

EPOS GNSS DATA AND PRODUCTS

Date	8 July 2019
Document title:	Report of NOA(NTUA) for the EPOS STRAINWEBTOOL
WP name and no	WP10 TCS GNSS Data and Products
Lead Partner	NOA
Main Author(s)	Dr. Dimitrios Anastasiou
Contributing author(s)	Mr. Xanthos Papanikolaou
Reviewer(s)	Dr Athanassios Ganas
Approved by	
Target audiences	EPOS WP10
Keywords	Strain rates, web application, open-access, GitHub, StrainTool
Deliverable nature	Report and software
Dissemination level	
Delivery date	1 July 2019
Version/Date	beta / 08-07-2019

Contents

1. Product Summary	3
2. General.....	3
3. Structure and background	3
4. Installation	4
5. User Guidelines	5
5.1 Input Files	5
5.2 Options and parameters.....	6
5.3 Output Files	8
6. Active Map visualize strain tensor results	9
7. Final note	9

1. Product Summary

Name of Product / Service: *StrainWebTool* beta for EPOS

This report describes the web application named *StrainWebTool*, that estimates strain tensor parameters using the *StrainTool* Software (Anastasiou et al., 2019a, 2019b).

This work is part of NOA contribution to EPOS-WP10.

The web address of the product / service is <http://83.212.103.160/StrainWebTool/inputs> (please note this is a temporary address).

2. General

StrainWebTool is a web application developed to estimate strain tensor parameters using StrainTool Software. The development of the application was based on Flask microframework for Python (<http://flask.pocoo.org/>). The Bootstrap open source toolkit (<https://getbootstrap.com/>) was used to enable a responsive web design and Leaflet open-source JavaScript library (<https://leafletjs.com/>) for producing interactive maps.

3. Structure and background

The application consists of three basic parts:

1. **webtool.py**: the main python source code file, that includes all necessary functions utilizing StrainTool software and enables the building of HTML templates.
2. **static files**: a folder including static files such as headers, footers, images, javascript source code files.
3. **templates**: a folder including the main templates for the application.

Three different HTML templates have been formulated to implement the application:

- [tmpl_inputs.html](#) is the template where the user uploads their input files.
- [tmpl_params.html](#) is the template where the user chooses the parameters that StrainTool will later on use to estimate strain tensor parameters
- [tmpl_results.html](#) is the template where the results are presented. The user can download result files and see the results visualized on an interactive map.

Each template consists of three basic columns.

The first column contains all input forms for the parameters needed to estimate strain tensors.

The second column contains the plot tools and options to generate GMT maps; these however are not active in the current beta version.

The third column is where the interactive map is placed and the results are visualized (Figure 1).

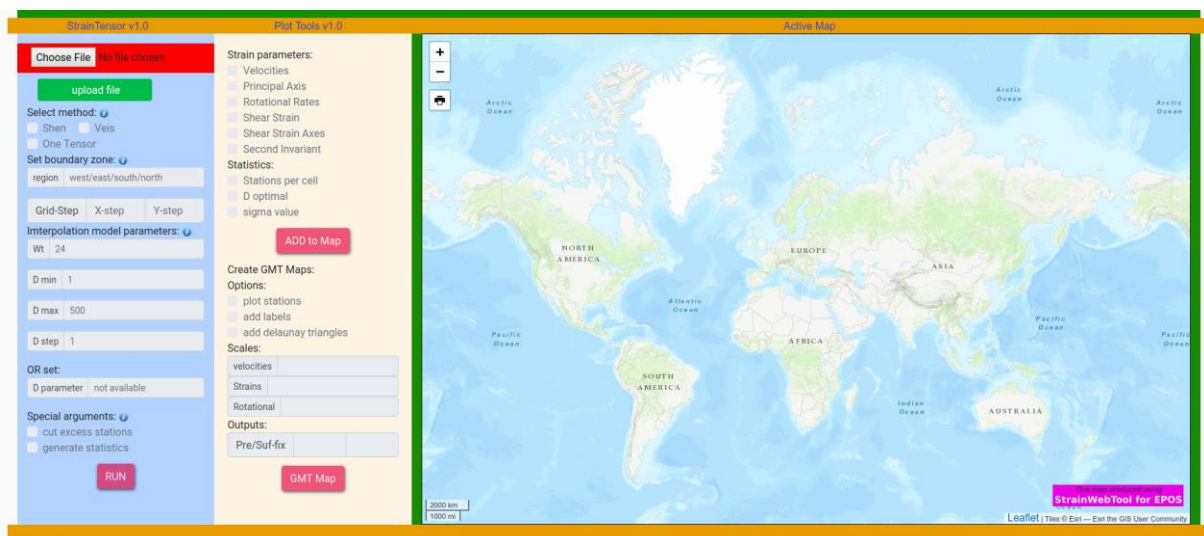


Figure 1: The main page of the application (accessed on 8 July 2019)

4. Installation

For the installation of *StrainWebTool* on a local Apache server, users can follow the next steps.

