

# StrainTool - Improving the Mapping of Tectonic Strain in Eurasia

D.Anastasiou<sup>1,2</sup>, X. Papanikolaou<sup>1,2</sup>, V. Kapetanidis<sup>1,3</sup>, V. Tsironi<sup>1</sup>, A. Ganas<sup>1</sup>,  
D. Paradissis<sup>2</sup>

<sup>1</sup>Institute of Geodynamics - National Observatory of Athens

<sup>2</sup>Dionysos Satellite Observatory - Higher Geodesy Laboratory,  
National Technical University of Athens

<sup>3</sup>Department of Geology and Geoenvironment, National and Kapodistrian University of Athens



<https://dsolab.github.io/StrainTool>



dganastasiou@gmail.com



# Presentation Structure

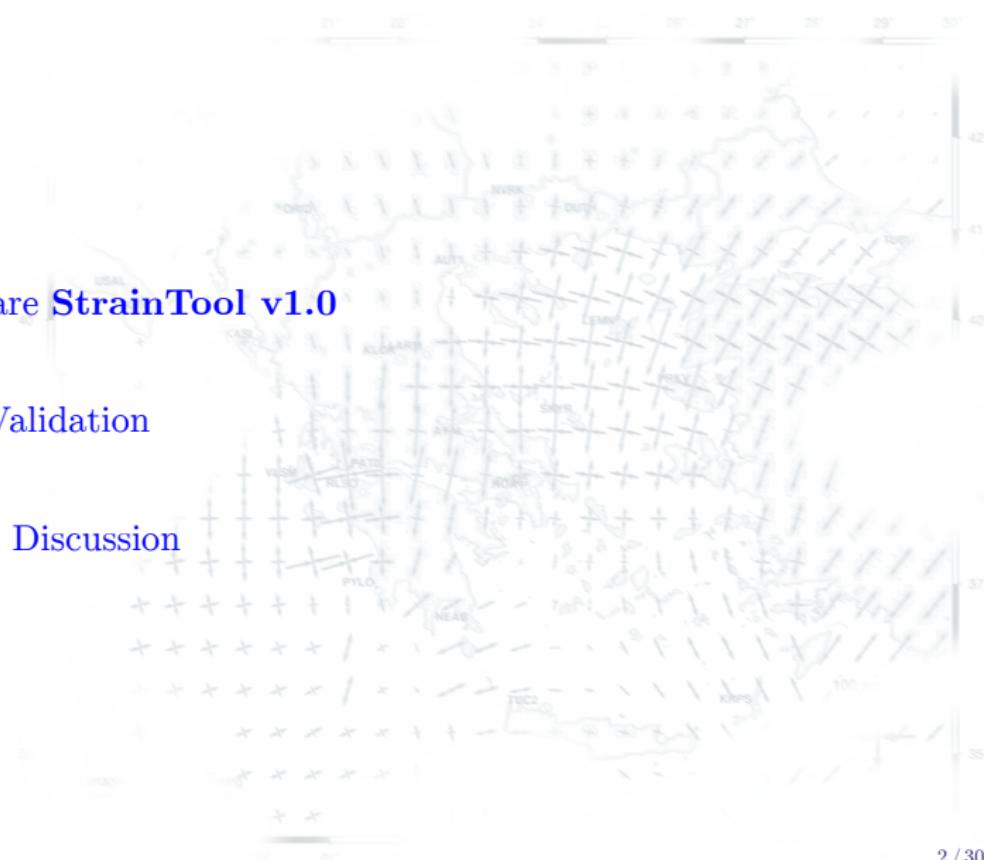
Introduction

Open Source Software **StrainTool v1.0**

Data analysis and Validation

Strain Analysis and Discussion

Conclusions



# 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.
- Python programming language: free, flexible and cross-platform-compatibility.
- Software's development was performed using Github.
- Input a list of data points along with their tectonic velocities.
- Estimate Strain Tensor parameters.

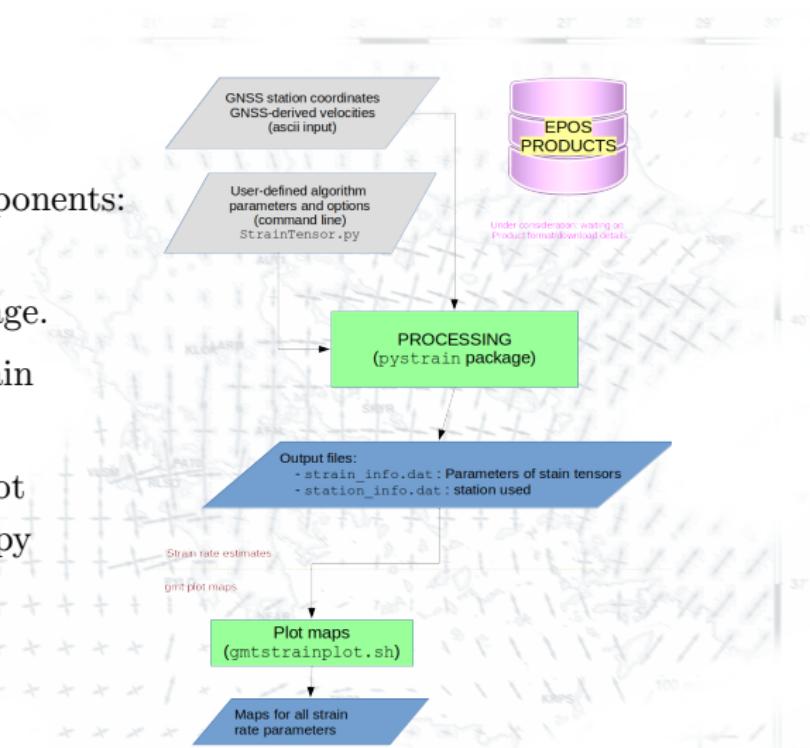
# Open Source Software **StrainTool v1.0**

StrainTool has three basic components:

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

**TODO:** structure design, add

strain\_stats.dat



# 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 estimated (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 cell center.

A minimum of three stations is required to compute the parameters.

**explain all parameters**

## Estimate strain tensor parameters

Assuming that there is variance information for the station velocities (and a Gaussian distribution), we can include 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.

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

# Shen Algorithm

Shen et al. 2015, propose a more elaborate approach.

