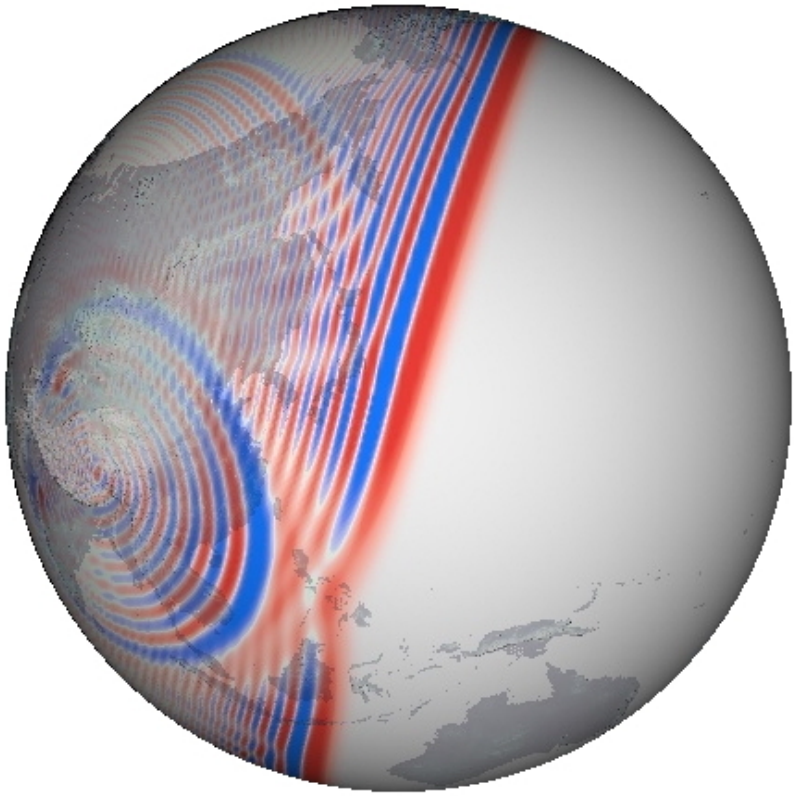
membraneSphere

User manual



membraneSphere
User Manual

Version 1.2

Table of Contents

Chapter 1

Introduction

`membraneSphere` is a software package to simulate waves on a spherical membrane. It implements the membrane wave method introduced by Tanimoto (1990). Membrane waves are an analogue for seismic surface waves. The zero-thickness sphere is discretized by a geodesic grid (Tape, 2003). Wave propagation on this sphere is solved by a finite-difference scheme for such hexagonal grids (Tape, 2003; Heikes & Randall, 1995a).

If you intent to use `membraneSphere`, please reference the following articles:

Tanimoto, T., 1990. *Modelling curved surface wave paths: membrane surface wave synthetics*, *Geophys. J. Int.,* **102**, 89–100.

Tape, C. H., 2003. *Waves on a Spherical Membrane*, M.Sc. thesis, University of Oxford, U.K.

Peter, D., C. Tape, L. Boschi and J. H. Woodhouse, 2007. *Surface wave tomography: global membrane waves and adjoint methods*, *Geophys. J. Int.,* , *171*: p. 1098-1117.

Chapter 2

Getting started

The codes are written in Fortran90 and require MPI compiler and libraries. Before compilation with make, you should run the configuration ./configure and use the appropriate compiler flags for your system (such as FC, MPIFC, FCFLAGS or LDFLAGS). If you want to use double precision instead of the default single precision in the package, you would add `--enable-double-precision` to the configuration. After the configuration, you can always modify the `Makefile` settings as well directly to fit your installation.

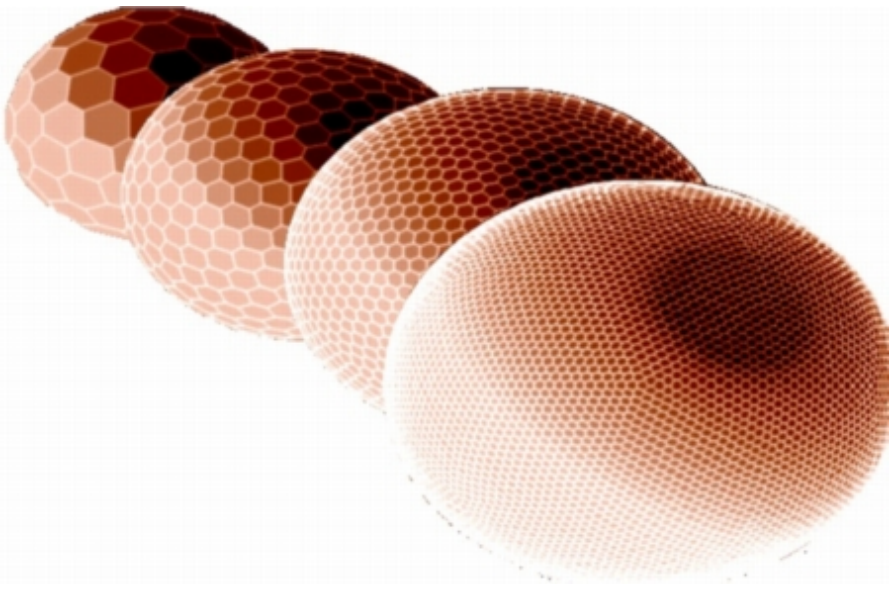
In `src/include/commonModules.f90` you can choose further details about the source and the filtering parameters prior to compilation. To create the binaries, use the command make.

