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 University of Athens

[3]Department of Geology and Geoenvironment, 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

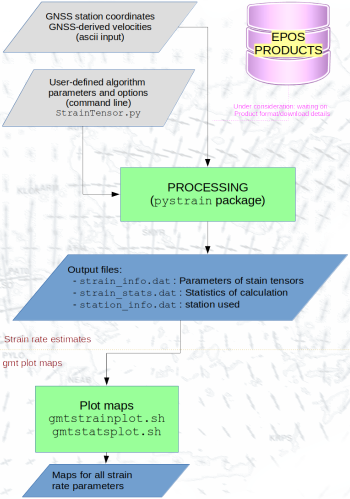
- StrainTool is a free and open-source software.
- Cooperation between the National Technical University of Athens (NTUA) and National Observatory of Athens (NOA) under EPOS-IP project.
- User-friendly software can be used directly by the scientific community.
- Pyhton programming language: free, flexible and cross-platform-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**

(Anastasiou et al., 2019)

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



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:

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

V_{x,S_1}, V_{y,S_1} : North-east velocity components for station S_1

$\Delta_{x_i}, \Delta_{y_i}$: Displacement components between station i and cell center.

U_x, U_y : Translation components.

ω : Total solid body rotation.

$\tau_x, \tau_{xy}, \tau_y$: Horizontal strain components.

A minimum of three stations is required to compute the 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.

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

Shen Algorithm

(Z. Shen et al., 2015)

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

Optimal smoothing parameter D

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

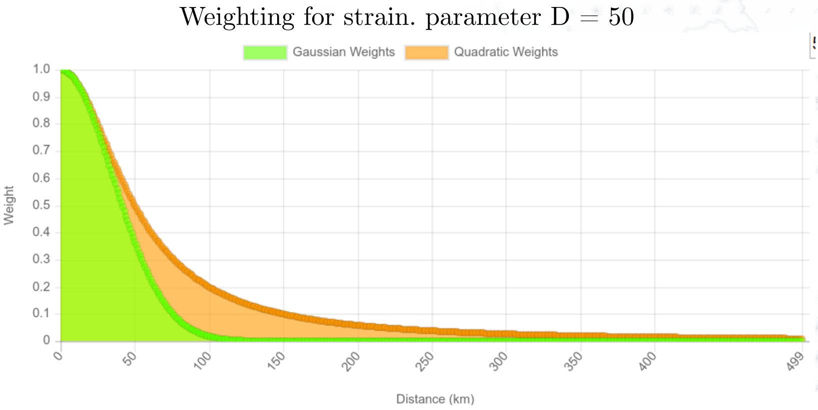
- Either pass in the parameter value as a command line option, or
- search for an optimal D value, given the range Dmin, Dmax, Dstep, and the limit value W

Searching for an optimal D value

1. If a Gaussian approach is selected, then Lmax = 2.15D (in km), while for the quadratic approach Lmax = 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

Distance-dependent weighting

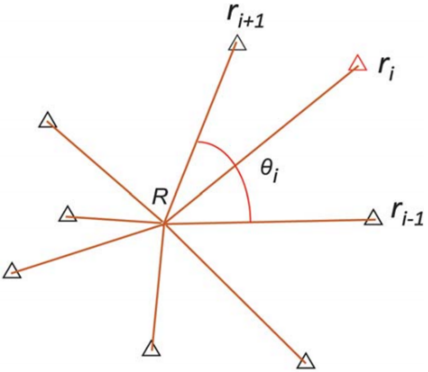


Gaussian: $L_i = exp(-\Delta R^2/D^2)$

Quadratic: $L_i = 1/(1 + \Delta R^2/D^2)$

Shen Algorithm

Spatial weights



(Z. Shen et al., 2015)

The azimuth span θ_i of geodetic data point r_i relative to interpolation site R. Triangles denote locations of geodetic data points near the interpolation site R.

