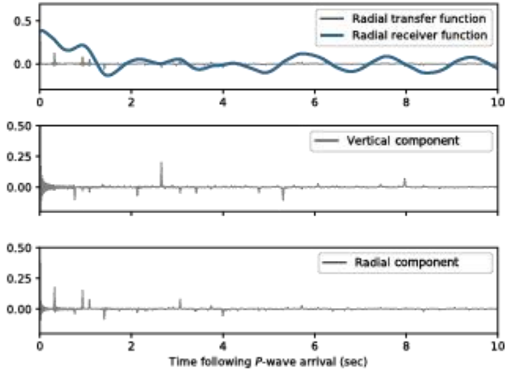
Tutorial 5: Modelling teleseismic receiver functions

OBS training workshop, VUW, April 14-16, 2025 https://github.com/paudetseis

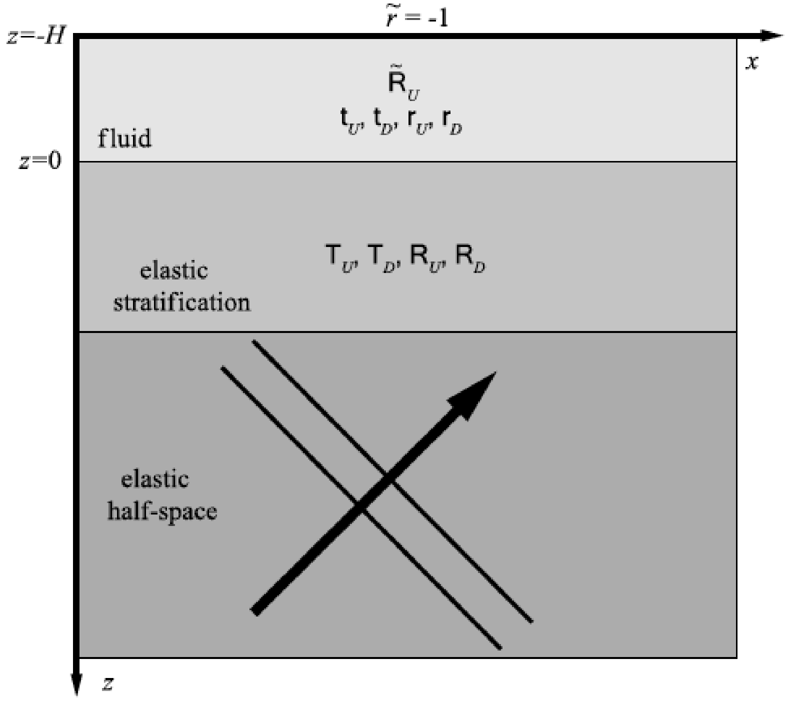
telewavesim



Software for teleseismic body wave modeling through stacks of anisotropic layers:

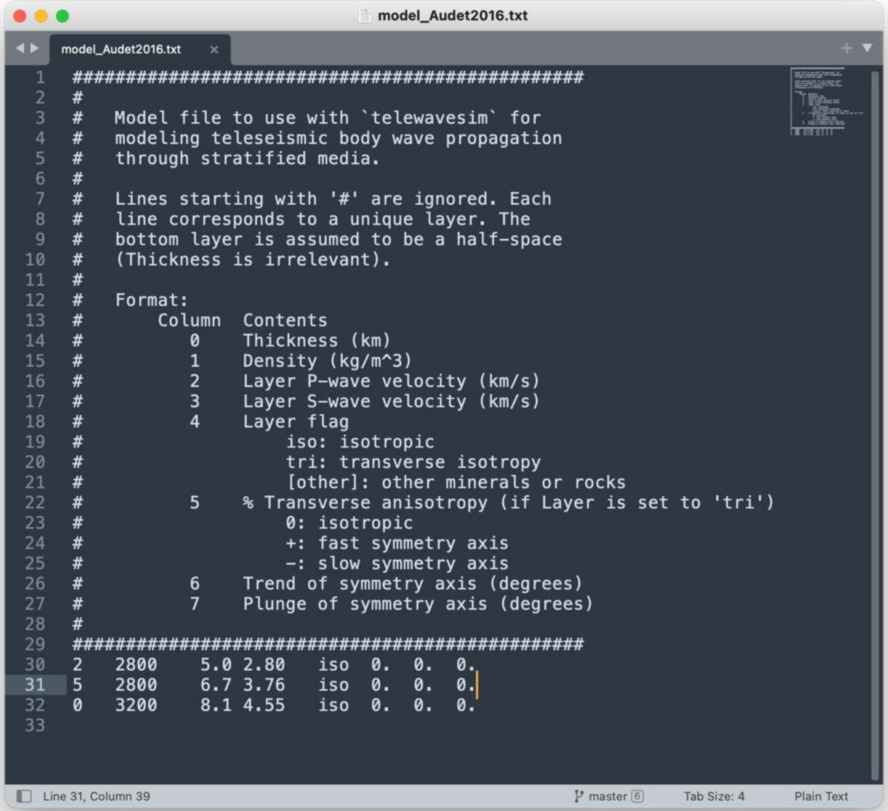
1. Calculates Green's functions for plane (body) waves
2. Can be used for receiver functions, shear wave splitting, etc.
3. Can include water (ocean) layer

