

GeophysicalModelGenerator.jl: A Julia package to visualise geoscientific data and create numerical model setups

Boris J. P. Kaus ¹, Marcel Thielmann  ², Pascal Aellig  ¹, Albert de Montserrat  ³, Luca de Siena  ^{4,1}, Jacob Frasukiewicz ¹, Lukas Fuchs ⁵, Andrea Piccolo ², Hendrik Ranocha ¹, Nicolas Riel ¹, Christian Schuler ¹, Arne Spang ², and Tatjana Weiler ²

¹ Johannes Gutenberg-University Mainz, Germany ² University of Bayreuth, Germany ³ ETH Zürich, Switzerland ⁴ Alma Mater Studiorum Bologna University, Italy ⁵ Goethe University Frankfurt, Germany

DOI: [10.21105/joss.06763](https://doi.org/10.21105/joss.06763)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: [Rachel Wegener](#) 

Reviewers:

- [@anowacki](#)
- [@Alexander-Barth](#)

Submitted: 15 March 2024

Published: 31 October 2024

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

Summary

Geoscientific data exists in a wide variety of formats. Yet, to make a consistent interpretation of a certain region, it is often helpful to jointly visualise all this data using the same coordinates and compare, for example, seismic tomography, surface geology, Moho (crust-mantle transition) depth, Earthquake locations, and GPS surface velocities. If one wishes to create mechanical or thermo-mechanical numerical models of the region, creating an input setup that honors these constraints is crucial. And since most numerical codes work in Cartesian boxes, it is helpful to have tools to project the data from geographic to Cartesian coordinates.

A significant challenge in doing this is that there is no standard format for geoscientific data. Seismic tomography, for example, may come in the form of ASCII data with lon/lat/depth axes or as NetCDF files, with the ordering of the data typically differing from one dataset to the other. In ideal cases, geological surfaces may be provided as GeoTIFF images. In many cases, however, the underlying data discussed in publications are not available in digital format and are only shown as figures in the paper. It is nevertheless still helpful to visualise these in 3D in the correct coordinates, along with more recent, digitally available datasets.

The aim of the GeophysicalModelGenerator.jl package is therefore twofold:

- 1) Simplify collecting and visualising a wide variety of geoscientific data that is provided as point (e.g., earthquake locations), surface (e.g., topography) or volumetric data (e.g., seismic tomography).
- 2) Create input setups for 2D or 3D numerical models.

Statement of need

GeophysicalModelGenerator.jl is a Julia ([Bezanson et al., 2017](#)) package that helps collect and visualise a wide variety of geophysical and geoscientific data coherently. It also simplifies the process of generating 2D or 3D models that can, for example, be used as input models in geodynamic simulations. It provides functions that transfer data from one format to the other, or project them from geographic Longitude/Latitude/Depth or UTM coordinates to Cartesian coordinates (kilometers). It allows performing tasks such as creating cross-sections through volumetric data, importing screenshots from published papers, downloading digital elevation

data and saving the resulting data in VTK format, which can, for instance, be visualised with open source tools such as [Paraview](#).

Most geoscientists tend to have their own Python/MATLAB/Bash visualisation, and many perform part of this job on a daily basis. Yet, having all functionality in one place in an easy-to-use package makes this more extendable and will likely facilitate sharing data along with their interpretations.

Related software packages

Perhaps the most widely used package in geophysics to create figures or maps is the Generic Mapping Tools package ([GMT](#)), which also provides a Julia interface [GMT.jl](#) ([Wessel et al., 2019](#)). It mostly focuses on generating maps and postscript/pdf images and is therefore not ideally suited for interactive 3D data visualisation, or to generate input models for numerical codes.

The [Geodynamic World Builder](#) is a C++ library to create model setups ([Fraters et al., 2019](#)). The focus is on generating input models for geodynamic simulations, such as subduction zones and related thermal structures. It has C and Fortran wrappers and can thus be embedded in geodynamic codes. Users of the Geodynamic World Builder have to generate JSON files to define the model geometry, which is less interactive than by using the Julia REPL. There is no straightforward way to integrate existing geophysical/geological data in the workflow and compare model results with them.

[geomIO](#) is a MATLAB-based toolbox that allows creating geodynamic input setups by drawing several cross-sections using vector software such as Inkscape, which is put together into 3D volumes ([Bauville & Baumann, 2019](#)). While it does allow the creation of sophisticated setups, data can only be taken into account by adding them as screenshots to Inkscape while drawing cross-sections. Its reliance on commercial software may be problematic for some users.