The weighting function

$$G_i = L_i Z_i$$

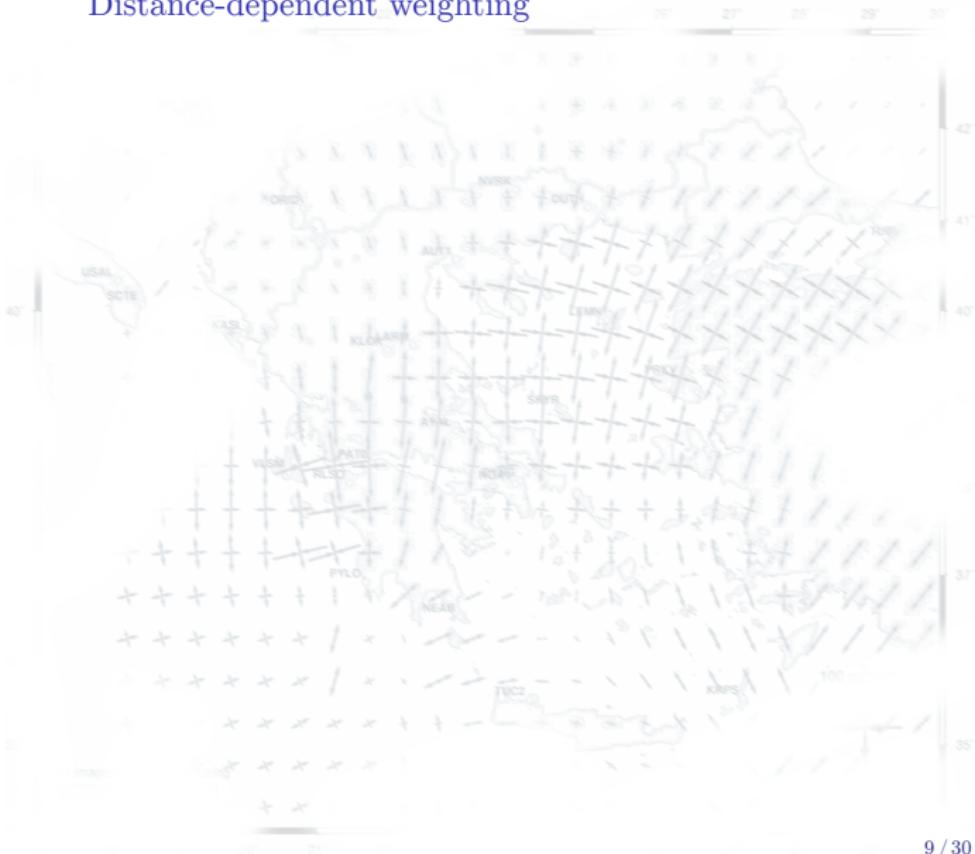
$L_i$  : distance-dependent weighting

$Z_i$  : spatial weighting

The final covariance matrix  $C_i = C_i G_i^{-1}$

# Shen Algorithm

## Distance-dependent weighting



# Shen Algorithm

## Optimal smoothing parameter D

Smoothing coefficient D needed to actually compute the distance-dependent weights  $L_i$ .

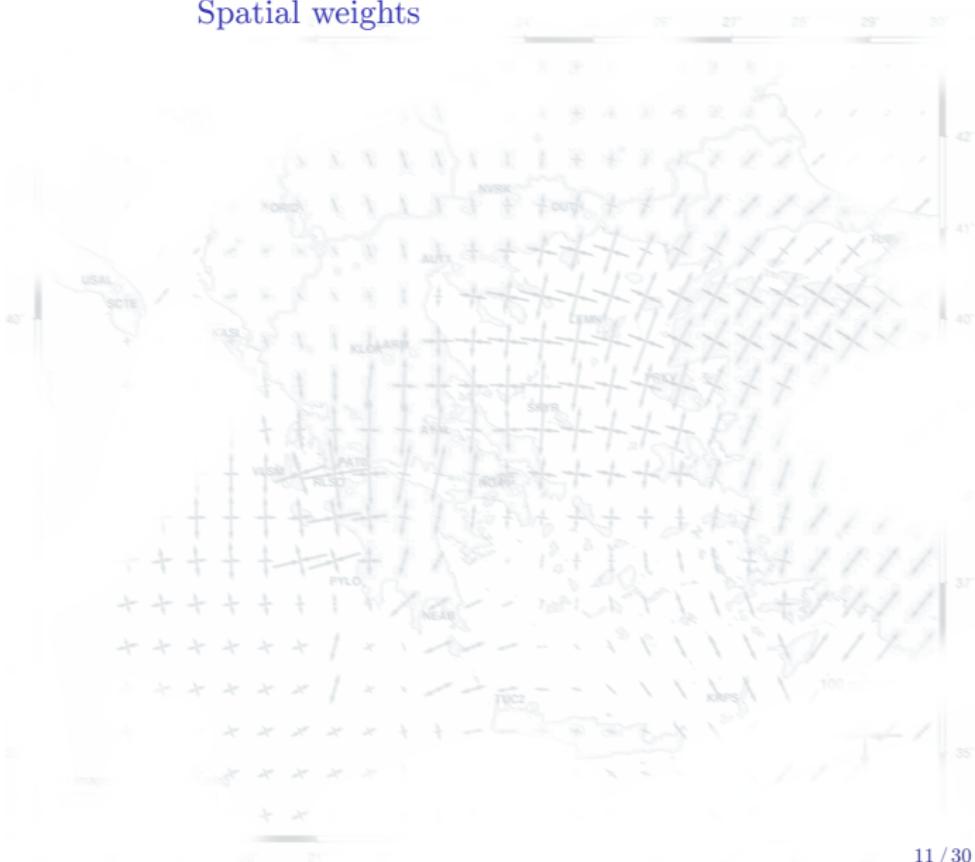
- Either pass in the parameter value as a command line option, or
- search for an optimal D value, given the range Dmin, Dmax, Dstep, and the limit value W

Searching for an optimal D value

1. If a Gaussian approach is selected, then  $L_{\text{max}} = 2.15D$  (in km), while for the quadratic approach  $L_{\text{max}} = 10D$  (in km).
2. Compute distance-dependent and spatial weights  $L_i$  and  $Z_i$  respectively, for every station
3. compute the sum  $W = 2 \sum Z_i L_i$ .
4. repeat the process for the next D value if absolute value W is smaller than Wt, else current D value is optimal.