Veis Algorithm

(Veis et al., 1992)

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^{o}
- Time (earthquakes) or space (faults) discontinuities are not included in the calculation

Strain Tensor parameters

Parameter	Formula
Maximum shear strain	$\tau_{max} = \sqrt{{\tau_{xy}}^2 + {e_{diff}}^2}$
Extension	$e_{max} = e_{mean} + \tau_{max}$
Compression	$e_{min} = e_{mean} - \tau_{max}$
Azimouth of e_{max}	$Az_{e_{max}} = 90 + \frac{-atan2(\tau_{xy}, e_{diff})}{2}$
Dilatation	$dil = \tau_x + \tau_y$
Second Invariand	$2nd\_inv = \sqrt{{\tau_x}^2 + {\tau_y}^2 + 2{\tau_{xy}}^2}$

where,

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

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)

Softwares used for validation:

(I) **VISR:** free, Linux-based, needs own scripting of plotting tools (Z.-K. Shen et al., 1996)

(II) **STIB:** needs request, python-based, Spakman & Nyst method (Masson et al., 2014)

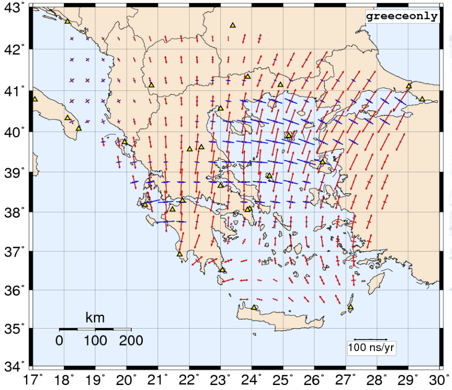
(III) **SSPX:** free, Mac-based, VISR method (Cardozo et al., 2008)

In this study we present results using CNRS (MIDAS).