In `Parameter_Input` choose the physical model (grid refinement level, simulation times and wave type) and set the source/receiver geometry. You can also place a single scatterer and/or use a heterogeneous background phase-velocity model. These parameters are read in when the program starts. They can be changed without the need of recompilation of the program.

Chapter 3

Using a mesh

There are data files containing the coordinations of grid points and cells for a geodesic grid (Tape, 2003) located in the directory `data/griddata/`. To choose a certain refinement level of the grid, you set the parameter **LEVEL** in the file `Parameter_Input` to a value between 0 and 6.



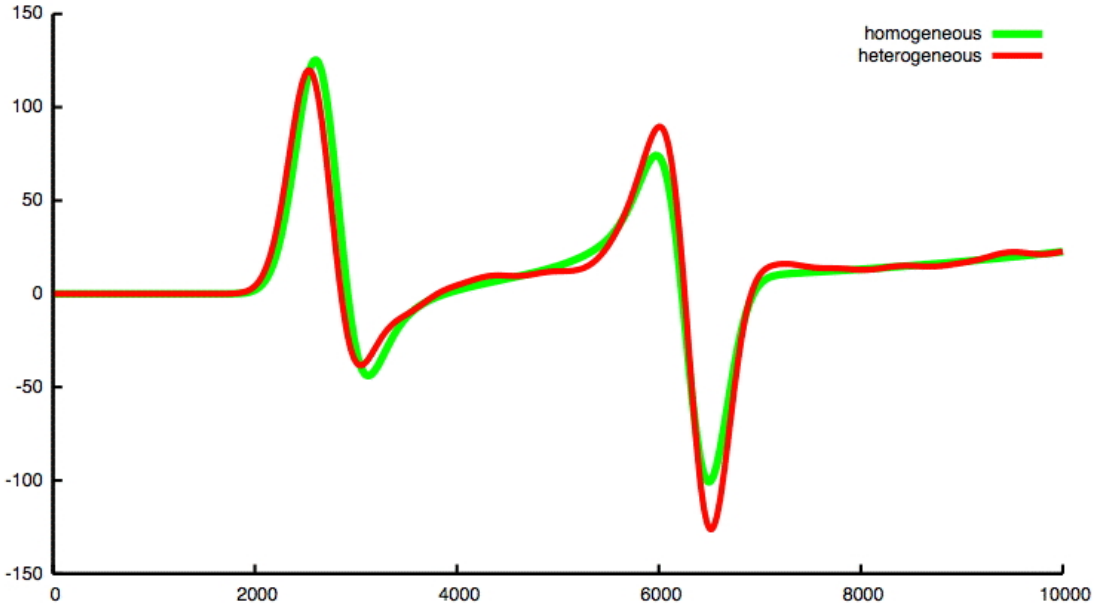
Chapter 4

Performing a simulation

The main executable is `propagation`. It calculates a simulation of membrane waves propagating over a sphere. It reads the main input file `Parameter_Input`:

- **LEVEL** sets the refinement level of the hexagonal grid. It can be an integer value between 0 (coarsest) to 6 (finest grid).
- **FIRSTTIME** determines the starting time (in seconds) of the simulation. In order to properly account for the analytical source (Tape, 2003), the default is set to -1000.0 s.
- **LASTTIME** determines the end time (in seconds) of the simulation.
- **CPHASE** sets the phase-velocity of the seismic wave which will be simulated. This value will determine the background phase-velocity in (km/s) set on the membrane.

  It is given in as a combination format, first of the type, either Rayleigh (R) or Love (L) wave, and the wave period (available wave periods in seconds are: 35, 37, 40, 45, 50, 60, 75, 100, 150, 200, 250 or 300), e.g. if one wants Love waves at 150 s period it is set to *L150*.
- **SOURCE** sets latitude and longitude (in degrees) for the source location.
- **RECEIVER** sets latitude and longitude (in degrees) for the receiver location. The simulation will output the wave displacements calculated for this location.



