `clear_shock_plots/` contains the PS and PNG plots produced by the analysis program. The files are named using the convention `yyyymmdd_HHMMSS_spacecraft` (e.g., `20041205_121212_ACE(.ps/.png)`). With the correct shock times and only the real shocks included, these are finished products ready to be uploaded to the database.

3 Using the toolkit for the shock database

With the help of this toolkit the user can perform the necessary analysis needed for the shock database. The following outlines a suggested routine to go through.

3.1 Identifying shock candidates in spacecraft data

There are two ways to do this. Originally the shock candidates were searched for visually, but nowadays this can be done using the automated machine learning algorithm IPSVM, which identifies potential shock candidates.

3.1.1 Visual search of shock candidates

Use CDAWeb (https://cdaweb.gsfc.nasa.gov) plots or other sources to plot the spacecraft data and search for shock candidates following the principles presented in Sections 1 and 2 of the IPShocks documentation (http://ipshocks.helsinki.fi/documentation). Write down the times of the discontinuities which fulfill the characteristics of fast forward (FF) or fast reverse (FR) shocks. Use the discontinuity or the rapid increase of the magnetic field to set the preliminary time of the shock.

3.1.2 Automated search of shock candidates

The operation of the IPSVM algorithm is not described here. The algorithm outputs the shock candidate times in the format `yyyy-mm-dd HH:MM:SS` (e.g., `2018-02-26 20:55:28`). The `shock_analysis.py` program reads the shock times from `shocks.dat` in the format `yyyy mm dd HH MM SS` (e.g., `2018 02 26 20 55 28`), i.e., a conversion is needed. To do the conversion for a large number of shock candidate times, use the tools found in the `/shock_time_formatting` folder:

1. Copy the output datetimes from IPSVM to `unformatted_shock_times.txt`.
2. Run the `shock_time_formatter.py` script.
3. Now formatted shock times should appear in the `formatted_shock_times.txt` file, ready to be copied to `shocks.dat` and analysed by the `shock_analysis.py` program. You should take a look to see if you find any duplicate times or times that are clearly from the same event (within a minute of each other) and remove them.

3.2 Running the preliminary set of analysis

The program is designed to be ran multiple times in an iterative way to improve shock times, drop bad shocks, etc. Many of the candidate times given by the detection algorithm are false positives (not a shock), hence it is not necessary to plot and look at them during the first run. For efficiency set the "Show plots for events" parameter from the `shock.dat` input file to 0 for the first run. This way the program will not stop and plot every event, needing the user to look at and close the plots. This allows for an easy preliminary run in the background.

3.2.1 Checking the new times

After the preliminary run is complete, you will have event times in `shocks_out.dat` that were classified as either an FF or FR shock, and events which did not pass the requirements were removed. Copy and paste these output times back to the input file `shocks.dat`. Set the "Show plots for events" parameter to 1 and run the program again.

3.2.2 Adjusting shock times

You can adjust the shock times using the "left" and "right" arrows to move the shock time (red line) left and right, respectively (one click moves the shock time one second). To save the new shock time press the "down" arrow. Now the new adjusted time will be saved as the output to the `shocks_output.dat` file. If no new time is saved or no adjustment is needed, you can close the plotting window and the old time will be kept as the output. If the shock time is changed, the program will need to be ran again because the downstream and upstream have shifted and the corresponding parameter values will need to be calculated again.

3.2.3 Checking for other issues

The shock analysis program is not foolproof. It may produce a positive result for an event which is not actually a shock but just happens to fulfill the shock criteria. When checking the times, exclude these kind of erroneous events. Also, data gaps or other issues may result in bad output values (NaNs in `shock_parameters.dat`). Check what problem is causing the bad values and exclude the event if the issue cannot be fixed. If there is no data for some parameter in the upstream or downstream analysis intervals, the analysis cannot be executed. For STEREO-B the velocity vector components are an exception, but for missing component data there has to be enough bulk speed data. Finally, bad data spikes which affect the results should be checked for and filtered.

3.3 Running the final set of analysis

Using the new accurate times which you have checked and corrected, you may run the final set of analysis. In this analysis run you can also classify which shocks are clear and which are unclear. By clicking the "up" arrow while a plot is displayed, the event will be marked as unclear and the parameters will be saved to the `unclear_shock_parameters.csv` file instead of the default `clear_shock_parameters.csv`. This can be a useful tool to classify questionable shocks from good ones but not necessary to use.

3.3.1 Sending the final results forward

Now the `clear_shock_parameters.csv` file is ready to be uploaded to the database along with the PS and PNG plots of the shocks.