- Install StrainTool Software with all prerequisites needed
- Install Flask web developer (via pip, e.g. pip install Flask)
- Install virtual-environment package. (on Ubuntu-based OS, e.g. apt-get install python-virtualenv)

Run the application:

- on the root folder create virtual environment. (`$> virtualenv flask`)
- activate flask virtual-environment. (`$> flask/bin/activate`)
- set your application variable (`$> export FLASK_APP=webtool.py`)
- run your server: (`$> flask run`)
- The application will run on your local server; you can browse it at: <http://127.0.0.1:5000/inputs>

5. User Guidelines

5.1 Input Files

To perform the computations, *StrainWebTool* needs an input file, that holds input data (Figure 2). Usually, this implies a list of GPS/GNSS stations with their ellipsoidal coordinates (aka longitude and latitude) and their respective tectonic velocities (usually estimated using position time-series) along with the corresponding standard deviation values.

The format of these files, should follow the convention:

```
station-name longitude latitude Ve Vn SigmaVe SigmaVn Sne time-span
string deg. deg. mm/yr mm/yr mm/yr mm/yr / dec. years
```

Station coordinates are provided in longitude/latitude pairs in decimal degrees.

Velocities and velocity standard deviations are provided in mm per years (mm/yr).

Sne is the correlation coefficient between East and North velocity components (Ve, Vn) and time-span is the total time span of the station timeseries in decimal degrees.

Note that at this point the last two columns (aka Sne and time-span) are not used, so they could have random values.

There are no strict formatting rules on how the individual elements should be printed (i.e. how many fields, decimal places, etc.). The only condition is that fields are separated by whitespace(s).

Note that the input file format is identical to what is used in *StrainTool* Software (Anastasiou et al., 2019b);

Users can browse its dedicated web page (<https://dsolab.github.io/StrainTool/>) for a more detailed description.

The screenshot shows the 'StrainTensor v1.0' web interface. At the top, there is a 'Choose File' button with the text 'No file chosen' next to it. Below this is a green 'upload file' button. The 'Select method:' section has three radio buttons: 'Shen', 'Veis', and 'One Tensor'. The 'Set boundary zone:' section has a 'region' label and a text input field containing 'west/east/south/north'. Below this are three buttons: 'Grid-Step', 'X-step', and 'Y-step'. The 'Interpolation model parameters:' section has four input fields: 'Wt' with the value '24', 'D min' with '1', 'D max' with '500', and 'D step' with '1'. The 'OR set:' section has a 'D parameter' label and a text input field containing 'not available'. The 'Special arguments:' section has two checkboxes: 'cut excess stations' and 'generate statistics'. At the bottom of the form is a red 'RUN' button.

Figure 2. Inputs required by the user

5.2 Options and parameters

After the uploading of the input-file, all the options for the estimation of strain tensor are unlocked. (Figure 3).

The **first** part is the selection of the method for strain estimation. If 'shen' is passed in, the estimation will follow the algorithm described in Shen et al, (2015), using a weighted least squares approach. If 'veis' is passed in, then the region is going to be split into delaneuy triangles and a strain estimated in each barycenter. Default is 'shen'. If 'One Tensor' checked, then only one strain tensor will be estimated, at the region's barycentre.

In the **second** part, user specifies the region as a rectangle and x-axis/y-axis grid steps. Any station falling outside this region will be omitted.

In the **third** part, the user selects the interpolation model parameters for 'shen' method. The options are:

- **Wt**: Let $W = \sum_i G_i$, the total reweighting coefficients of the data, and let W_t be the threshold of W . For a given W_t , the smoothing constant D is determined by $W_d = W_t$. It should be noted that W is a function of the interpolation coordinate, therefore for the same W_t assigned, D varies

spatially based on the in-situ data strength; that is, the denser the local data array is, the smaller is D , and vice versa. Default is $Wt=24$.

-D min: This is the lower limit for searching for an optimal d -param value. Unit is km. Default is $dmin=1$ km.

