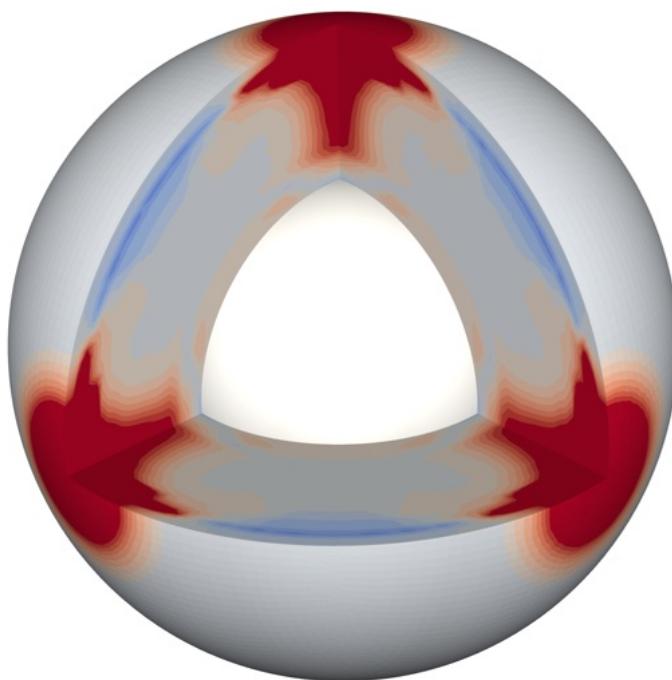
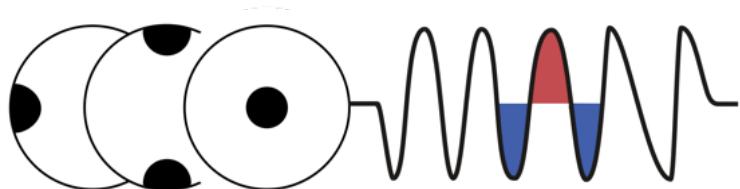


Ecoman

Exploring the COnsequences of Mechanical
Anisotropy in the maNtle



User Manual - Version 2.0

August 24, 2024

Manuele Faccenda (manuele.faccenda@unipd.it)

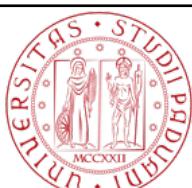
Brandon P. VanderBeek (brandongpaul.vanderbeek@unipd.it)

Albert de Montserrat Navarro

Jianfeng Yang



European Research Council
Established by the European Commission



Contents

1	Introduction	1
1.1	General information about ECOMAN	1
1.2	Why ECOMAN?	2
1.3	Referencing and useful links	4
1.4	Author contribution	5
1.5	Acknowledgements	5
2	Installation and compilation	6
2.1	Installation	6
2.2	Software dependencies	7
3	D-REX_S: single mantle aggregate LPO fabrics with D-Rex	8
3.1	Parameter input file	9
3.2	Fabric visualization	11
4	D-REX_M: multiple mantle aggregates LPO fabrics with D-Rex	18
4.1	Software structure and modus operandi	18
4.2	Coordinate systems	19
4.3	Input vtp*.h5 files	22
4.3.1	How saving information from your geodynamic model?	22
4.3.2	How often saving information from your geodynamic model?	23
4.3.3	Which is the model time interval over which D-REX_M should run?	23
4.4	Output Cijkl*.h5 files	24
4.5	Allocated memory	24
4.6	Parameter input file	25
4.7	Parallelization and its efficiency	29
5	EXEV: extrinsic elastic and viscous anisotropy	31
5.1	Effective Medium Theories	31

5.1.1	Smoothed Transversely Isotropic Long-Wavelength Equivalent (STILWE)	31
5.1.2	Differential Effective Medium (DEM) Theory	32
5.2	Extrinsic elastic anisotropy	33
5.2.1	STILWE modeling (active when spomod = 1)	34
5.2.2	DEM modelling (active when spomod > 1)	35
5.3	Extrinsic viscous anisotropy	39
5.3.1	Generate the viscous tensor database	39
5.3.2	Coupled mechanical simulations	42
5.3.3	Uncoupled mechanical simulations: visualizing the viscous tensors with VIZVISC	43
6	VIZTOMO: visualization of the D-REX_M output	48
6.1	Parameter input file	48
6.2	SPECFEM3D simulations within the domain built with VIZTOMO	53
7	SKS-SPLIT: synthetic SKS splitting	56
7.1	Parameter input file(s)	57
7.2	Compute SKS splitting	58
7.3	SKS splitting visualization	58
8	PSI_D: Platform for Seismic Imaging (Deterministic)	60
8.1	Overview	60
8.2	Installation	61
8.3	Coordinate system	61
8.4	Forward Modelling	62
8.4.1	Parameter file variables for forward modelling	62
8.5	Inverse Modelling	64
8.5.1	Parameter file variables for inverse modelling	65
8.5.2	Inversion Output	69
8.6	Examples	69
8.6.1	Sinking Slab	70
8.6.2	Subducting Plate	72
9	COOKBOOKS: some useful examples	75
9.1	Steady-state convection in 2D cartesian coordinates	76
9.2	Steady-state subduction in 2D polar coordinates	77
9.3	Steady-state global scale convection in 2D polar coordinates	80
9.4	Sinking slab in 3D cartesian coordinates	82
9.5	Sinking slab in 3D spherical coordinates	83

9.6 Steady-state global scale convection in 3D spherical coordinates	84
A1 Appendix	90

Font styles

Throughout the manual, we have done our best to use different font styles indicating:

- text
- **SOFTWARE**
- files, folders, URL links, commands
- **variables present within the codes**
- **scalar, vectors, 2nd and 4th order tensors**

Nomenclature

Here we provide explanations for the different symbols, acronyms and variables found in this guide.

Abbreviation	Description	Units
UM	Upper mantle	
MTZ	Mantle Transition zone	
UTZ	Upper Transition zone	
LTZ	Lower Transition zone	
LM	Lower Mantle	
LPO	Lattice Preferred Orientation	
SPO	Shape Preferred Orientation	
fse_{max}, fse_{min}	maximum, minimum semiaxis of the FSE	1
$\ln_{fse}, \ln_{fse \text{ NODE}}$	natural logarithm of the ratio between the major and minor semi-axes of the FSE computed for an aggregate or a node of the Eulerian grid	1
Vp_{max}, Vp_{min}	maximum, minimum P-wave velocity	m/s
Vp_{anis}	P-wave anisotropy	%
Vs_1, Vs_2	fast and slow S-wave components after splitting	m/s
dVs_{MAX}	maximum velocity difference between the fast and slow S-wave components	%
V_{SH}, V_{SV}	azimuthally averaged velocity of the S-wave polarized in the horizontal and vertical planes	m/s
ξ	radial anisotropy	1,%
G	azimuthal anisotropy	1,%
ψ	porosity	1
P	Pressure	Pa
Tk	Temperature	K
Fd	Fraction of dislocation creep deformation	1
\mathbf{V}, V_i	Velocity vector and its components	m/s
\mathbf{D}, D_{ij}	2 nd order velocity gradient tensor and its components	s^{-1}
\mathbf{F}, F_{ij}	2 nd order deformation gradient tensor and its components	1
FSE	2 nd order left stretch tensor, whose eigenvalues and eigenvectors describe the Finite Strain Ellipsoid	1
$\mathbf{C}, C_{\alpha\beta}$	4 th order elastic tensor and its components in Voigt notation	GPa
$\boldsymbol{\eta}, \eta_{\alpha\beta}$	4 th order normalized viscous tensor and its components in Voigt notation	1

The units on the rightmost column define variables when dimensional.

1 Introduction

1.1 General information about ECOMAN

ECOMAN is a software package for modelling strain-/stress-induced rock fabrics and testing the effects of the resulting elastic and viscous anisotropy on seismic imaging and mantle convection patterns.

ECOMAN includes programs for geodynamic modelling (**ECOMAN-geodynamics**) and seismological modelling (**ECOMAN-seismology**) that are mostly written in Fortran and Julia language, and where most of the routines are parallelized with a hybrid MPI and OpenMP scheme, providing strong scalability with increasing number of cores.

ECOMAN-geodynamics includes software for (Fig. 1.1):

1. Rock fabrics and mechanical properties simulations:

- **D-REX_S**: Single Aggregate Fabric. This program builds on the original D-REX and estimates the strain-induced LPO of a single mantle aggregate. It includes the necessary **MTEX Toolbox** libraries to generate pole figures of the (1) LPO and (2) isotropic/anisotropic elastic properties.
- **D-REX_M**: Multiple Aggregates Fabric. This program builds on the original D-REX and computes the strain-induced LPO of multiple mantle aggregates and output their (1) deformation gradient tensor \mathbf{F} and (2) elastic tensor \mathbf{C} as a function of the crystals orientation, volume fraction and of the local P-T conditions.
- **EXEV**: this program computes the EXtrinsic Elastic and Viscous anisotropy using Effective Medium Theories (STILWE [Backus, 1962] and DEM (e.g., [Mainprice, 1997])). It includes a parametrization of the viscous tensor evolution as a function of the deformation for 2 phases (matrix-inclusion) systems to be used in large-scale mantle convection simulations [de Montserrat and Faccenda, 2021].

2. Visualisation of the mechanical properties and data formatting in preparation for seismological synthetics:

- **VIZTOMO**: this program processes the **D-REX_M** output for (1) visualization of the aggregates elastic and deformational history (FSE) properties, and (2) preparing the initial setup for seismic wave propagation simulations (e.g., **SPECFEM3D,PSI_D**).
- **VIZVISC**: this program processes the **D-REX_M** output for visualization of the aggregates viscous anisotropy and deformational history (FSE) properties.

ECOMAN-seismology includes software for seismic forward and inverse modelling:

- **SKS-SPLIT**: this program estimates the SKS splitting at a grid of virtual seismic stations placed at the top of the **D-REX_M** model as a function of the back-azimuth using **FSTRACK** [Becker et al., 2006].
- **PSI_D**: this software performs isotropic and anisotropic P- and S- wave travel-time + S-wave splitting intensity inversions on either synthetic elastic models computed with **D-REX_M** and **VIZTOMO** or with real datasets, and utilizing the method of [VanderBeek and Faccenda, 2021, VanderBeek et al., 2023]. Other inversion methods to be added soon!.

1.2 Why ECOMAN?

The understanding of the Earth's interior is mainly based on seismological and geodynamic modelling, the first one providing important information about the present-day structure, while the second about the Earth's geological history. Seismological and geodynamical modelling are typically conducted separately, which creates difficulties in both interpreting seismic data in terms of geodynamic processes and in providing mantle structure constraints to geodynamic models. From this perspective, **ECOMAN** aims at linking seismology and geodynamics by providing a set of software to estimate the mantle elastic properties as a function of the deformation history and local P-T conditions. The modelled elastic properties can be used to generate tomographic models and run forward/inverse seismological problems (as it is done with real datasets), and the output compared with observations.

In addition, **ECOMAN** allows estimating the role of extrinsic viscous anisotropy in large-scale mantle convection models by providing a parametrization of the strain-induced evolution of the SPO-related fabrics in two-phase (matrix-inclusions) systems. The advantages and (real or potential) limitations of of **ECOMAN** are listed in section 3 of [Faccenda et al., 2024].

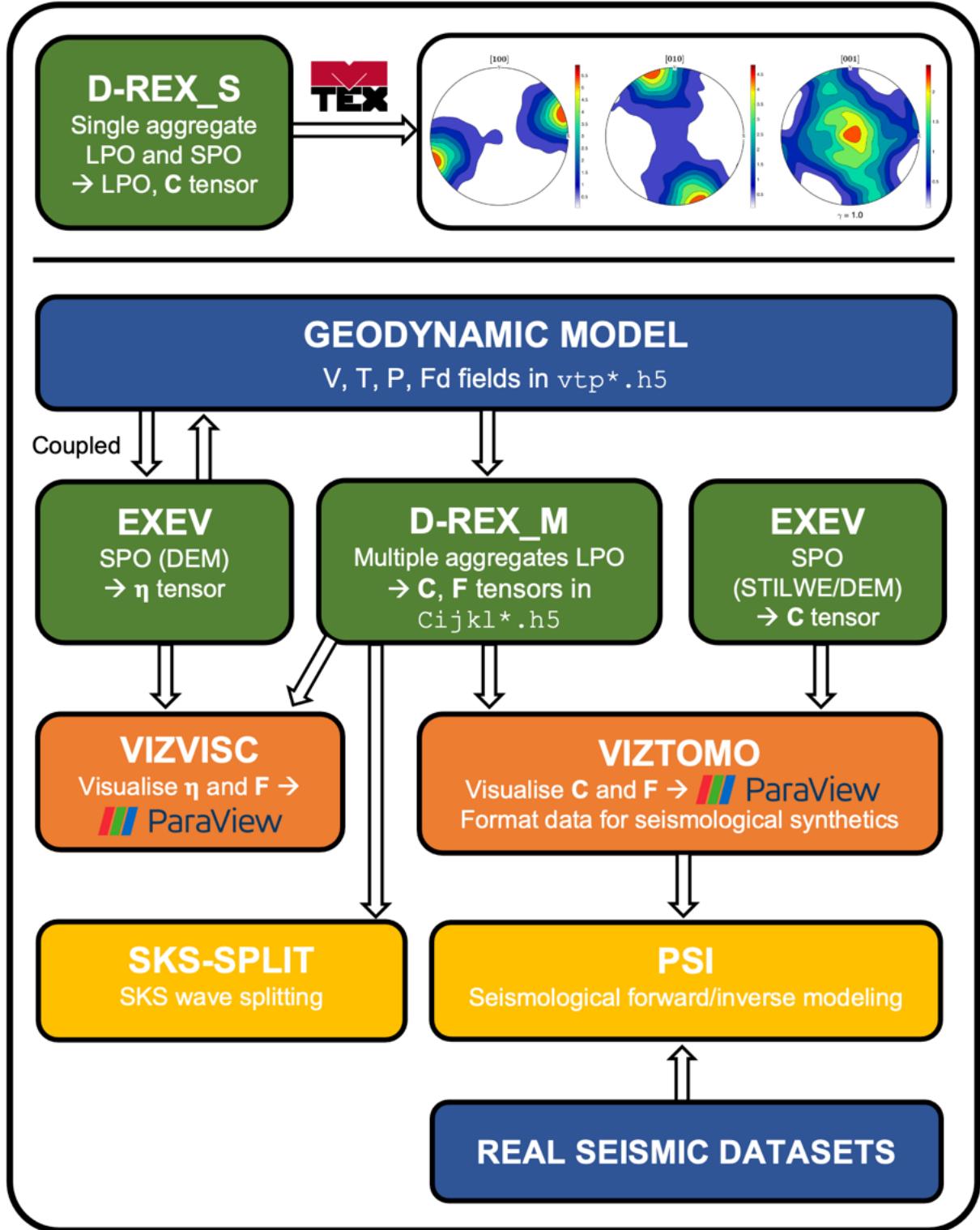


Figure 1.1: ECOMAN structure and flow chart. Coloured boxes denote programs that compute rock fabrics (green), post-process the elastic (C), viscous (η) and deformation gradient (F) tensors for visualisation and/or data formatting for seismological synthetics (orange), perform seismological forward/inverse modelling on synthetic or real datasets (yellow). Input data are from geodynamic modelling or real seismic datasets (blue). Visualisation of the mechanical properties and LPO can be done with the MTEX MATLAB toolbox for single crystal aggregates or the software Paraview for large-scale simulations.

1.3 Referencing and useful links

If you use **ECOMAN**, please cite one of the following paper that best fits your application.

1. [Faccenda et al., 2024]: main article about **ECOMAN**;
2. [Faccenda and Capitanio, 2013]: **D-REX_M** methodology explained in detail; applied to upper mantle fabrics.
3. [Faccenda, 2014]: Extension of **D-REX_M** to mantle fabrics of the MTZ and ULM; elastic tensors scaled by local P-T conditions.
4. [Bezada et al., 2016]: First example of synthetic P-wave travel-time tomography on mantle fabrics generated with **D-REX_M**.
5. [Hu et al., 2017]: **D-REX_M** goes spherical.
6. [Faccenda et al., 2019]: Evaluation of extrinsic elastic anisotropy in mantle aggregates.
7. [Ferreira et al., 2019], [Sturgeon et al., 2019]: Evaluation of lower mantle intrinsic and extrinsic fabrics to explain the observed seismic anisotropy.
8. [VanderBeek and Faccenda, 2021, VanderBeek et al., 2023]: Isotropic and anisotropic tomography with **PSI_D** on mantle fabrics generated with **D-REX_M**.
9. [de Montserrat and Faccenda, 2021]: submitted, viscous anisotropy due to extrinsic fabrics.

Useful references and links:

ECOMAN download: <https://github.com/ecoman-geos>

ECOMAN website: <https://newtonproject.geoscienze.unipd.it/ecoman/>

D-REX: <http://www.ipgp.fr/~kaminski/> [Kaminski et al., 2004]

FSTRACK: <https://github.com/thwbecker/fstrack> [Becker et al., 2006]

MTEX Toolbox: <https://mtex-toolbox.github.io/index.html> [Mainprice et al., 2011]

MMA_EoS: <https://bitbucket.org/chust/eos> [Chust et al., 2017]

1.4 Author contribution

Ecoman was developed by Manuele Faccenda (geodynamic modelling and visualization), Brandon P. VanderBeek (developer of **PSI_D** module and of body-wave anisotropic inversion method), with fundamental contributions by Albert de Montserrat (derived and tested the parametrization of the extrinsic viscous anisotropy) and Jianfeng Yang (testing and applications).

1.5 Acknowledgements

The development of **Ecoman** has been funded by the ERC StG 758199 NEWTON and the Università di Padova FACCPRAT12 granted to Manuele Faccenda.

2 Installation and compilation

Software included in the **ECOMAN** package run with the following compilers, libraries and software (see Table 2.1):

- Intel Fortran90 compiler
- Intel OpenMP, MPI compiler
- HDF5
- **ParaView** (version 5.4 or higher; <https://www.paraview.org/download/>)
- **MATLAB** (version R2018b or higher)
- **Julia** (tested version 1.7.x - 1.9.x)

Table 2.1: Required compilers and software for execution and visualization

Software	Compilers/Libraries/Software	Visualization Software
D-REX_S	Intel F90, HDF5	MATLAB (MTEX toolbox, version 5.* or higher)
D-REX_M	Intel F90/OpenMP/MPI, HDF5	-
VIZTOMO/VIZVISC	Intel F90/OpenMP, HDF5	ParaView
EXEV	Intel F90/OpenMP, HDF5	MATLAB
SKS-SPLIT	Intel F90, HDF5	MATLAB, ParaView
PSI_D	Julia	ParaView

2.1 Installation

1) Clone the source code(s):

D-REX_S, D-REX_M, EXEV, VIZTOMO, VIZVISC:

```
git clone https://github.com/ecoman-geos/ECOMAN2.0-geodynamics.git
```

SKS-SPLIT:

```
git clone https://github.com/ecoman-geos/ECOMAN2.0-seismology.SKS-SPLIT.git
```

PSI_D:

```
git clone https://github.com/ecoman-geos/ECOMAN2.0-seismology.PSI_D.git
```

- 2) To compile the F90 files, execute the `bash_compile` file present in each software directory as `./bash_compile`. The scripts should be compiled with the command `h5pfc` when the parallel HDF5 libraries are installed, or `h5fc` otherwise.
- 3) To run the F90 executables, see files `RUN` or `pbs_*`

2.2 Software dependencies

D-REX_M requires velocity field(s) and time constraints from a geodynamic model. To scale the elastic tensors by the local (P,T) conditions, pressure and temperature fields need to be additionally provided. Velocity, pressure and temperature field(s) and time are stored in `vtp*.h5` files.

Software **VIZTOMO**, **VIZVISC**, **SKS-SPLIT** and **PSI_D** require the pre-computation of an elastic and/or deformation history model, and thus should be run after the execution of **D-REX_M** (see Fig. 1.1). However, **PSI_D** can also be used with real seismic datasets

VIZVISC needs also the pre-computation of a database of viscous tensors through software **DEMviscous** included in directory `EXEV`.

3 D-REX_S: single mantle aggregate LPO fabrics with D-Rex

D-REX_S is a software that is designed for testing the evolution of mantle fabrics and related elastic properties of a single aggregate, as a function of the flow field, amount of strain, crystal plasticity, P-T conditions and SPO fabrics. It builds on the original **D-Rex** software [Kaminski et al., 2004].

Five different types of 2-phases mantle aggregates, hereafter named rocktypes, are available:

1 = Ol:Ens, for the upper mantle (UM: 0-410 km)

2 = Wd:Grt, for the upper transition zone (UTZ: 410-520 km)

3 = Rw:Grt, for the lower transition zone (LTZ: 520-660 km)

4 = Brd:MgO, for the lower mantle (LM: 660-2900 km)

5 = pPv:MgO, for the bottom of the lower mantle

The LPO evolution is computed for Olivine, Enstatite, Wadsleyite, Bridgmanite and post-Perovskite, while other major phases such as Garnet, Ringwoodite and MgO are considered to be isotropic and their distribution is set to random. Thus, no LPO is computed for rocktype **3**, and anisotropy in the LTZ arises only when SPO modeling is active.

The fabrics generated with **D-REX_S** can also be used to impose pre-existing microstructures on multiple crystal aggregates lying within a specific subdomain of large scale geodynamic models. The computed final LPO and F_{ij} are saved to `fossilfabric.h5`, and can be subsequently loaded during the **D-REX_M** simulations upon setting `fossilfabric > 0` in `drexm_input.dat` (see section 4.6).

COMPILE: `./bash_compile`

RUN: ./drexs drexs_input.dat

3.1 Parameter input file

The following information need to be set in the parameter input file `drexs_input.dat` (unless otherwise specified, the variable format is FORTRAN double precision):

A) Output file/directory name

- **output_name**: (String) if not existing, a new directory is created with output files saved each 0.1 strain increment. The output files are named as `[output_name] + [strain] + [.h5]` (example: if **output_name** = `UM`, **strain_max** = 1.0, the output files will be `UM_0.1.h5, UM_0.2.h5, ..., UM_1.0.h5`). When **sponod** > 0, an additional output file is generated at the end of the run named as `[output_name] + [SPO.h5]` (example: `UM_SPO.h5`).

B) Maximum strain

- **strain_max**

C) Velocity gradient tensor D_{ij}

- D_{11}, \dots, D_{32} : components of the velocity gradient tensor

D) LPO parameters:

- **size3**: (Integer) cubic root of total number of grains in the aggregate for each of the 2 mineral phases (e.g.: **size3** = 10 → 1000 crystals of phase 1 and 1000 crystals of phase 2).
- **rocktype**: (Integer) define the mantle aggregate (**1,2,3,4,5**)

For each of the 5 different types of aggregate¹:

- **xol**: volume fraction of Ol, Wd, Rw, Brd, pPv (in %)
- **stressexp**: stress exponent for non-Newtonian plasticity

¹Rw and Grt are assumed to be isotropic at MTZ depths, and therefore no LPO is computed for this aggregate. Consequently, only **xol** and **single_crystal_elastic_db** need to be set.

- **Mob**: efficiency of grain boundary migration
- **chi (χ)**: volume fraction threshold below which no dislocation creep occurs :e
- **lambda (λ)**: efficiency of grain nucleation
- **fractdislrock**: fraction of deformation accommodated by the anisotropic phase
- **tau1...tauN**: normalized CRSS of the anisotropic phase slip systems. For upper mantle aggregates (rocktype = 1), tau(1,1-4) are for olivine, while tau(1,5)=1 is for enstatite. See A1.1.
- **single_crystal_elastic_db**: (Integer) choose single crystal elastic tensor for phase 1 and 2 from those compiled in elastic_database.f90. See A1.2.

E) Operating modes:

- **ptmod**: (Integer)
 - 0** = use single crystal tensors at room P-T conditions
 - 1** = scale elastic moduli by local P-T conditions defined below
- **eosmod**: (Integer) set the domain lithology (**1**: Dunite; **2**: Harzburgite; **3**: Pyrolite; **4**: Basalt; **5**: Pyroxenite) for which the corresponding lookup tables of density, isotropic Vp and Vs computed with **MMA_EoS** are loaded (see Table A1.3). The isotropic Vp and Vs are used to compute the isotropic component of the elastic tensor scaled at the local P-T conditions.
- **pressure**: pressure (in GPa)
- **tkelv**: temperature (in Kelvin)
- **fractvoigt**: fraction of Voigt averaging scheme when calculating the aggregate elastic tensor in %. It varies from 0.0 (Reuss average) to 100.0 (Voigt average). When equal to 50.0, it yields the Hill average.
- **spomod**: (Integer) when > 0, computes extrinsic elastic anisotropy² saved in output file [**output_name**] + [SPO.h5]
 - 0** = no extrinsic elastic anisotropy
 - 1** = extrinsic elastic anisotropy due to grain-scale or rock-scale layering (STILWE model)

2 = extrinsic elastic anisotropy due to the presence of aligned inclusions (DEM model)

3 = extrinsic elastic anisotropy due to the presence of aligned inclusions (DEM model) superimposed over the LPO fabric.

3.2 Fabric visualization

Fabric visualization requires the installation of the MTEX toolbox (Mainprice et al. [2011]). A recently tested version is 5.10.2, but previous versions 5.* should work as well. Please refer to the MTEX download webpage (<https://mtex-toolbox.github.io/download>) for installation instructions.

The aggregate fabric can be visualized with the MATLAB script `/viz/read_Cij_LPO.m` that allows to plot pole figures of the LPO, Vp, Vs1 and dVs, for multiple output files, which then can be combined into animations, together with the possibility to display the evolving fabric strength (M-index, J-index), anisotropy and its components (obtained with tensor decomposition; Browaeys and Chevrot [2004]) as a function of strain.

The control parameters to be set in the first part of the script are (Fig. 3.1, Fig. 3.2, Fig. 3.3, Fig. 3.4, Fig. 3.5):

- **input_dir:** (String) path to D-REX_S output files directory.
- **output_dir:** (String) output directory where images and videos are saved.
- **fname0:** (String) first part of output file name (same as **output_name** in `drex_s_input.dat`).
- **min_strain, stp_strain, max_strain:** minimum, increment and maximum strain as indicated in the second part of the output file name to be visualized. When **min_strain** = **max_strain**, only a single output file is visualized.
- **plot_Cij_LPO:** (Boolean) when true, activates plotting pole figures for Vp, dVs and LPOs.
 - **plot_Voigt:** (Boolean) when true, plot Vp, Vs1, dVs due to aggregate LPO, Voigt average.
 - **plot_Reuss:** (Boolean) when true, plot Vp, Vs1, dVs due to aggregate LPO, Reuss average.
 - **plot_Mixed:** (Boolean) when true, plot Vp, Vs1, dVs due to aggregate LPO, mixed Voigt/Reuss average as specified by **fractvoigt** in `drex_s_input.dat`.

