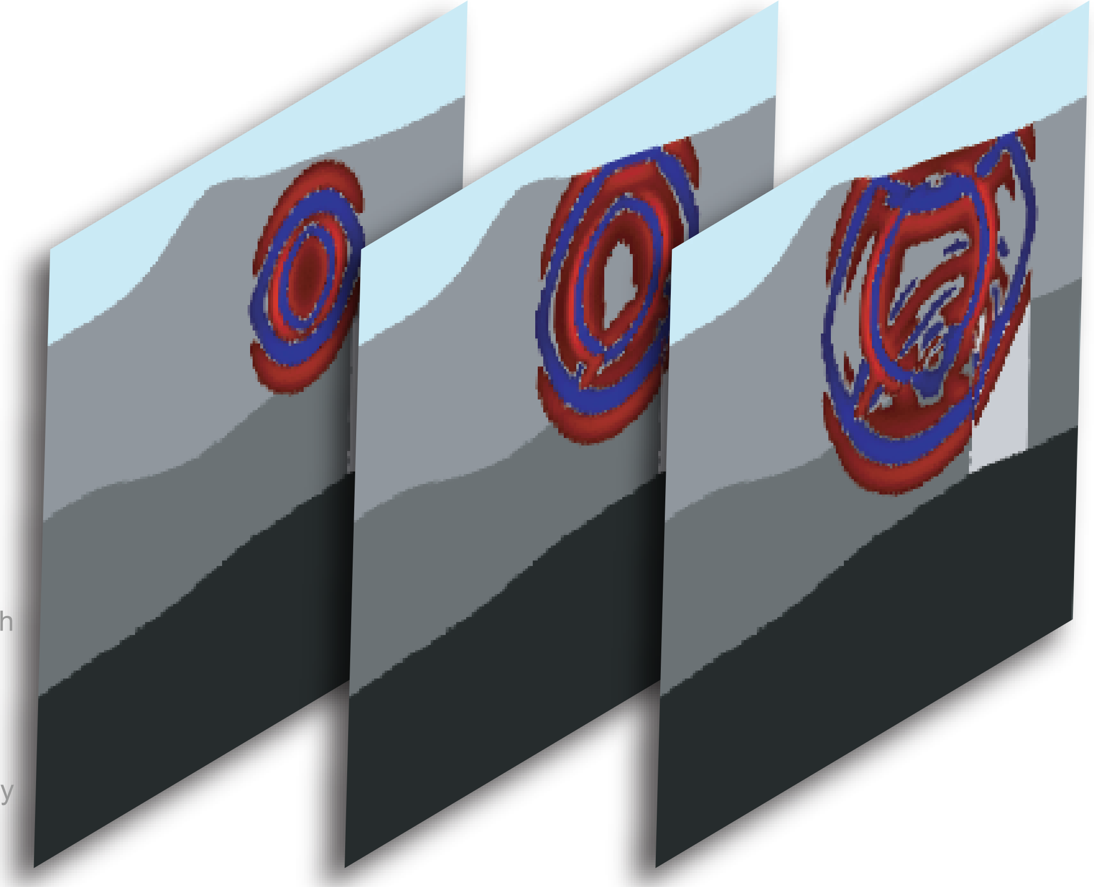
COMPUTATIONAL INFRASTRUCTURE FOR GEODYNAMICS (CIG)
PRINCETON UNIVERSITY (USA)
UNIVERSITY OF PAU, CNRS and INRIA (FRANCE)

SPECFEM 2D

User Manual

Version 6.1



Paul Cristini
Dimitri Komatitsch
Jesús Labarta
Nicolas Le Goff
Pieyre Le Loher
Roland Martin
Christina Morency
Carl Tape
Jeroen Tromp

SPECFEM2D
User Manual

Version 6.1

January 31, 2011

Authors

The SPECFEM2D package was first developed by Dimitri Komatitsch and Jean-Pierre Vilotte at IPG in Paris (France) from 1995 to 1997 and then by Dimitri Komatitsch at Harvard University (USA), Caltech (USA) and then University of Pau (France) from 1998 to 2005.

Since then it has been developed and maintained by a development team: in alphabetical order, Paul Cristini, Dimitri Komatitsch, Jesús Labarta, Nicolas Le Goff, Pieyre Le Loher, Roland Martin, Christina Morency, Carl Tape, Jeroen Tromp... (add other developers here in the future, several are currently missing).