# Shen Algorithm

## Spatial weights



## Veis Algorithm

The region is split into delaunay triangles at the barycenter of which a strain tensor is computed.

This approach uses only three points to calculate tensor parameters

### Assumptions:

- 2-dimensional deformation of earth's crust in time
- Crust is considered a thin deformable shell on a spherical earth
- Mapping distortions are ignored for regions with radius of less than  $5^{\circ}$
- Time (earthquakes) or space (faults) discontinuities are not included in the calculation

# Strain Tensor parameters

parameter	equation
maximum shear extension	$\tau_{max} = \sqrt{\tau_{xy}^2 + e_{diff}^2}$
compression	$e_{max} = e_{mean} + \tau_{max}$
Azimuth of $e_{max}$	$Az_{e_{max}} = 90 + \frac{-\text{atan2}(\tau_{xy}, e_{diff})}{2}$
dilatation	$dil = \tau_x + \tau_y$
Second Invariand	$2nd\_inv = \sqrt{\tau_x^2 + \tau_y^2 + 2\tau_{xy}^2}$

where,

$$e_{mean} = \frac{\tau_x + \tau_y}{2} \quad e_{diff} = \frac{\tau_x - \tau_y}{2}$$

TODO: make a table with param and math equation

# Input Datasets

**For validation:**

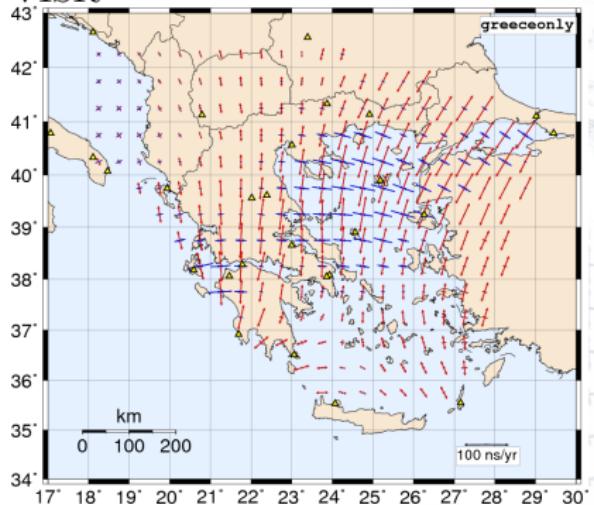
1. EPN network, solution C2010 - ETRF 2014 (299 stations)
2. EPOS network, INGV solution (571 stations)
3. EPOS network, CNRS solution (MIDAS) (452 stations)
4. network GREECE, NTUA reprocessing 2017 (153 stations)

# Validation

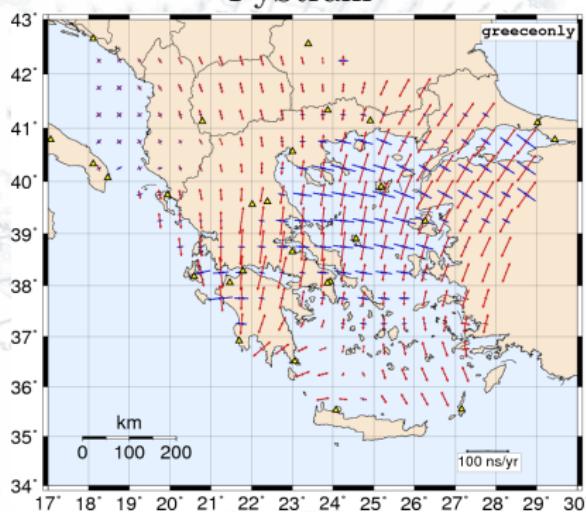
## $\epsilon_{\text{max}}$ - $\epsilon_{\text{min}}$ maps comparison