- **MANYRECEIVERS** can be either *true* or *false* depending if one wants to run a simulation with a geometrical optimization where many receivers were placed on the same latitude in order to perform a direct sensitivity kernel calculation (see Peter et al. 2007).

- **MANYNUMOFRECEIVERS** sets the number of many receivers to be used for the geometrical optimization from above. It is only required when MANYRECEIVERS is set to be true.

- **MANYKERNELS** can be either *true* or *false*. It performs a loop over the range of given epicentral distances to calculate sensitivity kernels.

- **KRNEPI** sets the range of epicentral distances, i.e. the minimum and maximum distance (in degrees) for the loop over many kernels. Only if MANYKERNELS is set to true it is required.

- **IMPORTKERNELSRECEIVERS** can be either *true* or *false*. It reads the different receiver locations from a file. The file name must be called "tmpReceiverStations***.dat"

- **SOURCE_TIME_SIGMA** sets the source time function parameter, see the source parameters explained in Tape (2003).
- **SOURCE_WIDTH_MU** sets the source width parameter, see the source parameters explained in Tape (2003).
- **FILTER_INITIALSOURCE** can be either *true* or *false*. It is used to filter the initial prescribed source, which can be useful when calculating kernels to filter out spurious frequency artefacts.
- **ADJOINT_TAPERSIGNAL** can be either *true* or *false*, applies tapering to the adjoint source.
- **WINDOWED_INTEGRATION** can be either *true* or *false*, does apply a time window to seismograms for adjoint kernel integrations.
- **WINDOW_START** start time of time window (requires WINDOWED_INTEGRATION being turned on). For major-arcs, good start times are R150: 4700.0; R200: 4400.0; R250: 4100.0; R300: 3800.0. Time window examples for L150 are 1. orbit: 0 to 4000; 2. orbit: 4000 to 8000; 3. orbit: 8000 to 13500; 4. orbit: 13500 to 17000.
- **WINDOW_END** end time of time window (requires WINDOWED_INTEGRATION being turned on).
- **ADJOINT_ANTIPODE_TIME** can be either *true* or *false*. For adjoint kernel simulations, if turned on the simulation time ends at the reference antipode time (overrides LASTTIME setting).
- **DELTA** can be either *true* or *false*. It is used if one want to place a single scatterer on the membrane.
- **DRADIUS** determines the radius size of the single scatterer (in km). If one wants only a single cell to be perturbed, it is set to a value smaller than the grid spacing, e.g. 0.1.
- **DTYPE** can be either *plateau* or *gaussian*. The perturbation is multiplied by a normalization function. This function is either given by a function set to 1 where the grid point location is inside the radius of the scatterer or 0 everywhere else

(*plateau*). The other normalization is given by a gaussian function with a smooth transition from 0 for locations further away than the radius to 1 at the center of the scatterer location (*gaussian*).

- **DPERTURBATION** sets the perturbation size (in km/s) of the phase velocity at the scatterer location.
- **DLOCATION** sets latitude and longitude (in degrees) of the scatterer location.
- **MOVEDELTA** can be either *true* or *false*. In case one wants to calculate a sensitivity kernel in a direct way (see Peter et al. 2007). It loops over many simulations with each time a slightly different scatterer location.
- **DLATSTART** determines the starting latitude (in degrees) of the scatterer location. It is only considered in case MOVEDELTA is set.
- **DLATEND** determines the latitude (in degrees) of the scatterer for the last simulation. It is only considered in case MOVEDELTA is set.
- **DLONEND** determines the longitude (in degrees) of the scatterer for the last simulation. It is only considered in case MOVEDELTA is set.
- **DINCREMENT** gives the incremental step size (in degrees) to take from one simulation to the next one. The scatterer location is moved by this amount, first for a loop over latitudes and then a loop over longitudes. It is only considered in case MOVEDELTA is set.
- **SECONDDELTA** can be either *true* or *false*. If set, a second scatterer is placed (details can be set in the file commonModules.f90) on the membrane together with the delta scatterer set above. It is only considered in case DELTA is set.
- **HETEROGENEOUS** can be either *true* or *false*. If set to true, a heterogeneous phase-velocity distribution must be specified which will be set as background map on the membrane. If not set, the membrane will have a uniform, constant phase-velocity over the whole sphere, with a value set to the wave type and period specified by parameter CPHASE.
- **BLKFILE** specifies the file name of a heterogeneous phase-velocity distribution. You have two possibilities:

(a) choose a block file e.g. "L150.crust.2degree.3.pcn.blk", then BLK/INV_PIXELSIZE must be set to the size of the blocks (e.g. 3 for a 3x3 degree pixel size).

