StrainTool - Improving the Mapping of Tectonic Strain in Eurasia

D.Anastasiou[1,2], X. Papanikolaou[1,2], V. Kapetanidis[1,3], V. Tsironi[1], A. Ganas[1], D. Paradissis[2]

[1]Institute of Geodynamics - National Observatory of Athens
[2]Dionysos Satellite Observatory - Higher Geodesy Laboratory, National Technical Univarsity of Athens
[3]Department of Geology, National and Kapodistrian University of Athens

https://dsolab.github.io/StrainTool dganastasiou@gmail.com

12HSTAM2019
INTERNATIONAL CONGRESS ON MECHANICS
22-25.09.2019 | THESSALONIKI, GREECE
in honor of Professor Ioannis Vardoulakis on the occasion of 10 years from his death

Presentation Structure

Introduction

- StrainTool is a free and open-source software.
- Cooperation between the National Technical University of Athens (NTUA) and National Observatory of Athens (NOA) under EPOS-IP project.
- User-friendly software can be used directly by the scientific community.
- Pyhton programming language: free, flexible and cross-platform-compatible nature.
- Software's development was performed using Github.
- Input a list of data points along with their tectonic velovcities.
- Estimate Strain Tensor parameters.

Open Source Software **StrainTool v1.0**

StrainTool has three basic components:

- **pystrain:** A python pachage.
- **StrainTensor.py:** the main executable.
- A list of shell scripts to plot results from StrainTensor.py

TODO: structure design

Python Package `pystrain`

`pystrain` the core part of the project.

Python functions and classes, enable computation of strain tensor.

The package includes:

- `iotools`: input/output classes to parse ASCII files.
- `geodesy`: functions for basic geodetic calculations.
- `grid.py`: a simple grid generator
- `strain.py`: main class and necessary functions for estimation of strain tensor parameters

Estimate strain tensor parameters

Strain tensor parameters aestimated (or calculated) by solving for the system:

$$\begin{bmatrix} V_{x,S_1} \\ V_{y,S_1} \\ \dots \\ V_{x,S_n} \\ V_{y,S_n} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta_{y_1} & \Delta_{x_1} & \Delta_{y_1} & 0 \\ 0 & 1 & -\Delta_{x_1} & 0 & \Delta_{x_1} & \Delta_{y_1} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 1 & 0 & \Delta_{y_n} & \Delta_{x_n} & \Delta_{y_n} & 0 \\ 0 & 1 & -\Delta_{x_n} & 0 & \Delta_{x_n} & \Delta_{y_n} \end{bmatrix} \begin{bmatrix} U_x \\ U_y \\ \omega \\ \tau_x \\ \tau_{xy} \\ \tau_y \end{bmatrix}$$

$\Delta_{x_i}, \Delta_{y_i}$ are the displacement components between station i and the point.
A minimum of three stations is required to compute the parameters.

Estimate strain tensor parameters

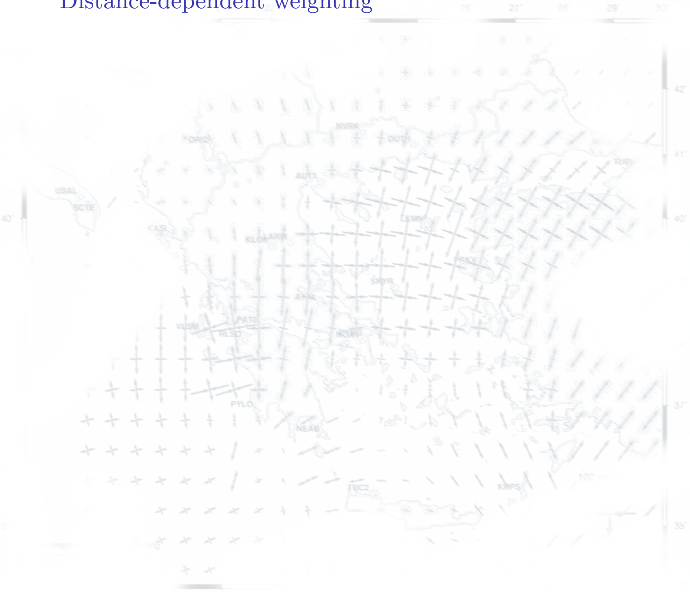
Assuming that there is a variance information for the station velocities (and a Gaussian distribution), we can add the covariance matrix C of the velocity data in the system. In the simplest case, C is a diagonal matrix, with the velocity component standard deviations as its elements.