TODO: datasets

VISR



PyStrain

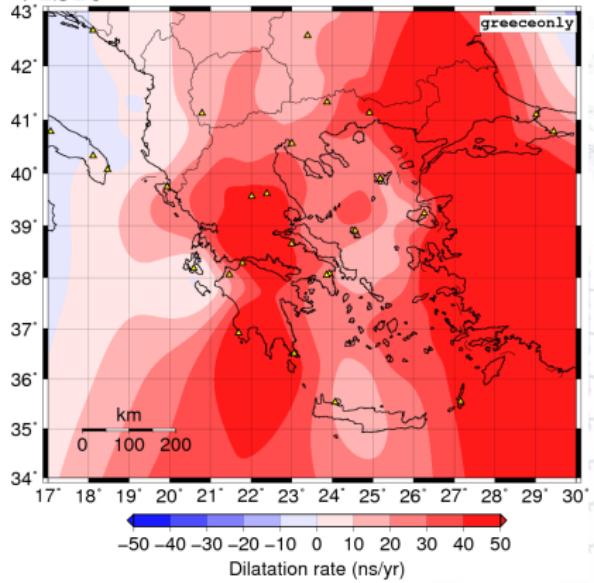


# Validation

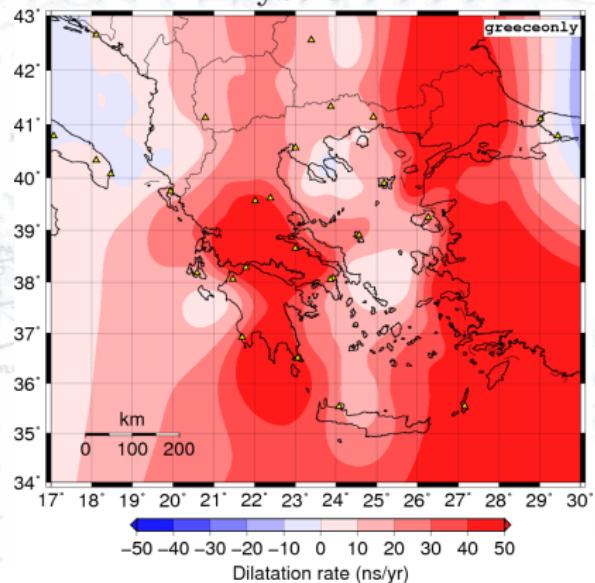
## dilatation maps comparison

maybe a wrong dataset

VISR

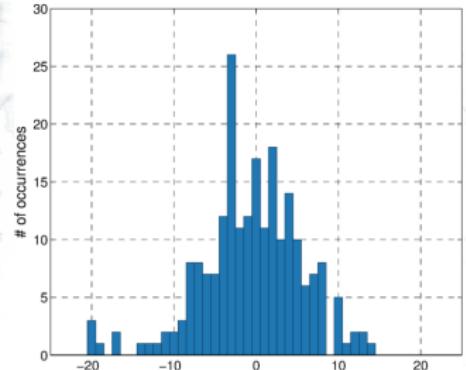
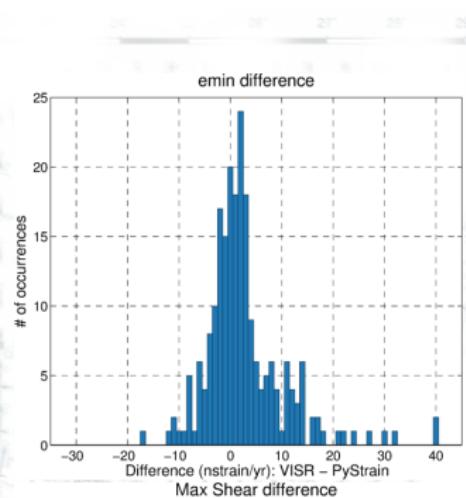
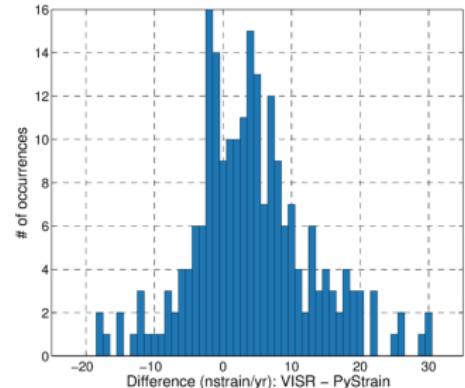
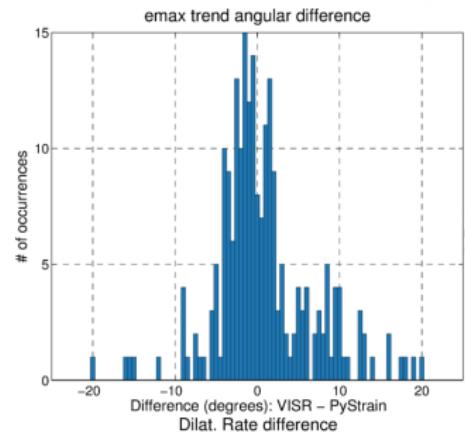


PyStrain



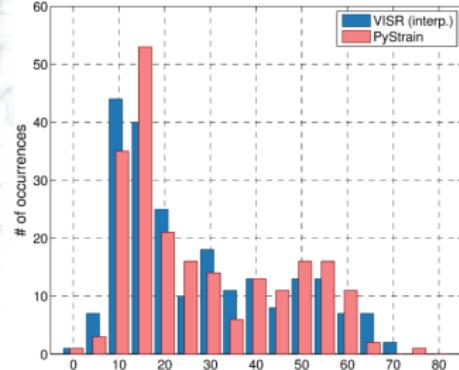
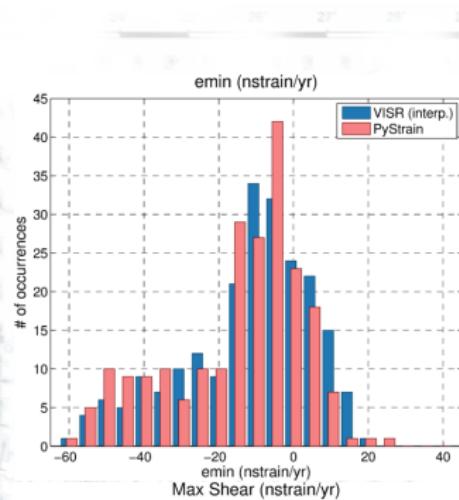
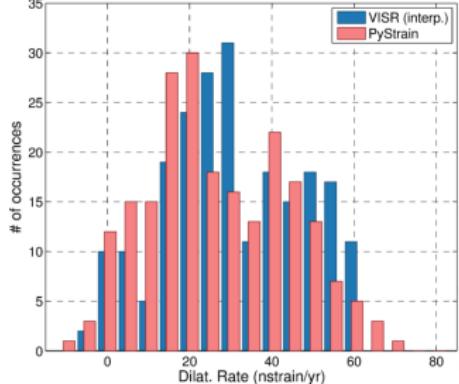
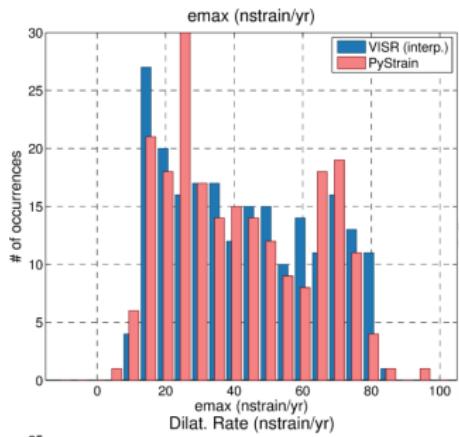
# Validation