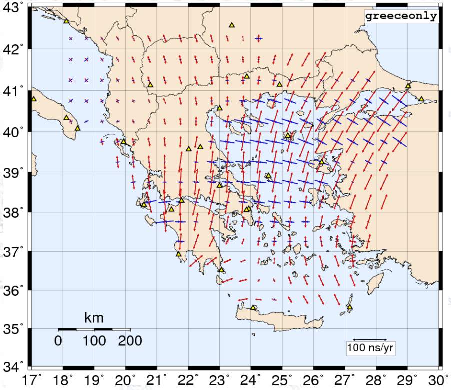
Validation

e_{max} - e_{min} maps comparison

VISR

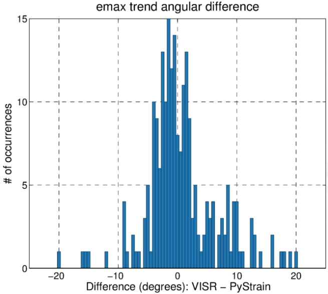


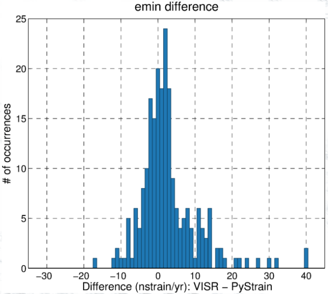
PyStrain

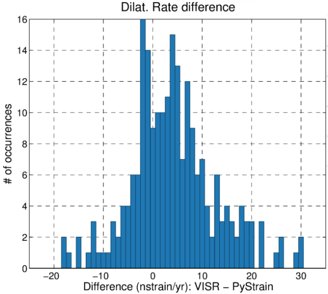


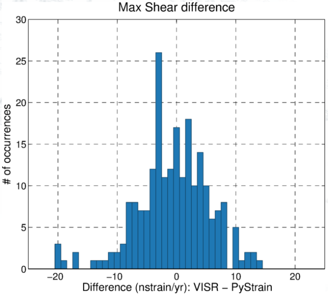
Validation

differences



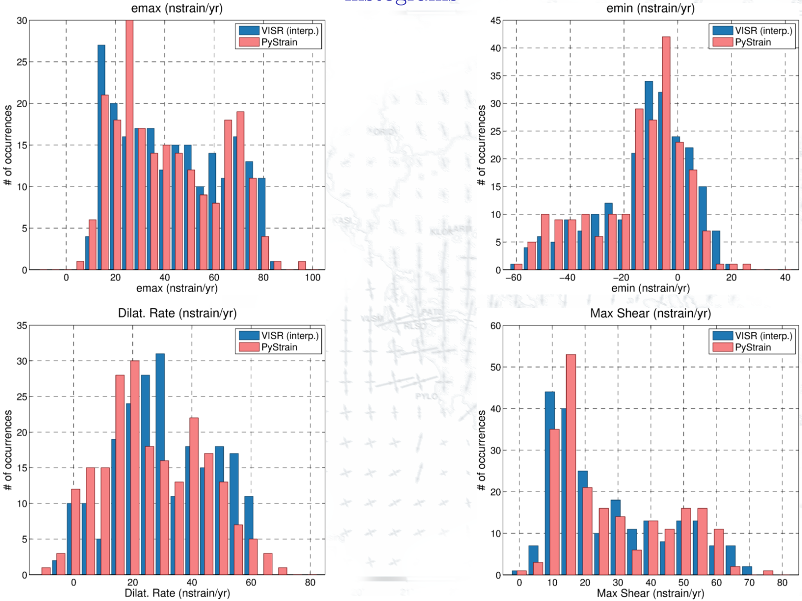






Validation

histograms



Strain Analysis

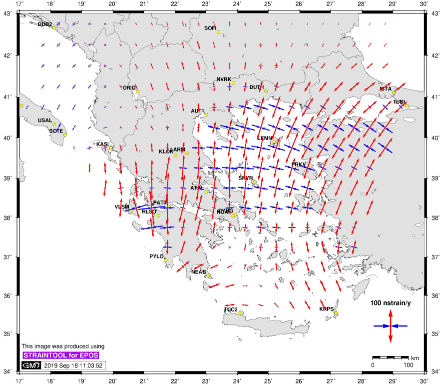
Model parameters for Shen algorithm
Data Set: CNRS Solution - MIDAS

- Wt=6
- dmin = 1km
- dmax = 500km
- dstep = 1km
- ltype = gaussian
- x step = y step = 0.5 °
- region = Greece: 18/30/34/43 - Italy: 4/18/32/48

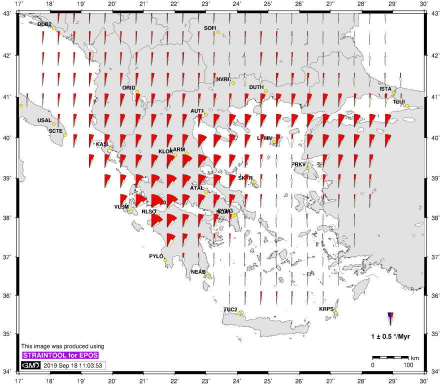
Strain Analysis

Greece region

Principal Axes of Strain Rates

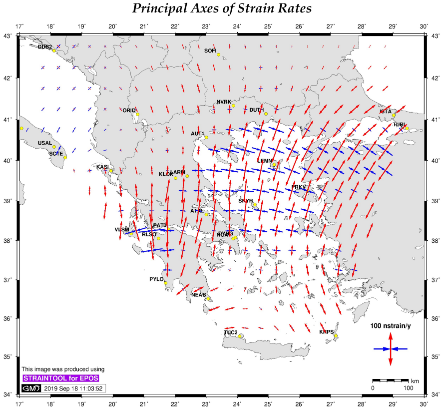


Rotational Rates

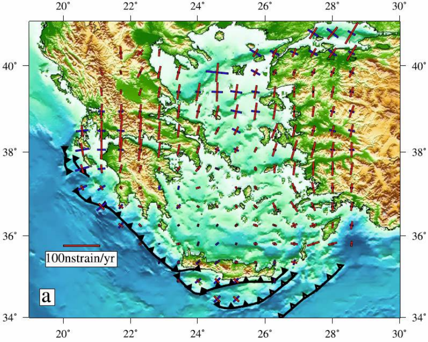


Product Validation

Greece region



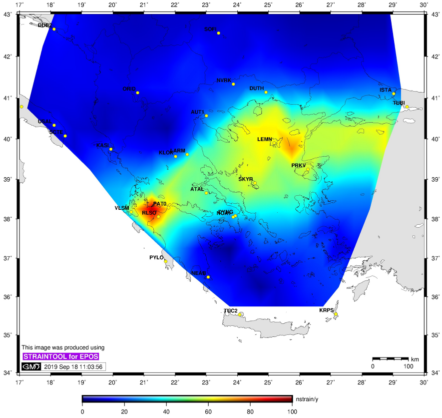
Floyd et al. (2010)