$$C = \sigma_0^2 \begin{bmatrix} (\frac{1}{\sigma_{V_{x_1 S_1}}})^2 & 0 & 0 & ... & 0 \\ 0 & (\frac{1}{\sigma_{V_{y_1 S_1}}})^2 & 0 & ... & 0 \\ 0 & 0 & (\frac{1}{\sigma_{V_{x_2 S_2}}})^2 & ... & 0 \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & (\frac{1}{\sigma_{V_{y_i S_i}}})^2 \end{bmatrix}$$

Shen Algorithm

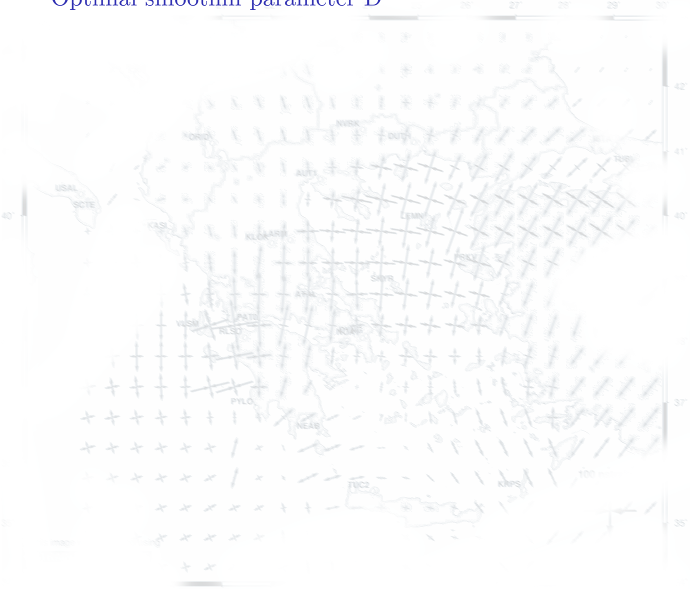
Shen Algorithm

Distance-dependent weighting



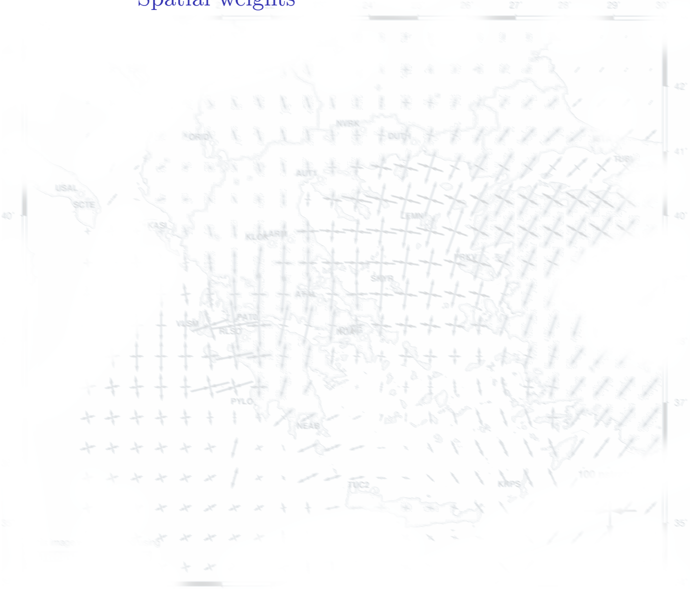
Shen Algorithm

Optimal smoothin parameter D



Shen Algorithm

Spatial weights



Veis Algorithm

Input Datasets

Validation

Conclusions

Thank you for your attention!