(b) choose a general spherical harmonics file e.g. "L150.crust.2degree.gsh" (derived by CRUST2.0 for Love waves at 150 s period), then BLK/INV_PIXELSIZE must be set to the degree of expansion you want (e.g. 12 for expansion up to degree 12).

(Make sure that the path to the file is given in the correct form: it might include to have a full path, e.g.
"data/phasedata/L150.crust.2degree.3.pcn.blk".)

It is only considered if HETERGENEOUS is set.

- **BLK/INV_PIXELSIZE** see above. It is only considered if HETERGENEOUS is set.
- **BLKVELOCITYREFERENCE** set the type of surface wave and period for which the heterogeneous distributions are given. The distributions are given as percentage of perturbations to this reference.
- **INV_DATA** specifies the file name with a list of source-station setups like e.g. "wei_sum.02.L0150.1.txt", used for the executable `heterogeneousPhaseshift` (otherwise not required) in order to calculate the phase-anomalies for each source-station pair.

  (Make sure that the path to the file is given in the correct form: it might include to have a full path, e.g.
  "data/phasedata/wei_sum.02.L0150.1.txt".)

  It requires a heterogeneous background phase-velocity map (see HETEROGENEOUS).
- **INV_OUTPUT** not required.
- **INV_VOXELSIZE** not required.
- **DATADIRECTORY** sets the name of the direction where output files are written to.
- **ADJOINTKERNEL** is optional and only considered for adjoint kernel calculations. It sets the file name for the output kernel values.

- **VERBOSE** can be either *true* or *false* depending if one wants the console output to be verbose.
- **SIMULATIONOUTPUT** can be either *true* or *false*. If true, the simulation will write out the complete forward wavefield at a regular time interval (details can be specified in file commonModules.f90).

  This output can be used to visualize the displacements as a movie.
- **PARALLELSEISMO** can be either *true* or *false*. If set to true, the code runs in parallel on using MPI to communicate. The executable must be run either by mpirun or mpiexec.

  (Additional detailed parameters can be set in the file `include/commonModules.f90`.)

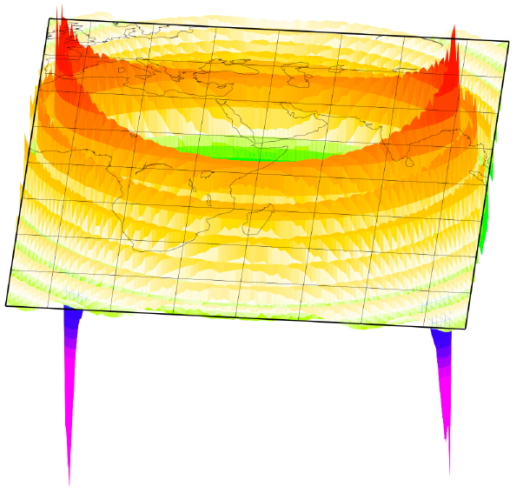
Main program output is the seismogram (format: time / displacement, with the time given in seconds), written in the data directory DATADIRECTORY specified in `Parameter_Input`.

For checking purposes, it also outputs the used phase map (format: longitude / latitude / phasespeed) and the phase speed squared at the vertices (format: verticeID / phasespeedsquared).

Visualization of the seismograms can be done with the GNUPLOT application. The `scripts/` folder holds a bash-script for visualization of the phase-map with GMT.