[GemPy](#) is a Python-based, open-source geomodeling library that can construct 3D geological models of folded structures, fault networks and unconformities while taking uncertainties into account ([De La Varga et al., 2019](#)). It focuses on creating geometric models with uncertainties rather than on integrating a wide variety of geoscientific datasets.

There are also a number of commercial software solutions:

- [Petrel subsurface software](#) (by Schlumberger), which is mostly used by the hydrocarbon industry and is particularly powerful in integrating seismic reflection and well-data.
- [GOCAD Mining Suite](#) (by MiraGeoscience) helps to generate geometric models of the subsurface in the vicinity of mines based on sparse geological measurements and drill hole data.
- [GeoModeller](#) (by Intrepid Geophysics) creates surface-near geometric geological models by implicit modelling of surface measurements while taking geophysical constraints into account.

In all cases, the commercial license fees are far beyond what most researchers can afford, even if reduced license fees are often available for academia. The closed-source nature of the software packages makes them non-extendable by the community.

The `GeophysicalModelGenerator.jl` package is already used to generate input models for the geodynamic codes [LaMEM](#) ([Kaus et al., 2016](#)), [JustRelax.jl](#), and [MagmaThermokinematics.jl](#). It is also used in a number of short courses and lectures at the Universities of Mainz, Heidelberg, Jena and Bologna, in hands-on tutorials during various workshops and in a number of recent publications to highlight 3D ([Gabrielli et al., 2023](#); [Napolitano et al., 2023](#)) and 4D ([De Siena et al., 2024](#)) geophysical and geological data.

Basic usage

The core of the package consists of the GeoData, UTMDATA, ParaviewData, and CartData structures which hold the 3D data along with coordinates (and potentially metadata) information. GeophysicalModelGenerator.jl can be installed using the built-in Julia package manager:

```
julia> ]
(@v1.10) pkg> add GeophysicalModelGenerator
```

which comes with a test-suite:

```
(@v1.10) pkg> test GeophysicalModelGenerator
```

and can be loaded with:

```
julia> using GeophysicalModelGenerator
```

As a first example, we will download a 3D seismic tomography dataset for the Alpine region (from (Paffrath et al., 2021)):

```
julia> Tomo_Alps_full = load_GMG(
    "https://zenodo.org/records/10738510/files/Paffrath_2021_SE_Pwave.jld2?download=1");
```

We can download the topography of the Alpine region with:

```
julia> Topo_Alps = load_GMG(
    "https://zenodo.org/records/10738510/files/AlpsTopo.jld2?download=1");
```

The seismic data covers a much wider region than the Alps itself, but in much of that region there is poor data coverage. We can, therefore, extract a part of the data that has coverage:

```
julia> Tomo_Alps = extract_subvolume(Tomo_Alps_full, Lon_level=(4,20),
    Lat_level=(36,50), Depth_level=(-600,-10));
```

At this stage, we can save the data to VTK format:

```
julia> write_paraview(Tomo_Alps, "Tomo_Alps");
julia> write_paraview(Topo_Alps, "Topo_Alps");
```

and open it with Paraview (see [Figure 1a](#)). We can create vertical and horizontal cross-sections through the data with:

```
julia> Cross_200km = cross_section(Tomo_Alps, Depth_level=-200, Interpolate=true);
julia> Cross_vert = cross_section(Tomo_Alps, Start=(5,47), End=(15,44));
julia> write_paraview(Cross_vert, "Cross_vert");
julia> write_paraview(Cross_200km, "Cross_200km");
```

and visualise them along with the volumetric data ([Figure 1a](#)).