## differences



# Validation

## histograms

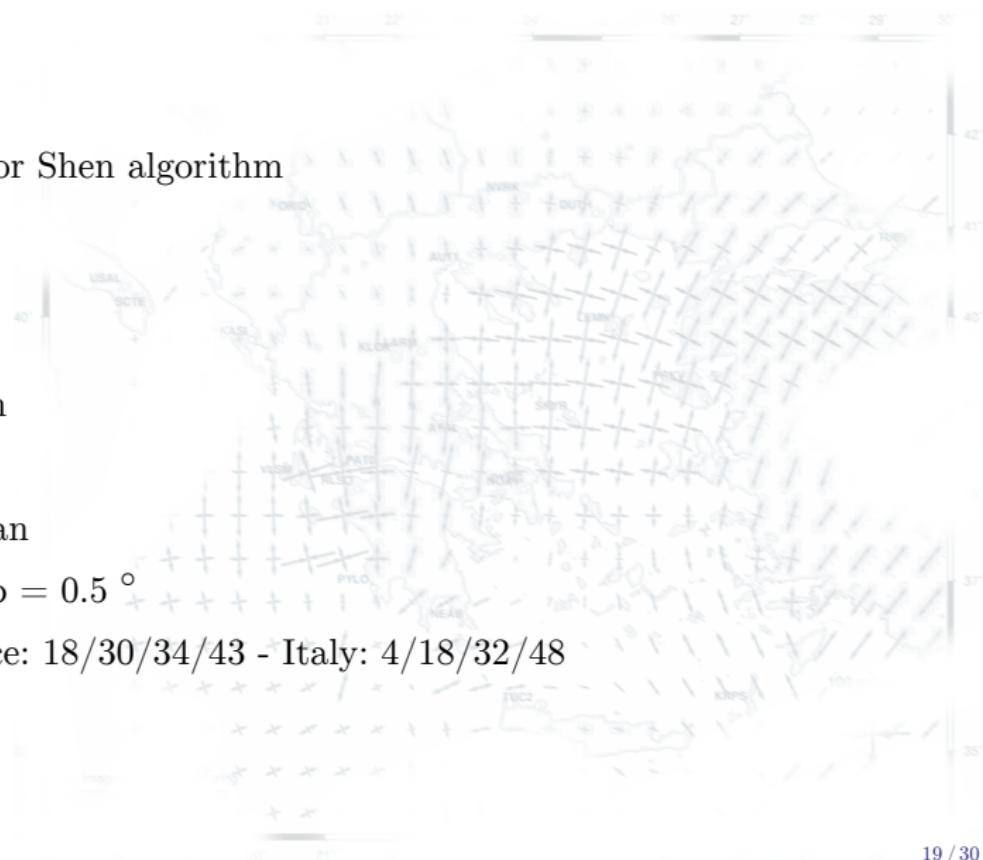


# Strain Analysis

Model parameters for Shen algorithm

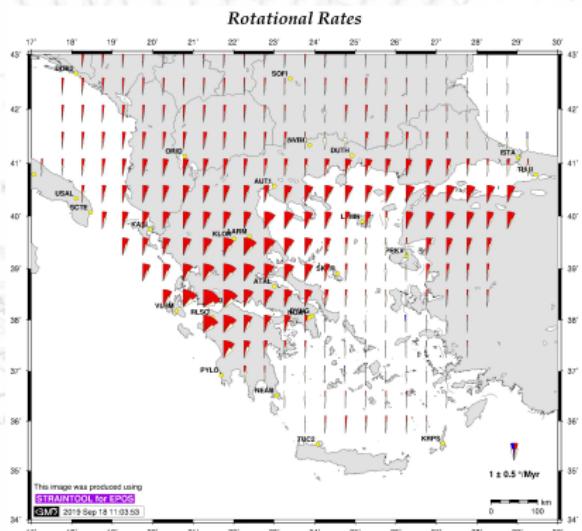
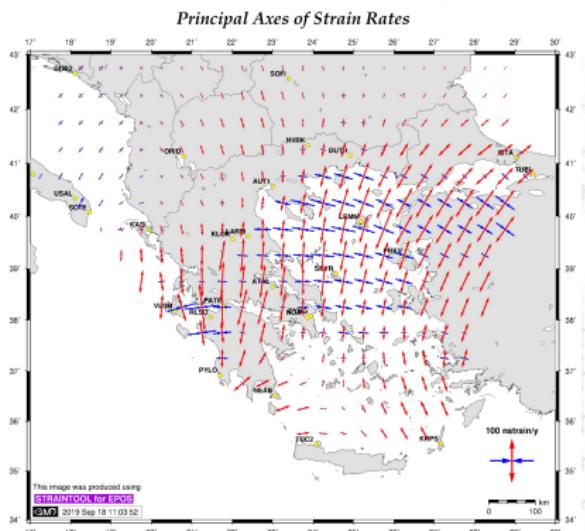
Data Set: MIDAS

- $W_t = 6$
- $d_{min} = 1\text{ km}$
- $d_{max} = 500\text{ km}$
- $d_{step} = 1\text{ km}$
- $ltype = \text{gaussian}$
- $x \text{ step} = y \text{ step} = 0.5^\circ$
- $\text{region} = \text{Greece: } 18/30/34/43 - \text{Italy: } 4/18/32/48$



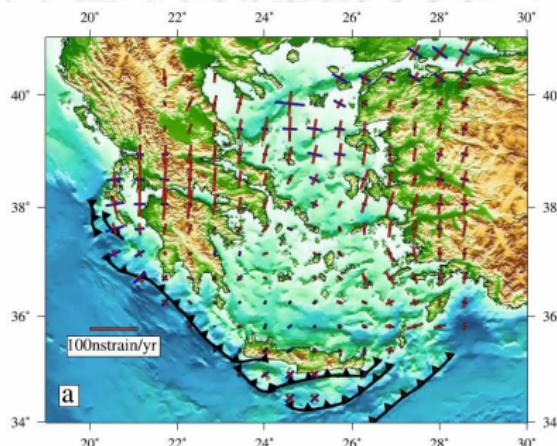
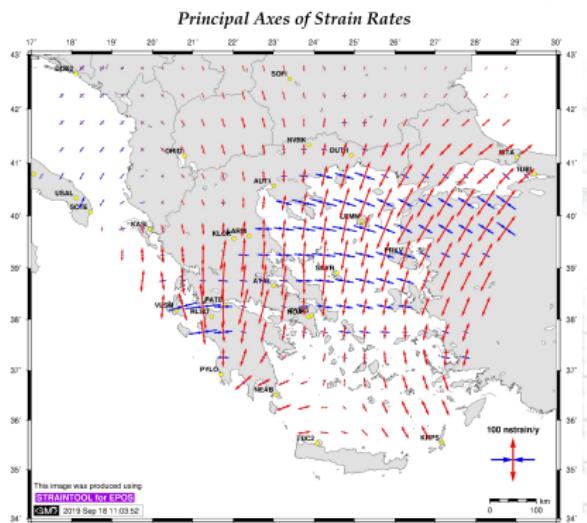
# Strain Analysis