-D max: This is the upper limit for searching for an optimal d -param value. Unit is km. Default is $dmax=500$ km.

-D step: This is the step size for searching for an optimal d -param value. Unit is km. Default is $dstep=2$ km.

-D parameter: This is the 'D' parameter for computing the spatial weights. If this option is used, then the parameters: $dmin$, $dmax$, $dstep$ and Wt are not used.

Finally, there are two special arguments as:

- **cut excess stations:** If this option is enabled, then any station (from the input file) outside the region limit (passed in via the 'region' option) is not considered in the strain estimation.

- **generate statistics:** This option will create an output file, named 'strain_stats.dat', where estimation info and statistics will be written.

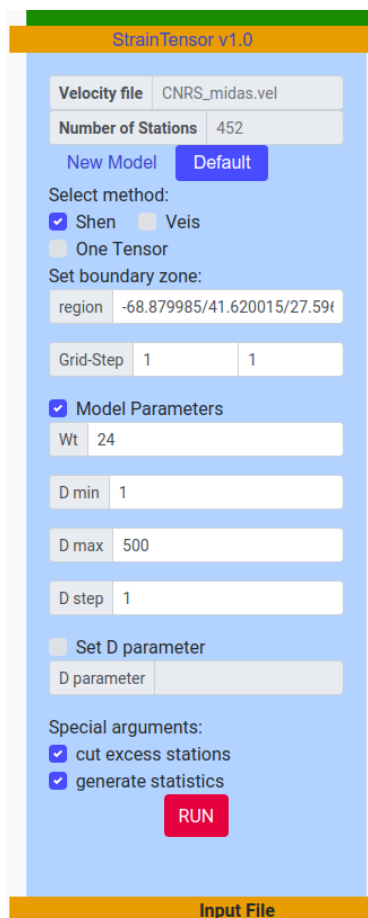


Figure 3. Graphic input interface for options and parameters

5.3 Output Files

The results of *StrainWebTool* are recorded in the following three files:

[strain_info.dat](#): This file includes strain tensor parameters, principal axis, rotational rates, dilatation etc.

The columns of the file are structured as below:

Col num	Column name	Units	description
1	Latitude	Deg	
2	Longitude	Deg	
3+4	vx +dvx	mm/y	
5+6	vy + dvy	mm/y	
7+8	w + dw	deg/My	Rotational rates
9+10	exx + dextr	nstrain/y	Strain Tensor parameters
11+12	exy + dexy	nstrain/y	
13+14	eyy + deyy	nstrain/y	
15+16	emax + demax	nstrain/y	E _{max}
17+18	emin + demin	nstrain/y	E _{min}
19+20	shr + dshr	nstrain/y	Shear Strain
21+22	azi + dazi	deg	Azimuth of E _{max}
23+24	dilat + ddilat	nstrain/y	Dilatation
25+26	sec.inv + dsec.inv	nstrain/y	Second Invariant

[station_info.dat](#) : Stations' data used for the calculation of strain tensor are written at this file.
Format is: Code / Longitude / Latitude / V_e / V_n

[strain_stats.dat](#) : Parameters and statistics used to modify grid, calculate Strain tensors etc.

6. Active Map visualize strain tensor results

For results visualization, the application uses an active map developed using the *Leaflet javascript* library. In this map, user can choose to plot principal axes, shear strain, dilatation or second invariant results. In addition, user can add as separate layer the stations and their respective velocities used to estimate strain tensors (Figure 4).