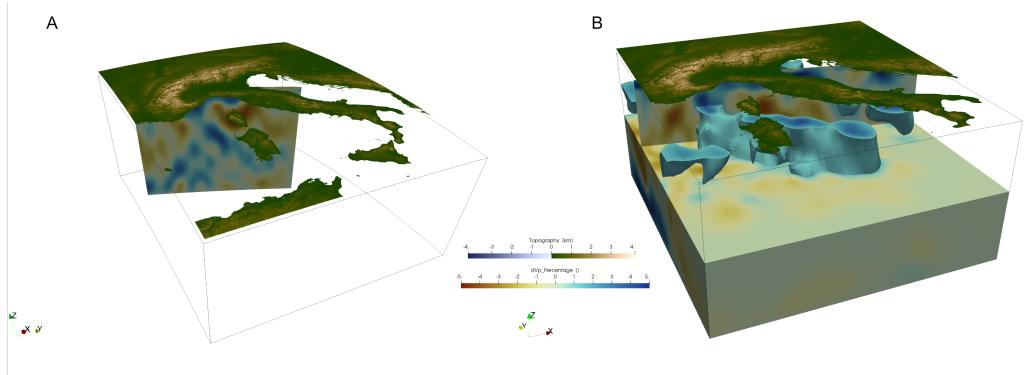


Figure 1: Example of visualising 3D seismic data of the Alps, using a) geographic coordinates (GeoData) or b) Cartesian coordinates (CartData) projected from geographic coordinates. The topography and several slices through the 3D seismic tomography P-wave model of (Paffrath et al., 2021) are shown.

One complication with geographic data is that Paraview does not have native support for geographic coordinates; accordingly, it is not always straightforward to use the built-in tools, for example, to create slices through the data. In addition, many numerical models work in (orthogonal) Cartesian rather than in spherical coordinates, which appears to be a good first-order approximation for many geodynamic applications (Macherel et al., 2024).

GeophysicalModelGenerator.jl includes tools to transfer the data from geographic to Cartesian coordinates, which requires defining a projection point along which the projection is performed:

```
julia> proj = ProjectionPoint(Lon=12.0,Lat =43)
ProjectionPoint(43.0, 12.0, 255466.98055255096, 4.765182932801006e6, 33, true)
```

We can now project the topography with:

```
julia> Topo_cart = convert2CartData(Topo_Alps, proj);
```

which returns a CartData (Cartesian data) structure. The disadvantage of doing this projection is that the resulting Cartesian grid is no longer strictly orthogonal, which is a problem for some Cartesian numerical models (e.g., those using a finite difference discretisation). We can project the data on an orthogonal grid as well, by first creating appropriately sized orthogonal grids for the tomography and topography:

```
julia> Tomo_rect = CartData(xyz_grid(-550.0:10:600, -500.0:10:700, -600.0:5:-17));
julia> Topo_rect = CartData(xyz_grid(-550.0:1:600, -500.0:1:700, 0));
```

Next, we can project the data to the orthogonal grids with:

```
julia> Topo_rect = project_CartData(Topo_rect, Topo_Alps, proj);
julia> Tomo_rect = project_CartData(Tomo_rect, Tomo_Alps, proj);
julia> write_paraview(Tomo_rect,"Tomo_rect");
julia> write_paraview(Topo_rect,"Topo_rect");
```

We can now use the built-in tools of Paraview to visualise the data (see Figure 1 b), and use this as inspiration to create an initial numerical model setup. It is also possible to interpolate other seismic tomography datasets to the same grid and subsequently compute a *votemap* to count how many tomographic models contain a specific seismic anomaly (see, e.g., Shephard et al., 2017).

Examples of usage

`GeophysicalModelGenerator.jl` comes with build-in (CI/CD) tests and [tutorials](#) that explain the most important use cases, from importing data to generating input model setups for numerical simulations. In the following, we present a number of examples that illustrate various aspects of the package. Many additional tutorials are available in the online documentation.

Visualise data from the Alps

The European Alps are among the best-studied mountain belts on the planet and have therefore been the focus of numerous geological and geophysical studies. Different seismic tomography models have been published (using different parameterisations and datasets), but those do not necessarily agree with each other.

In [Tutorial_AlpineData.jl](#), users learn how to load the topography of the region, import Moho data, load and visualise GPS vectors, import and plot earthquake locations, along with cross-sections through the model ([Figure 2](#)).