²The SPO modeling is explained in detail in section 5.2

- **plot_Phase1LPO**: (Boolean) when true, plot LPO pole figures of main anisotropic phase, together with a plot of the M-index and J-index.
- **plot_Phase2LPO**: (Boolean) when true, plot LPO pole figures of enstatite, together with a plot of the M-index and J-index (only for UM aggregates).
- **makevideo**: (Boolean) when true, generate animations of Vp, dVs and LPO pole figures evolution as specified by the 5 **plot_*** operational modes defined above.
- **plot_azirad**: (Boolean) when true, plot azimuthal and radial P and S anisotropy, and η parameter, as a function of strain for the tensors activated with **plot_Voigt**, **plot_Reuss**, **plot_Mixed**.
- **plot_dec**: (Boolean) when true, plot (i) the norm of the total anisotropy component of the tensor over the norm of the full tensor (in %), and (ii) the norm of each anisotropic component of the tensor normalized over the norm of the total anisotropic component (in %). These have been obtained by decomposition (Browaeys and Chevrot [2004] of the elastic tensor (Mixed average) and saved in file anisdec.h5].
- **plot_SPO**: (Boolean) when true, plot Vp, dVs due to SPO from the [**output_name**] + [SPO.h5] output file.
- **printmod**: (Boolean) when true, save images in *.png format.

```

9      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10     %%%%%%%%%%%%%% Set input parameters %%%%%%%%%%%%%%
11     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12
13     %Path to D-REX_S output files directory
14     input_dir = 'Atype_/';
15     %Output directory, where images and videos are saved
16     output_dir = input_dir;
17     %First part of the D-REX_S output file name (usually same as output_dir)
18     fname0 = 'Atype_';
19
20     %Min/Step/Max strain as filenumber of D-REX_S output files
21     min_strain = 0.1;
22     stp_strain = 0.1;
23     max_strain = 1.0;
24
25     %Choose what to plot/save (No = false; Yes = true)
26     plot_Cij_LPO = true; %Activate plotting tensors and LPO. Set to false if we
27
28     plot_Voigt = true; %Plot Vp, dVs due to aggregate LPO, Voigt average
29     plot_Reuss = false; %Plot Vp, dVs due to aggregate LPO, Reuss average
30     plot_Mixed = false; %Plot Vp, dVs due to aggregate LPO, mixed Voigt/Reuss ;
31     plot_Phase1LPO = false; %Plot LPO pole figures of main anisotropic phase
32     plot_Phase2LPO = false; %Plot LPO pole figures of enstatite (only for upper
33
34     makevideo = false; %Make .mp4 movie of the LPO evolution
35
36     plot_azirad = false; % Plot azimuthal, radial anisotropy as a function of :
37
38     plot_dec = false; % Plot components of elastic tensor as a function of strain
39
40     plot_SPO = false; %Plot Vp, dVs when an SPO model is superimposed on the
41
42     printmod = true; %Save the MatLab figures
43
44     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
45     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
46     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

Figure 3.1: Control parameters in `read_Cij_LPO.m` for visualizing the D-REX_S output. In this particular case: generate pole figures of Vp, Vs1 and dVs for the elastic tensor obtained with the Voigt averaging scheme.

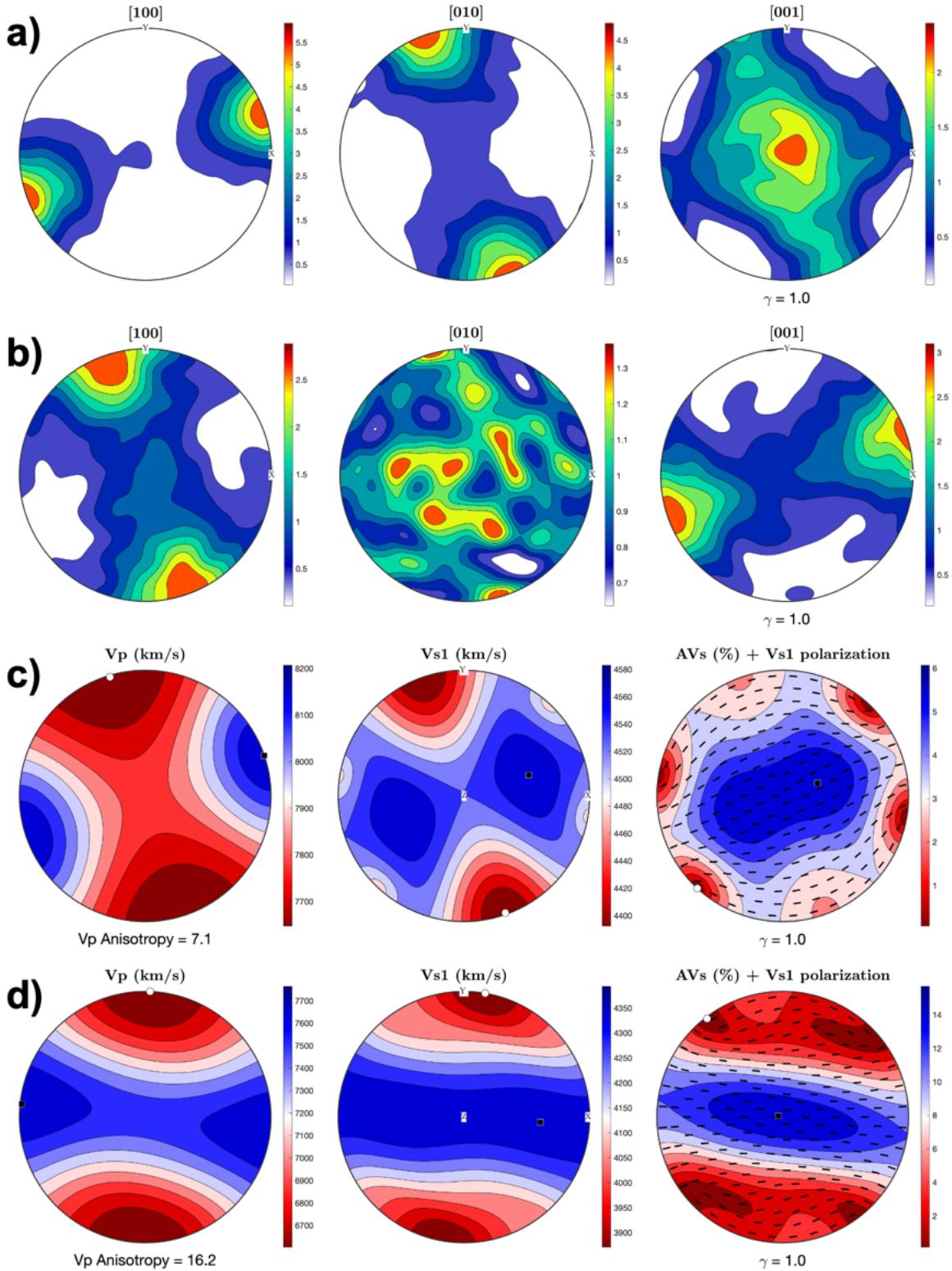


Figure 3.2: Example of UM fabric after right-lateral simple shear strain of 1.0 with $\text{Ol:Ens} = 70:30$, $\text{Mob} = 10$, $\lambda = 5$, $\chi = 0.3$. (A) Olivine (A-type) LPO; (B) Enstatite LPO; (C) Vp, Vs1, and dVs + polarization direction of the fast shear wave component for the Mixed (Hill average) elastic tensor resulting from the LPO fabrics in (A) and (B); (D) same as (C) but with superimposed an SPO fabric due to 5% of melt-filled inclusions (10:10:1) oriented at -30° from the principal stress axis (which is at 45° from the horizontal plane).

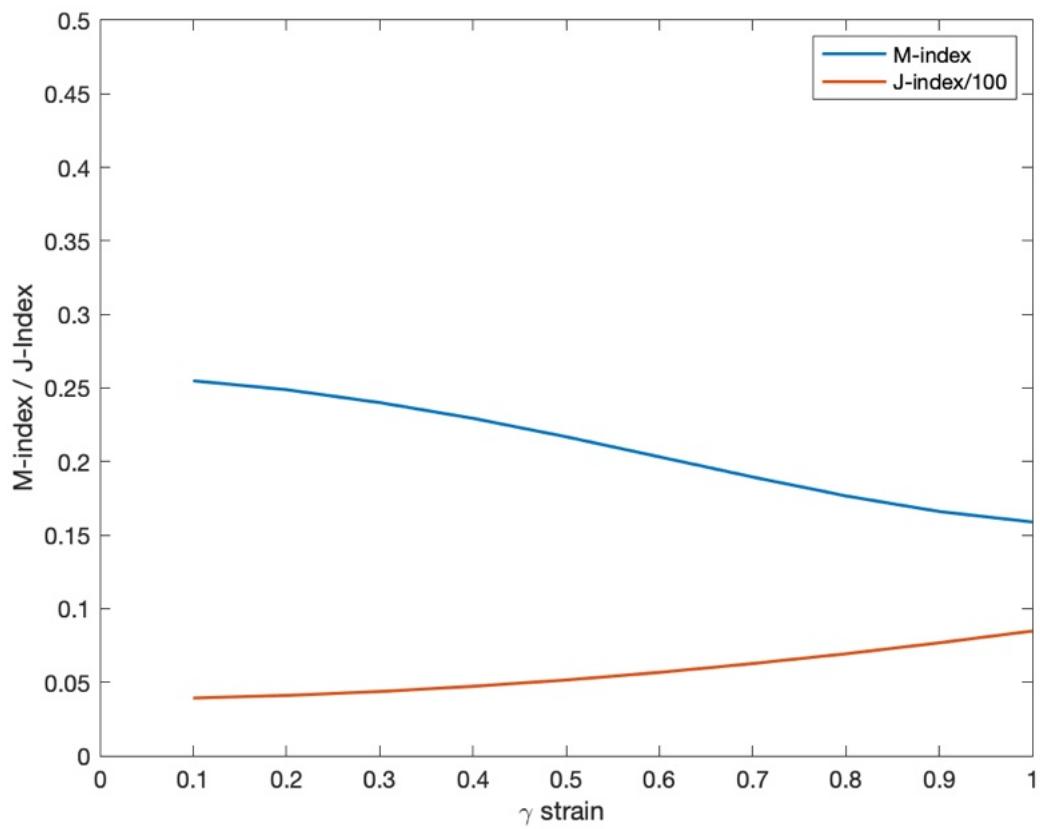


Figure 3.3: Evolution of M-index and J-index for an A-type olivine fabric.

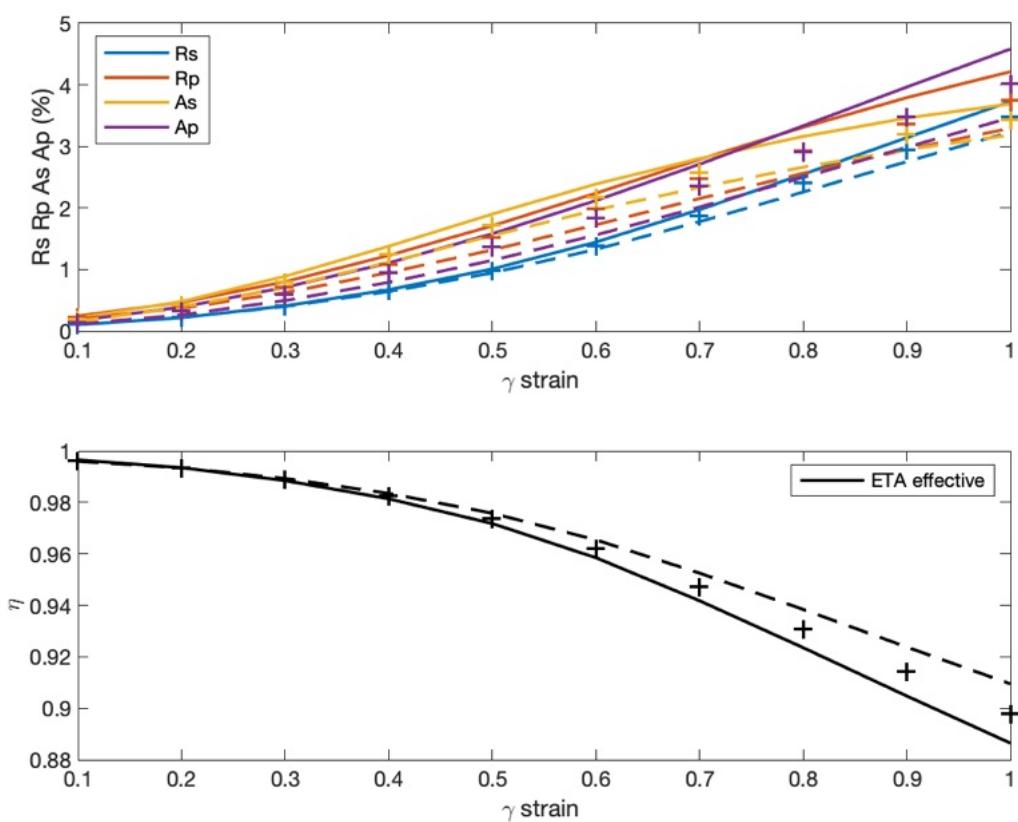


Figure 3.4: Evolution of radial and azimuthal P and S-wave anisotropy, and η parameter for Voigt (continuous lines), Reuss (dashed lines) and Hill (+ symbols) averages.

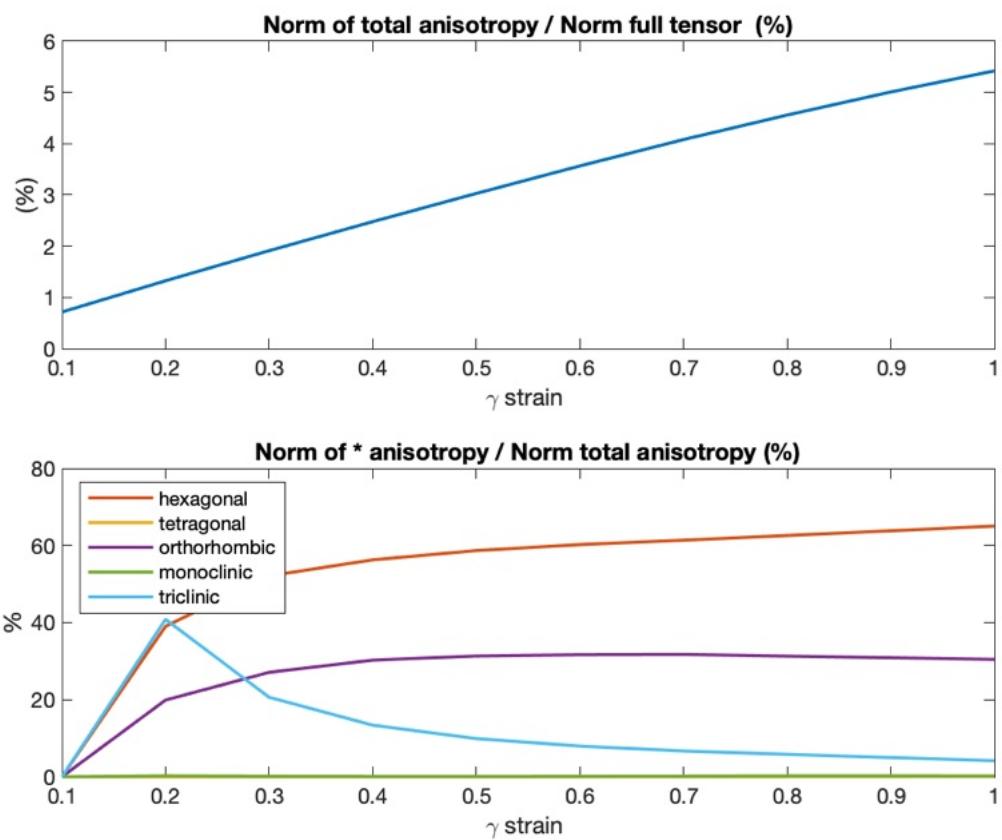


Figure 3.5: Evolution of (top) the norm of the total anisotropy component of the tensor over the norm of the full tensor (in %), and (bottom) the norm of each anisotropic component of the tensor normalized over the norm of the total anisotropic component (in %), computed for the elastic tensor obtained with the Mixed average.

4 D-REX_M: multiple mantle aggregates LPO fabrics with D-Rex

D-REX_M is a software that computes the evolution of the LPO and related elastic properties of multiple mantle aggregates, as a function of the flow field, deformation mechanisms and P-T conditions given by large-scale geodynamic simulations, and of the single crystal plastic and elastic properties. It builds on the original **D-Rex** software [Kaminski et al., 2004]. The additional features included in **D-REX_M** are listed in section 2.1.2 of [Faccenda et al., 2024].

COMPILE: `./bash_compile`

RUN: `mpirun -np nprocs ./drexm drexm_input.dat`

4.1 Software structure and modus operandi

D-REX_M is structured as the following:

1. the **D-REX_M** model parameters are loaded from `drexm_input.dat`, and the Eulerian grid (where the Eulerian fields are defined) and Lagrangian grid (mantle aggregates distribution) initialized accordingly.
2. the **V** (velocity), **P** (pressure), **Tk** (temperature), **Fd** (fraction of deformation accommodated by dislocation creep) Eulerian fields of the geodynamic model are loaded from the input `vtp*.h5` files.
3. firstly, the mantle aggregates are advected backward in time for a desired time span (note that is different from the original D-Rex where the particles are advected backward in time for a desired amount of strain).
4. successively, the mantle aggregates are advected forward and the strain-induced

LPO and F_{ij} evolution are simultaneously calculated. The forward run status can be checked in `cycle.txt` displaying the processed `vtp*.h5` file number and the terminated iterations/cycles.

5. output `Cijkl*.h5` files are generated containing, among other infos, the elastic tensor \mathbf{C} , density and the deformation gradient tensor \mathbf{F} of each aggregate.

4.2 Coordinate systems

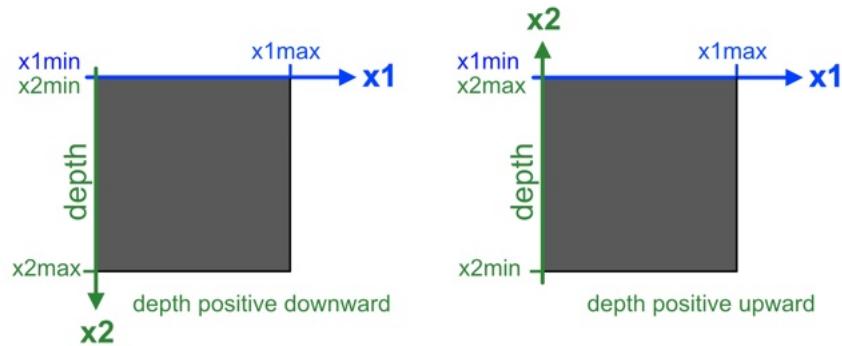
The coordinate system of the **D-REX_M** computational domain is defined by axis 1, axis 2 and, in 3D, axis 3 (Figs. 4.1, 4.2).

In cartesian coordinates, axis 1 is the horizontal direction, axis 2 depth, and, in 3D, axis 3 is the second horizontal direction.

In polar coordinates, axis 1 is longitude (in degrees), axis 2 the radial distance, and in 3D, axis 3 is colatitude (in degrees).

The vertical axis 2 can be positive either upward or downward.

2D cartesian coordinate system - D-REX_M / VIZTOMO / VIZVISC



2D polar coordinate system - D-REX_M / VIZTOMO / VIZVISC

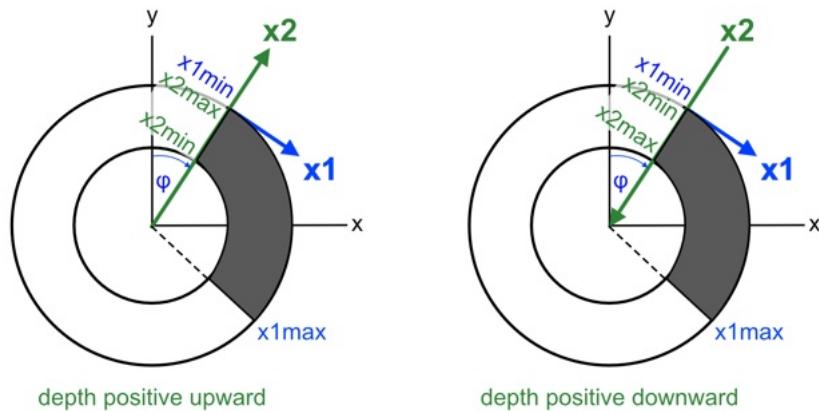
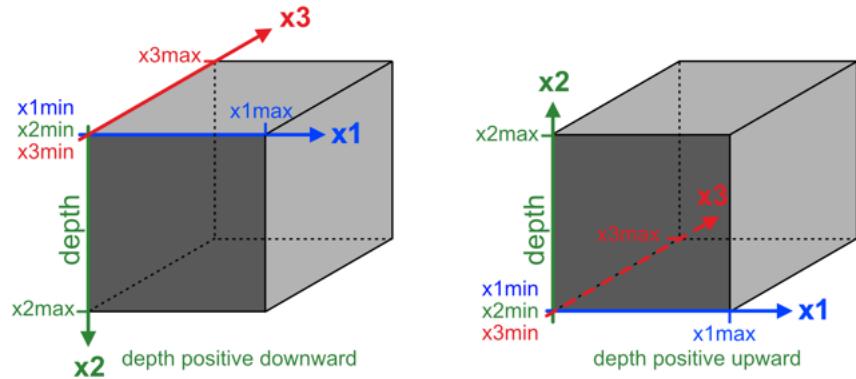


Figure 4.1: Cartesian and polar coordinate systems for 2D models valid for **D-REX_M**, **VIZTOMO** and **VIZVISC**.

3D cartesian coordinate system D-REX_M / VIZTOMO / VIZVISC



3D spherical coordinate system D-REX_M / VIZTOMO / VIZVISC

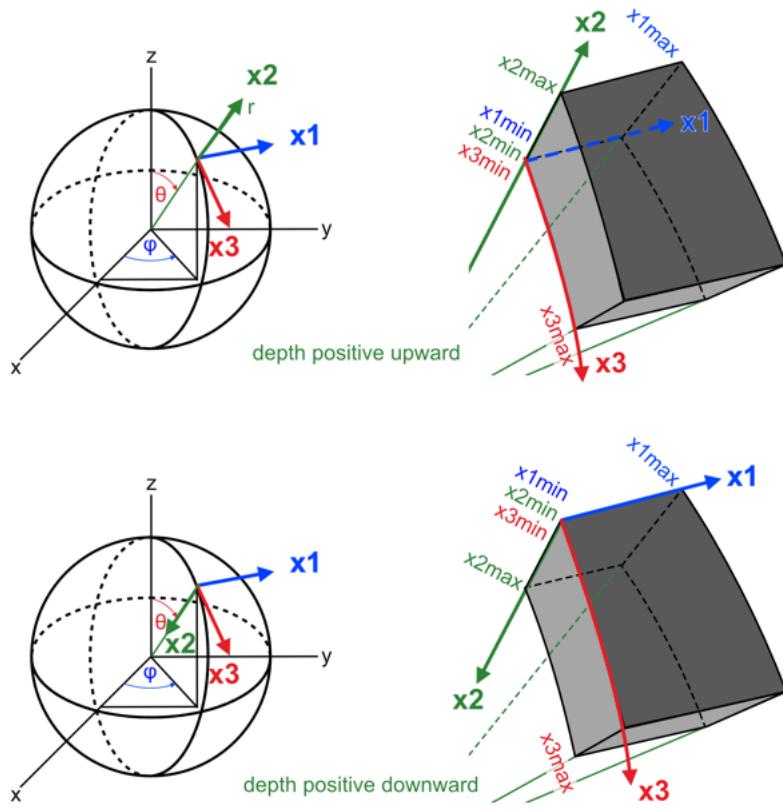


Figure 4.2: Cartesian and spherical coordinate systems for 3D models valid for **D-REX_M, VIZTOMO and VIZVISC**.

4.3 Input vtp*.h5 files

In order to perform the operations mentioned in the previous section, input files named `vtp*.h5`, where * is the file number, must be provided containing information about the geodynamic model evolution.

Essential information the input `vtp*.h5` files should contain is:

- the total elapsed time `timesum` and the timestep `dt` (time interval over which the fields are representative; normally it is the difference with the total elapsed time of the previous input `vtp*.h5` file, if any. See section 4.3.2) of the geodynamic model as a HDF5 attribute `/Time`.
- the components of the Eulerian velocity vector field **V** saved in `/Nodes/V1`, `/Nodes/V2`, and in 3D, `/Nodes/V3`.

In addition, the following Eulerian fields can be provided:

- when the geodynamic model is thermo-mechanical, the temperature **Tk** and total pressure **P** fields saved in `/Nodes/Tk` and `/Nodes/P`. These fields will be used to compute the elastic tensors as a function of the local P-T conditions.
- when the rheological model is based on multiple visco-plastic deformation mechanisms, the fraction of deformation accommodated by dislocation creep saved in `/Nodes/Fd`, where $Fd = nu_{disl} / nu_{eff}$, $0 \leq Fd \leq 1$, nu_{disl} is the viscosity calculated with the dislocation creep flow law, nu_{eff} is the effective viscosity which is typically calculated with the harmonic average of each of the viscosities representing a different deformation mechanism.

Summarizing, the time infos and the velocity field are essential, while the other 3 fields are optional and depend on the type of (mechanical vs. thermomechanical) geodynamic and (single vs. multiple visco-plastic deformation mechanisms) rheological models.

4.3.1 How saving information from your geodynamic model?

Examples of how to build the input file in HDF5 format with **MATLAB** are provided in directory `/D-REX_M/make_HDF5_vtp_files`.

Indexing: since the **V**, **P**, **Tk**, **Fd** Eulerian fields are loaded as 1D HDF5 datasets, they must be provided as 1D arrays where the nodes are indexed going through the grid axes sequenced as: 2D models: axis 2, axis 1. That is, Y, X axes in cartesian coordinates, and Radial, Long. axes in polar coordinates. 3D models: axis 2, axis 1, axis 3. That

is, Y, X, Z axes in cartesian coordinates, and Radial, Long., Colat. axes in spherical coordinates.

Units: values in attribute /Time and **V** as well as those of variables (i.e., **timemax** or distances) in the input file `drexm_input.dat` should be provided using consistent units for time and length, and which can be of any type (e.g., dimensional (i.e., [cm,yr] or [m,s]) or adimensional). In contrast, the **P** and **Tk** Eulerian fields must always be in Pa and K.

The Eulerian fields of the geodynamic model must be defined on the same node positions of the Eulerian grid defined in `drexm_input.dat`. The Eulerian grid must be regular (i.e., same node spacing along a given axis; if other types of grid are required, please contact the software developer).

For 3D global-scale simulations, the Yin-Yang Eulerian grids [Kageyama and Sato, 2004] are used to ensure a more homogeneous sampling of the spherical domain (see section 9.6).

In cartesian coordinates, the velocity field can also be defined on staggered additional nodes, which allows to compute the velocity gradient tensor with a numerical resolution that is double than when the velocity vector components are all defined on the same basic node. In 2D, V1 is shifted by $+\Delta x_2/2$ and V2 by $+\Delta x_1/2$ with respect to the basic node. In 3D, V1 is shifted by $+\Delta x_2/2$ and $+\Delta x_3/2$, V2 by $+\Delta x_1/2$ and $+\Delta x_3/2$, V3 by $+\Delta x_1/2$ and $+\Delta x_2/2$ with respect to the basic node.

4.3.2 How often saving information from your geodynamic model?

Time-dependent flows: From our experience, for Earth-like deformation rates it is sufficient to save the **V**, **P**, **Tk** and **Fd** fields averaged every 100/200 kyr, a time span over which these fields do not vary substantially. This will prevent outputting too many files from the geodynamic model. The code will then calculate an advection timestep which fulfill the Courant criterion and iterate for several cycles upon reaching the timestep defined in /Time.

Steady-state flows: in this case, it is clear that only a single `vtp*.h5` file is needed.

4.3.3 Which is the model time interval over which D-REX_M should run?

[Boneh and Skemer, 2014] showed that a cumulative strain of 5 is sufficient to reset any previous LPO and develop a new one. This is indeed the maximum cumulative strain that the original **D-Rex** consider for LPO calculation of each mantle aggregate.

For typical mantle deformation rates of $10^{-15} - 10^{-14} \text{ s}^{-1}$, this amount of deformation is acquired in about 3 – 30 Myr. This is the model time interval during which input `vtp*.h5` files must be saved in your time-dependent geodynamic model. Thus, saving an input `vtp*.h5` file every 200 kyr over a time span of 20 Myr, implies saving and processing 100 input files.

There exist cases where the model time interval could be larger, such as when it is desired to model the fabrics forming at oceanic ridges and successively advected for several tens of Myr within the rigid lithosphere till the subduction zone. However, steady-state corner flow is often assumed for this particular set of models, which implies that only a single `vtp*.h5` file is needed.

4.4 Output `Cijkl*.h5` files

Output files includes, for each aggregate, its rocktype, position, density, elastic tensor **C** and deformation gradient tensor **F**. When available, the aggregate P-T conditions are additionally saved. The **C** tensor is calculated as a function of the crystals orientation, volume fraction, modal composition of the aggregate (e.g., 70:30=Ol:Ens in hartzburgitic mantle), single crystal elastic properties scaled by the local P-T conditions, and with a Reuss-Voigt averaging scheme.

An output `Cijkl*.h5` file is always generated at the end of the simulation, whereby the aggregates, after the backward and forward advection episodes, display the same initial (regular) distribution. Output files at intermediate steps can be generated by setting **OutputStep** smaller than the total number of computing cycles of the forward advection scheme, with the characteristic that the aggregate distribution will be irregular.

4.5 Allocated memory

Memory is allocated in large part to track the mineral aggregates properties and, to a minor extent, for storing the loaded Eulerian node fields. Thus in case of 100s of thousands or millions of aggregates, problems related to insufficient memory allocation may occur.

For a 3D model, the memory allocated for the Eulerian fields is up to $24 * 8$ bytes per node. Thus, around 200 MB of memory is necessary every 1 million nodes per each MPI process.

For each mineral aggregates, the allocated memory is up to $(75 + 20 * \text{size3}) * 8$ bytes, thus very sensitive to the number of crystals. When **size3** = 8, there are $512 * 2$ crystals per aggregate, and thus 82.5 kB per aggregate. Hence, over 80 GB of memory is necessary every 1 million aggregates.

4.6 Parameter input file

The following information need to be set in the parameter input file `drexm_input.dat` (unless otherwise specified, the variable format is double precision):

MPI proc distribution along axis

- **nproc1, nproc2, nproc3**: (Integer) number of MPI subdomains (processes) along axes 1, 2 and 3

(WARNING: the number of MPI processes `nproc1*nproc2*nproc3` must be > 1)

A) Input and output directories/files

- **input_dir**: (String) path to the input `vtp*.h5` files directory (the path should end with “/”)
- **output_dir**: (String) path to the directory where the output `Cijkl*.h5` file is saved (the path should end with “/”)
- **Tinit, Tstep, Tend**: (Integers) initial, increment and final number of input `vtp*.h5` files. Set **Tinit = Tend** for steady-state models.
- **OutputStep**: (Integer) number of cycles interval after which an output file is generated during forward advection.
- **timemax**: time span (in Myr for dimensional time) for steady-state models

B) Define the computational domain:

- **dimensions**: (Integer)

2 = 2D model

3 = 3D model

- **cartspsher**: (Integer)

1 = cartesian coordinate system

2 = polar coordinate system

- **basicstag**: velocity field defined on: (Integer)

1 = basic nodes

2 = staggered nodes¹

Eulerian grid (i.e., where the **V**, **P** and **Tk** fields are defined):

- **x1min, x1max, nx1, x1periodic**: define Eulerian axis 1: min, max coordinates (in unit length in cartesian coord.; in degrees in polar/spherical coord.), number of nodes (Integer), periodic (1) or not (0) (Integer).
- **x2min, x2max, nx2, x2periodic**: define Eulerian axis 2: min, max coordinates (in unit length), number of nodes (Integer), periodic (1) or not (0) (Integer).
- **x3min, x3max, nx3, x3periodic**: define Eulerian axis 3: min, max coordinates (in unit length in cartesian coord.; in degrees in spherical coord.), number of nodes (Integer), periodic (1) or not (0) (Integer).

Lagrangian grid (i.e., the initial and final position of the mantle aggregates):

- **mx1min, mx1max, mx1stp**: define particle distribution along axis 1: min, max coordinates (in unit length in cartesian coord.; in degrees in polar/spherical coord.), spacing (in unit length, also for polar coordinate system)
- **mx2min, mx2max, mx2stp**: define particle distribution along axis 2: min, max coordinates, spacing (in unit length)
- **mx3min, mx3max, mx3stp**: define particle distribution along axis3: min, max coordinates (in unit length in cartesian coord.; in degrees in spherical coord.), spacing (in unit length, also for polar coordinate system)

C) LPO parameters:

- **size3**²: (Integer) cubic root of total number of grains in the aggregate for each of the 2 mineral phases (e.g.: **size3** = 10 → 1000 crystals of phase 1 and 1000 crystals of phase 2).

For each of the 5 different rocktype (Ol:Ens; Wd:Grt; Rw:Grt³; Brd:MgO; pPv:MgO⁴):

- **xol**: volume fraction of Ol, Wd, Rw, Brd, pPv (in %)
- **minx2, maxx2**: min, max vertical distribution (in unit length)
- **stressexp**: stress exponent for non-Newtonian plasticity
- **Mob**: efficiency of grain boundary migration

¹The staggered nodes can be defined only for cartesian grids

²The size of the allocated memory is mostly sensitive to this parameter, be careful!

- **chi** (χ): volume fraction threshold below which no dislocation creep occurs
- **lambda** (λ): efficiency of grain nucleation
- **fractdislrock**: fraction of deformation accommodated by the anisotropic phase. It assumes values between 0.0 (no deformation) and 1.0 (all deformation accommodated by the anisotropic phase). It should be always 1.0 for UM aggregates where both olivine and enstatite are anisotropic.
- **tau1...tauN**: normalized CRSS of the anisotropic phase slip systems. For upper mantle aggregates, one additional tau5=1 is added for enstatite. See A1.1.
- **single_crystal_elastic_db**: (Integer) choose single crystal elastic tensor for phase 1 and 2 from those compiled in `elastic_database.f90`. See A1.2.

D) Pre-existing (fossil) fabric. The fabric (given by the LPO and F) computed for a single aggregate in directory /D-REX_S is assigned to all aggregates within a given subdomain defined below:

- **fossilfabric**: (Integer)
 - 0** = no pre-existing fabric
 - 1** = the pre-existing fabric is set after backward advection and before computing the LPOs
 - 2** = the pre-existing fabric is set at the end of the model replacing the computed LPOs
- **mx1minfab, mx1maxfab**: distribution of aggregates with pre-existing fabric along axis 1 (in unit length in cartesian coord.; in degrees in polar/spherical coord.)
- **mx2minfab, mx2maxfab**: distribution of aggregates with pre-existing fabric along axis 2 (in unit length)
- **mx3minfab, mx3maxfab**: distribution of aggregates with pre-existing fabric along axis 3 (in unit length in cartesian coord.; in degrees in spherical coord.)

In polar coordinates, the loaded LPO and F are rotated according to the aggregate longitude and colatitude. Thus, when computing the single aggregate fabric with

⁴Rw and Grt are considered as isotropic at MTZ, and therefore only **Xol,minx2,maxx2** and **single_crystal_elastic_db** need to be set

⁴the occurrence of pPv is set according to the parametrized phase boundary by Oganov and Ono [2004b], and therefore **minx2,maxx2** should not be set. Hence, **ptmod** = 2

/D-REX_S, bear in mind that axis 1, 2 and 3 should correspond to longitude, depth and colatitude, respectively.

E) Operating modes:

- **fsemod**: (Integer)

0 = compute strain-induced F and LPO evolution

1 = compute only F evolution, and **fractdislmod**, **fabrictransformmod** and **ptmod** are set to 0.

- **uppermantlemod**: (Integer)

0 = F, LPO and advection for all aggregates.

1 = F and LPO only for upper mantle aggregates; advection for all aggregates. This mode is suggested when focusing on UM anisotropy only. The elastic tensor of deeper mantle aggregates are computed assuming isotropy. As a consequence, the execution of the run is much faster.

- **fractdislmod**: (Integer)

0 = 100% of the deformation is accommodated by dislocation creep.

1 = use only fraction of deformation accommodated by dislocation creep as interpolated from the **Fd** field. The **Fd** field need to be provided in input **vtp*.h5** files.

- **fabrictransformmod**: (Integer)

0 = no phase transformation.

1 = retain LPO after phase transformation, simulating axisymmetric topotactic growth of the new phases.

2 = reset LPO after phase transformation

- **ptmod**: (Integer)

0 = use single crystal tensors at room P-T conditions. Choose this when your geodynamic model is purely mechanical and there no is information about the local P-T conditions.

1 = scale elastic moduli and density by local P-T conditions, phase transitions occur at depths specified in top and bot; thus, the phase boundaries are horizontal. **Tk** and **P** fields need to be provided in input **vtp*.h5** files.

2 = scale elastic moduli and density by local P-T conditions, phase transitions occurs at density crossovers ($\text{Ol}+\text{Ens} \rightleftharpoons \text{Wd}+\text{Grt}$ at 3650 kg/m^3 ; $\text{Wd}+\text{Grt} \rightleftharpoons \text{Rw}+\text{Grt}$ at 3870 kg/m^3 ; $\text{Rw}+\text{Grt} \rightleftharpoons \text{Brd}+\text{MgO}$ at 4150 kg/m^3). This mode allows to model deflections of phase boundaries in proximity of mantle downwellings/upwellings. **Tk** and **P** fields need to be provided in input `vtp*.h5` files.

The occurrence of pPv is determined with the parametrized phase transition by Oganov and Ono [2004a], thus **Tk** and **P** fields are necessary. In general, whenever the computational domain includes layers deeper than the upper mantle, **ptmod** > 0 is the preferred operational mode.

In order to minimize the contrast in elastic properties among rocktypes and arising from contrasts in experimental single crystal elastic tensors and their P-T derivatives (see A1.2), the isotropic component of the aggregate elastic tensor is replaced with that computed with **MMA_EoS** for a given lithology when **ptmod** > 0.

- **eosmod**: (Integer) set the domain lithology (**1**: Dunite; **2**: Harzburgite; **3**: Pyrolite; **4**: Basalt; **5**: Pyroxenite) for which the corresponding density, Vp and Vs lookup tables computed with **MMA_EoS** are loaded (see Table A1.3).
- **fractvoigt**: fraction of Voigt averaging scheme when calculating the aggregate elastic tensor in %. It varies from 0.0 (Reuss average) to 100.0 (Voigt average). When equal to 50.0%, it yields the Hill average.

4.7 Parallelization and its efficiency

D-REX_M routines are parallelized using a hybrid MPI and OpenMP scheme to take advantage of multi-CPU nodes and multicore architectures of modern HPC clusters. The parallel efficiency is close to 1 for most routines, with the update of the LPO and Fij tensor during the forward advection is the most time-consuming part of the run (Fig. 4.3). The small performance degradation is due to the initialization of the Eulerian/Lagrangian grids and arrays, and to I/O operations (i.e., loading infos from input `vtp*.h5` files, writing the output to `Cijkl*.h5`) which are executed serially within each process. As a result, the efficiency of the time-dependent flow models (such as the 3D sinking slab model) is lower than that of steady-state models (the 3D global flow model) because of the larger I/O operations ((i.e., loading infos from input `vtp*.h5` files and compute the velocity gradient tensors, advection timestep, etc.).

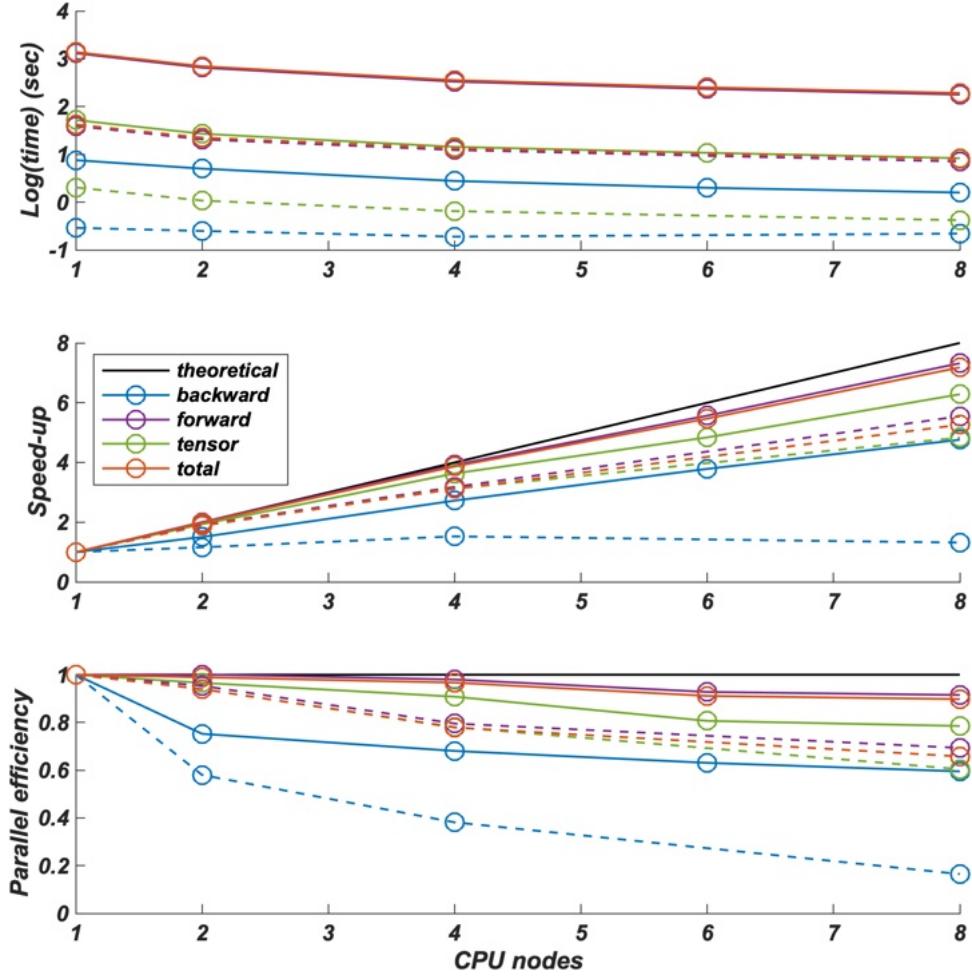


Figure 4.3: Runtime (top), speedup (centre) and parallel efficiency (bottom) of **D-REX_M** for initial backward advection of aggregates (backward), forward advection and update of the LPO and Fij tensor (forward), full elastic tensor computation and output generation (tensor), and entire run (total). Results shown for two models included in the cookbooks: a 3D global convection model (steady-state flow, 96 timesteps, 1327606 aggregates, LPO computed only for 260474 upper mantle aggregates) and the 3D sinking slab model (dashed lines; time-dependent flow, 20 timesteps, 38509 aggregates, LPO computed for 25177 upper mantle and 6587 upper mantle transition zone aggregates). Runs performed on a HPE Superdome Flex (8 CPUs, 28 cores Intel Xeon(R) PLATINUM 8180 @ 2.50GHz) using from 1 to 8 nodes.

5 EXEV: extrinsic elastic and viscous anisotropy

The EXEV software includes routines to compute extrinsic (SPO-related) fabrics and associated elastic and viscous tensors through effective medium theories

5.1 Effective Medium Theories

5.1.1 Smoothed Transversely Isotropic Long-Wavelength Equivalent (STILWE)