Method



- Telewavesim is based on a reflectivity formulation (matrix propagator method).
- The model is built by specifying, at minimum, each layer's thickness, Vp, Vs, and density
- Additional properties include:
  - Layer anisotropy from elastic tensors of rocks, oriented in space
  - Hexagonal anisotropy (transverse isotropy)
- It is possible to include a homogeneous water layer

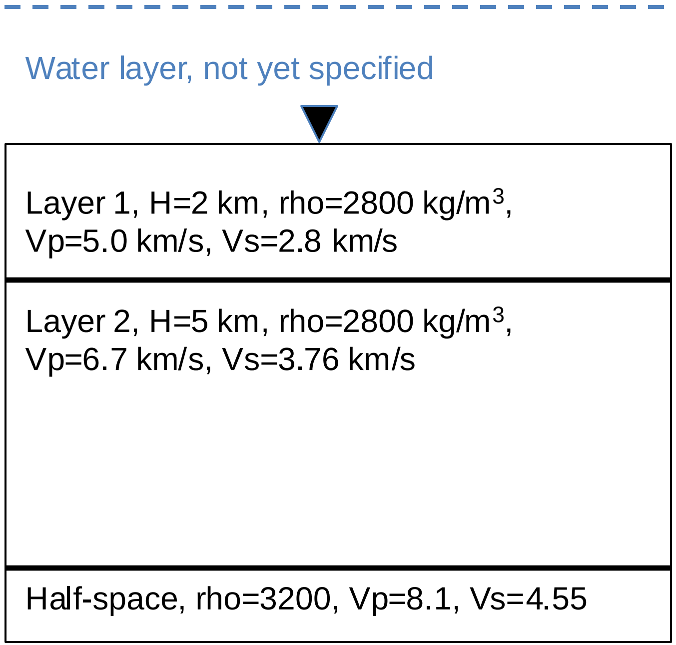
Model design



model_Audet2016.txt

```
#############################################
#
#   Model file to use with `telewavesim` for
#   modeling teleseismic body wave propagation
#   through stratified media.
#
#   Lines starting with '#' are ignored. Each
#   line corresponds to a unique layer. The
#   bottom layer is assumed to be a half-space
#   (Thickness is irrelevant).
#
#   Format:
#       Column  Contents
#          0    Thickness (km)
#          1    Density (kg/m^3)
#          2    Layer P-wave velocity (km/s)
#          3    Layer S-wave velocity (km/s)
#          4    Layer flag
#                   iso: isotropic
#                   tri: transverse isotropy
#                   [other]: other minerals or rocks
#          5    % Transverse anisotropy (if Layer is set to 'tri')
#                   0: isotropic
#                   +: fast symmetry axis
#                   -: slow symmetry axis
#          6    Trend of symmetry axis (degrees)
#          7    Plunge of symmetry axis (degrees)
#
#############################################
2   2800    5.0 2.80    iso  0.  0.  0.
5   2800    6.7 3.76    iso  0.  0.  0.
0   3200    8.1 4.55    iso  0.  0.  0.
```