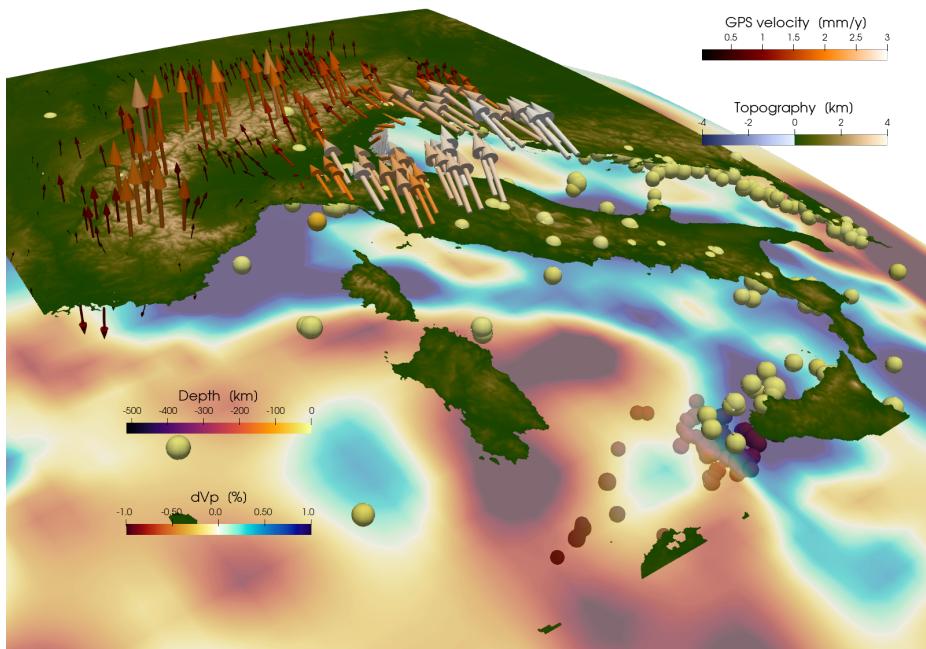


Figure 2: Example of combined data of the Alps, which shows the GPS surface velocity (arrows), topography, earthquake locations (colored dots) and cross-sections through a recent anisotropic P-wave tomography model by ([Rappisi et al., 2022](#)).

La Palma volcanic eruption

The 2019 Cumbre Vieja eruption in La Palma, Canary Islands, was accompanied by seismic activity. In [Tutorial_LaPalma.jl](#), users learn to generate a Cartesian block model of the island, import seismicity and use that to generate a 3D volumetric seismic activity map ([Figure 3](#)).

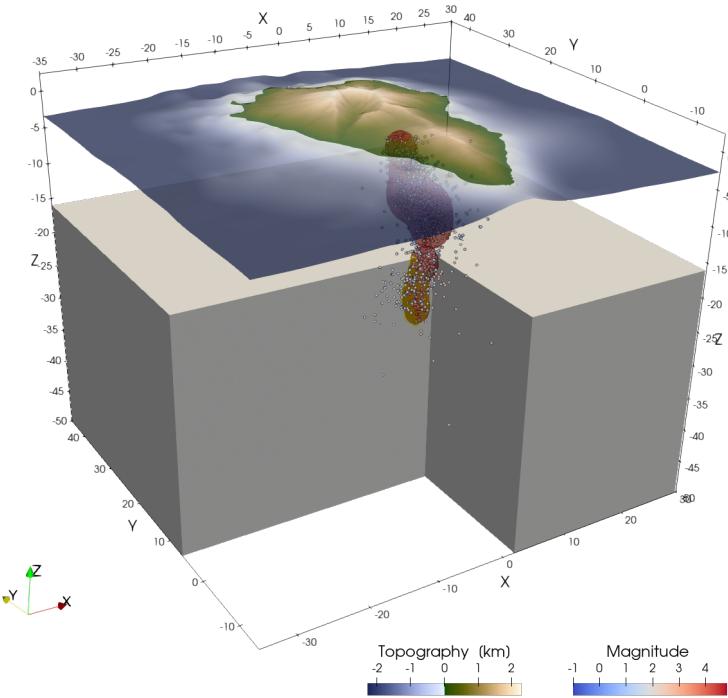


Figure 3: Example of a model of La Palma that shows seismicity during the 2019 Cumbre Viejo eruption.

Jura mountains

The Jura mountains are a small-scale fold and thrust belt located in Switzerland and France. Thanks to seismic cross-sections and boreholes, a lot of information is available about its structure at depth. This information was used to generate extensive 3D models of the subsurface, including thickness maps of various geological units, generate a new geological map of the region, and create balanced reconstructions (Schori, 2021).

In [Tutorial_Jura.jl](#), users learn how to drape the geological map over the topography, import surfaces from GeoTIFF images (such as basement topography), and include screenshots from geological cross-sections. The data is rotated and transferred to Cartesian coordinates such that we obtain a 3D block model that is perpendicular to the strike of the mountain range ([Figure 4](#)).