## Greece region



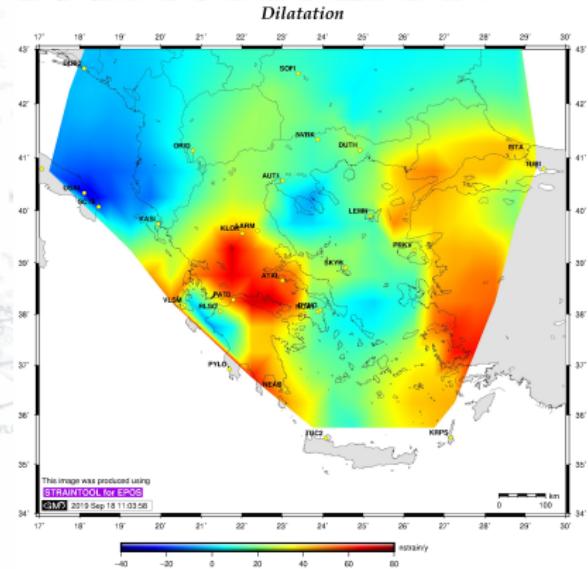
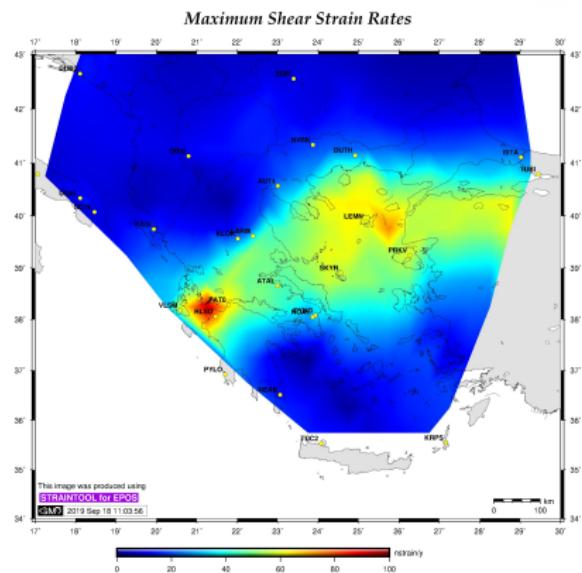
# Product Validation

## Greece region



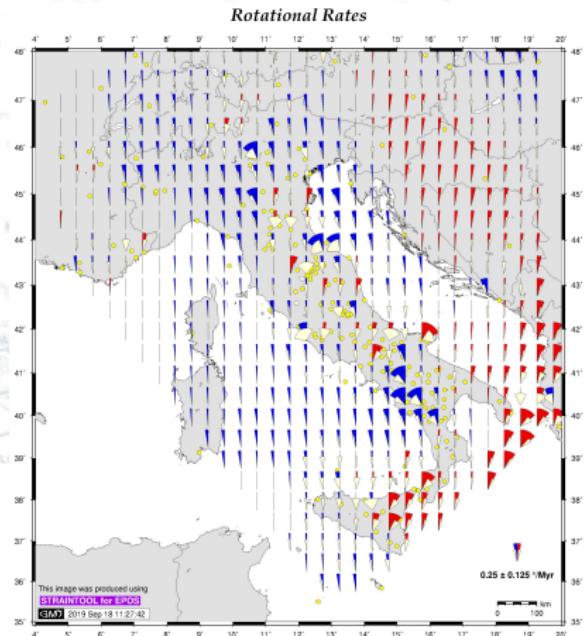
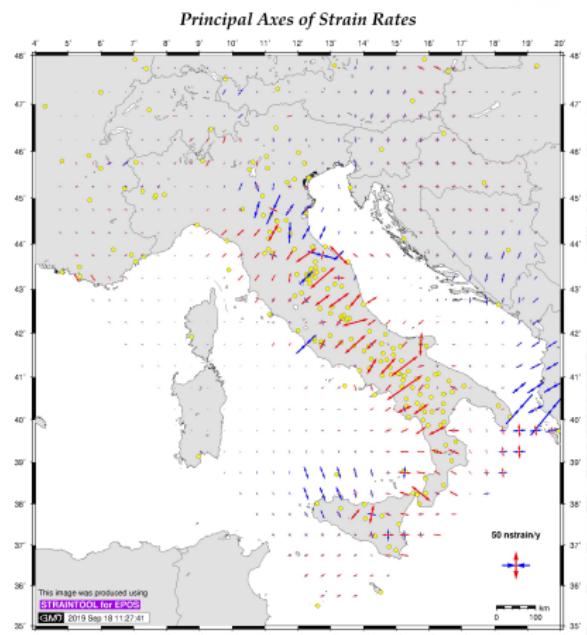
# Strain Analysis

## Greece region



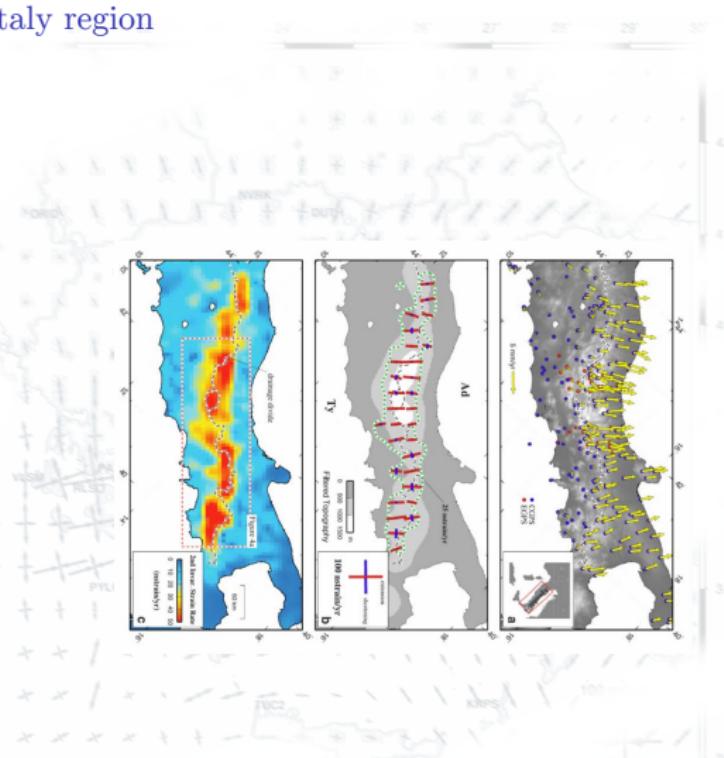
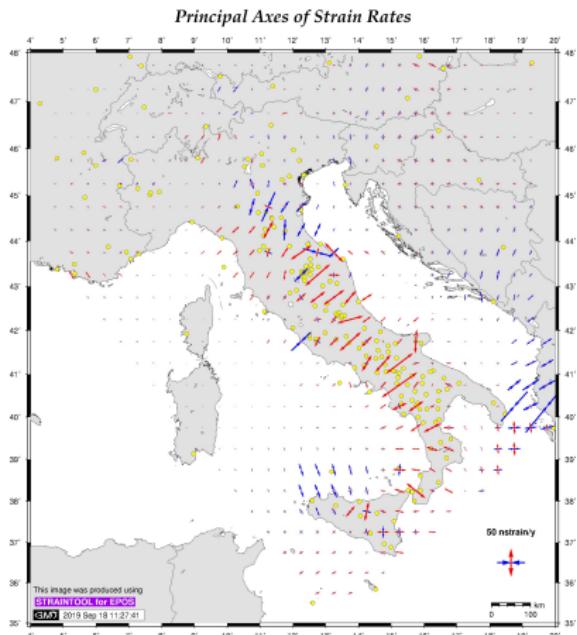
# Strain Analysis