Line 31, Column 39 | master 6 | Tab Size: 4 | Plain Text



Model design

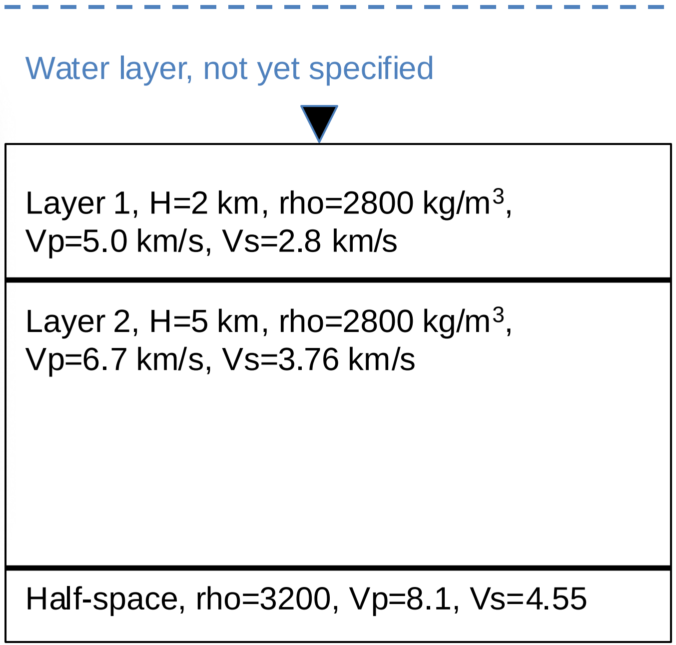
model_design.py

```python
from telewavesim import utils

# Load model from file
model = utils.read_model('model_Audet2016.txt')

# Define the same model using a Python script
thick = [2., 5., 0]         # Half-space layer thickness is irrelevant
rho = [2800., 2800., 3300.]   # kg/m^3
vp = [5.0, 6.7, 8.1]          # km/s
vs = [2.8, 3.76, 4.55]        # km/s
flag = ['iso', 'iso', 'iso']  # All layers are isotropic
model = utils.Model(thick, rho, vp, vs, flag)
```

pycodestyle(w:1), no newline at end of file, Line 12, Column 46 Tab Size: 4 Python



Experimental setup

- For each experiment, we must specify the sampling (in sec), number of points, and wave type ('P' or 'S'): `dt, npts, wtype`.

- To include a water layer, we specify its thickness (depth below the seafloor, km) and geophysical properties (density, kg/m^3, and P-wave velocity, km/s): `dp, rhof, c`

- Then, we select the source-station geometry (slowness, s/km, and back-azimuth, degrees): `slow, baz`

- The final step is to run the corresponding function:

```
trxyz = ut.run_plane(model, slow, npts, dt, baz=baz, wvtype=wvtype,
obs=True, dp=dp, c=c, rhof=rhof)
```

Receiver functions

- The result is a stream containing 3 traces: N-E-Z. To obtain receiver functions, we first calculate transfer functions:

```
tfs = ut.tf_from_xyz(trxyz, pvh=False)
```

- The stream tfs contains the radial (`tfs[0]`) and transverse (`tfs[1]`) transfer functions. Receiver functions are simply filtered transfer functions:

```
tfs.filter('bandpass',freqmin=f1, freqmax=f2, corners=2, zerophase=True)
```

- To model RFs from different slowness or back-azimuth values, we simply call the function inside a loop:

```
slow = [0.04, 0.04, 0.06]
baz = [10., 90., 95.]
trR = Stream(); trT = Stream()
for ss, bb in zip(slow, baz):
    trxyz = ut.run_plane(model, ss, npts, dt, baz=bb, wvtype=wvtype, obs=False)
    tfs = ut.tf_from_xyz(trxyz, pvh=False)
    trR.append(tfs[0]); trT.append(tfs[1])
```