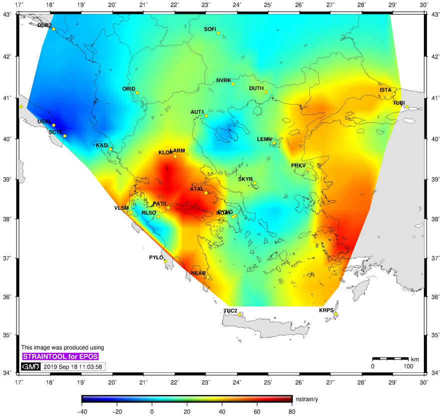
Strain Analysis

Greece region

Maximum Shear Strain Rates



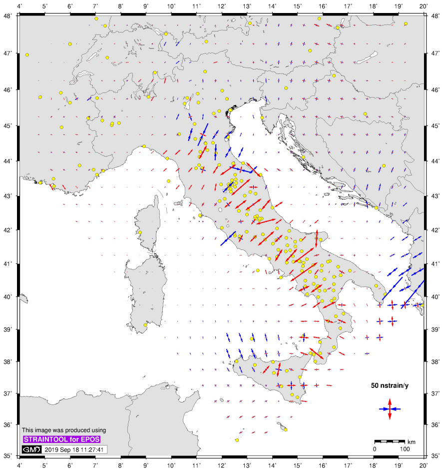
Dilatation



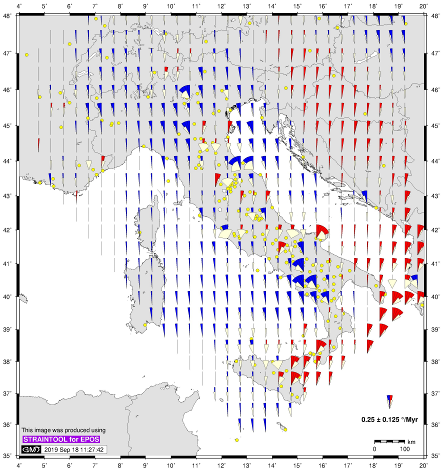
Strain Analysis

Italy region

Principal Axes of Strain Rates

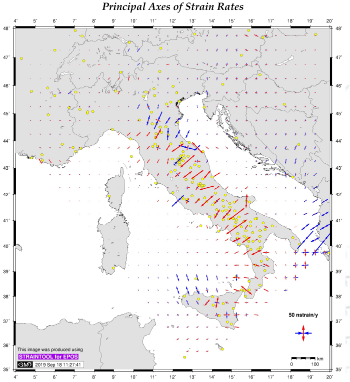


Rotational Rates

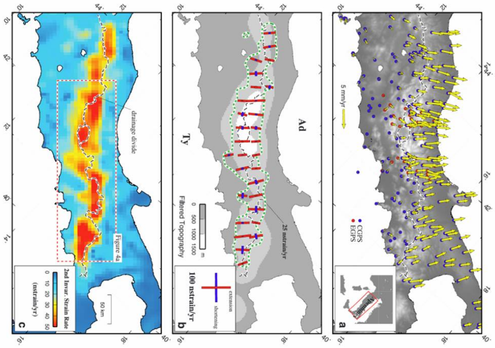


Product Validation

Italy region

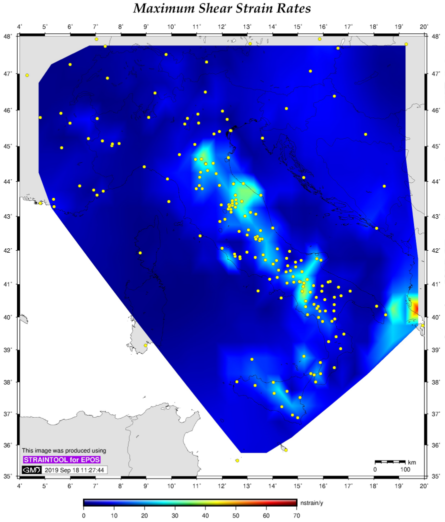


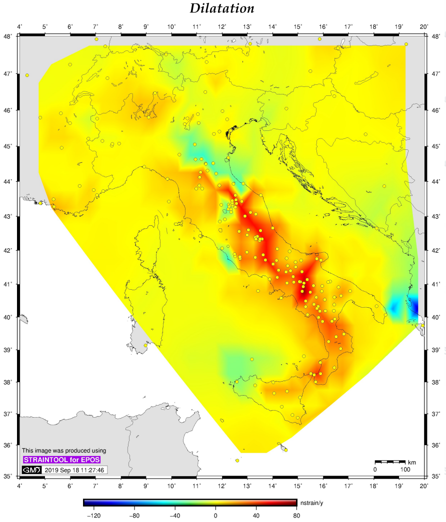
D'Agostino (2014)