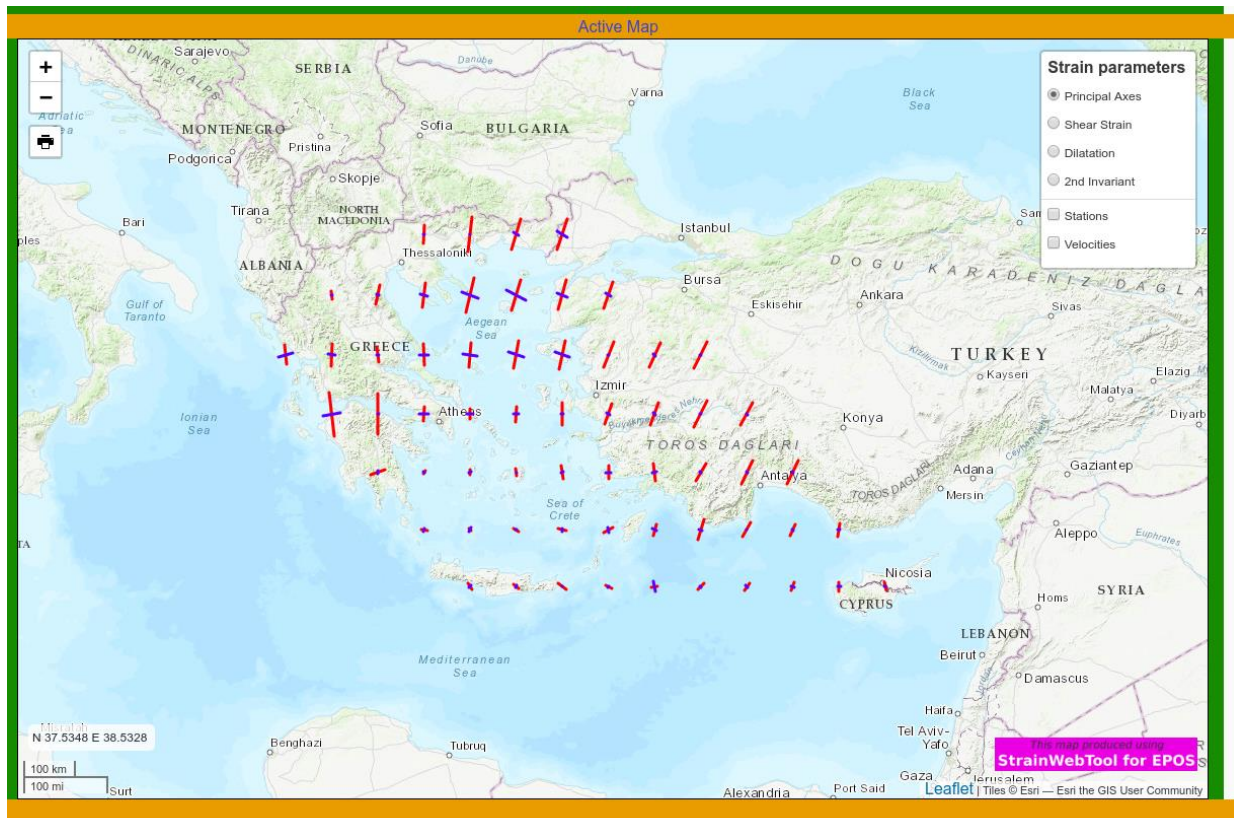


Figure 4. An example of graphic output showing principal axes of strain.

7. Final note

Due to the fact that the STRAINTOOL project is still under development, minor parts of what is described above may be subject to change. However, the Project comes with comprehensive documentation where the user can be informed of any alterations.

References

Anastasiou D., Ganas A., Legrand J., Bruyninx C., Papanikolaou X., Tsironi V. and Kapetanidis V. 2019a. Tectonic strain distribution over Europe from EPN data. EGU General Assembly 2019, Geophysical Research Abstracts, Vol. 21, EGU2019-17744-1

Anastasiou D., Papanikolaou X., Ganas A., Paradissis D., 2019b. StrainTool: A software package to estimate strain tensor parameters (Version v1.0) Zenodo. DOI: [10.5281/zenodo.3239497](https://doi.org/10.5281/zenodo.3239497)

Contribution to EPOS-IP WP10 STRAIN PRODUCT, Task 10.6 GNSS Products - Guidelines for DDSS Strain-rate derivation maps, A. Ganas, K. Chousianitis, version: 20 December 2016.

Contribution to EPOS-IP WP10 STRAIN PRODUCT, Task 10.6 Report 6 strain-tool-epos, Xanthos Papanikolaou, Dimitris Anastasiou, A. Ganas, version: 23 February 2018 and 20 July 2018.

Python Software Foundation. Python Language Reference, version 2.7. Available at <http://www.python.org>

Shen, Z.-K., M. Wang, Y. Zeng, and F. Wang, 2015. Strain determination using spatially discrete geodetic data, Bull. Seismol. Soc. Am., 105(4), 2117-2127, doi: 10.1785/0120140247

Veis, G., Billiris, H., Nakos, B., and Paradissis, D. 1992. Tectonic strain in Greece from geodetic measurements, C. R. Acad. Sci. Athens, 67:129–166

[The Generic Mapping Tools – GMT](#)

Flask web development <http://flask.pocoo.org/>

Bootstrap 4.2 <https://getbootstrap.com/>

Leaflet 1.4.0 <https://leafletjs.com/>

[END OF REPORT]