Shock analysis toolkit
Documentation

Erkka Lumme, Timo Mäkelä

July 31, 2024

1 Introduction

The Shock Analysis Toolkit (IPSKIT) enables user to analyse fast interplanetary shocks detected in spacecraft data. The toolkit provides all the necessary automatised parts for the analysis. It produces an output data file and .ps- and .png-plots for the analysed shocks. This output can be uploaded to Database of Heliospheric Shock Waves (https://ipshocks.helsinki.fi). Basics of the analysis method are described at (https://ipshocks.helsinki.fi/documentation).

This toolkit was originally developed by Erkka Lumme in 2017 and was written in IDL. Later in 2024 it was translated to Python by Timo Mäkelä. During the translation PSP and 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.

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 shock_analysis.py.

2.2.1 Input

shocks.dat contains all the necessary input data for the analysis. The file contains the times of the shocks (see example file in requirements package). The times of the shocks are given in the precision of seconds. 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 defines 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. If 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. Turn it back for subsequent runs.

3. The use of 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 median filtered value is calculated (default filter length is 5). When calculating the medians, missing points are substituted by their respective median values. If actual value P and median filtered values $\langle P \rangle$ fill the condition $|P - \langle P \rangle| > tol * \langle P \rangle$, data point is classified as a spike and its value is set to NaN. Variable *tol* depends on the given input filter parameter in shocks.dat. Default values for it are: [Filter length, Np , Tp , V] = [5, 0.75, 1.5, 0.2]. If V (velocity magnitude) has a spike, corresponding velocity vector components Vx, Vy, Vz are also set to NaN.
   - 4-element float array = user-defined filter. If user gives 4-element array of values (e.g. [7, 0.5, 1.0, 0.1]) data is filtered in the same way as the default filter case but the user can now specify specific tolerances for different data series and adjust the size of the median window.

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 has only a few spikes, and its good resolution mitigates the effect of individual spikes.
- **For ACE and Ulysses plasma data, the filter is too sensitive** and should 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 proper data points.
- **OMNI data has already been despiked by the team.** However, if some spikes have passed the filter, the user should carefully try applying the filter, first using default settings. Then, if the default settings do not work, try to modify the settings.

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

2.2.2 Output

Analysis program produces two output files as well as plots of the shocks. Output is found in:

clear_shock_parameters.csv

shock_parameters.dat

shock_out.dat

clear_shock_plots/

clear_shock_parameters.csv includes the results of the analysis in the same format and order that is used in the shock database website. shock_parameters.dat is nicer to look at when taking a look at the results of the analysis.

shock_out.dat has the same format 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.

Directory *clear_shock_plots/* contains the .ps- and png-plots produced by the analysis program. The files are named with convention: ”yyyymmdd_hourminsec_spacecraft” (e.g. ”20041205_121212_ACE(.ps /.png)”). With 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 user can do the necessary analysis needed for the shock database. Here it is outlined a suggested routine to follow:

3.1 Identify shock candidates in the spacecraft data

There are two ways to do this. Originally the shock candidates were searched visually, but now also an automated machine learning algorithm (IPSVM) for searching potential shocks exists.

3.1.1 Visual search of shock candidates

Use CDAWeb plots or other sources to plot the spacecraft data and search for shock candidates following the principles in (http://ipshocks.helsinki.fi/documentation) sections 1 and 2. Write down the times of the discontinuities which fulfil the characteristics of fast forward (FF) or fast reverse shocks (FR). 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 the shock candidates

The operation of the IPSVM algorithm is detailed in its own documentation. The automated shock detection algorithm outputs shock candidate times in the format (YYYY-MM-DD HH:MM:SS) (eg. 2018-02-26 20:55:28). The shock_analysis.py reads the shock times from shocks.dat in the format (YYYY MM DD HH MM SS) (eg. 2018 02 26 20 55 28) hence 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 shocktimes should appear in the formatted_shock_times.txt, ready to be copied to shocks.dat and analyzed by the shock_analysis.py program. You can take a look if you see any duplicate times or times that are clearly from the same event (within a minute of each other) and remove them already.

3.2 Run 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 ” from the shock.dat input file to 0 for the first run. This way the program wont 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 Check the new times

Now after the preliminary run is complete, you will have event times in shocks_out.dat that were classified as either FF or FR shock and events which didnt pass the requirements were removed. Copy and paste these output times back to the input file shocks.dat. Turn on the plotting this time and run the program again.

3.2.2 Adjusting shocktimes

The user can adjust the shock times using the ”left” and ”right” arrows to move the shock time (red line) left and right. (one click is 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 / no adjustment is needed, you can close the plotting window and the old time will be output. If the shock time is changed, the program will need to be ran again because the downstream and upstream have shifted and corresponding parameter values will need to be calculated again.

3.2.3 Check for other issues

Shock analysis program is not foolproof. It may produce positive result for an event which is not 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 is the problem causing the bad values and exclude the event if the issue cannot be fixed. (If there is no data for some parameter in the up- stream or downstream analysis intervals, the analysis cannot be executed. For STEREO B spacecraft 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 and filtered (see 3.1.1, part 5).

3.3 Run 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 you can also classify which shocks are clear and 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 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 Send the final results forward

Now the clear_shock_parameters.csv is ready to be uploaded to the database along with the png. and ps. plots