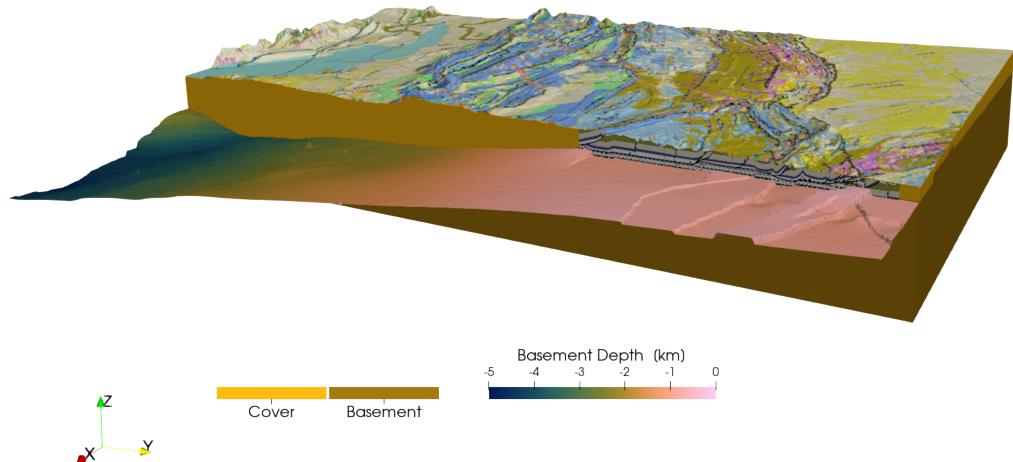


Figure 4: Example of creating a 3D Cartesian block model that runs perpendicular to the Jura mountains, combining surface geology, with screenshots from interpreted cross-sections (in the center right), and digital data of the the basement topography (using data of [Schori, 2021](#)).

Slab model setup

In `Tutorial_NumericalModel_3D.jl`, users learn how to generate a 3D geodynamic model setup with subducting slabs, a mid-oceanic ridge and an overriding cratonic lithosphere. The thermal structure of the oceanic slab increases away from the ridge until the trench, following a halfspace cooling analytical solution. In contrast, the thermal structure of the subducted part of the slab is based on an analytical solution that takes heating from the surrounding hot mantle into account (after [McKenzie, 1969](#)). Between the mantle and the trench, the slab uses a mixture of these two thermal models. A weak zone is added above the slab (to facilitate subduction in numerical models). A sedimentary wedge is situated at the continental margin, and a grid-like pattern is put on top of the oceanic slab to simplify tracking deformation throughout the simulation ([Figure 5](#)).

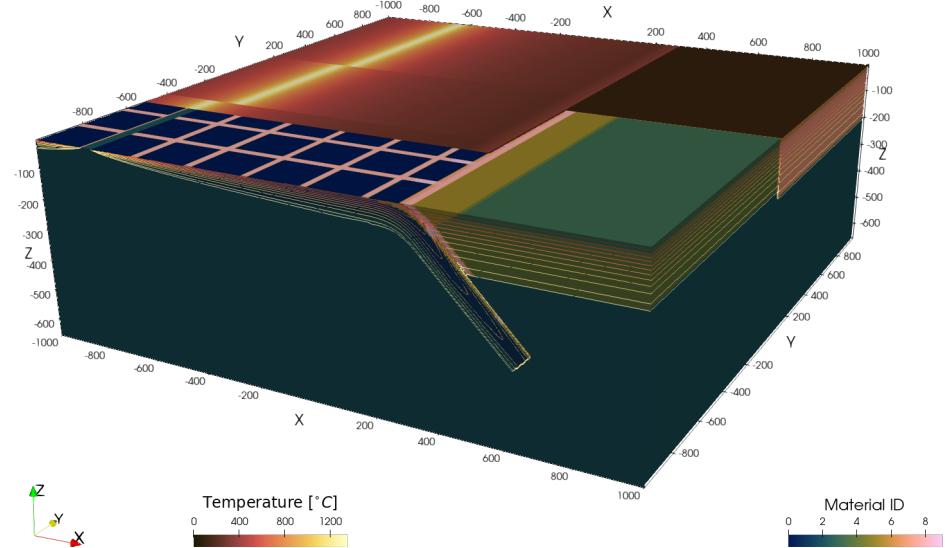


Figure 5: Example of a geodynamic setup of a subducting oceanic plate beneath a continental lithosphere with a smoothly bending slab.

Acknowledgements

We acknowledge funding by 1) the European Research Council as part of Consolidator Grant 771143 (MAGMA), 2) the German Research Foundation through DFG grants KA3367/10-1, TH2076/7-1 (both part of the SPP 2017 4D-MB project) and Emmy Noether grant TH2076/8-1, 3) the German Ministry of Science and Education (BMBF) through project DEGREE, and 4) by the CHEESE-2p Center of Excellence, co-funded by both EuroHPC-JU and the BMBF.