Strain Analysis

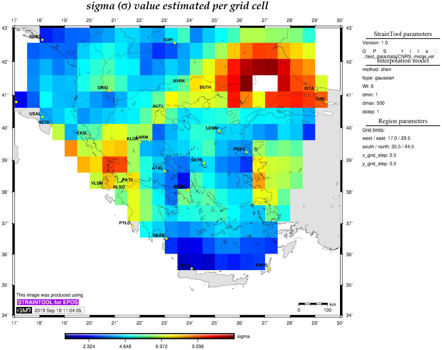
Italy region

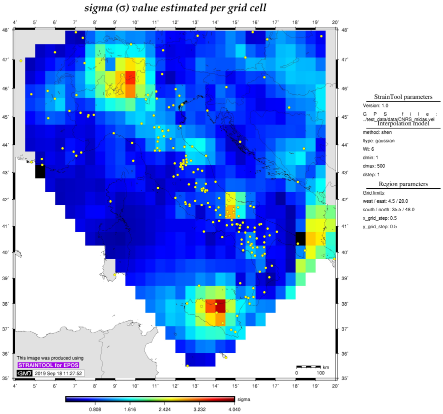




Statistics

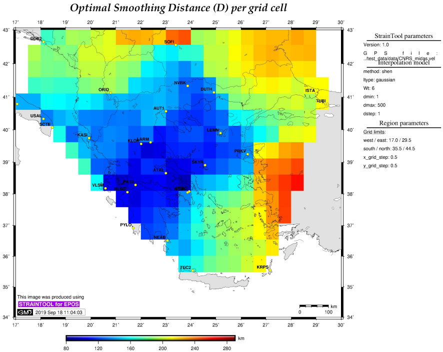
Sigma value

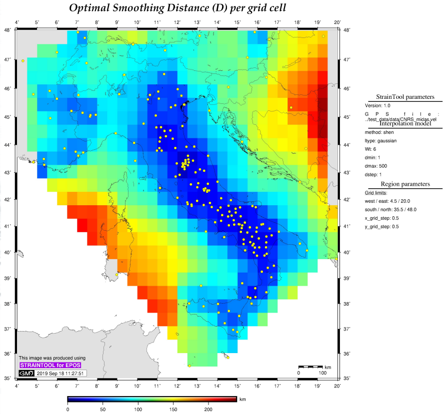




Statistics

D optimal smoothing parameter





Conclusions

- 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.
- Our results reproduce the gross features of tectonic deformation in both Italy and Greece, such as:
  - NE-SW extension across the Apennines.
  - N-S extension in Central Greece.
  - NE-SW compression acrossthe area of Albania's shoreline.
- 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.

Thank you for your attention!

References I

Anastasiou, D., X. Papanikolaou, A. Ganas, and D. Paradissis (July 2019). *StrainTool: A software package to estimate strain tensor parameters.* DOI: 10.5281/zenodo.1297565 (cit. on p. 4).

Cardozo, N. and R.W. Allmendinger (2008). "SSPX: A program to compute strain from displacement / velocity data". In: *Computer and Geoscience* 35.1, pp. 1343–1357 (cit. on p. 14).

D'Agostino, N. (2014). "Complete seismic release of tectonic strain and earthquake recurrence in the Apennines (Italy)". In: *Geophysical Research Letters* 41.4, pp. 1155–1162. DOI: 10.1002/2014GL059230 (cit. on p. 23).

Floyd, M. A. et al. (2010). "A new velocity field for Greece: Implications for the kinematics and dynamics of the Aegean". In: *Journal of Geophysical Research: Solid Earth* 115.B10, n/a–n/a. ISSN: 2156-2202. DOI: 10.1029/2009JB007040 (cit. on p. 20).

Masson, F., M. Lehujeur, Y. Ziegler, and C. Doubre (Jan. 2014). "Strain rate tensor in Iran from a new GPS velocity field". In: *Geophysical Journal International* 197.1, pp. 10–21. ISSN: 0956-540X. DOI: 10.1093/gji/ggt509 (cit. on p. 14).

Shen, Zheng-Kang, David D. Jackson, and Bob X. Ge (1996). "Crustal deformation across and beyond the Los Angeles basin from geodetic measurements". In: *Journal of Geophysical Research: Solid Earth* 101.B12, pp. 27957–27980. DOI: 10.1029/96JB02544 (cit. on p. 14).

Shen, Zheng-Kang, Min Wang, Yuehua Zeng, and Fan Wang (July 2015). "Optimal Interpolation of Spatially Discretized Geodetic Data". In: *Bulletin of the Seismological Society of America* 105.4, pp. 2117–2127. ISSN: 0037-1106. DOI: 10.1785/0120140247 (cit. on pp. 8, 11).

References II

Veis, G., H. Billiris, B. Nakos, and D. Paradissis (1992). "Tectonic strain in Greece from geodetic measurements". In: *C.R.Acad.Sci.Athens* 67, pp. 129–166 (cit. on p. 12).