STRESSINVERSE

User guide

Version 1.1.3, last update 23.11.2020

Author: Václav Vavryčuk (vv@ig.cas.cz)

Short description

STRESSINVERSE is a Matlab software package for an iterative joint inversion for stress and fault orientations from focal mechanisms. The code needs just the Matlab kernel. No toolboxes are required. The inversion is based on Michael's method (1984, 1987) in which an instability criterion proposed by Lund & Slunga (1999) is incorporated.

Details about the method and its accuracy are published in Vavryčuk (2014).

Primary reference

Vavryčuk, V., 2014. Iterative joint inversion for stress and fault orientations from focal mechanisms, *Geophysical Journal International*, **199**, 69-77, doi: 10.1093/gji/ggu224.

Other references

Lund, B., Slunga, R., 1999. Stress tensor inversion using detailed microearthquake information and stability constraints: Application to Olfus in southwest Iceland, *J. Geophys. Res.* **104**, 14.947-14.964.

Michael, A.J., 1984. Determination of stress from slip data: Faults and folds, *J. Geophys. Res.* **89**, 11.517-11.526.

Michael, A.J., 1987. Use of focal mechanisms to determine stress: A control study, *J. Geophys. Res.* **92**, 357-368.

Run the code

The software is structured in the following directories: **Data, Documentation, Figures**, **Output**, **Programs**.

The stress inversion is run by a script called **StressInverse.m** stored in the Programs directory.

Control parameters

Parameters controlling the inversion are defined in the script called **input_parameters.m** (Data). The main control parameters are:

- input_file (the name of the input data),
- output_file (the name of the file with the results),
- principal_mechanisms_file (the name of the file with calculated principal focal mechanisms)
- N_noise_realizations (number of realizations of random noise for accuracy estimates),
- mean_deviation (estimate of accuracy of input focal mechanisms in degrees).

Input data

Input focal mechanisms should be stored in the Data directory. The conjugate solutions can be mixed randomly. The strike and dip angles need not describe the orientation of true fault planes.

The name of the input file is defined in ‘input_parameters.m’.

The input data are formed by three columns:

- strike (1st column), dip (2nd column), rake (3rd column).

Output

Two output files are stored in the Output directory. They have an extension '.mat' and '.dat'. The **'.mat'** output file includes:

- sigma_1, sigma_2 and sigma_3 (azimuth and plunge of principal stress directions),
- shape ratio,
- mechanisms (strike, dip and rake of focal mechanisms with a correctly selected fault plane),
- principal_mechanisms (strike, dip and rake of two principal focal mechanisms found for the retrieved optimum stress tensor),
- friction (mean friction on the faults).

The **'.dat'** output file is the ASCII file and contains three columns with:

- strike, dip, rake

These angles describe the focal mechanisms with fault orientations determined by the code.

Additionally, an ASCII file with two **principal focal mechanisms** calculated for the found optimum stress tensor is created. The file contains two rows (each row for one principal focal mechanism) with the following angles:

- strike, dip, rake

Figures

Figures are stored in the Figure directory. The code generates the following figures:

- P_T_axes.png (focal sphere with the P and T axes and with the optimum principal stress axes),
- Mohr_circles.png (Mohr's circles with fault planes),
- stress_directions.png (confidence limits of the principal stress axes),
- shape_ratio.png (histogram of the shape ratio).

Example

As an example, we provide the input file called '**West_Bohemia_mechanisms.dat**' with focal mechanisms from the West Bohemia swarm area (for the analysis of these data, see Vavryčuk 2014). The output files are: '**West_Bohemia_output.ma**t', '**West_Bohemia_output.dat**' and **‘'West_Bohemia_principal_mechanisms.dat’**.

To run the inversion just call StressInverse.m.

Python code

The Python codes are saved with DOS newlines (CRLF), you can use _dos2linux.sh code to convert them to have Unix newlines (LF).

You need to have installed these python libraries: matplotlib, numpy, scipy, os

Codes were tested on both Windows and Linux systems, Python 2.7 and 3.5, Python and IPython shells.

Copyright