Introduced by [Backus, 1962], the STILWE is widely used to estimate the elastic properties of a stack of isotropic layers whose thickness is much smaller than the seismic wavelength. The equivalent elastic moduli of the homogeneous transversely isotropic medium are algebraic combinations of algebraic combinations of two elastic parameters (λ , μ or $\theta = Vs^2/Vp^2$). For a VTI medium, the elastic constants can be computed by using averages $\langle \cdot \rangle$ of μ and θ as [Backus, 1962]:

$$A = 4(N - S) + R^{-1}(1 - 2T)^2 \quad (1)$$

$$C = R^{-1} \quad (2)$$

$$F = R^{-1}(1 - 2T) \quad (3)$$

$$L = \langle 1/\mu \rangle^{-1} \quad (4)$$

$$N = \langle \mu \rangle \quad (5)$$

where $R = \langle \theta/\mu \rangle$, $S = \langle \theta\mu \rangle$, $T = \langle \theta \rangle$. The range of μ and θ for which the STILWE is stable is $0 \leq \theta \leq 3/4$ and $0 \leq \mu \leq +\infty$.

5.1.2 Differential Effective Medium (DEM) Theory

The tensorial formulation for DEM is [McLaughlin, 1977]:

$$\frac{d\mathbf{C}^{\text{DEM}}}{dV} = \frac{1}{1-V}(\mathbf{C} - \mathbf{C}^{\text{DEM}})\mathbf{A}^i \quad (6)$$

$$\mathbf{A}^i = [\mathbf{I} + \mathbf{G}^s(\mathbf{C}^i - \mathbf{C}^{\text{DEM}})]^{-1} \quad (7)$$

where \mathbf{C}^i and \mathbf{C}^{DEM} are the 4th-order elastic tensors of the inclusion and of the effective medium, respectively, V is the volume fraction of the inclusion, \mathbf{A}^i is the ratio of strain inside the inclusion to the strain in the host medium, \mathbf{I} is the symmetric fourth-rank unit tensor, \mathbf{G}^s is the symmetric Green's interaction tensor [Mainprice, 1997],[Hornby et al., 1994]:

$$G_{ijkl}^s = \frac{1}{2}(G_{ikjl} + G_{jkl}) \quad (8)$$

$$G_{ijkl} = \frac{1}{4\pi} \int_0^\pi \sin(\theta) d\theta \int_0^{2\pi} (K_{ij}^{-1} x_k x_l) d\phi \quad (9)$$

Here, the Green's interaction tensor G depends on the inclusion geometry and matrix elastic tensor through the Christoffel stiffness tensor $K_{ik}(x) = C_{ijkl}^{\text{DEM}} x_j x_l$ and directions $x_1 = \sin\theta\cos\phi/a_1$, $x_2 = \sin\theta\sin\phi/a_2$ and $x_3 = \cos\theta/a_3$, where a_1, a_2, a_3 are the semiaxes of the ellipsoidal inclusion. Eq. S3.2.1a is solved with a 1st-order in time, 4th-order in space Runge-Kutta method by setting the matrix as the initial effective medium and then by progressively increasing the volume fraction of the inclusion. It is important to note that, for any inclusion concentration, the host medium is always fully interconnected while the inclusions remain isolated.

The DEM solution for an aggregate of spherical inclusions is shown in Fig. 5.1, and benchmarked with the analytical solution [McLaughlin, 1977]:

$$\frac{dK}{dV} = \frac{K - K^{\text{DEM}}}{1 - V} \frac{K + K^*}{K^{\text{DEM}} + K^*}$$

$$\frac{d\mu}{dV} = \frac{\mu - \mu^{\text{DEM}}}{1 - V} \frac{\mu + \mu^*}{\mu^{\text{DEM}} + \mu^*}$$

$$\text{where } K^* = \frac{4}{3}\mu, \mu^* = \frac{1}{6}\mu \frac{9K + 8\mu}{K + 2\mu}$$

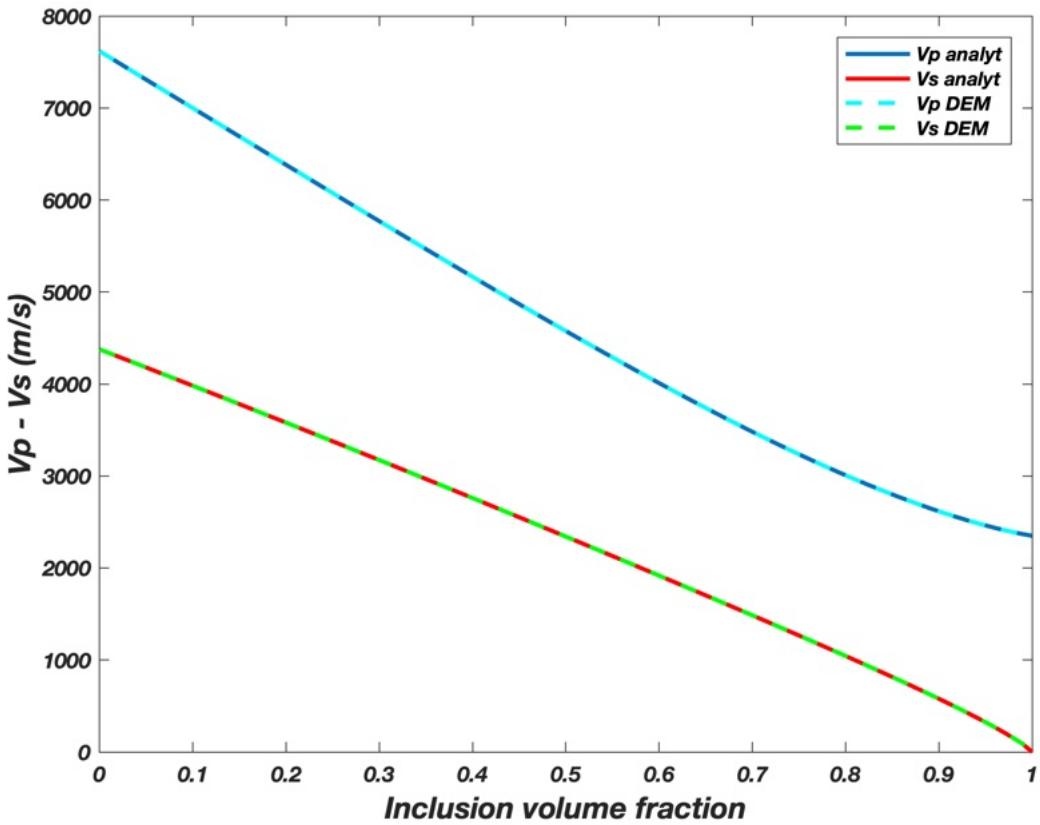


Figure 5.1: Comparison between analytical and DEM modeling solutions for spherical inclusions

5.2 Extrinsic elastic anisotropy

Seismic anisotropy can be generated by the strain-induced lattice preferred orientation (LPO) of minerals with anisotropic elastic properties (as modelled with D-Rex) and/or by the shape-preferred orientation (SPO) of isotropic compositional heterogeneities. The former anisotropy is referred to as intrinsic mechanical anisotropy, while the latter is referred to as extrinsic mechanical anisotropy. Extrinsic anisotropy occurs when (i) the size of the SPO is much smaller than the wavelength of the seismic signal and (ii) the contrast in isotropic elastic properties between the compositional domains is very large. When these conditions are satisfied, seismology fails to distinguish a finely layered and strongly heterogeneous isotropic medium from a smooth intrinsic anisotropic medium.

Within the Earth, extrinsic anisotropy can be related to either (1) the presence of a free gas or liquid phase included in elongated and preferentially oriented grain boundaries, pores, cracks, and porosity bands or (2) grain-scale (micrometer to centimeter) and/or rock-scale (centimeter to kilometer) SPO of compositionally distinct domains.

Modelling of extrinsic elastic anisotropy due to SPO is activated with parameter **spomod** in `drex_s_input.dat` (when using **D-REX_S**) or `viztomo_input.dat` (when using **VIZTOMO**):

0 = no extrinsic anisotropy is computed, and the original elastic tensor computed as a function of the LPO evolution is retained.

1 = extrinsic elastic anisotropy computed with the STILWE (Smooth Transversely Isotropic Long-Wavelength Equivalent medium; [Backus, 1962]), which calculates the elastic properties for a layered medium with ≥ 2 materials. The original elastic tensor computed in **D-REX_M** is ignored. The newly computed elastic tensor is then rotated with the layering normal to the shortest semiaxis of the FSE (e.g., Olson et al. [1984]).

2 = extrinsic elastic anisotropy computed with the DEM (Differential Effective Medium; [Mainprice, 1997, McLaughlin, 1977]), which calculates the elastic properties of a 2-phase matrix-inclusions system. The shape of the inclusions is ellipsoidal and can be defined in `spo_input.dat` together with their volume fraction. The elastic tensors of the 2 media are constant and the original elastic tensor computed in **D-REX_M** is ignored. The inclusions are oriented parallel to σ_1 , the maximum deviatoric stress, and normal to σ_3 , the minimum deviatoric stress, \pm an angle defined by **phi_spo**.

3 = same as **spomod = 2**, but the elastic tensor computed in **D-REX_M** is retained for the matrix.

In **VIZTOMO** the SPO fabrics are computed for aggregates with $\ln_{fse} \geq \text{ln_fse_min}$, implying that the rocks have accumulated a certain amount of deformation to generate the SPO fabrics.

In order to have the possibility to compute seismological synthetics (e.g., SKS-splitting) with the new elastic tensors, these are saved in a `Cijkl*.h5` file placed in a new directory `output_dir/SPO/`.

5.2.1 STILWE modeling (active when **spomod = 1**)

The STILWE can be used to model grain-/rock-scale layering when the P-T conditions of each aggregate are known (implying that the geodynamic model is thermo-mechanical and that **ptmod > 0** when running **D-REX_M**, or **D-REX_S**). Grain-scale or rock-scale layering can be evaluated for 5 lithologies with lookup tables of density and isotropic elastic moduli $f(P,T)$ generated with MMA_EoS [Chust et al., 2017]. It is possible to simultaneously compute grain-scale and rock-scale SPO by setting

spograinmod > 0 and choosing a rock mixture in volfractrock. When **ptmod = 0**, the constant $C_{\alpha\beta}$ of Medium 1 and 2 defined in `spo_input.dat` will be used with a relative volume fraction **Vmax** for Medium 2.

When using the STILWE, the LPO fabrics are ignored.

The SPO modeling parameters to be set in `spo_input.dat` are the following:

Set type of layered model:

- **spograinmod**: when > 0, computes the grain-scale SPO for a single lithology or lithology mixture as indicated in volfractrock. The allowed lithologies are Harzburgite, Pyrolite, Basalt and Pyroxenite, as Dunite does not generate grain-scale SPO. The sum of volfractrock must be 100 (%). It requires **spomod = 1**, **ptmod > 0** and **sporockmod = 0**.
- **sporockmod**: when > 0, computes the rock-scale SPO for the lithology mixture indicated in volfractrock. The sum of volfractrock must be 100 (%) and there should be ≥ 2 lithologies with volfractrock > 0. It requires **spomod = 1**, **ptmod > 0** and **spograinmod = 0**.
- **volfractrock(1, ..., 5)**: volume fractions in % of the 5 different lithologies (**1**: Dunite; **2**: Harzburgite; **3**: Pyrolite; **4**: Basalt; **5**: Pyroxenite). The sum must be 100%.

5.2.2 DEM modelling (active when **spomod > 1**)

The presence of a liquid phase (i.e., with null shear modulus) must be always evaluated with the DEM model (**spomod = 2** or **3**), as using the STILWE produces infinite elastic anisotropy.

As the inclusions are oriented relative to the principal stresses, when using **VIZTOMO** the flow field V from the input `vtp*.h5` file needs to be loaded. As such, a copy of the "present-day" `vtp*.h5` (where * is given by **Tend** as defined in `viztomo_input.dat`) must be placed in **cijkl_dir**.

When using the DEM, the matrix elastic tensor can be the constant $C_{\alpha\beta}$ of Medium 1 (**spomod = 2**) or the one from the **D-REX_M** strain-induced LPO modeling (**spomod = 3**).

The inclusions elastic tensor is always given by the constant $C_{\alpha\beta}$ of Medium 2, and their volume fraction is given either by **Vmax** or, when **meltspomod > 0**, by the porosity of the geodynamic model (see Figs. 9.2, 9.3).

The SPO modeling parameters to be set in `spo_input.dat` are the following (unless otherwise specified, the variable format is double precision):

Assign geodynamic model porosity (valid only when using **VIZTOMO** and not for the single aggregate fabric modelled with **D-REX_S**):

- **meltspomod**: (Integer) when > 0 , it requires an input file from which the porosity distribution in the computational domain is loaded. The inclusions volume fraction is given by the loaded fluid/melt fraction interpolated to the aggregate position.
- **meltfilename**: (String) name of the input HDF5 file with the geodynamic model porosity distribution. This file must be located in **cijkl_dir** and can be created with `/VIZTOMO/makeporosityfile.m`.

Set constant elastic tensors (in Voigt notation) and density for the 2 media:

- **Cback**: constant $C_{\alpha\beta}$ moduli of Medium 1 (Matrix for DEM)
- **Cinc**: constant $C_{\alpha\beta}$ moduli of Medium 2 (Inclusions for DEM)

Set DEM parameters:

- **esa1, esa2, esa3**: define shape of the ellipsoidal inclusions defining the relative length of the 3 semiaxes (or better to sigma 1, 2, 3). These semiaxes coincide initially with axes 1, 2 and 3 and then rotated such that **esa1** and **esa3** are parallel to, respectively, σ_1 and σ_3 . The aspect ratio should not exceed 1000.
- **vmax**: maximum inclusions volume fraction ($0 < \text{vmax} < 1$)
- **vstp**: volume fraction increments of the inclusions in the DEM model. The larger **vstp**, the less accurate is the computation. **vstp** should not be larger than 0.01.

Cracks/porosity bands orientation relative to σ_1 :

- **phi_spo** (ϕ_{spo}): when different from 0, it allows to rotate around σ_2 the fluid-filled cracks/porosity bands by an angle (in $^\circ$) with respect to σ_1 and in the direction of vorticity. As an example, porosity bands in melt-bearing mantle samples forming at relatively large strain are orientated at about -25° from σ_1 during simple shear deformation (the minus sign means that the rotation is against the vorticity) (Katz et al. [2006] and references therein).

See 5.1 for the different combinations of SPO model parameters, and Fig 9.3 for an application to a 2D oceanic ridge setting.

The program `/EXEV/DEMelastic/DEMelastic.f90` can be used to compute the two-phase component elastic properties for a range of inclusion ellipsoidal shapes and volume fractions as defined in `/EXEV/DEMelastic/dem_input.dat` (COMPILE: `./bash_compile`, RUN: `./demelastic`).

The elastic tensors properties saved in file `elastictensordem.h5` can then be plotted in Flinn diagrams using `/EXEV/DEMelastic/readDEMelastic.m` (Fig. 5.2).

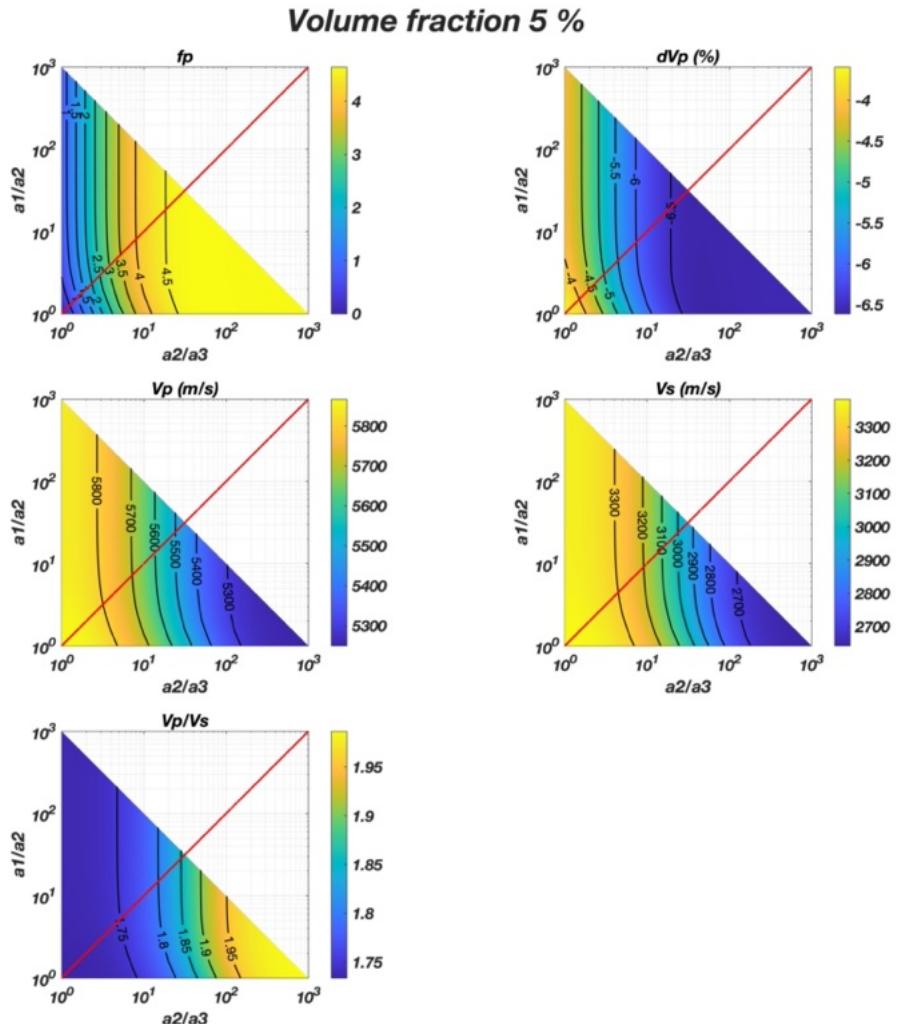


Figure 5.2: Flinn diagram showing the elastic properties of an aggregates with 5% in volume of melt-filled ellipsoidal inclusions with different shapes defined by the semiaxes ratios $a1/a2$ and $a2/a3$ and matrix-inclusion physical properties as defined in `dem_input.dat`. From top to bottom, left to right: P-wave anisotropy, isotropic P-wave anomaly w.r.t. the dry sample, isotropic P- and S-wave velocities, and their ratio.

Table 5.1: Different SPO models as a function of operating mode parameters

spomod	ptmod ¹	spograinmod	sporocksmod	meltspomod	Type of model
1	0	-	-	-	STILWE with 2 constant $C_{\alpha\beta}$ from <code>spo_input.dat</code>
1	1	1	0	-	STILWE with grain-scale SPO $f(P,T)$
1	1	0	1	-	STILWE with rock-scale SPO $f(P,T)$
2	-	-	-	0	DEM with constant $C_{\alpha\beta}$, ellipsoidal shape and $\psi^2 = \mathbf{Vmax}$ from <code>spo_input.dat</code>
2	-	-	-	1	DEM with constant $C_{\alpha\beta}$ and ellipsoidal shape from <code>spo_input.dat</code> . ψ from geodynamic model
3	-	-	-	0	DEM with $C_{\alpha\beta}$ from LPO for matrix, while constant $C_{\alpha\beta}$ for inclusions, ellipsoid shape and $\psi = \mathbf{Vmax}$ defined in <code>spo_input.dat</code>
3	-	-	-	1	DEM with $C_{\alpha\beta}$ from LPO for matrix, while constant $C_{\alpha\beta}$ for inclusions, ellipsoid shape defined in <code>spo_input.dat</code> . ψ from geodynamic model

¹ to be set in the `drex_s_input.dat` or `viztomo_input.dat` files.

² porosity.

The '-' symbol means that the corresponding parameter is ignored

5.3 Extrinsic viscous anisotropy

The modelling of extrinsic viscous anisotropy related to rock-scale and grain-scale fabrics relies on the DEM modelling and a geometric parametrization extrapolated from 3D high-resolution simulations of deforming matrix-inclusions aggregates [de Montserrat and Faccenda, 2021].

The DEM modelling is used to generate a database of viscous tensors for different matrix-inclusions viscosity contrasts, relative volume fraction and inclusions aspect ratios. Given the aggregate bulk FSE, the geometric parametrization provides an estimate of the inclusions average aspect ratio and orientation, thus allowing the mapping of the viscous tensor from the database and its rotation with respect to the FSE axes orientation.

Thus, the first modelling phase requires generating the DEM database (stored in file `viscoustensorDEM.h5`), which, together with the geometric parametrization, can be then exploited to estimate the local viscous tensor due to SPO-related fabrics in every point of the computational domain. The viscous tensor can be returned on the fly when the proper functions are called from a running geodynamic model (coupled mechanical simulations). Alternatively, its anisotropic properties can be computed for post-processing visualization (uncoupled mechanical simulations).

It is important to note that the values of the aggregate viscous tensor are always normalized with respect to the isotropic matrix viscosity η_{mat} . Hence, to retrieve the total viscous tensor:

$$\eta_{tot} = \eta_{DEM} \cdot \eta_{mat}$$

5.3.1 Generate the viscous tensor database

The main file `DEMviscous.f90` can be compiled and run as:

DIRECTORY: /EXEV/DEMviscous

COMPILE: ./bash_compile

RUN: ./demviscous

The DEM modeling parameters to be set in the parameter input file `dem_input.dat` are the following (unless otherwise specified, the variable format is double precision):

- **devmod:** (Integer)

0 = compute total viscous tensor

>0 = compute deviatoric viscous tensor

Set viscosities for the 2 media:

- **isotropicmod**: (Integer)
 - 0 = viscous tensors as defined in Voigt notation by η_{mat} and η_{inc}
 - >0** = force the viscous tensors to be isotropic with the range of viscosity contrasts defined by **etacontrmin**, **etacontrmax**, **etacontrstp**
- **etacontrlogscalemod**: (Integer), when > 0, the following viscosity contrasts must be defined as \log_{10} .
- **etacontrmin**, **etacontrmax**, **etacontrstp**: define min, max and increment of isotropic viscosity contrasts.
- **η_{mat}** : $\eta_{\alpha\beta}$ moduli of Medium 1 (Matrix for DEM)
- **η_{inc}** : $\eta_{\alpha\beta}$ moduli of Medium 2 (Inclusions for DEM)

Define inclusions shape range:

- **axislogscalemod**: (Integer), when > 0, the following semiaxes lengths must be defined as \log_{10} .
- **esa1**, **esa2**, **esa3**: define shape of the ellipsoidal inclusions defining the range of lengths for, respectively, the maximum, intermediate, minimum semiaxes normalized to that of the smallest one. Thus, **esa3**, the length of semiaxis 3, should be always 1. The range of lengths should be given as the min. max. and increment lengths.

Define array of inclusions volume fraction for which the viscous tensor is saved:

- **Volsavemin**: minimum inclusions volume fraction (**0 < Volsavemin < Vsavemax**). It must be a multiple of **Vstpdem**.
- **Volsavemax**: maximum inclusions volume fraction (**Volsavemin < Volsavemax < 1**). It must be a multiple of **Vstpdem**.
- **Volsavestp**: volume fraction increments of the inclusions in the DEM model. It must be a multiple of **Vstpdem**.

Set volume fraction increment for DEM solution

- **Vstpdem**: it is recommended to keep it as low as possible, especially for large aspect ratios, as the larger **Vstpdem**, the less accurate is the computation. However, bear in mind that the smallest the volume increment, the larger the num-

ber of iterations and the longer the computational time. As a rule of thumb, **vstpdem** should not be larger than 0.01, and a good choice is 0.005.

The viscous tensor database is saved in the output file EXEV/DEMviscous/viscoustensorDEM.h5 that must be then copied/moved in the **VIZVISC** directory.

The viscous tensors properties saved in file viscoustensorDEM.h5 can then be plotted in Flinn diagrams using /EXEV/DEMviscous/readDEMviscous.m.

In VIZVISC/viscoustensorDEM.h5 we provide such a database for deviatoric viscous tensors that can be readily used and computed in the parameter ranges:

$$\begin{aligned} \log_{10}(etac) &= -3 : 1 : 3, \\ \log_{10}(esa1) &= 0 : 0.1 : 3, \\ \log_{10}(esa2) &= 0 : 0.1 : 3, \\ \log_{10}(esa3) &= 0, \\ Volumefraction &= 0.05 : 0.05 : 0.3. \end{aligned}$$

5.3.2 Coupled mechanical simulations

DIRECTORY: /EXEV

In order to update the \mathbf{F} tensor and extract the associated $\boldsymbol{\eta}$ tensor from the database, the file `viscoustensor.f90` must be compiled together with `moduledem.f90`. This file includes subroutine `viscoustensor()` to be called from the large-scale geodynamic model by inputting the following variables:

x1m = axis 1 coordinate. In polar coordinates, longitude in radians (double)

x3m = axis 3 coordinate. In polar coordinates, colatitude in radians (double). In 2D cylindrical/annulus, set to pi/2.0

cartspsher = cartesian (1) or polar (2) domain (integer)

yinyang = Yin-Yang grid for 3D global simulations, active when = 2 (integer)

fse0 = particle Fij (3x3 double) in cartesian coordinates

l = particle velocity gradient (3x3 double) in cartesian coordinates

dt = timestep (double)

etac = matrix-inclusions isotropic viscosity contrast: $\eta_i nc / \eta_m at$ (double). For hard inclusions (**etac** > 1), allowed values are **etac**= 10, 100, 1000

volinc = inclusions volume fraction (double). Allowed values are **volinc** = 0.1, 0.2, 0.3.

The function returns as an output:

fse = updated particle \mathbf{F} tensor (3x3 double) in cartesian coordinates

etavoigt = mapped viscous tensor (6x6 double)

5.3.3 Uncoupled mechanical simulations: visualizing the viscous tensors with VIZVISC

DIRECTORY: /VIZVISC

COMPILE: ./bash_compile

RUN: ./vizvisc vizvisc_input.dat

VIZVISC is a software pretty much similar to **VIZTOMO**. It includes routines for post-processing the `Cijkl*.h5` output files generated with **D-REX_M** and for visualizing the η anisotropy and FSE tensors with **ParaView**. The viscous tensor η due to extrinsic viscous anisotropy is extracted from a database of tensors according to the FSE geometry, the matrix-inclusions viscosity contrast (**etac**) and the inclusions volume fraction (**volinc**). Thus, pre-computation of the viscous tensor database as explained in section 5.3.1 is required before running **VIZVISC** (or one can use the one provided as explained above). The viscous anisotropy is visualized on an eulerian grid defined below, as a function of the radial and azimuthal viscous anisotropy (which are analogous to those defined for elastic anisotropy) computed in the FSE reference frame.

The parameter input file is `vizvisc_input.dat`, where the following parameters can be set (unless otherwise specified, the variable format is double precision):

A) Input and output directories/files

- **cijkl_dir**: (String) path to input directory where to load the **D-REX_M** `Cijkl*.h5` output files (the path should end with “/”)
- **output_dir**: (String) path to output directory where to save the visualization output (the path should end with “/”), which can be or not the same as **cijkl_dir**
- **Tinit,Tstep,Tend**: (Integers) initial, increment and final number of input `Cijkl*.h5` files to be processed. When **Tinit = Tend**, only a single `Cijkl*.h5` file is processed.

B) Define viscosity contrast and inclusions volume fraction

- **etac**: isotropic viscosity contrast: η_{inc}/η_{mat} . For hard inclusions (**etac** > 1), allowed values are **etac**= 10, 100, 1000
- **volinc**: inclusions volume fraction. Allowed values are **volinc** = 0.1, 0.2, 0.3.

C) Visualize properties of Lagrangian aggregates:

- **Lagrangian**: (Integer) when > 0 , activates visualization of Lagrangian aggregates. The datasets saved in file `lagrangian*.h5` can be loaded on PARAVIEW with the corresponding file `XDMF.lagrangian*.xmf`. Vectors can be visualized applying the Glyph filter
- **ln_fse_min**: assumes values ≥ 0.0 , and indicates the threshold of $\ln_{fse} = \ln(fse_{max}/fse_{min})$ above which aggregates properties are visualized. It allows to skip aggregates that experienced low deformation and are nearly isotropic (try with 0.5, for example, as suggested by Becker et al. [2006]). When = 0.0, all aggregates are displayed.
- **uppermantlemod**: (Integer) when > 0 , displays only upper mantle aggregates.
- **rocktypemod**: (Integer) when > 0 , displays the rocktype of the aggregates.
- **fse3Dmod**¹: (Integer) when > 0 , the left stretch tensor **FSE** of ALL aggregates are saved in file `3Dfse*.h5`, and can be loaded on **ParaView** with `XDMF.3Dfse*.xmf`. Applying the TensorGlyph filter, the 3D FSE are displayed. The ellipsoids can be colored according to the rocktype or fse_{max} (suggestion: set the scale factor equal to the initial spacing of the aggregates defined in `drexm_input.dat`) (see Fig. 5.3C).
- **fsemiminmod**: (Integer) when > 0 , saves the fse_{min} vector scaled by \ln_{fse} .
- **fsemamaxmod**: (Integer) when > 0 , saves the fse_{max} vector scaled by \ln_{fse} .

D) Eulerian gridding: interpolate the $\boldsymbol{\eta}$ and **FSE** tensors of Lagrangian crystal aggregates to an Eulerian grid and visualize the anisotropic properties of $\boldsymbol{\eta}$ at each grid node only when $\ln_{fse} \text{ NODE} > \text{ln_fse_min}$. The coordinate system is the same as the **D-REX_M** model and described in Figs. 4.1, 4.2).

- **Eulerian**: (Integer) when > 0 , activates Eulerian gridding and visualization of Eulerian nodes elastic properties. The datasets saved in files named `eulerian*.h5` can be loaded on PARAVIEW with the corresponding files `XDMF.eulerian*.xmf`.
- **n1first,n1last,nx11**: define Eulerian grid axis 1: min, max coordinates (in m or m' in cartesian coord.; in degrees in polar/spherical coord.), number of nodes ($2 \leq \text{Integer} \leq \text{number of aggregates along axis 1}$).
- **n2first,n2last,nx21**: define Eulerian grid axis 2: min, max coordinates, number of nodes ($2 \leq \text{Integer} \leq \text{number of aggregates along axis 2}$).

¹Visualizing the 3D ellipsoid requires lots of computer memory, so be careful!

- **n3first,n3last,nx31**: define Eulerian grid axis 3: min, max coordinates (in m or m' in cartesian coord.; in degrees in spherical coord.), number of nodes ($2 \leq$ Integer \leq number of aggregates along axis 3).
- **radialmod**: (Integer) compute the radial anisotropy as a scalar when $\ln_{\text{fse NODE}} > \ln_{\text{fse_min}}$:
 - 0** = no radial anisotropy
 - 1** = radial anisotropy computed as $\xi = N/L$
 - 2** = radial anisotropy computed as $\xi(\%) = \sqrt{(N/L) - 1} \cdot 100$
where $N = \frac{1}{8}(\eta_{11} + \eta_{22}) - \frac{1}{4}\eta_{12} + \frac{1}{2}\eta_{66}$, $L = \frac{1}{2}(\eta_{44} + \eta_{55})$
- **azimod**: (Integer) compute the azimuthal anisotropy when $\ln_{\text{fse NODE}} > \ln_{\text{fse_min}}$.
In 2D, only the magnitude G of the azimuthal anisotropy is saved. In 3D, it is possible to visualize also the azimuth of the fast V_{SV} indicated by a vector that can be plotted with the Glyph filter of **ParaView**. The 3D vector is scaled by G , and saved in files `XDMF.azianis*.xmf` and `azianis*.h5`.
 - 0** = no azimuthal anisotropy
 - 1** = azimuthal anisotropy computed as $G = \sqrt{G_C^2 + G_S^2}$
 - 2** = azimuthal anisotropy computed as $G(\%) = \sqrt{\eta_{55}/\eta_{44}} - 1 \cdot 100$
where $G_C = \frac{1}{2}(\eta_{55} - \eta_{44})$; $G_S = \eta_{54}$
- **aziscalex1,aziscalex2,aziscalex3**: set position of scale bar for azimuthal anisotropy in 3D models (in m or m' in cartesian coord.; in degrees in polar/spherical coord. for **aziscalex1** and, in 3D, **aziscalex3**). The bar length correspond to $G = 1\%$ and it is oriented parallel to x1 axis.

Figure 5.3 shows the viscous anisotropy in the FSE reference frame for the steady-state global scale convection model in 2D polar coordinates explained in section 9.3. The run can be executed after having computed the velocity field with `cartesiancells.m` (that returns file `vtp0001.h5`), and the F field with **D-REX_M** (that returns file `Cijk10001.h5`), as:

```
RUN: ./vizvisc ../cookbooks/2Dpolar_convection/vizvisc_input.dat
```

For the other examples shown in the cookbooks chapter, it is straightforward to prepare the input file `vizvisc_input.dat`, with several parameters similar to those

found in `viztomo_input.dat`.

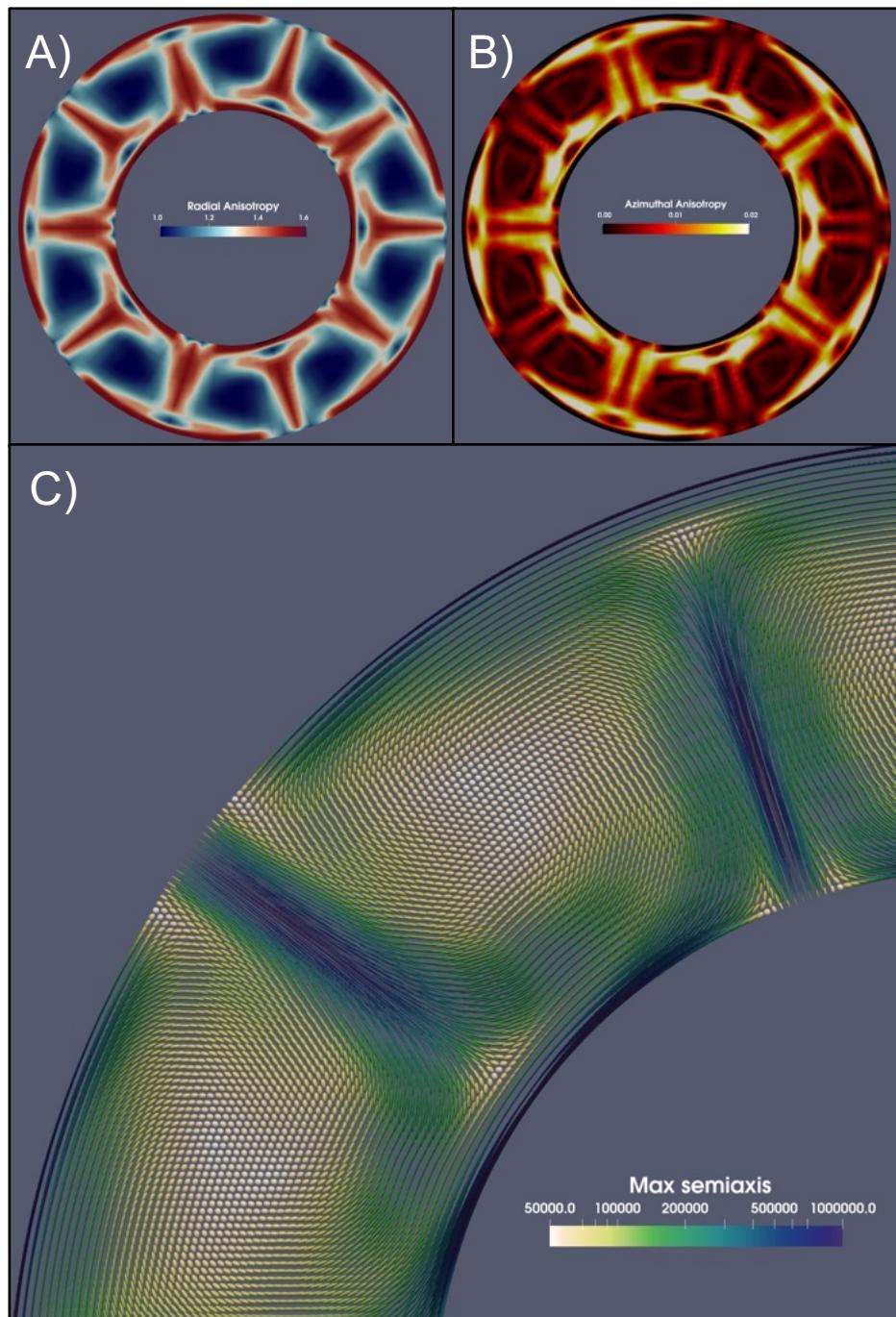


Figure 5.3: Steady-state convection patterns in polar coordinates created with `polarcells.m`. A) Radial and B) Azimuthal viscous anisotropy when `etac` = 0.1 and `volinc` = 0.2; C) the shape and orientation of the FSE ellipsoids plotted with the **ParaView** tool `TensorGlyph`. The ellipsoids are color-coded according to the length of the maximum semiaxis upscaled by 50 km.

6 VIZTOMO: visualization of the D-REX_M output

VIZTOMO is a software that includes routines for post-processing the `Cijkl*.h5` output files generated with **D-REX_M** and for visualizing the elastic tensor **C** and left stretch tensor **FSE** with **ParaView**. The software also allows computing extrinsic elastic anisotropy (which can or not be superimposed over the LPO fabrics) and provides grid structured distributions of elastic tensors for seismological model simulations.

COMPILE: `./bash_compile`

RUN: `./viztomo viztomo_input.dat`

6.1 Parameter input file

The parameter input file is `viztomo_input.dat`, where the following parameters can be set (unless otherwise specified, the variable format is double precision):

A) Input and output directories/files

- **cijkl_dir:** (String) path to input directory where to load the `Cijkl*.h5` files generated by **D-REX_M** (the path should end with “`/`”)
- **output_dir:** (String) path to output directory where to save the visualization output (the path should end with “`/`”), which can be or not the same as **cijkl_dir**
- **Tinit,Tstep,Tend:** (Integers) initial, increment and final number of input `Cijkl*.h5` files to be processed. When **Tinit = Tend**, only a single `Cijkl*.h5` file is processed.

B) Visualize **C** and **FSE** properties of Lagrangian aggregates:

- **Lagrangian**: (Integer) when > 0, activates visualization of Lagrangian aggregates. The datasets saved in file `lagrangian*.h5` can be loaded on **ParaView** with the corresponding file `XDMF.lagrangian*.xmf`. Vectors can be visualized applying the Glyph filter.
- **ln_fse_min**: assumes values ≥ 0.0 , and indicates the threshold of $\ln_{fse} = \ln(fse_{max}/fse_{min})$ above which aggregates properties are visualized. It allows to skip aggregates that experienced low deformation and are nearly isotropic (try with 0.5, for example, as suggested by Becker et al. [2006]). When = 0.0, all aggregates are displayed.
- **uppermantlemod**: (Integer) when > 0, displays only upper mantle aggregates.
- **rocktypemod**: (Integer) when > 0, displays the rocktype of the aggregates.
- **fse3Dmod**¹: (Integer) when > 0, the left stretch tensor **FSE** of ALL aggregates are saved in file `3Dfse*.h5`, and can be loaded on **ParaView** with `XDMF.3Dfse*.xmf`. Applying the TensorGlyph filter, the 3D FSE are displayed. The ellipsoids can be colored according to the rocktype or fse_{max} (suggestion: set the scale factor equal to the initial spacing of the aggregates defined in `drexm_input.dat`) (see Fig. 5.3C).
- **fseminmod**: (Integer) when > 0, saves the fse_{min} vector scaled by \ln_{fse} .
- **fsemamaxmod**: (Integer) when > 0, saves the fse_{max} vector scaled by \ln_{fse} .
- **TIaxismod**: (Integer) when > 0, saves the symmetry axis of the Transverse Isotropic component of the full elastic tensor. The length of the vector is proportional to the ratio of the Euclidean norm of the hexagonal part of the tensor over the Euclidean norm of the full tensor (in %; details of the tensor decomposition are provided in Browaeys and Chevrot [2004]). In addition, it saves as scalar fields the ratio in % of the Euclidean norm of the anisotropic part of the tensor over the Euclidean norm of the full tensor (**perc_anis**), as well as the ratio of the Euclidean norm of each anisotropic component of the tensor over the Euclidean norm of the total anisotropic components (**perc_hexa**, **perc_ortho**, **perc_tetra**, **perc_mono**, **perc_tri**).
- **vpmmaxmod**: (Integer) when > 0, saves the direction of Vp_{max} . The vector is scaled by:

$$Vp_{anis} = (Vp_{max} - Vp_{min}) / (Vp_{max} + Vp_{min}) * 200 \text{ (in \%)}$$
- **dvsmaxmod**: (Integer) when > 0, saves the direction of maximum Vs splitting. The vector is scaled by:

$$dVs_{MAX} = (Vs_1 - Vs_2) / (Vs_1 + Vs_2) * 200 \text{ (in \%)}.$$

C) SPO: Extrinsic elastic anisotropy²

- **spomod:** (Integer) when > 0, computes extrinsic elastic anisotropy for aggregates with $\ln_{fse} > \text{ln_fse_min}$ using two Effective Medium Theoretical models: STILWE and DEM.

0 = no extrinsic elastic anisotropy

1 = extrinsic elastic anisotropy due to grain-scale or rock-scale layering (STILWE model). The LPO fabrics are ignored.

2 = extrinsic elastic anisotropy due to the presence of aligned inclusions (DEM model). The LPO fabrics are ignored.

3 = extrinsic elastic anisotropy due to the presence of aligned inclusions (DEM model) superimposed over the LPO fabrics.

D) Eulerian gridding: interpolate the **C** and **FSE** tensors of Lagrangian crystal aggregates to an Eulerian grid and visualize the **C** properties at each grid node. **FSE** is used to visualize a given elastic properties only when $\ln_{fse \text{ NODE}} > \text{ln_fse_min}$. The coordinate system is the same as the **D-REX_M** model and described in Figs. 4.1, 4.2).

- **Eulerian:** (Integer) when > 0, activates Eulerian gridding and visualization of Eulerian nodes elastic properties. The datasets saved in files named `eulerian*.h5` can be loaded on **ParaView** with the corresponding files `XDMF.eulerian*.xmf`.
- **n1first,n1last,nx11:** define Eulerian grid axis 1: min, max coordinates (in unit length in cartesian coord.; in degrees in polar/spherical coord.), number of nodes ($2 \leq \text{Integer} \leq \text{number of aggregates along axis 1}$).
- **n2first,n2last,nx21:** define Eulerian grid axis 2: min, max coordinates (always in unit length), number of nodes ($2 \leq \text{Integer} \leq \text{number of aggregates along axis 2}$).
- **n3first,n3last,nx31:** define Eulerian grid axis 3: min, max coordinates (in unit length in cartesian coord.; in degrees in spherical coord.), number of nodes ($2 \leq \text{Integer} \leq \text{number of aggregates along axis 3}$).
- **vpvsmmod:** (Integer) compute Vp and Vs

0 = no Vp, Vs velocities saved.

1 = isotropic Vp, Vs velocities.

¹Visualizing the 3D ellipsoid requires lots of computer memory, so be careful!

²The SPO modeling is explained in detail in section 5.2

2 = Vp, Vs velocities along the direction given by (**cosx1,cosx2,cosx3**). This is useful to see what seismic waves “see” when propagating along certain directions.

- **dvpvsmod**: (Integer) compute dVp and dVs anomalies and requires **vvpvsmod** > 0.

0 = no dVp, dVs anomalies saved.

1 = dVp, dVs computed with respect to the average Vp,Vs velocity at a given depth.

2 = dVp, dVs computed with respect to the Vp,Vs along the reference vertical profile of Eulerian nodes specified by the horizontal node index **nx1ref** (and **nx3ref**, in 3D).

- **zoeppritzmod**: (Integer) compute transmitted/reflected energies for each couple of adjacent horizontal layers of the Eulerian grid for an incident P wave as dictated by the Zoeppritz’s equation [Aki and Richards, 2002]. The incidence angle (angle between the P wave and the vertical axis) is given by:

$$\psi = \text{atan2}(\sqrt{\text{cosx1}^2 + \text{cosx3}^2}, \text{cosx2}).$$

When the incidence angle is larger than the critical angle, an error message appears and the run is exited. The P wave is assumed travelling along the vertical axis positive direction. Four datasets are produced: **ERpp** (P-P reflected energy), **ERps** (P-S reflected energy), **ETpp** (P-P transmitted energy), **ETps** (P-S transmitted energy). The energies are proportional to the square of the reflection/transmission coefficients and are normalized with respect to the incoming P-wave energy, such that their sum is 1. It is clear that the vertical resolution of the model grid affects the values of these energies, as a low resolution tends to smooth eventual sharp variations in acoustic impedance.

0 = no **ERpp**, **ERps**, **Etpp**, **ETps** energies saved.

1 = **ERpp**, **ERps**, **Etpp**, **ETps** computed with isotropic Vp, Vs.

2 = **ERpp**, **ERps**, **Etpp**, **ETps** computed with Vp, Vs along the direction given by (**cosx1,cosx2,cosx3**).

- **cosx1,cosx2,cosx3**: direction angles (in degrees) of the incident seismic waves with respect to axes 1, 2 and 3, from which direction cosines are computed. The sum of the squared direction cosines must be 1.0 (i.e., should yield a unit vector), otherwise an error message is displayed and the run exited when **vvpvsmod** = 2 or **zoeppritzmod** > 0.

- **nx1ref**: (Integer) node index along horizontal axis 1 for the reference Vp, Vs profile when **dvpvsmod** = 2.
- **nx3ref**: (Integer) node index along horizontal axis 3 for the reference Vp, Vs profile when **dvpvsmod** = 2.
- **radialmod**: (Integer) compute the radial anisotropy as a scalar when $\ln_{fse} \text{NODE} > \ln_{fse} \text{min}$:
 - 0** = no radial anisotropy
 - 1** = radial anisotropy computed as $\xi = V_{SH}^2 / V_{SV}^2 = N/L$
 - 2** = radial anisotropy computed as $\xi(\%) = \sqrt{(N/L) - 1} \cdot 100$

where $N = \frac{1}{8}(C_{11} + C_{22}) - \frac{1}{4}C_{12} + \frac{1}{2}C_{66}$, $L = \frac{1}{2}(C_{44} + C_{55})$
- **azimod**: (Integer) compute the azimuthal anisotropy when $\ln_{fse} \text{NODE} > \ln_{fse} \text{min}$.
 In 2D, only the magnitude G of the azimuthal anisotropy is saved. In 3D, it is possible to visualize also the azimuth of the fast V_{SV} indicated by a vector that can be plotted with the Glyph filter of **ParaView**. The 3D vector is scaled by G, and saved in files XDMF.azianis*.xmf and azianis*.h5.
 - 0** = no azimuthal anisotropy
 - 1** = azimuthal anisotropy computed as $G = \sqrt{G_C^2 + G_S^2}$
 - 2** = azimuthal anisotropy computed as $G(\%) = \sqrt{C_{55}/C_{44}} - 1 \cdot 100$

where $G_C = \frac{1}{2}(C_{55} - C_{44})$; $G_S = C_{54}$
- **aziscalex1, aziscalex2, aziscalex3**: set position of scale bar for azimuthal anisotropy in 3D models (in unit length in cartesian coord.; in degrees in polar/spherical coord. for **aziscalex1** and, in 3D, **aziscalex3**). The bar length correspond to $G = 1\%$ and it is oriented parallel to x1 axis.
- **reflectmod**: (Integer) when > 0 , the Eulerian grid is reflected with respect the given axis. The size of the eulerian grid should be adjusted accordingly (i.e., if the **D-REX_M** axis 1 is X(0,1000 km), and it is desired to reflect with respect to n1first, the new axis 1 should be X(-1000 km, 1000 km)).
 - 0** = no Eulerian grid reflection
 - 1** = the Eulerian grid is reflected with respect to n1first
 - 1** = the Eulerian grid is reflected with respect to n1last
 - 3** = the Eulerian grid is reflected with respect to n3first

3 = the Eulerian grid is reflected with respect to n3last

- **replicateZmod**: (Integer) when > 0, it works only for 2D models and activates the replication of the 2D model along the third dimension as indicated by **n3first, n3last, nx31**.
- **specfem3Dmod**: (Integer) when > 0, save the Eulerian grid to be loaded in SPECFEM3D cartesian or globe version
- **psitomomod**: (Integer) when > 0, save the Eulerian grid to be loaded in **PSI_D**.

Figs. 6.1, 6.2, 7.1, 9.4, 9.5, 9.6, 9.7 show some of the Eulerian and Lagrangian fields that can be visualized in 2D and 3D geodynamic models.

In order to better understand the utility of **reflectmod** and **replicateZmod**, Fig. 6.2 shows, instead, the isotropic P-wave velocity for a 2D subduction model in polar coordinates (section 9.2) extending in longitude from 0° to 40° (A), the same model reflected with respect to **x1min** (B), and then replicated along colatitude from 70° to 110° with 201 nodes (C).

6.2 SPECFEM3D simulations within the domain built with VIZTOMO

To be finalized, coming soon!.

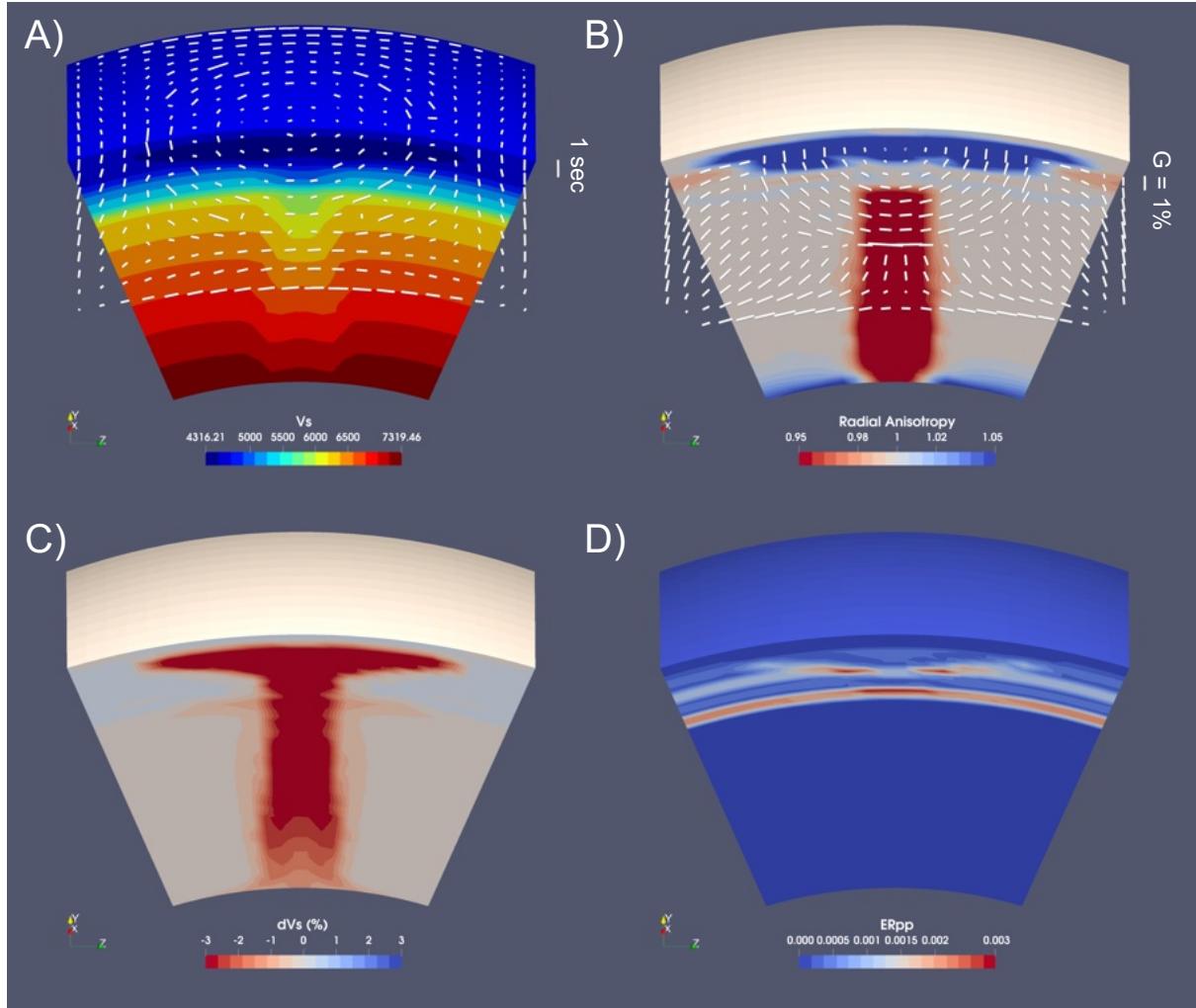


Figure 6.1: Example of a 3D thermo-mechanical model of an axi-symmetric thermal plume rising from the CMB in spherical coordinates. Only half of the eulerian grid generated with **VIZTOMO** is showed. A) S-wave isotropic velocity together with SKS splitting patterns at the surface (see chapter 7); B) Radial anisotropy together with azimuthal anisotropy patterns (white bars) at 180 km depth; C) S-wave velocity anomalies (%); D) Reflected P-P energy (note the lateral variations of the 410 and 660 km discontinuities as a function of the temperature anomaly).

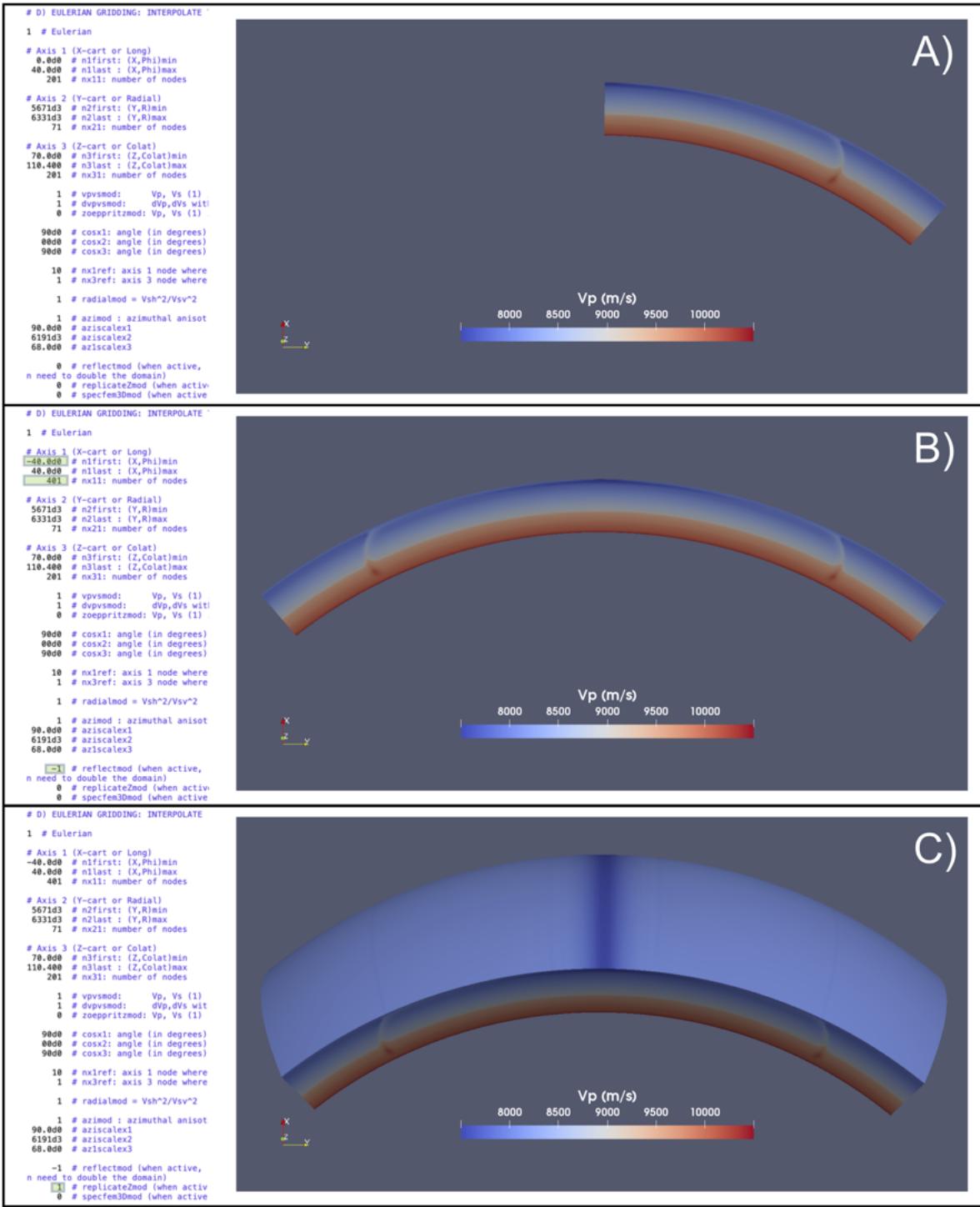


Figure 6.2: Isotropic P-wave velocity of a 2D petrological-thermo-mechanical model of subduction in polar coordinates (see section 9.2) (A). The same model reflected with respect to the oceanic ridge (B) and then replicated along the ridge axis (C). Right column: input variables in `viztomo_input.dat`; the variations with respect to the configuration in the case right above is highlighted in pale green. Left column: P-wave velocity.

7 SKS-SPLIT: synthetic SKS splitting

Synthetic SKS splitting is computed with routines included in FSTRACK (Becker et al. [2006]). An additional routine (savstack.f90) is provided to define a grid of seismic stations and, per each seismic station, build a stack of mantle horizontal layers by averaging elastic tensors and densities of crystal aggregates that are located close to the vertical projection of the station at depth and that are loaded from the `Cijkl*.h5` output file generated with **D-REX_M**.

COMPILE:

- Untar `fstrack.tar.gz`: `tar -zxvf fstrack.tar.gz`
- Execute the Makefile in `fstrack/single_layer`¹: `make`
- Execute the Makefile in `fstrack/multi_layer`: `make`
- copy the binary file `anicake` from `/fstrack/bin` directory to the **SKS-SPLIT** directory containing the bash file `pbs_sks`. Directory `/fstrack` can be now deleted, if wanted.
- Execute `./bash_compile` in **SKS-SPLIT**

RUN: submit the `pbs_sks` bash file which consecutively runs:

```
./stack_calc args (generates a stack of horizontal layers with different elastic tensors)  
mpiexec -np nprocs ./split_calc args (computes splitting paramters averaged over back-azimuth)
```

¹Depending on the Intel Fortran Compiler and environment settings, you need to add `F77=ifort,F90=ifort,CC=icc` to each of the two Makefile. This is done in the FSTRACK version present in this package

7.1 Parameter input file(s)

In **SKS-SPLIT**, open the bash file `pbs_sks` and set the directory **cijkl_dir** (String) where the `Cijkl*.h5` file that you want to process is and its filenumber **timestep** (4 digit Integer). Other parameters are rarely modified, and thus here not explained (but please feel free to contact the software developer for more infos). The number of processes (`nprocs`) is defined by the run control parameters (shown as an example in `pbs_sks`). At first, the code `stack_calc`, executed by one process, generate a 2D virtual grid of seismic stations and the underlying vertical stack of homogeneous elastic horizontal layers. Successively SKS splitting is computed by `nprocs` OpenMPI processes by executing `split_calc` (modified after Thorsten Becker).

Set the following parameters in file `stack_input.dat` (to be placed in the **cijkl_dir** directory) to build the grid of virtual seismic stations (in 2D models, ignore **nsx3**, **i3first**, **i3last**):

- **nsx1**: Integer. Number of seismic stations along horizontal axis 1 of the domain defined below
- **nsx3**: Integer. Number of seismic stations along horizontal axis 3 of the domain defined below
- **depthaxis**: Integer
 - 0: vertical axis positive downward
 - 1: vertical axis positive upward

Define the domain where to set the grid of seismic stations, and the volume enclosing the Lagrangian crystal aggregates sampled by the SKS waves. As SKS waves are mostly sensitive to anisotropy at upper mantle depths (Sieminski et al. [2008]), the vertical extent of the domain is usually that of the upper mantle. The coordinate system is the same as the **D-REX_M** model and described in Figs. 4.1, 4.2).

- **i1first**, **i1last**: seismic stations grid and Lagrangian domain axis 1: min, max coordinates (in unit length in cartesian coord.; in degrees in polar/spherical coord.).
- **i2first**, **i2last**: Lagrangian domain axis 2: min, max coordinates (in unit length).

- **i3first, i3last**: seismic stations grid and Lagrangian domain axis 3: min, max coordinates (in unit length in cartesian coord.; in degrees in spherical coord.).
- **maxdist**: maximum horizontal distance of mantle aggregates to be used for the interpolation of density and elastic tensor from the vertical projection at depth of the seismic station (in unit length). As the Fresnel zone of SKS waves is 100-200 km in the upper mantle, maxdist is typically set to 50-100 km.
- **minlayer**: minimum thickness of each mantle layer (in unit length). This is to avoid a very slow computation resulting from the formation of too many horizontal mantle layers when Lagrangian aggregates are densely packed. The max. number of layers set in anicake.F is 500.
- **xscale**: scaling factor used to convert depth in km as Depth [km] = Depth [geodynamic model units] · **xscale**. As an example, if depth is defined in meters in the geodynamic model, **xscale** must be set equal to 10^{-3} .

7.2 Compute SKS splitting

By running `./calc_split`, the seismic stations position and vertical stack of averaged elastic tensors scaled by density are saved in directory `cijkl_dir/seismic_stations`, while the apparent SKS splitting computed as a function of the back-azimuth every 5 degrees are saved in files `cijkl_dir/splitting/split*.dat.gz`, where * is the seismic station number.

The azimuthally averaged fast azimuth and delay times, together with standard deviations and station position, are saved in `cijkl_dir/splitting/split.0.*.staat`, where * is the `Cijkl*.h5` file number.

7.3 SKS splitting visualization

Files `split.0.*.staat` can be processed with file `/viz/read_sks_splitting.m`, by setting the input parameters. The script plots the SKS splitting averages for a quick view, and generates `*.xmf` and `*.h5` files for visualization in Paraview via the Glyph plot filter. In this way, the SKS data can be superimposed over the geodynamic model and/or the Eulerian grids generated with **VIZTOMO** (Fig. 7.1).

Examples of `stack_input.dat` and `read_sks_splitting.m` are available for each of the models shown the cookbooks chapter 9.

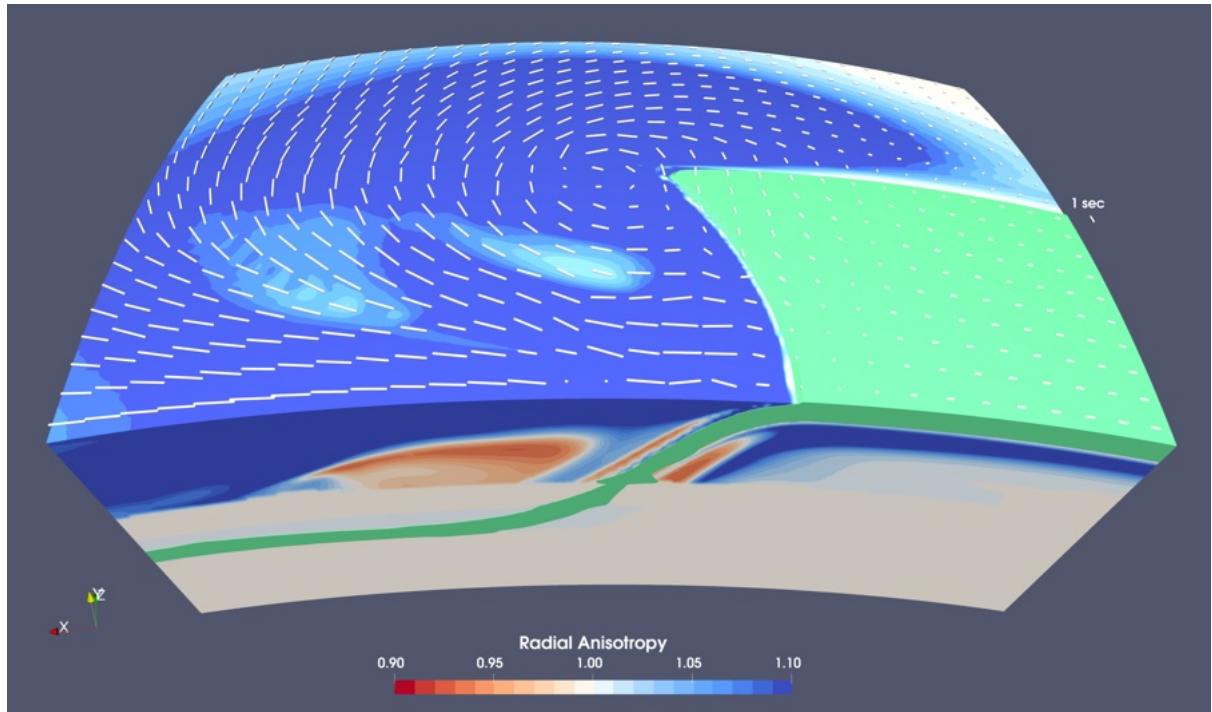


Figure 7.1: Example of a 3D thermo-mechanical model of subduction in spherical coordinates. The front wall is the subduction zone mid-plane, and it is a symmetric boundary. The green surface encloses material with +1% V_p anomalies which are related to the cold slab temperatures (note the thicker anomaly around the 410 km discontinuity, which is due to the upwelling of the $Ol \rightleftharpoons Wd$ phase transition). The volume is color coded according to radial anisotropy and it has been generated with **VIZTOMO** after running **D-REX_M**. The white bars are the mean SKS splitting for each of the seismic stations computed with the **SKS-SPLIT** software.

8 **PSI_D**: Platform for Seismic Imaging (Deterministic)

8.1 Overview

The Platform for Seismic Imaging (**PSI_D**) is a Julia package for performing forward and deterministic inverse modelling of seismic datasets. Another stochastic inversion methodology is under development and planned for a future release (see Del Piccolo et al., 2023). The primary motivation for its development was to facilitate the conversion of mantle geodynamic models into realistic seismic velocity models that may in turn be compared to imaging results from real arrays. This allows one to better understand how geodynamic features may be represented in seismic images and to identify potential artefacts—particularly those arising from neglecting anisotropy [e.g. Bezada et al., 2016, VanderBeek and Faccenda, 2021, VanderBeek et al., 2023]. To this end, synthetic seismic datasets can be generated directly from elastic models produced via **VIZTOMO** (Chapter 6) and subsequently inverted for both isotropic and/or anisotropic parameters. Note that only **VIZTOMO** models in spherical coordinates are supported at this time.

At present, **PSI_D** can predict P and S travel-times, splitting intensity, and shear wave splitting parameters in a ray-theoretical framework. Forward calculations are supported for models parameterized by (1) isotropic body wave velocities, (2) Thomsen parameters describing hexagonally anisotropic media, and (3) the full 21-component elastic tensor. Travel-time and splitting intensity datasets can be inverted for both isotropic and hexagonally anisotropic 3D models with arbitrary symmetry axis orientations. The inversion is performed using an iterative Gauss-Newton approach to solve for the model perturbations that minimize the sum of the squared data residuals in addition to constraints on the model norm and Laplacian. For a complete description of the methodology see VanderBeek and Faccenda [2021] and VanderBeek et al. [2023].

8.2 Installation

PSI_D, may be installed from Julia's package manager,