References

- Bauville, A., & Baumann, T. S. (2019). geomIO: An open-source MATLAB toolbox to create the initial configuration of 2-D/3-D thermo-mechanical simulations from 2-D vector drawings. *Geochemistry, Geophysics, Geosystems*, 20(3), 1665–1675. <https://doi.org/10.1029/2018GC008057>
- Bezanson, J., Edelman, A., Karpinski, S., & Shah, V. B. (2017). Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1), 65–98. <https://doi.org/10.1137/141000671>
- De La Varga, M., Schaaf, A., & Wellmann, F. (2019). GemPy 1.0: Open-source stochastic geological modeling and inversion. *Geoscientific Model Development*, 12(1), 1–32. <https://doi.org/10.5194/gmd-12-1-2019>
- De Siena, L., Amoruso, A., Petrosino, S., & Crescentini, L. (2024). Geophysical responses to an environmentally-boosted volcanic unrest. *Geophysical Research Letters*, 51(5), e2023GL104895. <https://doi.org/10.1029/2023GL104895>
- Fraters, M., Thieulot, C., Berg, A. van den, & Spakman, W. (2019). The Geodynamic World Builder: A solution for complex initial conditions in numerical modeling. *Solid Earth*, 10(5),

1785–1807. <https://doi.org/10.5194/se-10-1785-2019>

- Gabrielli, S., Akinci, A., De Siena, L., Del Pezzo, E., Buttinelli, M., Maesano, F. E., & Maffucci, R. (2023). Scattering attenuation images of the control of thrusts and fluid overpressure on the 2016–2017 central italy seismic sequence. *Geophysical Research Letters*, 50(8), e2023GL103132. <https://doi.org/10.1029/2023GL103132>
- Kaus, B. J., Popov, A. A., Baumann, T., Püsök, A., Bauville, A., Fernandez, N., & Collignon, M. (2016). *Forward and inverse modelling of lithospheric deformation on geological timescales*. *NIC Series*, 48, 299–306. ISBN: 978-3-95806-109-5
- Macherel, E., Räss, L., & Schmalholz, S. M. (2024). 3D stresses and velocities caused by continental plateaus: Scaling analysis and numerical calculations with application to the Tibetan plateau. *Geochemistry, Geophysics, Geosystems*, 25(3), e2023GC011356. <https://doi.org/10.1029/2023GC011356>
- McKenzie, D. P. (1969). Speculations on the consequences and causes of plate motions. *Geophysical Journal International*, 18(1), 1–32. <https://doi.org/10.1111/j.1365-246X.1969.tb00259.x>
- Napolitano, F., Gabrielli, S., De Siena, L., Amoroso, O., & Capuano, P. (2023). Imaging overpressurised fracture networks and geological barriers hindering fluid migrations across a slow-deformation seismic gap. *Scientific Reports*, 13(1), 19680. <https://doi.org/10.1038/s41598-023-47104-w>
- Paffrath, M., Friederich, W., Schmid, S. M., Handy, M. R., AlpArray, the, & Group, A.-S. D. W. (2021). Imaging structure and geometry of slabs in the greater Alpine area – a P-wave travel-time tomography using AlpArray Seismic Network data. *Solid Earth*, 12(11), 2671–2702. <https://doi.org/10.5194/se-12-2671-2021>
- Rappisi, F., VanderBeek, B. P., Faccenda, M., Morelli, A., & Molinari, I. (2022). Slab geometry and upper mantle flow patterns in the Central Mediterranean from 3D anisotropic P-wave tomography. *Journal of Geophysical Research: Solid Earth*, 127(5), e2021JB023488. <https://doi.org/10.1029/2021JB023488>
- Schori, M. (2021). *The Development of the Jura Fold-and-Thrust Belt: Pre-existing Basement Structures and the Formation of Ramps* [PhD, University of Fribourg (Switzerland)]. <https://doi.org/10.51363/unifr.sth.2022.001>
- Shephard, G. E., Matthews, K. J., Hosseini, K., & Domeier, M. (2017). On the consistency of seismically imaged lower mantle slabs. *Scientific Reports*, 7(1), 10976. <https://doi.org/10.1038/s41598-017-11039-w>
- Wessel, P., Luis, J. F., Uieda, L., Scharroo, R., Wobbe, F., Smith, W. H. F., & Tian, D. (2019). The Generic Mapping Tools Version 6. *Geochemistry, Geophysics, Geosystems*, 20(11), 5556–5564. <https://doi.org/10.1029/2019GC008515>