## Italy region



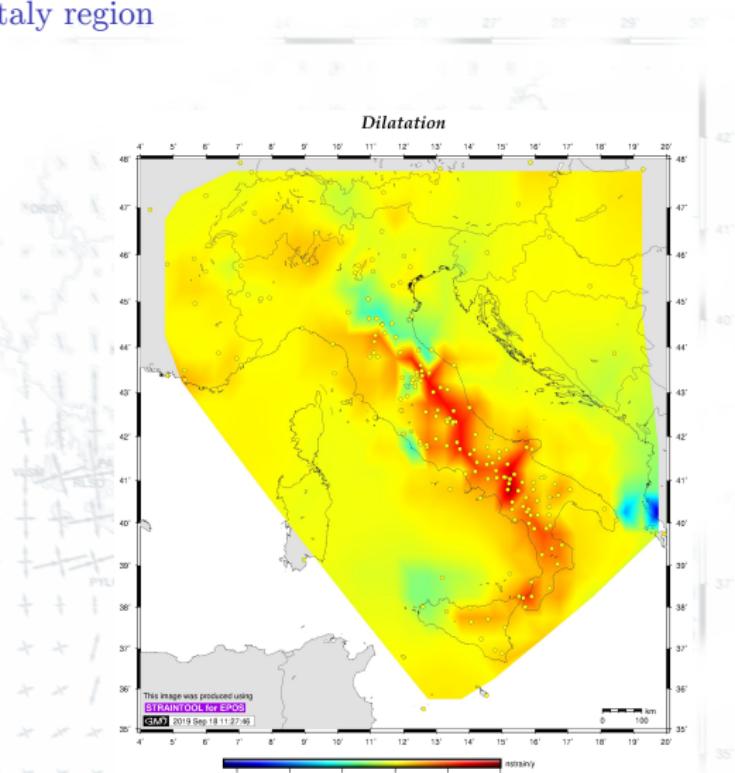
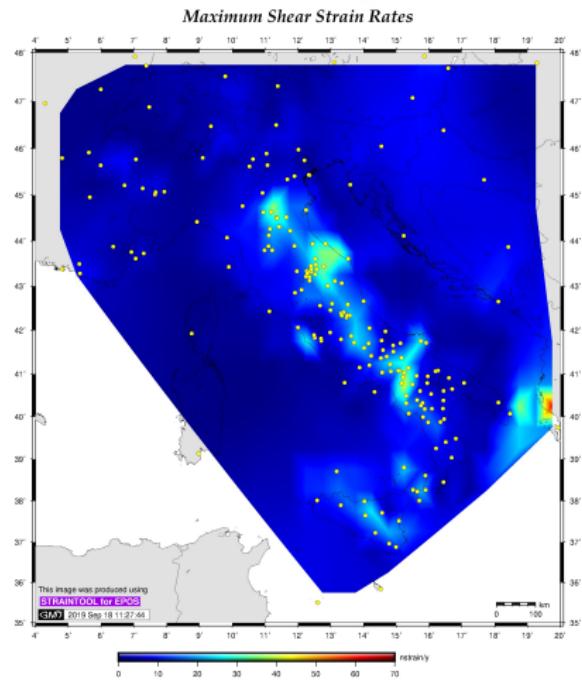
# Product Validation

## Italy region



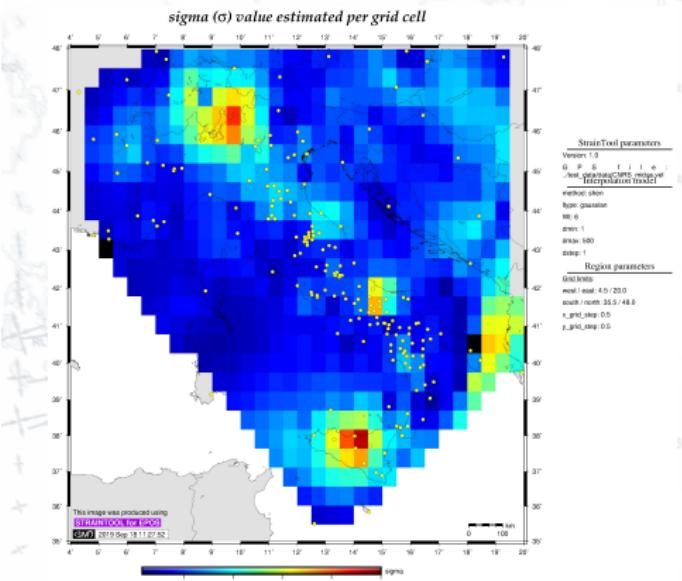
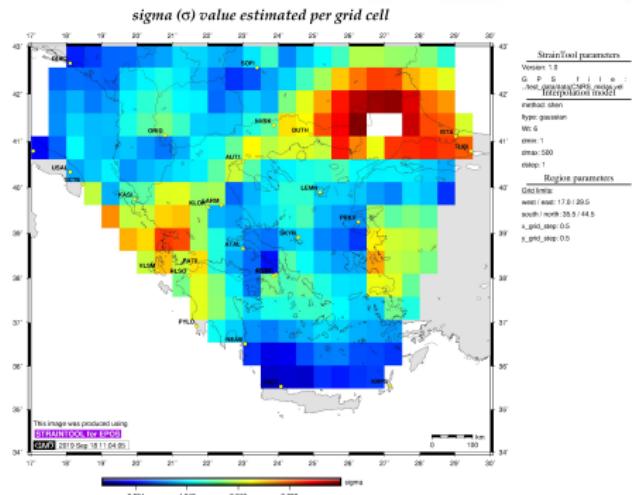
# Strain Analysis