```
julia> using Pkg  
julia> Pkg.add(url="https://github.com/bvanderbeek/PSI_D.git")
```

Attention non-Windows users! For travel-time and ray path calculations, **PSI_D** uses the **TauP Toolkit** [Crotwell et al., 1999]. For the Julia wrapper to this Java program to function properly, the environment variable `JULIA_COPY_STACKS = yes` must be defined. This may be done within the shell configuration file or in Julia prior to loading the **PSI_D** package,

```
julia> ENV["JULIA_COPY_STACKS"] = "yes"
```

Attention Mac users! To avoid Java related segmentation faults, Julia must be started with the flag `--handle-signals=no`. Note that this may cause multithreading in Julia to crash.

A pure Julia implementation of the travel-time calculations and ray tracing is planned for a future release to avoid these Java-related inconveniences.

8.3 Coordinate system

The coordinate system follow the usual convention in seismology, upon which the axes 1, 2 and 3 correspond to the geographical coordinates: East (longitude, measured relative to the Greenwich meridian), North (latitude, measured relative to the equator) and radial direction (measured relative to the Earth's surface) (Fig. 8.1).

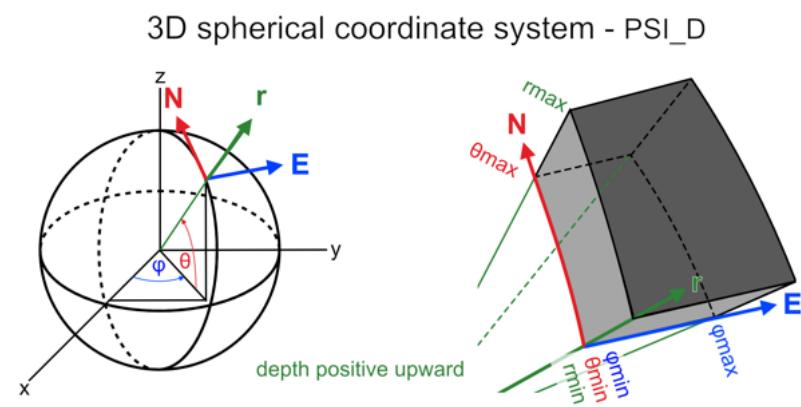


Figure 8.1: Coordinate system valid for **PSI_D**.

8.4 Forward Modelling

All the parameters required to make predictions are defined in a TOML file (see <https://toml.io/en/>). An example parameter file is located in: examples/SinkingSlab/psi_input/psi_parameters_synthetic.toml. The TOML table headings and variables are detailed below (8.4.1).

Once the parameter file is defined, predictions can be made by calling the forward problem function with the parameter file as input, e.g.,

```
julia> psi_forward("psi_input/psi_parameters_synthetic.toml")
```

Upon completion of the forward calculation, the predictions will be written to csv-formatted files in the output directory specified in the parameter file. A separate file is generated for each observation and phase (i.e., P or S) pair. The structure of these files is detailed below. A copy of the parameter file used to call `psi_forward` will also be saved.

8.4.1 Parameter file variables for forward modelling

1. [Output]

- `output_directory`: Path (String) to the directory where results are written.
- `tf_time_stamp`: Flag (Boolean) that when `true` will write results to a timestamped folder (`yyymmdd_hhmmss`) in `output_directory`. This is for convenience if one does not want to define unique names for every run.

At least one of the following observations types should be defined,

2.1. [Observations.TravelTime.CompressionalWave]

- `filename`: Path (String) to file containing P-wave travel-time data. Data should be written to a csv-formatted file with the following columns, travel-time (s), error (s), dominant waveform period (s; this will be used in the future for finite-frequency kernels), IASPEI recognized phase name, source ID (Integer), receiver ID (String), Channel (String; not currently used for any calculations but for reference).

2.2. [Observations.TravelTime.ShearWave]

- `filename`: Defined as above. However, for anisotropic forward calculations an additional 8th column containing the polarization azimuth measured counter-clockwise positive from the Q-channel (in ray-aligned QTL-coordinates) should be defined. If not present, shear waves are assumed to be radially polarized.

2.3. [Observations.SplittingIntensity.ShearWave]

- filename: Defined as above for S-wave travel-times (replacing the travel-time and error columns with the splitting intensity values).

2.4. [Observations.SplittingParameters.ShearWave]

- filename: Defined similar to the S-wave travel-time file. However, the first two columns contain the split time (s) and fast-azimuth (radians), respectively. The 3rd and 4th columns contain the errors in the split time and fast-azimuth.

3.1 [Model]

- parameterisation: The type of parameterization (String) used to define the velocity model. Options are "IsotropicVelocity", "HexagonalVectoralVelocity" for hexagonal anisotropic models with arbitrary symmetry axis orientations, or "ElasticVoigt" when models are defined using the 21-component elastic tensor.
- theModel: Path (String) to the csv-file containing model data. At present, models must be defined on a regular grid in spherical coordinates. Each model file must contain 3 header lines,

Header 1: Earth radius (km), longitude of model origin (i.e. center; decimal degrees), latitude of model origin (i.e. center; decimal degrees), rotation of model from north (radians; for future use);

Header 2: Number of nodes in longitude, number of nodes in latitude, number of nodes in radial direction

Header 3: Half-arc width of model in east-west direction (degrees), half-arc width of model in north-south-direction, maximum depth of model (km).

For IsotropicVelocity models, the values are printed first by increasing longitude, then increasing latitude, and lastly decreasing elevation. Each data row contains the following columns, longitude (decimal degrees), latitude (decimal degrees), elevation (km), Vp (km/s), Vs (km/s).

For HexagonalVectoralVelocity models, an additional 4th header line containing a single Boolean is required specifying if the exact (1) or weak (0) form of Thomsen's phase velocity equations are used in the forward calculation. Data rows are printed as for IsotropicVelocity with columns 4-11 defining α (i.e. symmetry axis P-velocity; km/s), β (i.e. symmetry axis P-velocity; km/s), f the anisotropic fraction, ϕ the symmetry axis azimuth (radians), θ the symmetry axis elevation (radians), and the ratios between the anisotropic strength and the Thomsen parameters, $r_e = \epsilon/f$, $r_\eta = (\epsilon - \delta)/f$, and $r_\gamma = \gamma/f$.

For ElasticVoigt models, an additional 4th header line containing a single

Boolean is required specifying if the elastic constants are density-normalized (1) or not (0). The 21 elastic coefficients should be written to columns 4-24 in the order, $c_{11}, c_{12}, \dots, c_{16}, c_{22}, c_{23}, \dots, c_{26}, \dots, c_{66}$. When density-normalized, the units should be m^2/s^2 . If not density normalized, the 25th-column should contain the density (kg/m^3) and the elastic coefficients given in GPa.

3.2 [Model.Aquisition]

- `source_data`: Path (String) to csv-file with seismic source data. The data columns are organized as follows: numeric ID, longitude (decimal degrees), latitude (decimal degrees), elevation (km).
- `receiver_data`: Path (String) to csv-file with seismic receiver data. The file should be formatted the same as `source_data` with the exception that the receiver ID will be interpreted as a string.

3.3 [Model.Methods.TauP]

- `reference_model`: Name (String) of the reference 1D model used for computing ray paths with **TauP**. Any built-in **TauP** model names are valid (e.g., "iasp91", "ak135") and custom velocity profiles may also be used provided they comply with **TauP** format requirements.
- `DL`: Ray path discretization interval (km). **TauP** ray paths are resampled at this level. An appropriate spacing is one that allows the details of the 3D velocity model to be accurately interpolated to the ray path.

8.5 Inverse Modelling

All the parameters required to run the inverse problem are defined in a TOML file similar to that used for the forward modelling (8.4). An example parameter file for performing an inversion is located in: examples/SinkingSlab/psi_input/psi_parameters_iso_inverse.toml. The TOML table headings and variables are detailed below (8.5.1).

Once the parameter file is defined, an inversion is run by calling the inverse problem function with the parameter file as input, e.g.,

```
julia> psi_inverse("psi_input/psi_parameters_iso_inverse.toml")
```

Upon completion, a number of outputs will be generated and stored in the directory specified in the parameter file. These are described below in section 8.5.2.

8.5.1 Parameter file variables for inverse modelling

All the forward modelling parameters detailed in 8.4.1 must also be defined in the inversion parameter file in addition to the following:

1. [Model]

If the starting model for the inversion is read from a file (as in the forward modelling) then no additional parameters are required under this TOML heading. **Note!** Only `IsotropicVelocity` and `HexagonalVectoralVelocity` parameterizations are allowed for inverse modelling. It's also possible to use the 1D TauP velocity profile as the starting model in which case the following additional definitions are required.

- `theModel`: This should be defined as an empty string (" ") to trigger the use of the reference 1D TauP velocity model.

1.1. [Model.Parameters]

When using `HexagonalVectoralVelocity` default ratios between the Thomsen parameters ($\epsilon, \eta = \epsilon - \delta, \gamma$) and the anisotropic strength (f) must be defined.

- `ratio_epsilon`: The ratio ϵ/f .
- `ratio_eta`: The ratio η/f .
- `ratio_gamma`: The ratio γ/f .

1.2: [Model.CoordinateSystem]

Because the coordinate system parameters are no longer read from the model file header, they must be defined.

- `R_0`: The reference Earth radius (km).
- `Lon_0`: Longitude origin (i.e. center) of model domain (decimal degrees).
- `Lat_0`: Latitude origin (i.e. center) of model domain (decimal degrees).
- `Rot`: Rotation of model with respect to North (degrees). Rotated coordinate systems are not yet implemented and this value should remain '0'.

1.3. [Model.Mesh]

Define the size of the forward model domain.

- `DX_1`: Longitudinal half-width of model domain (decimal degrees).
- `DX_2`: Latitudinal half-width of model domain (decimal degrees).
- `DX_3`: Maximum depth (positive) of model domain (km).
- `NX_1`: Number of nodes in the longitudinal direction.

- NX_2: Number of nodes in the latitudinal direction.
- NX_3: Number of nodes in depth.

2. [Invert]

All of the inversion parameters are defined under this heading in the TOML file. Excluding any of the parameters defined below from an inversion simply requires that it's not defined in the parameter file.

2.1. [Invert.SourceStatics] Source statics can be defined for each observation type using a separate heading as shown below.

2.1.1. [Invert.SourceStatics.TravelTime] Define the above heading in the TOML parameter file to invert for travel-time source static terms. A unique static for every source-observation type-phase tuple is included. Under this heading the following variables must be defined:

- phases: A vector of strings listing the phase types ("CompressionalWave" or "ShearWave") to include.
- damping: Vector of damping values that limit the norm of the static perturbations for each phase.
- tf_jump: Vector of booleans to minimise the cumulative (true) or incremental (false) static perturbation for each phase.

These variables can be repeated under [Invert.SourceStatics.SplittingIntensity] to include splitting intensity source statics.

2.2. [Invert.ReceiverStatics]

Receiver statics are implemented in the same way as the source statics with the same variables defined under the headings [Invert.ReceiverStatics.ObservationType] where ObservationType is a place-holder for TravelTime or SplittingIntensity.

2.3. [Invert.Velocity]

Under the Velocity sub-heading are defined isotropic and anisotropic seismic velocity inversion parameters.

2.3.1. [Invert.Velocity.Isotropic]

- parameterisation: Here we define the type of parameterization to use for the isotropic velocity fields. Currently, the only option is "InverseIsotropicSlowness". In the future, one may want to invert directly for velocity (rather than slowness) or relative velocity perturbations (i.e. dlnV).
- coupling_option: This parameter is for future use in joint P and S velocity inversions. It will describe how to couple P and S velocity perturbations. The

only option currently implemented is '0' which does not enforce any coupling between P and S velocity perturbations.

2.3.1.1. [Invert.Velocity.Isotropic.Mesh]

Define the grid on which the isotropic inversion parameters are discretized. The inversion grid is separate from the forward model grid such that we can choose to solve for heterogeneity on a different scale than the starting model. This is useful when the starting model contains *a priori* fine-scale structure (e.g., sediment basins). The variables for this field are the same as those described under [Model.Mesh] (i.e. type, DX_1, DX_2, DX_3, NX_1, NX_2, NX_3).

2.3.1.2. [Invert.Velocity.Isotropic.P]

Define the regularization parameters for P-velocity perturbations. If this heading is not defined, will not invert for P-velocity perturbations.

- damping_weight: Damping value that limits the norm of the velocity perturbations.
- tf_min_cumulative: Boolean to minimise cumulative (true) or incremental (false) norm of the of velocity perturbations.
- smoothing_weights: Smoothing weights to enforce spatially smooth velocity perturbations in x1-, x2-, and x3-directions.
- tf_smooth_cumulative: Boolean to minimise cumulative (true) or incremental (false) Laplacian of the velocity perturbations with respect to the starting model.

2.3.1.3. [Invert.Velocity.Isotropic.S]

Define the regularization parameters for S-velocity perturbations. If this heading is not defined, will not invert for S-velocity perturbations. Variables for this section are the same as those detailed in [Invert.Velocity.Isotropic.P].

2.4. [Invert.Velocity.Anisotropic]

Under this heading are the variables relevant to anisotropic inversions.

2.4.1. [Invert.Velocity.Anisotropic.Mesh]

Define the grid on which the isotropic inversion parameters are discretized. This is done as described under [Invert.Velocity.Isotropic.Mesh].

2.4.1.1. [Invert.Velocity.Anisotropic.Orientations]

Define the regularization parameters for hexagonal symmetry axis inversions. If this heading is not defined, will not solve for any anisotropic parameters.

- parameterisation: Here we define the type of parameterization to use for the

isotropic velocity fields. Currently, the only option is "InverseAzRadVector" which will solve for the azimuthal and radial (i.e. vertical) components of the hexagonal symmetry axis. The magnitude of the symmetry axis is directly proportional the P- and S-anisotropic fractions via the `ratio_epsilon`, `ratio_eta`, and `ratio_gamma` forward model parameters.

- `damping_weight`: Damping value that limits the norm of the anisotropy perturbations.
- `tf_min_cumulative`: Boolean to minimise cumulative (true) or incremental (false) norm of the of anisotropy perturbations.
- `smoothing_weights`: Smoothing weights to enforce spatially smooth anisotropy perturbations in x1-, x2-, and x3-directions.
- `tf_smooth_cumulative`: Boolean to minimise cumulative (true) or incremental (false) Laplacian of the anisotropy perturbations with respect to the starting model.

3. [Solver]

- `type`: Type of solver for performing the inverse problem. Only the LSQR-based method "SolverLSQR" is implemented. L-BFGS and gradient descent methods planed for future releases.
- `atol`: Stopping tolerance for LSQR algorithm (see lsqr documentation from IterativeSolvers.jl package). Default is `1.0e-6`.
- `conlim`: Convergence tolerance for LSQR algorithm (see lsqr documentation from IterativeSolvers.jl package). Default is `1.0e8`.
- `maxiter`: Maximum number of LSQR iterations allowed.
- `tf_jac_scale`: Boolean that if true, applies Levenberg-Marquardt style scaling of the regularization constraints. It is recommended that this parameter is true for most inversions.
- `nonliniter`: Maximum number of non-linear iterations (i.e. forward calculations and LSQR calls). Non-linear anisotropic inversions typically converge within 6 iterations. Iterations may stop before this criteria if the reduction in the sum of the squared residuals with respect to the prior iteration does not decrease significantly (as determined by an F-test at the 95% confidence level).

8.5.2 Inversion Output

When an inversion is run via