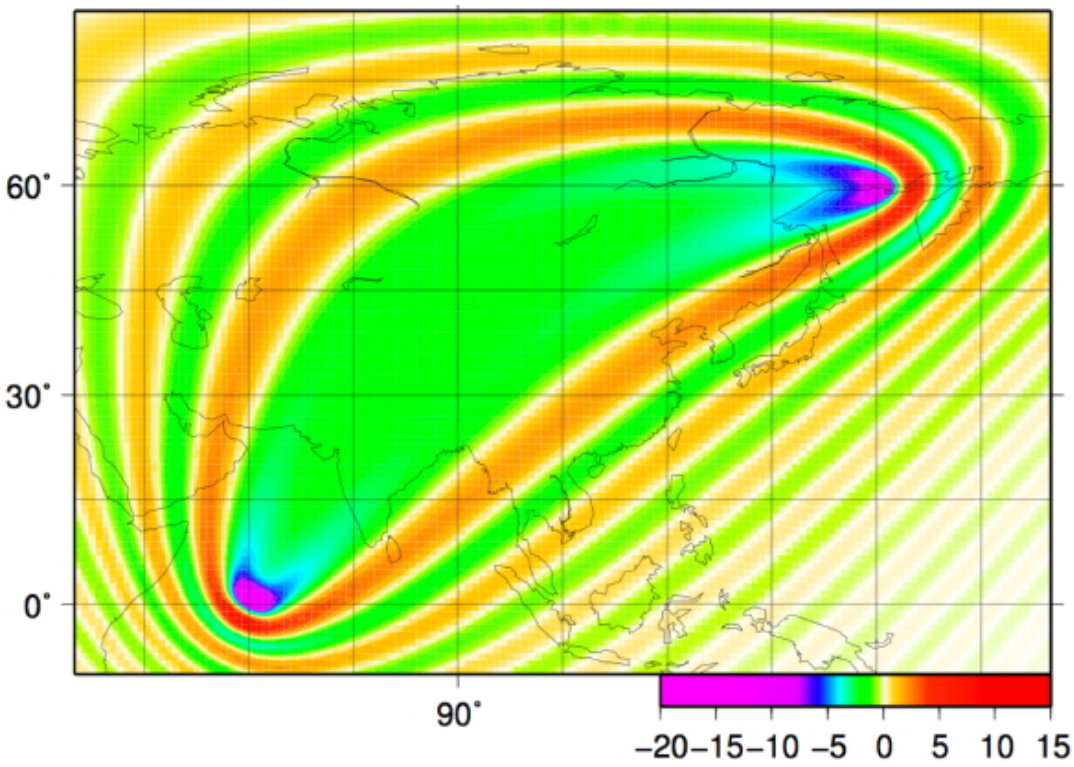
Chapter 5

Computing a sensitivity kernel



You can construct sensitivity kernels by membrane waves. The executable `adjointMethod` performs a simulation, where no scatterer is present, and calculates the adjoint source. Then it makes a time-reversed simulation and computes the kernel values.

It reads the main input file `Parameter_Input`. For details, look in the former section such as parameters regarding filtering and time windowing. More detailed and specific parameters related to the adjoint method can also be found in the file `commonModules.f90`, located in the `src/includes/` source directory.



The kernel values are written to an output file in the data directory and with a name specified in file `Parameter_Input`. The output file format is: longitude / latitude / kernelvalue / vertexID. Kernel values are evaluated at the vertices of the (triangular) spherical membrane grid only.

To visualize the kernel, there are two bash-scripts in the `scripts/` folder using GMT. These scripts `gmtplot_2Dkernel.sh` (plot as 2D map) and `gmtplot_3Dkernel.sh` (plot in 3D perspective) interpolate the kernel file and output cross-sections at different longitudes as well.

Appendix A

Utilities

The source `timelag.f90` calculates the phase shift/time shift of two seismograms. This utility program calculates the time lag between two seismograms.

It reads only the input file `Timelag_Input`. Other parameters (especially the filter parameters) are taken from `commonModules.f90`.

The time shift result is written to the console.

Appendix B

Post-processing scripts

There are several scripts to visualize your results. They are located in the directory `scripts/`.

- The *gmtplot* scripts use the GMT (Generic Mapping Tools) data processing and display software package Version 4+ They also make use of the color table files in the directory.
- The *gnuplot* scripts use the Gnuplot plotting utility Version 4+
- The shell scripts employ the utilities Gawk Version 3+ and Python Version 2+

Appendix C

Copyright

Appendix D

Notes and Acknowledgement

The parallel computing is done using the standard message passing interface (MPI). It is freely available under https://www.open-mpi.org.

Bibliography

[1] Heikes, R. and D. A. Randall, 1995. Numerical integration of the shallow-water equations on a twisted icosahedral grid.1. Basic design and results of tests, *Mon. Weather Rev.*, **123**, 1862–1880.

[2] Heikes, R. and D. A. Randall, 1995. Numerical integration of the shallow-water equations on a twisted icosahedral grid.2. A detailed description of the grid and an analysis of numerical accuracy, *Mon. Weather Rev.*, **123**, 1881–1887.

[3] Peter, D., C. Tape, L. Boschi and J. H. Woodhouse, 2007. Surface wave tomography: global membrane waves and adjoint methods, *Geophys. J. Int.,* , *171*: p. 1098-1117.

[4] Tanimoto, T., 1990. Modelling curved surface wave paths: membrane surface wave synthetics, *Geophys. J. Int.,* **102**, 89–100.

[5] Tape, C. H., 2003. *Waves on a Spherical Membrane*, M.Sc. thesis, University of Oxford, U.K.