## Italy region



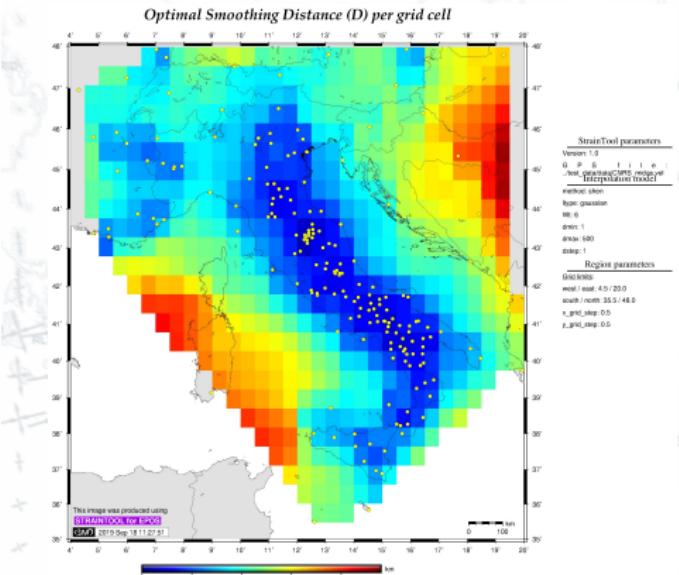
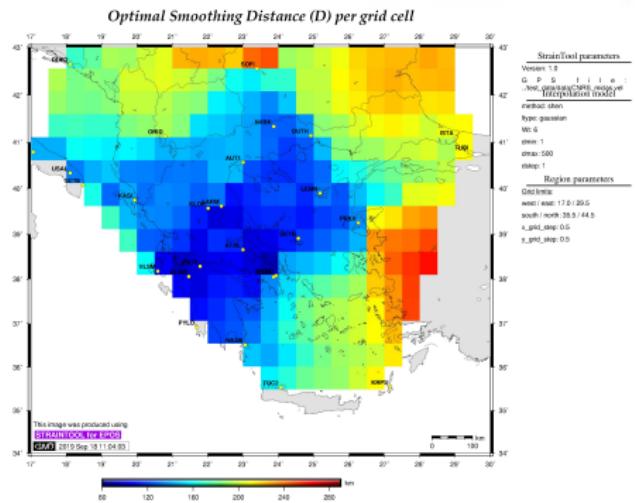
# Statistics

## Sigma value



# Statistics

## D optimal smoothing parameter



# Conclusions

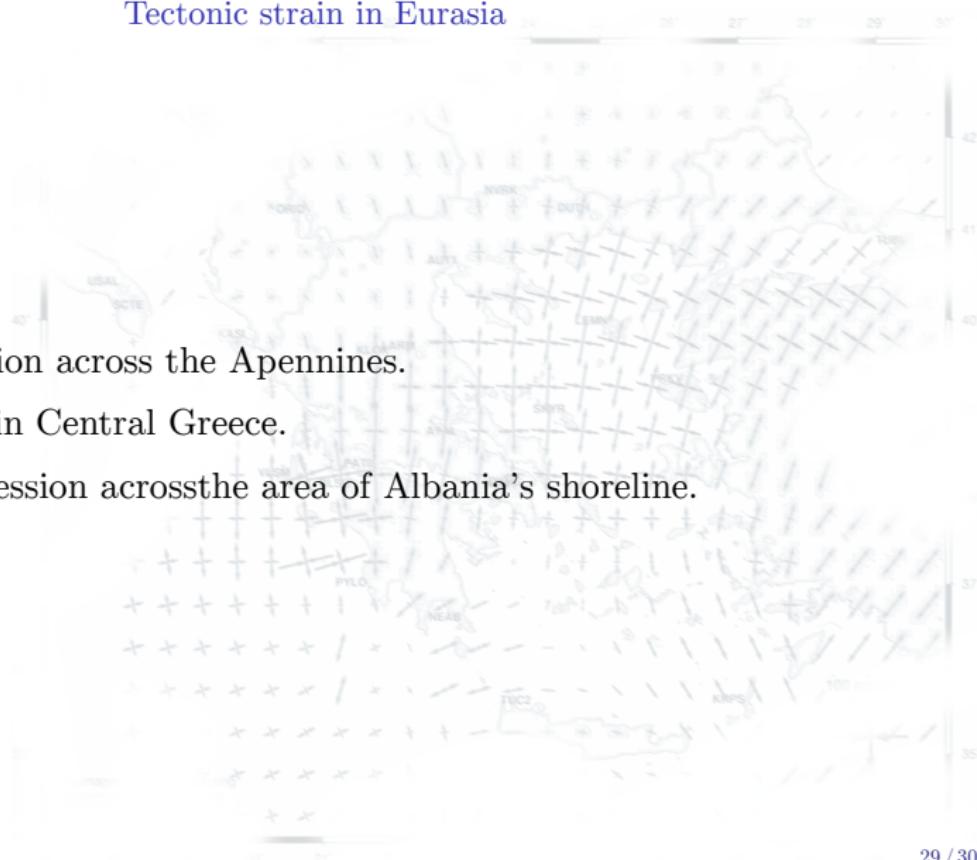
## StrainTool open-source software v1.0

- We propose a new, open-source tool to estimate strain in geodesy and geodynamics, the **STRAINTOOL**.
- Free, flexible and cross-platform-compatibility.
- Use different algorithms to estimate strain tensor parameters.
- We validated our calculations using two open-source algorithms recommended by EPOS-IP, namely the VISR and STIB as well as the SSPX software suite.
- Our results reproduce the gross features of tectonic deformation in both Italy and Greece, such as NE-SW extension across the Apennines and N-S extension in Central Greece.
- It is anticipated that the significant increase of GNSS data amount associated with the operational phase of EPOS in the forthcoming years will be of great value to perform an unprecedented, reliable strain rate computation over the Eurasian plate.

# Conclusions

## Tectonic strain in Eurasia

- NE-SW extension across the Apennines.
- N-S extension in Central Greece.
- NE-SW compression across the area of Albania's shoreline.



Thank you for your attention!