```
julia> psi_inverse("psi_parameters.toml")
```

the following outputs are generated:

- `FinalModel.vts`: VTK file that can be visualized in **Paraview**. This file contains both the starting model and the solution from the last iteration.
- `RSJS_ParameterField.vts`: VTK file that can be visualized in **Paraview**. This file contains the row-sum of the squared Jacobian elements (i.e. the squared derivative weight sum) and illustrates the data coverage for a given parameter field.
- `chi_squared.txt`: This file summarizes the data fit at each iteration. The first line of the file contains the critical F-value at which point iterations are stopped. The following lines list the iteration number and the corresponding F test statistic.
- `RES_ObservationType_PhaseType.dat`: Data (`csv`) files containing the residuals for each observation computed through the final iteration model. Separate files are generated for each observation type-phase pair. Files are formatted as described in 8.4.1 with the first column containing the residual (instead of the observed value).
- `Parameters.dat`: A `csv` file containing the final model data in a format that can be used as a starting model for subsequent inversions (format is described in 8.4.1 under `[Model]`).
- `Statics_Sources.dat`: A `csv` file containing the final source statics. The columns are organized as follows, (1) source ID, Observation Type, Phase Type, static value.
- `Statics_Receiver.dat`: A `csv` file containing the final receiver statics. The columns are organized as follows, (1) receiver ID, Observation Type, Phase Type, static value.
- `psi_parameters.toml`: A copy of the parameter file used to run the inversion.

8.6 Examples

Best way to get started using **PSI_D** is by looking through the examples. Two examples are included in the package repository and described below.

8.6.1 Sinking Slab

This example demonstrates how to generate a synthetic tomography model from a fully anisotropic elastic geodynamic model generated via **VIZTOMO**. The target model is a sinking slab generated by the Cookbook example 9.5 and is located in the **PSI_D** repository under examples/SinkingSlab.

The SinkingSlab example include the following directories and files:

jl-files/ Contains **Julia** scripts for generating synthetic data and running **PSI_D**

- `mk_inputs.jl`: Create synthetic array of seismic receivers and teleseismic sources for tomography.
- `mk_synthetics.jl`: Runs the forward calculation (`psi_forward`) to generate synthetic observation files.
- `run_inversion.jl`: Runs the inverse problem (`psi_inverse`).
- `example_SKS_SplittingParameters.jl`: Provides a short tutorial on how to generate SKS splitting parameters.

psi_input/ Contains input data for **PSI_D**.

- `psitomo0020.dat`: The **VIZTOMO** elastic model used for creating synthetic observations.
- `ak135_no_crust.tvel`: A modified version (no crustal velocities) of the AK135 1D reference earth used as a starting model for the inversions.
- `psi_parameters_synthetic.toml`: The **PSI_D** parameter file referenced in `mk_synthetics.jl` to call `psi_forward` and generate synthetic data.
- `psi_parameters_iso_inverse.toml`: The **PSI_D** parameter file for running isotropic inversion via `run_inversion.jl`.
- `psi_parameters_ani_inverse.toml`: The **PSI_D** parameter file for running an anisotropic inversion via `run_inversion.jl`.

psi_output/ The location where the example will write inversion results.

visualization/ Contains state-files that can be loaded to **Paraview** to automatically generate 3D plots of the isotropic (`render_solution_ISO_TTP.pvsm`) and anisotropic (`render_solution_ANI_TTP_TTS_SI.pvsm`) inversion results. Screen shots of the isotropic and anisotropic inversion results and predicted SKS split are provided in figures.

Here are the steps to reproduce the SinkingSlab synthetic tomography results:

1. Build required input data. **VIZTOMO** generates a target model (`psi_input/psitomo0020.dat`) but we must define the seismic array and source distribution used in the tomographic reconstruction. These data are generated by running the `jl-files/mk_inputs.jl`. This script will create `psi_input/Sources.dat` and `psi_input/Receivers.dat` files and the place-holder observation files `psi_input/DUMMY_TravelTime_CompressionalWave.dat`, `psi_input/DUMMY_TravelTime_ShearWave.dat`, and `psi_input/DUMMY_SplittingIntensity_ShearWave.dat` (these contain P and S travel-times and S splitting intensity observation files for every source-receiver pair). The synthetic array consists of a regular grid of 625 receivers extending $\pm 10^\circ$ in longitude and $\pm 10^\circ$ in latitude. In total, 24 teleseismic sources are used for imaging; 12 at a range of 50° and 12 at 80° from the model center and equally distributed in back-azimuth.
2. Generate synthetic observations. Using the inputs generated in step 1, we create synthetic travel-time and splitting intensity datasets by running `jl-files/mk_synthetics.jl`. This will create an output directory `psi_output/SYN_SinkingBlock` with the synthetic data files. This takes approx. 6 minutes to complete.
3. Perform an isotropic inversion of the P-wave travel-times. Open `jl-files/run_inversion.jl` and set the variable `the_parameter_file` to "`psi_parameters_iso_inverse.toml`". Then run the **Julia** script. The isotropic tomography results will be output in `psi_output/ISO_TTP`. This inversion completes in approx. 2 minutes.
4. Perform an anisotropic inversion of P and S travel-times and S splitting intensity data. Repeat step 3 but first redefine `the_parameter_file` to "`psi_parameters_ani_inverse.toml`". The anisotropic tomography results will be output in `psi_output/ANI_TTP_TTS_SI`. This inversion completes in approx. 18 minutes.
5. Visualize the results. The inversion will generate vtk-files that may be visualized using **Paraview**. From **Paraview**, one can load the state-files `render_solution_ISO_TTP.pvsm` or `render_solution_ANI_TTP_TTS_SI.pvsm` to automatically generate some 3D plots of the isotropic and anisotropic inversions, respectively. However, upon loading the state file, one should select "Search files under a specified directory" in the dialog box and then specify the path the appropriate output folder in `psi_output/`. The expected results of the isotropic and anisotropic inversions are shown in Fig. 8.2.

8.6.2 Subducting Plate

In addition to the Sinking Slab example, we also include the files required to reproduce the anisotropic tomography results for the subducting plate shown in Figure 7.1 (or Figure 9 of the ECOMAN publication) and reproduced below in Fig. 8.3. The inputs required to run the inversion are located in `examples/SubductingPlate`. The reference **VIZTOMO** model file is not included due to its large size. To run the inversion, do `julia jl-files/run_inversion.jl`. The results will be stored in `psi_output/ANI_TTP_TTS`. This inversion should finish in approx. 15 minutes. Once finished, load the state-file `visualization/render_model.pvsm` in **Paraview** (be sure to update the path to the appropriate output directory when prompted by the **Paraview** dialog box) and the tomographic results should appear as show in Fig. 8.3.

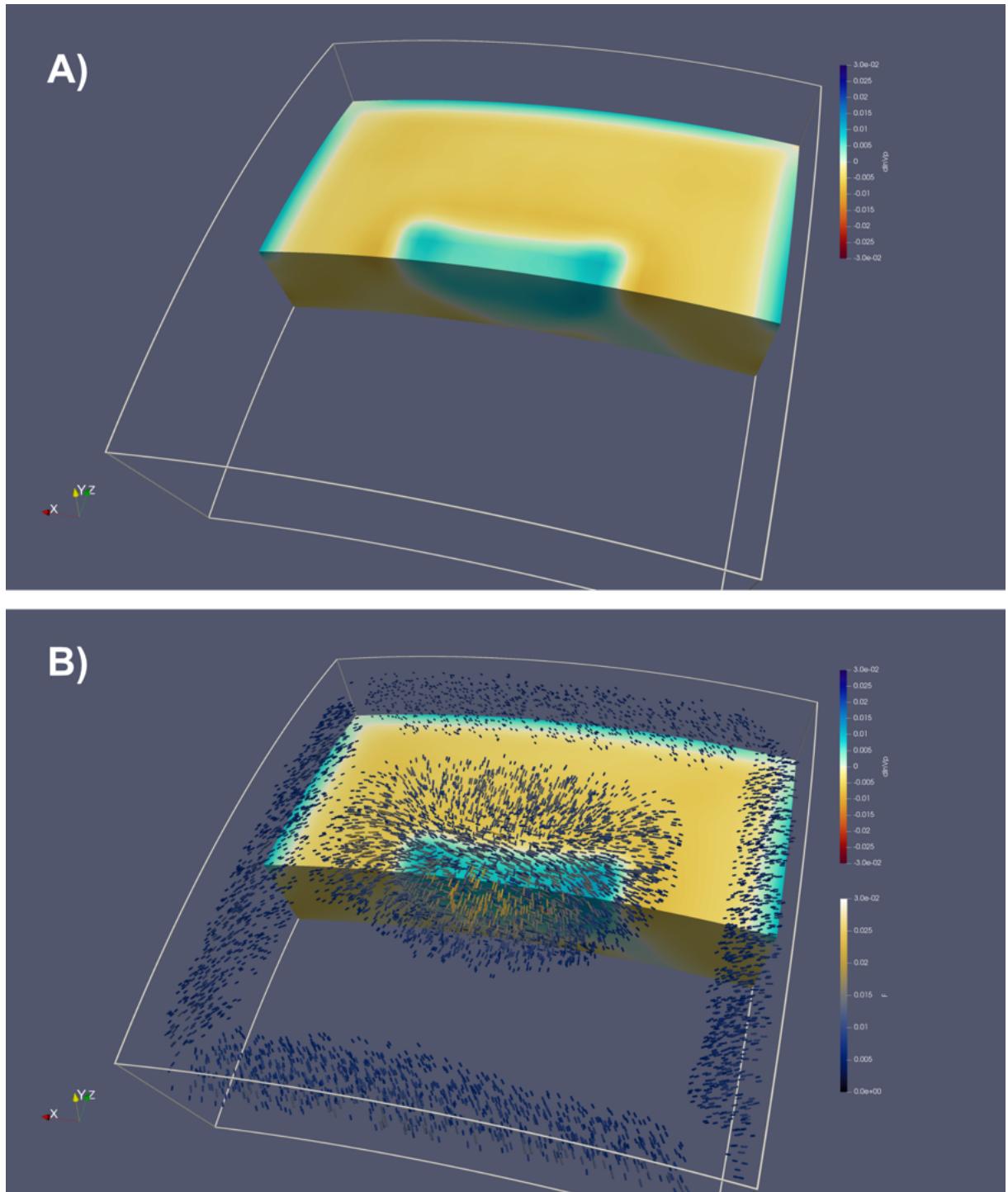


Figure 8.2: Sinking Slab tomography example. A) Isotropic inversion of P-wave traveltimes. B) Anisotropic inversion of P and S travel-times and S splitting intensity. In both panels, the recovered isotropic P-wave anomaly is contoured and the model is sliced through the center of the sinking slab at 300 km depth and in the east-west direction; full extent of the model is outlined. In B), a random distribution of recovered hexagonal symmetry axes (i.e. fast axes) are plotted and colored by the anisotropic strength. Symmetry axes are shown only where the anisotropic strength is >0.5%.

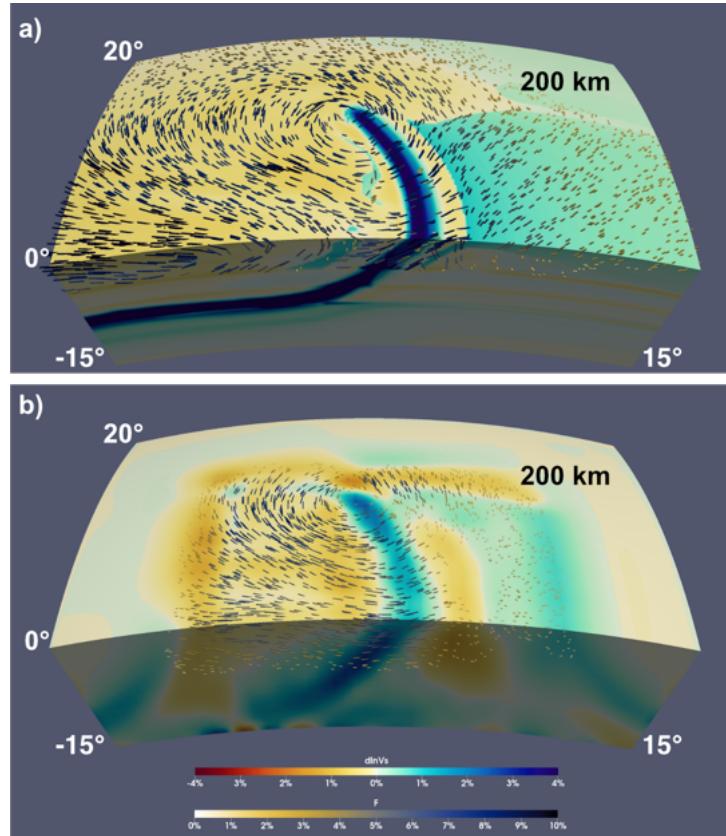


Figure 8.3: Synthetic subducting plate example. (a) Target anisotropic model generated from VIZTOMO. (b) Recovered anisotropic model obtained by inverting synthetic teleseismic P and S (relative) travel times computed from the model in (a). In both panels, the isotropic S-wave velocity perturbations are computed with respect to the far-field 1D velocity profile. Quivers parallel the hexagonal symmetry axis and are scaled and coloured by the anisotropic strength. The top surface shown is located at 200 km depth while the full model extends from 0-1000 km along the radial direction, 85°-115° along longitude, and 0-20° along latitude.

9 COOKBOOKS: some useful examples

The following cookbooks are provided as examples to facilitate the user configuring **D-REX_M**, **VIZTOMO** and **SKS-SPLIT** runs in different geodynamic model domains. The input parameter files (`*_input.dat`) should be structured as provided, that is, preserving the same sequence of parameters. However, not all the parameters need to be considered, as some of them are only used when required (i.e., in 2D, the 3rd dimension is ignored; the size of the domain where to set a fossil fabric is considered only when `fossifabric` is > 0 ; visualization of the Lagrangian and Eulerian fields is active only when `Lagrangian` and `Eulerian` are > 0 , respectively; etc.).

In order to minimize the size of the `vtp*.h5` files required to compute mantle fabrics, the examples below have been generated by low- to medium-resolution geodynamic models, and, in several cases, where the flow fields have reached or are assumed to have reached steady-state conditions. For each cookbook, we provide parameter input files for software **D-REX_M**, **VIZTOMO**, **VIZVISC** and **SKS-SPLIT**.

From each of the software directory execute:

D-REX_M: `./drexm .../cookbooks/DIRECTORY/drexm_input.dat`

VIZTOMO: `./vitzomo .../cookbooks/DIRECTORY/vitzomo_input.dat`

VIZVISC: `./vizvisc .../cookbooks/DIRECTORY/vizvisc_input.dat` (provided only for /2Dpolar_convection)

SKS-SPLIT: `./calc_split_from_flow_model1`

¹Open the bash file `pbs_sks`, and at lines 20 and 22 set, respectively, the directory `cijkl_dir` (String) where the `Cijkl*.h5` file that you want to process is (and where the `stack_input.dat` file has been copied), and its filenumber `timestep` (4 digit Integer)

9.1 Steady-state convection in 2D cartesian coordinates

DIRECTORY: /2Dcartesian_convection

A 2D incompressible flow field in cartesian coordinates is given by the following analytical solution, whereby the radial and tangential velocities are:

$$V_x(x, y) = -V_{x0} \sin(k_x 2\pi/W) \cos(k_y \pi/H)$$

$$V_y(x, y) = V_{y0} \cos(k_x 2\pi/W) \sin(k_y \pi/H)$$

where x and y are the horizontal and vertical distance, V_{x0} and V_{y0} a constants, k_x and k_y are the horizontal and vertical wave numbers, W and H are the domain width and height.

The \mathbf{v} field is computed with the **MATLAB** script `cartesiancells.m` and saved to `vtp0001.h5` as a function of the numerical resolution, horizontal and vertical wave numbers. The model is mechanical, and thus `ptmod = 0` in `drexm_input.dat`. The mantle fabrics computed with **D-REX_M** in steady-state conditions for 5 Myr and visualized with **VIZTOMO** are shown in Fig. 9.1.

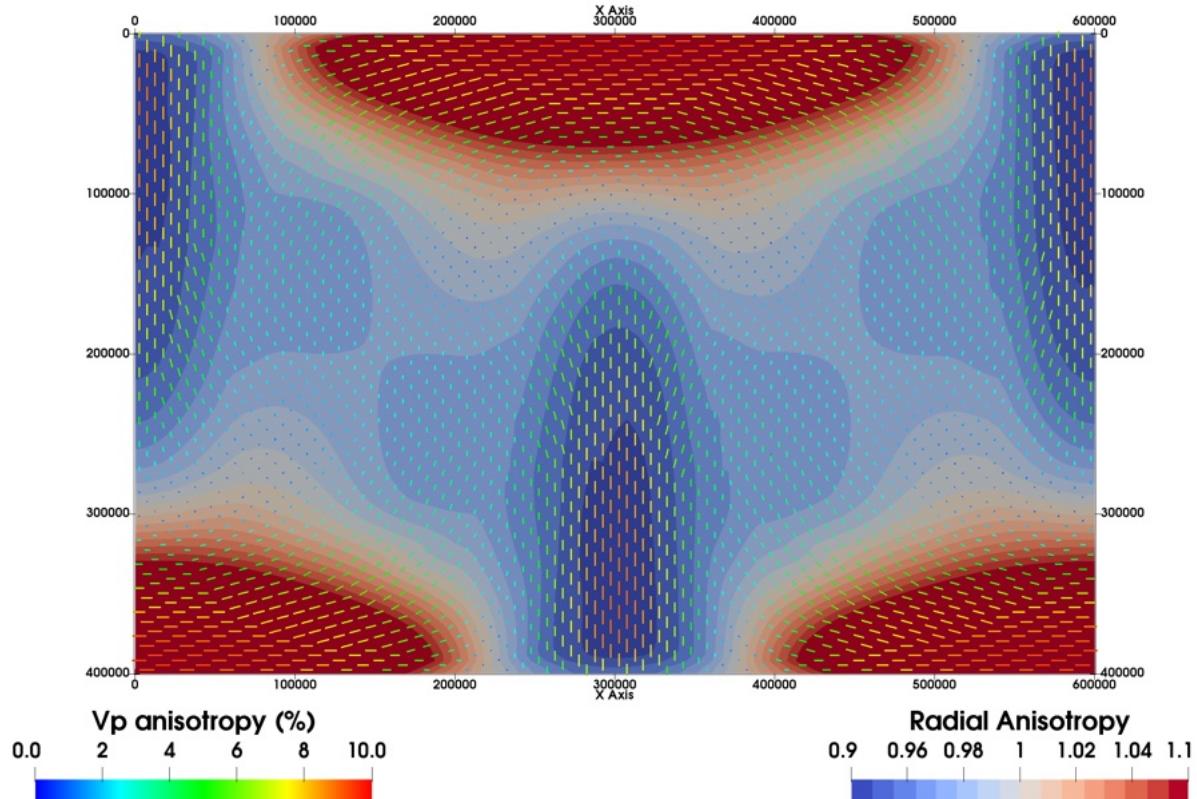


Figure 9.1: Steady-state convection pattern in cartesian coordinates created with `cartesiancells.m`. Radial anisotropy with superimposed $V_{p\max}$.

9.2 Steady-state subduction in 2D polar coordinates

DIRECTORY: /2Dpolar_subduction

Modeling the elastic properties in oceanic settings is fundamental to constrain the development of seismic anisotropy with formation, ageing and subduction of the oceanic lithosphere.

Oceanic plate formation at the ridge and subsequent subduction at the trench is modelled in 2D polar coordinates with a modified version of the petrological-thermo-mechanical code **I2VIS** [Gerya and Yuen, 2003]. The domain is $(\phi, r) = (0 - 40^\circ, 5671 - 6371 \text{ km})$ and discretized with 1001x351 nodes, the ridge-trench distance is 25° , and a constant plate velocity of 4 cm/yr is imposed. The bottom boundary is open, and free slip is imposed on the other 3 boundaries. Free-surface is simulated with a 30 km thick air-layer.

The initial temperature field is defined by a constant temperature in the air layer (273 K), the half-space cooling model in the upper (20 Myr) and lower (0-62.5 Myr) plates, and a 0.5 K/km adiabat with a mantle potential temperature of 1540K.

The rheology is visco-plastic, and a composite low-T/diffusion/dislocation creep mechanism accommodates viscous deformation. Partial dry melting is simulated beneath the ridge according to parametrized solidus/liquidus curves (Fig. 9.2).

The mantle fabrics are computed assuming steady-state conditions for 100 Myr, which is the time required by the crystal aggregates to move from the trench to the subduction zone, including the initial upwelling and final downwelling episodes. The isotropic P-wave distribution for an initial crystal aggregate spacing of 5 km is shown in Fig. 6.2 .

In order to check the role of melt-bearing cracks in seismic anisotropy, the mantle porosity has been saved in `dsd080melt.h5` and the SPO modelling has been activated in `viztomo_input_spo.dat` as shown in Fig. 9.3 for different SPO parameters defined in **VIZTOMO**/`spo_input.dat`. By setting `spomod = 3` in `viztomo_input.dat` and copying files `dsd080melt.h5` and `vtp0080.h5` in the output directory where the file `Cijk10080.h5` is located, it is then possible to superimpose the effect of melt-bearing cracks on that due to LPO.

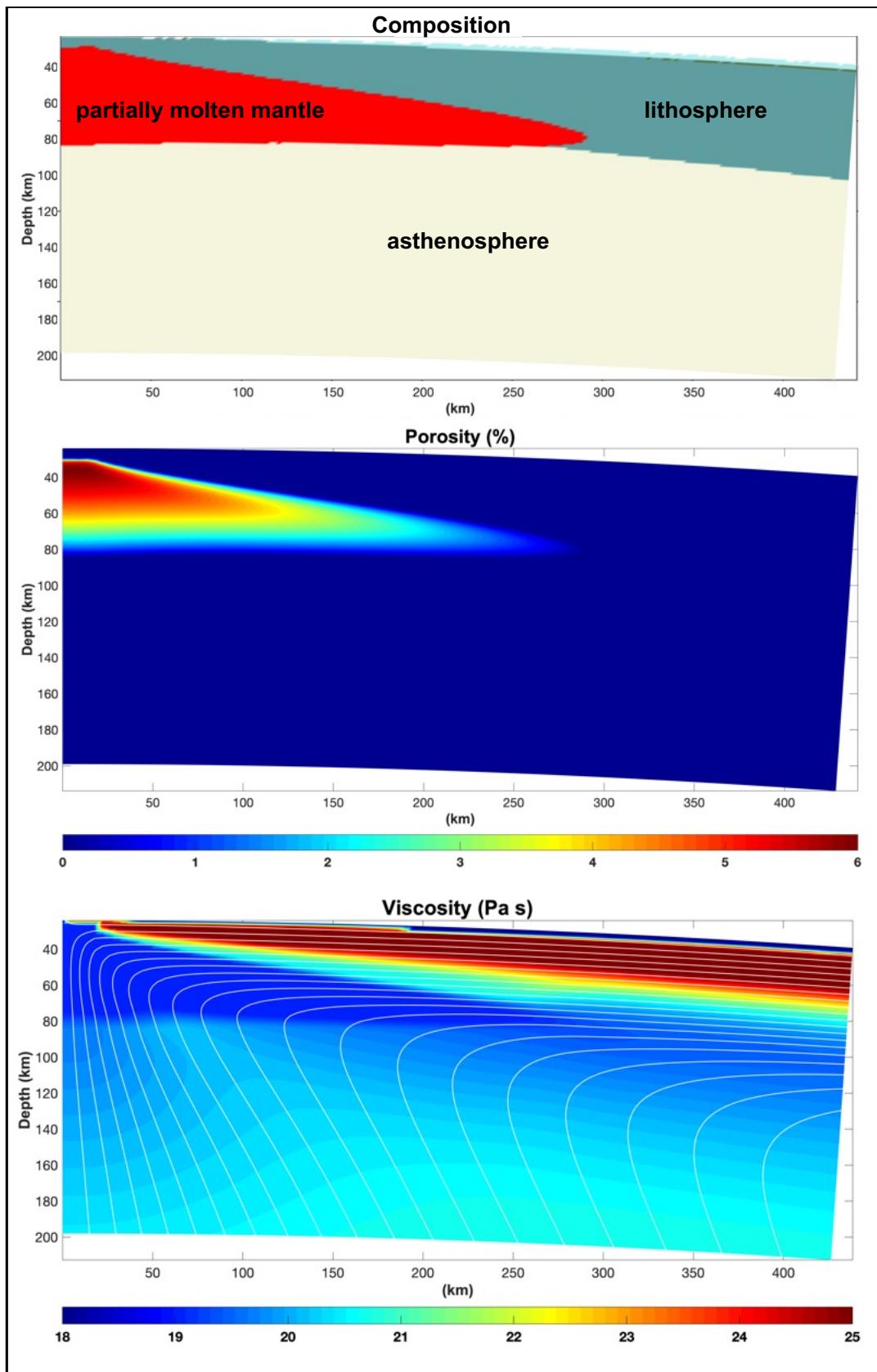


Figure 9.2: Close up of composition, melt fraction and viscosity (with superimposed the streamlines, white lines) of a 2D model of ridge dynamics and oceanic plate formation in polar coordinates.

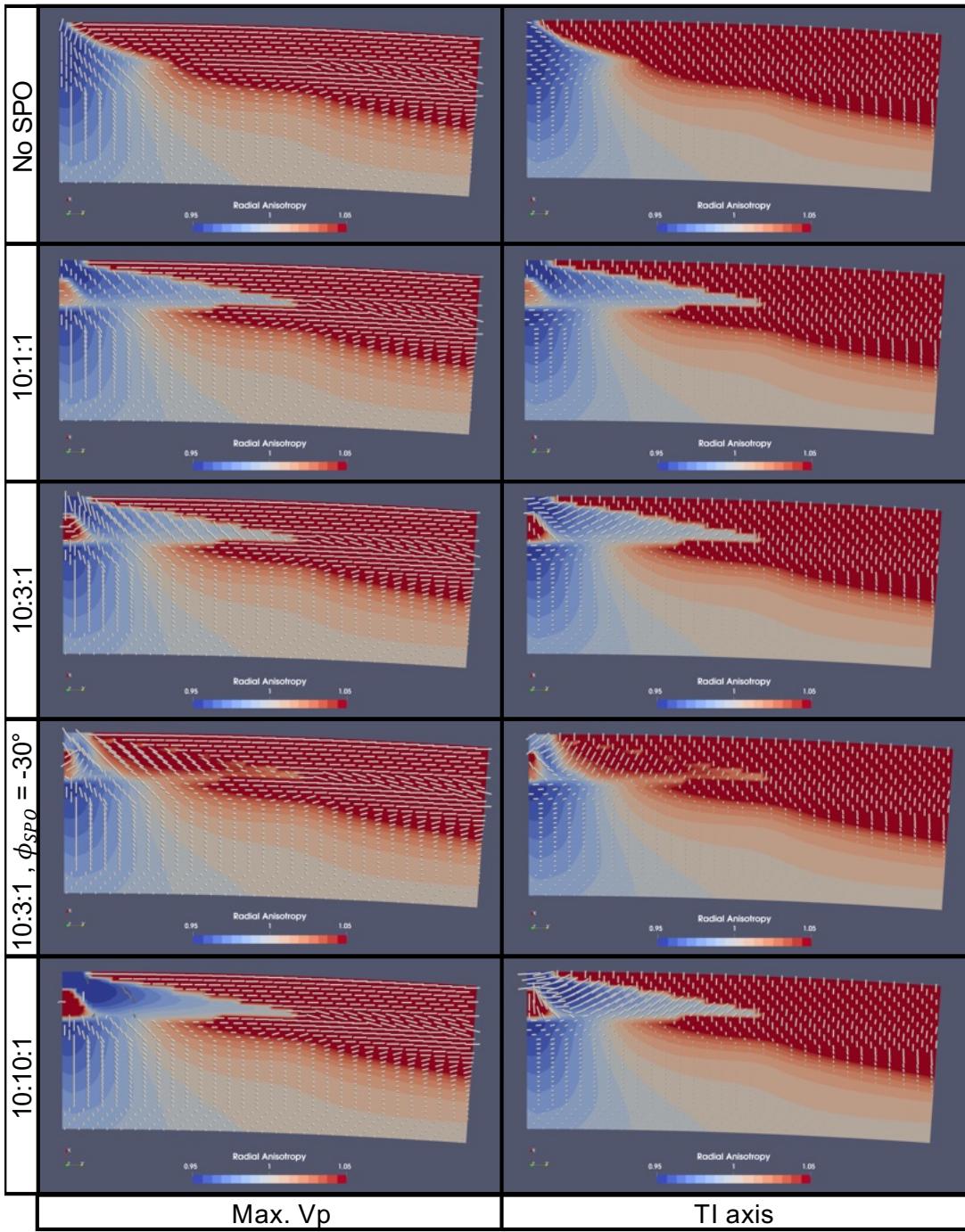


Figure 9.3: Radial anisotropy with superimposed the orientation of Vp_{max} (left) and TI axis (right) for the model in Fig. 9.2. In the top row the anisotropy is due to exclusively the strain-induced LPO (AG-type olivine fabric) estimated with **D-REX_M**, while in the underneath rows the LPO fabrics are superimposed with SPO fabrics due to the presence of melt. SPO control parameter used are **sponmod = 3**; **meltspomod = 1**. The inclusions aspect ratio is indicated on the leftmost column; ϕ_{SPO} is zero everywhere except in the second last row (-30°). Note that for the penny-shaped inclusions (bottom row) the Vp_{max} orientation is randomly oriented in the inclusion flat plane. The TI axis is normal to the flat inclusions, and parallel when cylindrical (second row).

9.3 Steady-state global scale convection in 2D polar coordinates

DIRECTORY: /2Dpolar_convection

A 2D incompressible flow field in cylindrical coordinates is given by the following analytical solution, whereby the radial and tangential velocities are:

$$\begin{aligned} V_r(\phi, r) &= V_{r0} \cos(k_\phi \phi / 2) \sin(k_r \pi r') \\ V_\phi(\phi, r) &= -V_{\phi0}(r) \sin(k_\phi \phi / 2) \cos(k_r \pi r') \\ V_{\phi0}(r) &= 2V_{r0}/k_\phi [\tan(k_r \pi r') + k_r \pi r / \Delta R], \\ r' &= (r - R_{min}) / \Delta R, \end{aligned}$$

where ϕ and r are longitude and radial distance, V_{r0} a constant, k_ϕ and k_r are the tangential and radial wave numbers, $\Delta R = R_{max} - R_{min}$, with R_{min} and R_{max} the bottom and top radial coordinate.

Pressure distribution is:

$$P(r) = P_0 + (R_{max} - r) / \Delta R \cdot P_1$$

Temperature distribution is:

$$\begin{aligned} T(\phi, r) &= T_c(r) + dT(\phi, r) \\ T_c(r) &= T_0 + (R_{max} - r) / \Delta R \cdot T_1 \\ dT(\phi, r) &= dT_{max} * V_r(\phi, r) / V_{r0} \end{aligned}$$

where $P_0, P_1, T_0, T_1, dT_{max}, V_{r0}$ are constants.

The **v**, **p** and **T** fields are computed with the **MATLAB** script `polarcells.m` and saved to `vtvp0001.h5` as a function of the numerical resolution, radial and tangential wave numbers, and P-T conditions. The mantle fabrics computed with **D-REX_M** in steady-state conditions for 100 Myr and visualized with **VIZTOMO** are shown in Fig. 9.4 In global convection models, the vertical boundary should be open, i.e., the domain should be periodic along the tangential direction, thus:

```
dimensions = 2
cartspher = 2
x1periodic = 1
```

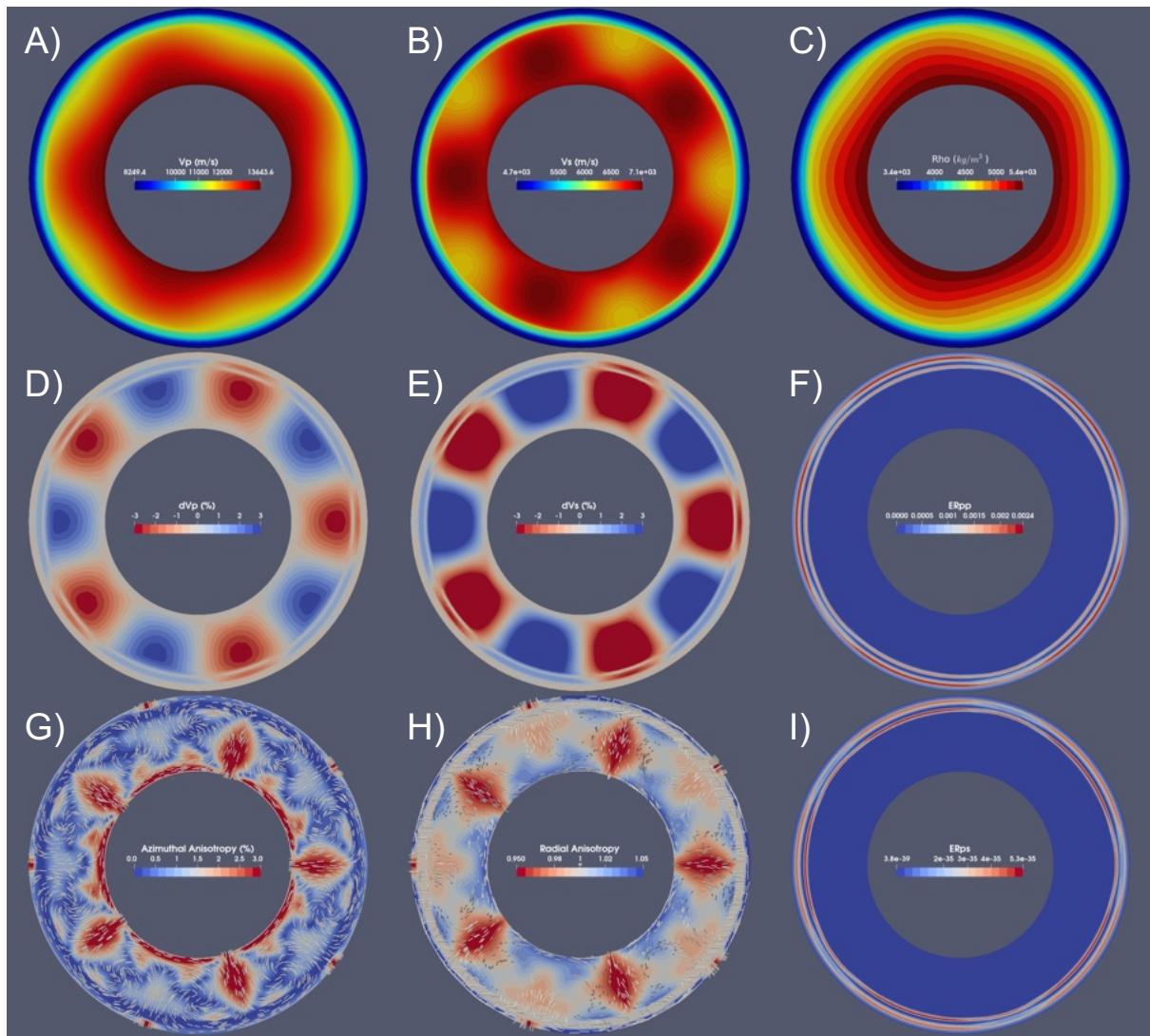


Figure 9.4: Steady-state convection patterns in polar coordinates created with `polarcells.m`. A) P-wave velocity; B) S-wave velocity; C) Density; F) and I) Normalized P-P and P-S reflected energy; D) and E) P- and S-wave anomalies with respect horizontally averaged velocities. G) Azimuthal anisotropy with superimposed fse_{max} ; H) Radial anisotropy with superimposed Vp_{max} .

9.4 Sinking slab in 3D cartesian coordinates

DIRECTORY: /3Dcartesian_sinkingslab

The flow field of a sinking cold horizontal slab is computed with the petrological-thermo-mechanical code **I3MG** [Gerya and Yuen, 2003]. The domain is $(x, y, z) = (2000, 1000, 2000) \text{ km}$ and discretized with $37 \times 37 \times 37$ nodes, and free slip is imposed on all boundaries. The initial temperature field is defined by the half-space cooling model in the 90 km thick, 80 Myr old plate, a constant temperature of 473 K in the 100 km thick, 800 km long and 400 km wide slab centered with respect to the horizontal plane and initially at depth of 200-300 km, and a 0.5 K/km adiabat in the surrounding mantle with a potential temperature of 1540K.

The rheology is visco-plastic, and a composite low-T/diffusion/dislocation creep mechanism accounts for viscous deformation. An increase in viscosity is set at the 660 km depth discontinuity.

The mantle fabrics are computed using 20 vtp*.h5 files, for a total of 2 Myr. An initial (fossil) fabric is imposed on the horizontal slab with fast directions lying in the horizontal plane at 45° from the x and z axes (Fig. 9.5).

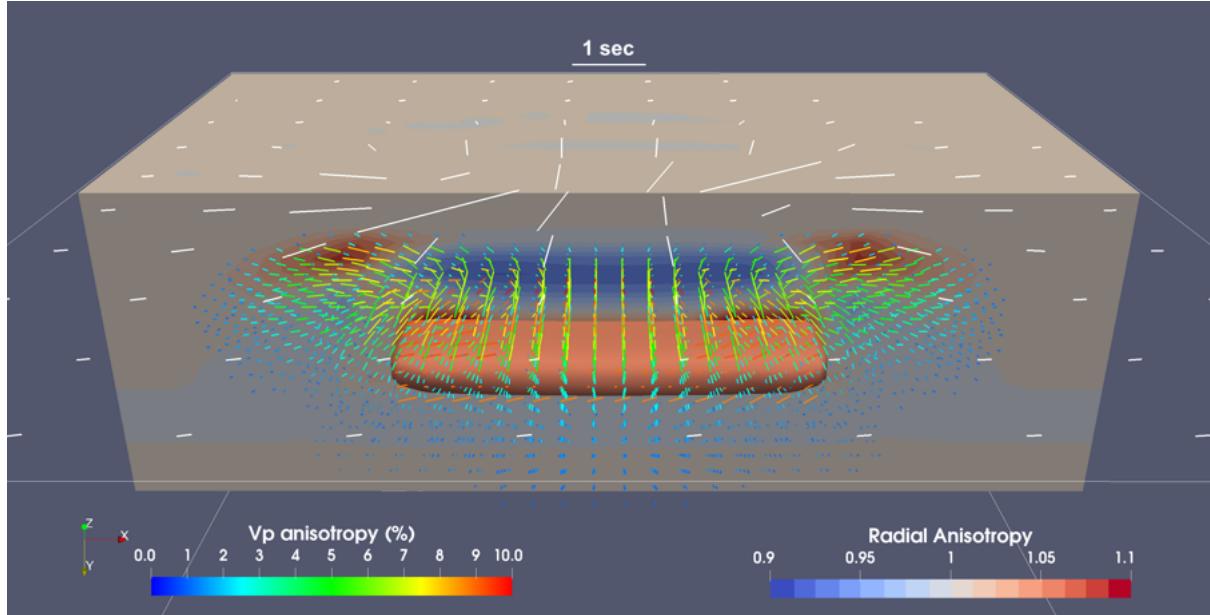


Figure 9.5: Time-dependent flow model of a sinking horizontal slab (contoured by the pink surface) in 3D cartesian coordinates. Radial anisotropy is shown on upper half of the image, while Vp_{\max} in the lower half. The white bars at the surface show SKS splitting, with length proportional to the delay time. The fossil fabric is shown by the orange bars close to the slab. Areas of sub-horizontal flow are characterized by positive radial anisotropy and sub-horizontal Vp_{\max} .

9.5 Sinking slab in 3D spherical coordinates

DIRECTORY: /3Dspherical_sinkingslab

An analogous model setup for the sinking slab in cartesian coordinates has been run in spherical coordinates using a modified version of **I3MG**. Differently from the previous example, the horizontal coordinates in the input files are now given in degrees rather than meters. The domain is $(\phi, r, \theta) = (80 - 100^\circ, 5371 - 6371\text{km}, 80 - 100^\circ)$, and the slab is 8° long and 4° wide (Fig. 9.6). This model is used as an example for the seismic inversions performed with (see ??).

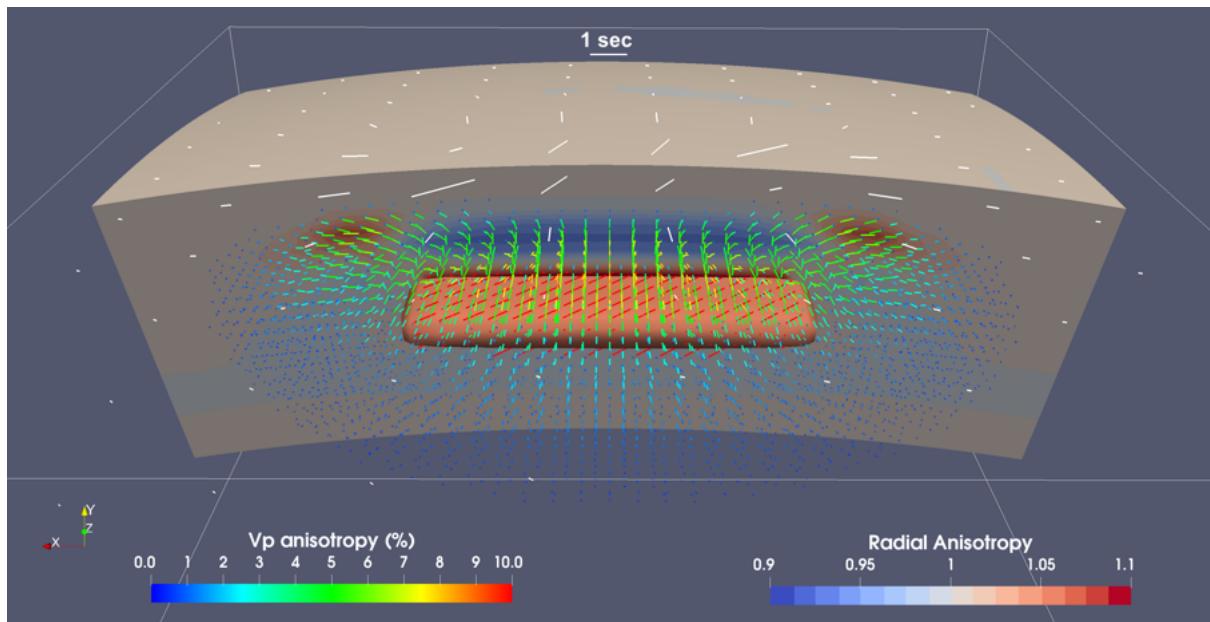


Figure 9.6: Time-dependent flow model of a sinking horizontal slab (contoured by the pink surface) similar to 9.5 but in spherical coordinates.

9.6 Steady-state global scale convection in 3D spherical coordinates

DIRECTORY: /3Dspherical_global

The global scale flow field has been simulated with a modified version of **I3MG** that solves the fundamental conservation equations of continuum mechanics in spherical coordinates using the Yin-Yang grid, an overset grid composed of two identical component grids that are combined in a complementary way to cover a spherical surface with partial overlap on their boundaries [Kageyama and Sato, 2004]. Each component grid is a low-latitude part of the latitude-longitude grid, and therefore the grid spacing is quasi-uniform.

The model is non-dimensional and discretized with 149x101x53x2 nodes, the initial temperature field is characterized by a conductive thermal gradient plus a thermal perturbation that varies along the radial direction with a sinusoidal pattern and laterally with cubic symmetry. The Rayleigh number is 14000, which yields a steady-state flow pattern.

In spherical coordinates, the mantle fabrics and their advection is computed by means of the Yin-Yang grid. The grids span over $-3\pi/4 - d\phi \leq \phi \leq 3\pi/4 + d\phi$ and $\pi/4 - d\theta \leq \theta \leq 3\pi/4 + d\theta$, where $d\phi$ and $d\theta$ are small buffers that ensure grids overlap, and usually taken equal to the grid spacing.

The longitudinal, radial and colatitudinal velocity components (in m/s or its adimensional counterpart), as well P,T and Fd, of the geodynamic model should then be interpolated to the nodes of the two grids as shown in the **MATLAB** script:

/D-REX_M/make_HDF5_vtp_files/makevtpfiles_yinyang.m.

In global convection models, the vertical boundaries should be open, thus **x1periodic** and **x3periodic** must be set to 1. As such, the Yin-Yang grid is activated in the **D-REX_M** parameter input file by setting:

```
dimensions = 3
cartspher = 2
x1periodic = 1
x3periodic = 1
```

The **D-REX_M** parameter input file is configured with adimensional **timemax** and distances, and such that the fabrics are computed only for upper mantle aggregates and without scaling the single crystal elastic tensors by the local P-T conditions. With an aggregate spacing of 80-90 km, about 1.3 millions of aggregates are distributed throughout the mantle, such that about 115 GB of memory needs to be allocated. The

azimuthal anisotropy related to these fabrics is shown in Fig. 9.7, while the P-wave anomalies are shown in the manual cover image.

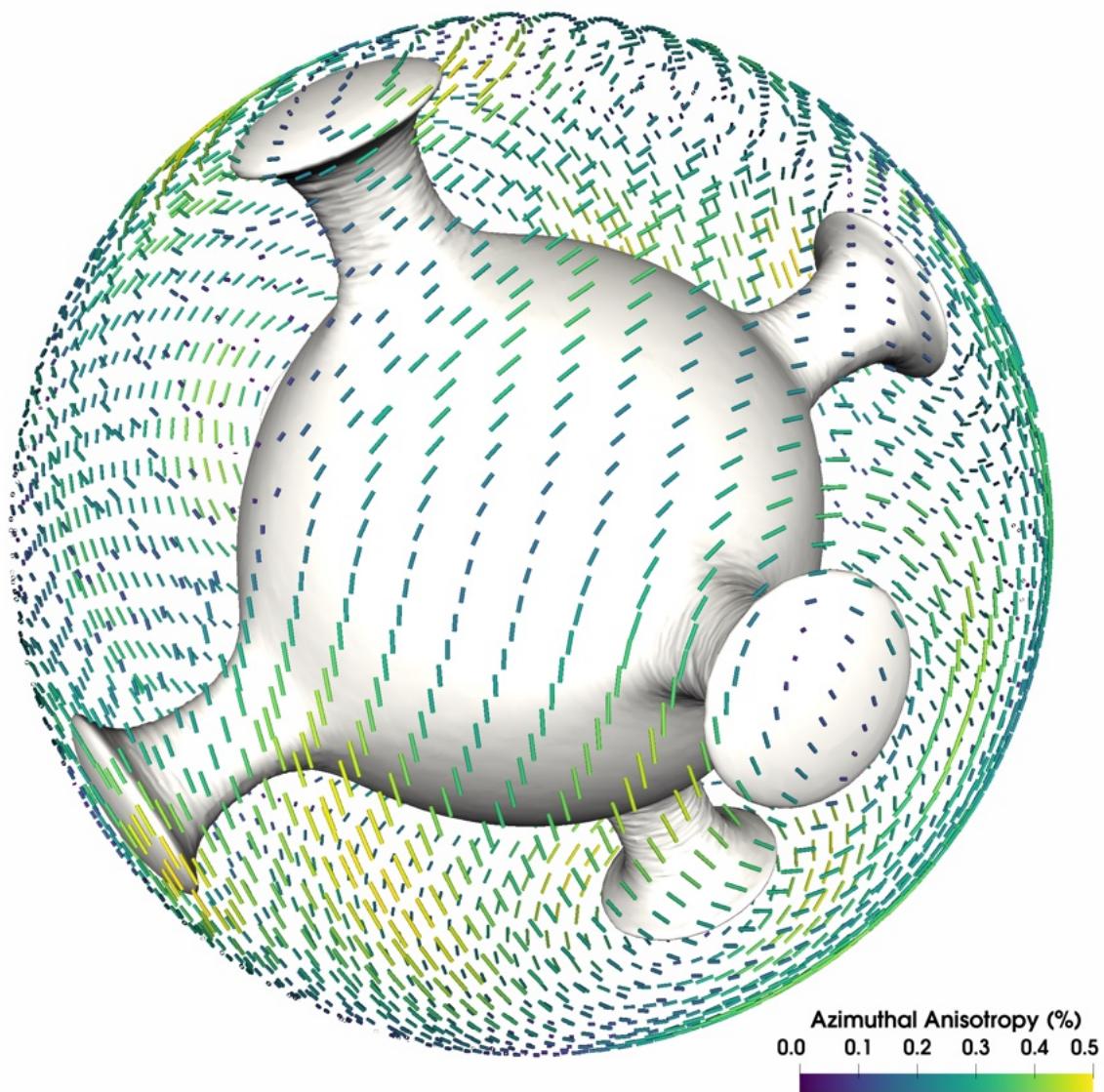


Figure 9.7: Steady-state global convection patterns in 3D spherical coordinates. The contours are $T = 0.5$. The bars indicate azimuthal anisotropy in the upper mantle.

Bibliography

- K. Aki and P. Richards. *Quantitative seismology, 2nd Ed.* University Science Books, 2002.
- G. Backus. Long-wave elastic anisotropy produced by horizontal layering. *J. Geophys. Res.*, 67(11):4427–4440, 1962. doi: 10.1029/JZ067i011p04427.
- T. Becker, V. Schulte-Pelkum, D. Blackman, J. Kellogg, and R. O’Connel. Mantle flow under the western united states from shear wave splitting. *Earth Planet. Sci. Lett.*, 247:235–251, 2006.
- M. Bezada, M. Faccenda, and D. Toomey. Representing anisotropic subduction zones with isotropic velocity models: A characterization of the problem and some steps on a possible path forward. *Geochem. Geophys. Geosyst.*, 2016. doi: 10.1002/2016GC006507.
- Y. Boneh and P. Skemer. The effect of deformation history on the evolution of olivine cpo. *Earth Planet. Sci. Lett.*, 406:213–222, 2014.
- T. Browaeys and S. Chevrot. Decomposition of the elastic tensor and geophysical applications. *Geophys. J. Int.*, 159:667–678, 2004. doi: 10.1111/j.1365-246X.2004.02415.x.
- M. Chai and J. Brown. The elastic constants of a pyrope-grossular-almandine garnet to 20 gpa. *Geophys. Res. Lett.*, 24(5):523–526, 1997.
- M. Chai, J. Brown, and L. Slutski. The elastic constants of an aluminous orthopyroxene to 12.5 gpa. *J. Geophys. Res.*, 102(14):779–785, 1992.
- Z. Chemia, D. Dolejš, and G. Steinle-Neumann. Thermal effects of variable material properties and metamorphic reactions in a three-component subducting slab. *J. Geophys. Res.*, 120:6823–6845, 2015.
- T. Chust, G. Steinle-Neumann, D. Dolejs, B. Schuberth, and H.-P. Bunge. Mma-eos: A computational framework for mineralogical thermodynamics. *J. Geophys. Res.*, 122: 9881–9920, 2017. doi: 10.1002/2017JB014501.

- A. de Montserrat and M. Faccenda. Extrinsic anisotropy of two-phase newtonian aggregates. fabric characterisation and parametrisation. *J. Geophys. Res.*, 126, 2021. doi: <https://doi.org/10.1029/2021JB022232>.
- M. Faccenda. Mid mantle seismic anisotropy around subduction zones. *Phys. Earth Planet. Int.*, 227:1–19, 2014. doi: 10.1016/j.pepi.2013.11.015.
- M. Faccenda and F. Capitanio. Seismic anisotropy around subduction zones: insights from three-dimensional modeling of upper mantle deformation and sks splitting calculations. *Geochem. Geophys. Geosyst.*, 2013. doi: 10.1002/ggge.20055.
- M. Faccenda, A. Ferreira, N. Tisato, C. Lithgow-Bertelloni, L. Stixrude, and G. Pennacchioni. Extrinsic anisotropy in a compositionally heterogeneous mantle. *J. Geophys. Res.*, 124, 2019. doi: 10.1029/2018JB016482.
- M. Faccenda, B. VanderBeek, A. de Montserrat, J. Yang, F. Rappisi, and N. Ribe. Econman: an open-source package for geodynamic and seismological modeling of mechanical anisotropy. *Solid Earth, egusphere-2024-299*, 2024.
- A. Ferreira, M. Faccenda, W. Sturgeon, S.-J. Chang, and L. Schardong. Ubiquitous lower-mantle anisotropy beneath subduction zones. *Nat. Geo.*, 12:301–306, 2019. doi: 10.1038/s41561-019-0325-7.
- T. Gerya and D. Yuen. Characteristics-based marker-in-cell method with conservative finite-differences schemes for modeling geological flows with strongly variable transport properties. *Phys. Earth Planet. Sci.*, 140:293–318, 2003.
- M. Hirschmann, T. Kogiso, M. Baker, and E. Stopler. Alkalic magmas generated by partial melting of garnet pyroxenite. *Geology*, 31:481–484, 2003.
- B. Hornby, L. Schwartz, and J. Huson. Anisotropic effective-medium modeling of the elastic properties of shales. *Geophysics*, 59:1570–1583, 1994.
- J. Hu, M. Faccenda, and L. Liu. Subduction-controlled mantle flow and seismic anisotropy in south america. *Earth Planet. Sci. Lett.*, 470:3–24, 2017. doi: 10.1016/j.epsl.2017.04.027.
- D. Isaak. High-temperature elasticity of iron-bearing olivines. *J. Geophys. Res.*, 97: 1871–1885, 1992.
- J. Jackson, S. Sinogeikin, and J. Bass. Sound velocities and single-crystal elasticity of orthoenstatite to 1073k at ambient pressure. *Phys. Earth Planet. Int.*, 171:1–12, 2007.
- A. Kageyama and T. Sato. “yin-yang grid”: An overset grid in spherical geometry. *Geochem. Geophys. Geosys.*, 5, 2004. doi: 10.1029/2004GC000734.

- E. Kaminski, N. Ribe, and T. Browaeys. D-rex, a program for calculation of seismic anisotropy due to crystal lattice preferred orientation in the convective upper mantle. *Geophys. J. Int.*, 158:744–752, 2004. doi: 10.1111/j.1365-246X.2004.02308.x.
- B. Karki, R. Wentzcovitch, S. de Gironcoli, and S. Baroni. High-pressure lattice dynamics and thermoelasticity of mgo. *Phys. Rev. B*, 61:8793–8800, 2000.
- R. Katz, M. Spiegelman, and B. Holtzman. The dynamics of melt and shear localization in partially molten aggregates. *Nature*, 442:676–679, 2006.
- P. Kelemen and M. Ghiorso. Assimilation of peridotite in zoned calc-alkaline plutonic complexes: evidence from the big jim complex, washington cascades. *Contrib. Min.*, 94:12–28, 1986.
- D. Mainprice. Modelling the anisotropic seismic properties of partially molten rocks found at mid-ocean ridges. *Phys. Rev. B*, 279(1-4):161–179, 1997.
- D. Mainprice, R. Hielscher, and H. Schaeben. Calculating anisotropic physical properties from texture data using the mtex open source package. In D. Prior, E. Rutter, and D. Tatham, editors, *Deformation Mechanisms: Rheology and Tectonics: Microstructures. Mechanics and Anisotropy*, pages 175–192. Geological Society, London, 2011.
- R. McLaughlin. A study of the differential scheme for composite materials. *Int. J. Eng. Sci.*, 15:237–244, 1977.
- A. Oganov and S. Ono. Theoretical and experimental evidence for a post-perovskite phase of mgsio₃ in earth's d layer. *Nature*, 430:445–448, 2004a.
- A. Oganov and S. Ono. Theoretical and experimental evidence for a post-perovskite phase of MgSiO₃ in earth's d" layer. *Nature*, 430:445–448, 2004b.
- P. Olson, D. Yuen, and D. Balsiger. Mixing of passive heterogeneities by mantle convection. *J. Geophys. Res.*, 89(B1):425–436, 1984. doi: 10.1029/JB089iB01p00425.
- A. Sieminski, H. Paulssen, J. Trampert, and J. Tromp. Finite-frequency sks splitting: Measurement and sensitivity kernels. *Bull. Seismol. Soc. Am.*, 98(5):1797–1810, 2008. doi: 10.1785/0120070297.
- S. Sinogeikin and J. Bass. Elasticity of pyrope and majorite – pyrope solid solutions to high temperatures. *Earth Planet. Sci. Lett.*, 203:549–555, 2002.
- S. Sinogeikin, J. Bass, and K. Katsura. Single-crystal elasticity of ringwoodite to high pressures and high temperatures: implications for 520 km seismic discontinuity. *Phys. Earth Planet. Int.*, 136:41–66, 2003.

- W. Sturgeon, A. Ferreira, M. Faccenda, S.-J. Chang, and L. Schardong. On the origin of radial anisotropy near subducted slabs in the midmantle. *Geochem. Geophys. Geosyst.*, 2019. doi: 10.1029/2019GC008462.
- B. VanderBeek and M. Faccenda. Imaging realistic upper mantle anisotropy with teleseismic p-wave delays: Insights from tomographic reconstructions of subduction simulations. *Geophys. J. Int.*, 225:2097–2119, 2021.
- B. VanderBeek, R. Lo Bue, F. Rappisi, and M. Faccenda. Imaging upper mantle anisotropy with travel-time and splitting intensity observations from teleseismic shear waves: Insights from tomographic reconstructions of subduction simulations. *Geophys. J. Int.*, 235:2640–2670, 2023.
- R. Wentzcovitch, B. Karki, M. Cococcioni, and S. de Gironcoli. Properties of $MgSiO_3$ -perovskite: Insights on the nature of the earth's lower mantle. *Phys. Rev. Lett.*, 92, 2004.
- R. Workman and S. Hart. Major and trace element composition of the depleted morb mantle. *Earth Planet. Sci. Lett.*, 231:53–72, 2005.
- C. Zha, T. Duffy, H. Mao, R. Downs, R. Hemley, and D. Weidner. Single-crystal elasticity of βMg_2SiO_4 , to the pressure of the 410km seismic discontinuity in the earth's mantle. *Earth Planet. Sci. Lett.*, 147:E9–E15, 1997.
- C. Zha, T. Duffy, R. Downs, H. Mao, and R. Hemley. Brillouin scattering and x-ray diffraction of san carlos olivine: direct pressure determination to 32 gpa. *Earth Planet. Sci. Lett.*, 159:25–33, 1998.
- Z. Zhang, L. Stixrude, and J. Brodholt. Elastic properties of $MgSiO_3$ -perovskite under lower mantle conditions and the composition of the deep earth. *Earth Planet. Sci. Lett.*, 379:1–12, 2013.
- Z. Zhang, S. Cottaar, L. Tao, S. Stackhouse, and B. Militzer. High-pressure, temperature elasticity of *fe*- and *al*-bearing $MgSiO_3$: Implications for the earth's lower mantle. *Earth Planet. Sci. Lett.*, 434:264–273, 2016.
- W.-Y. Zhou, M. Hao, J. Zhang, B. Chen, R. Wang, and B. Schamndt. Constraining composition and temperature variations in the mantle transition zone. *Nat. Coom.*, 13:1094, 2022. doi: 10.1038/s41467-022-28709-7.

A1 Appendix

Table A1.1: Slip systems of abundant anisotropic mantle phases present in **D-REX_S** and **D-REX_M**

Slip system [uvw](hkl)	Olivine	Enstatite	Wadleyite	Bridgmanite	post-Perovskite
[100](010)	★	-	★	★	★
[100](001)	★	-	★	★	★
[100](011)	-	-	★	-	-
[100](021)	-	-	★	-	-
[010](100)	-	-	-	★	★
[010](001)	-	-	-	★	★
[001](100)	★	★	-	★	★
[001](010)	★	-	★	★	★
[001]{110}	-	-	-	★	★
<110>(001)	-	-	-	★	-
<110>{110}	-	-	-	★	★
1/2<111>{101}	-	-	★	-	-

Table A1.2: Elastic moduli (GPa) and their P-T derivatives of the mineral phases used in this study. Temperature derivatives are scaled by 10^{-2} GPaK $^{-1}$, while cross derivatives by 10^{-3} K $^{-1}$

Mineral	Idx*	Moduli	C11	C22	C33	C12	C13	C23	C44	C55	C66	Source [†]
Olivine	1	M	320.71	197.25	234.32	69.84	71.22	74.80	63.77	77.67	78.36	(1)
		$\partial M/\partial P$	5.41	5.26	3.78	1.88	1.53	1.60	1.08	1.42	1.80	(2)
		$\partial M/\partial T$	-4.02	-3.10	-3.53	-1.14	-0.96	-0.72	-1.26	-1.30	-1.56	(1)
Enstatite	2	M	236.90	180.50	230.40	79.60	63.20	56.80	84.30	79.40	80.10	(3)
		$\partial M/\partial P$	10.27	8.87	11.7	6.22	6.63	7.26	1.23	0.75	2.78	(3)
		$\partial^2 M/\partial P^2$	-0.47	-0.38	-0.53	-0.33	-0.26	-0.31	0	0	0	(3)
		$\partial M/\partial T$	-3.64	-3.43	-5.70	-1.52	-2.29	-1.63	-1.23	-1.58	-1.51	(4)
Wadsleyite	3	M	370.50	367.70	272.40	65.60	95.20	105.10	111.20	122.50	103.10	(5)
		$\partial M/\partial P$	5.21	6.83	8.06	4.06	3.30	3.30	1.39	0.00	1.88	(5)
		$\partial M/\partial T$	-4.02	-3.10	-3.53	-1.14	-0.96	-0.72	-1.26	-1.30	-1.56	(1)
Wadsleyite Fe = 0.94 H ₂ O = 0.15 wt%	4	M	341.00	358.00	224.00	75.00	99.00	102.00	106.00	109.00	90.80	(6)
		$\partial M/\partial P$	7.80	5.37	10.5	4.60	2.33	4.20	0.87	0.84	2.43	(6)
		$\partial^2 M/\partial P^2$	-0.26	0	-0.46	-0.11	0	-0.14	0	0	-0.053	(6)
		$\partial M/\partial T$	-2.90	-3.60	-4.00	-0.60	-1.90	-1.20	-1.80	-1.10	-1.30	(6)
Ringwoodite	5	M	329.00	-	-	118.00	-	-	130.00	-	-	(7)
		$\partial M/\partial P$	6.20	-	-	0.80	-	-	2.80	-	-	(7)
		$\partial M/\partial T$	-4.90	-	-	-0.70	-	-	-1.30	-	-	(7)
Garnet	6	M	299.10	-	-	106.70	-	-	93.70	-	-	(8)
		$\partial M/\partial P$	6.54	-	-	2.87	-	-	1.72	-	-	(8)
		$\partial M/\partial T$	-3.05	-	-	-0.58	-	-	-0.71	-	-	(9)
Bridgmanite	7	M	484.00	542.00	477.00	146.00	146.00	162.00	195.00	172.00	151.00	(9)
		$\partial M/\partial P$	3.40	5.30	5.20	3.30	2.40	2.50	1.40	0.80	1.40	(10)
		$\partial M/\partial T$	-1.80	-3.30	-2.20	-0.40	-0.20	-0.50	-2.20	-0.60	-1.60	(10)
		$\partial^2 M/\partial P\partial T$	-0.017	-0.016	-0.016	-0.006	-0.006	-0.007	-0.0005	-0.0005	-0.004	(10)
Bridgmanite P ₀ = 34 GPa T ₀ = 1500 K	8	M	581.42	669.06	626.31	249.02	212.93	239.01	228.63	220.26	183.89	(11)
		$\partial M/\partial P$	3.73	5.54	5.47	3.30	2.48	2.53	1.38	0.85	1.56	(11)
		$\partial M/\partial T$	-3.99	-5.53	-4.32	-1.76	-0.64	-0.26	-2.12	-1.25	-1.96	(11)
		$\partial^2 M/\partial P\partial T$	-0.16	-0.044	-0.051	-0.017	-0.06	-0.035	-0.017	-0.044	-0.042	(10)
Bridgmanite P ₀ = 48 GPa T ₀ = 1500 K	9	M	601.83	698.17	652.27	271.81	236.00	265.58	233.53	205.41	190.43	(11)
		$\partial M/\partial P$	3.73	5.54	5.47	3.30	2.48	2.53	1.38	0.85	1.56	(11)
		$\partial M/\partial T$	-3.14	-5.94	-4.53	-1.94	-0.55	-0.59	-2.15	-1.26	-2.50	(11)
		$\partial^2 M/\partial P\partial T$	-0.66	-0.026	-0.034	-0.158	-0.008	-0.160	-0.001	-0.036	-0.002	(11)
Ferropericlase	10	M	300.0	-	-	93.60	-	-	147.0	-	-	(12)
		$\partial M/\partial P$	9.56	-	-	1.45	-	-	1.03	-	-	(12)
		$\partial M/\partial T$	-5.98	-	-	0.89	-	-	-0.88	-	-	(12)
		$\partial^2 M/\partial P\partial T$	0.56	-	-	0.06	-	-	0.20	-	-	(12)
post- Perovskite P ₀ = 99.95 GPa T ₀ = 1500 K	11	M	1124.75	884.22	1124.32	350.69	287.53	461.05	254.57	231.19	344.05	(13)
		$\partial M/\partial P$	5.85	2.70	5.44	3.06	3.00	2.33	1.37	1.27	2.06	(13)
		$\partial M/\partial T$	1.01	-5.24	-4.90	-0.92	-0.51	-1.78	-0.81	-2.33	-2.19	(13)
		$\partial^2 M/\partial P\partial T$	-0.937	0.709	0.797	0.60	-0.801	0.692	-0.107	-0.186	-0.16	(13)

* Index to be set in **single_crystal_elastic_db**

[†] (1) Isaak [1992]; (2) Zha et al. [1998]; (3) Chai et al. [1992]; (4) Jackson et al. [2007]; (5) Zha et al. [1997]; (6) Zhou et al. [2022]; (7) Sinogeikin et al. [2003]; (8) Chai and Brown [1997]; (9) Sinogeikin and Bass [2002]; (10) Wentzcovitch et al. [2004]. Derivatives at (100 GPa, 300 K); (11) Zhang et al. [2013].

Derivatives extrapolated by interpolation; (12) Karki et al. [2000]; (13) Zhang et al. [2016]. Derivatives extrapolated by interpolation;

Table A1.3: Composition (in mol %) used in **MMA_EoS** to compute lookup tables for different mantle and basalt compositions

Lithology	Idx*	SiO ₂	MgO	FeO	CaO	Al ₂ O ₃	Na ₂ O	Source [†]
Dunite	1	40.43	42.45	16.07	0.51	0.51	0.04	(1)
Hartzburgite	2	36.54	56.17	5.71	0.99	0.59	0.00	(2)
Pyrolite	3	38.71	49.85	6.17	2.94	2.22	0.11	(3)
Basalt	4	51.76	15.11	6.59	14.39	10.39	1.76	(2)
Pyroxenite	5	46.46	16.99	7.95	11.70	15.48	1.43	(4)

* Index to be set in **eosmod**

† (1) Kelemen and Ghiorso [1986]; (2) Chemia et al. [2015]; (3) Workman and Hart [2005]; (4) Hirschmann et al. [2003];