Contents

Chapter 1

Introduction

SPECFEM2D facilitates 2D simulations of acoustic, (an)elastic, and poroelastic seismic wave propagation. With version 6.1, the 2D spectral-element solver accommodates regular and unstructured meshes, generated for example by Cubit (`http://cubit.sandia.gov`), Gmsh (`http://geuz.org/gmsh`) or GiD (`http://www.gid.cimne.upc.es`). Even mesh creation packages that generate triangles, for instance Delaunay-Voronoi triangulation codes, can be used because each triangle can then easily be decomposed into three quadrangles by linking the barycenter to the center of each edge; while this approach does not generate quadrangles of optimal quality, it can ease mesh creation in some situations and it has been shown that the spectral-element method can very accurately handle distorted mesh elements.

The solver has adjoint capabilities and can calculate finite-frequency sensitivity kernels for acoustic, (an)elastic, and poroelastic media. The package also considers 2D SH and P-SV wave propagation. Finally, the solver can run both in serial and in parallel. See SPECFEM2D (`http://www.geodynamics.org/cig/software/packages/seismo/specfem2d`) for the source code.

The SEM is a continuous Galerkin technique, which can easily be made discontinous [Bernardi et al., 1994, Chaljub, 2000, Kopriva et al., 2002, Chaljub et al., 2003, Legay et al., 2005, Kopriva, 2006, Wilcox et al., 2010, Acosta Minolia and Kopriva, 2011]; it is then close to a particular case of the discontinuous Galerkin technique [Reed and Hill, 1973, Arnold, 1982, Falk and Richter, 1999, Hu et al., 1999, Cockburn et al., 2000, Giraldo et al., 2002, Rivière and Wheeler, 2003, Monk and Richter, 2005, Grote et al., 2006, Ainsworth et al., 2006, Bernacki et al., 2006, Dumbser and Käser, 2006, De Basabe et al., 2008, de la Puente et al., 2009, Wilcox et al., 2010, De Basabe and Sen, 2010, Étienne et al., 2010], with optimized efficiency because of its tensorized basis functions [Wilcox et al., 2010, Acosta Minolia and Kopriva, 2011]. In particular, it can accurately handle very distorted mesh elements [Oliveira and Seriani, 2011].

Note that in most geological models in the context of seismic wave propagation studies (except for fault dynamic rupture studies, in which very high frequencies need to be modeled near the fault, see e.g. de la Puente et al. [2009]) a discontinous mesh is not needed because material property contrasts are not drastic and thus a continuous formulation is sufficient.

1.1 Citation

If you use this code for your own research, please cite at least one article written by the developers of the package, for instance:
Tromp et al. [2008] or Vai et al. [1999], Lee et al. [2009a, 2008, 2009b], Komatitsch et al. [2010a,b, 2009], Liu et al. [2004], Chaljub et al. [2007], Komatitsch and Vilotte [1998], Komatitsch and Tromp [1999], Komatitsch et al. [2004], Morency and Tromp [2008] and/or other articles from (`http://web.univ-pau.fr/~dkomati1/publications.html`)

If you use the kernel capabilities of the code, please cite at least one article written by the developers of the package, for instance:
Tromp et al. [2008] or Liu and Tromp [2006], Morency et al. [2009]

If you use the SCOTCH / CUBIT non-structured capabilities, please also cite:
Martin et al. [2008]

The corresponding BibTEX entries may be found in file `doc/USER_MANUAL/bibliography.bib`.

1.2 Support

This material is based upon work supported by the USA National Science Foundation under Grants No. EAR-0406751 and EAR-0711177, by the French CNRS, French INRIA Sud-Ouest MAGIQUE-3D, French ANR NUMASIS under Grant No. ANR-05-CIGC-002, and European FP6 Marie Curie International Reintegration Grant No. MIRG-CT-2005-017461. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the USA National Science Foundation, CNRS, INRIA, ANR or the European Marie Curie program.

Chapter 2

Getting Started

The SPECFEM2D software package comes in a gzipped tar ball. In the directory in which you want to install the package, type

```
tar -zxvf SPECFEM2D_6.1.1.tar.gz
```

The directory `SPECFEM2D-6.1.1/` will then contain the source code. In the following, we will refer to this directory as the root directory `SPECFEM2D/`.

To configure the software for your system, run the `configure` shell script. This script will attempt to guess the appropriate configuration values for your system. However, at a minimum, it is recommended that you explicitly specify the appropriate command names for your Fortran90 compiler:

```
./configure FC=ifort
```

To optimize compilation of the executables on your specific system, please follow these steps:

- if you want to run in parallel, i.e., using more than one processor core, then you would type

  ```
  ./configure FC=ifort MPIFC=mpif90 --with-mpi
  ```

  The SPECFEM2D software package relies on the SCOTCH library to partition meshes. The SCOTCH library [Pellegrini and Roman, 1996] provides efficent static mapping, graph and mesh partitioning routines. SCOTCH is a free software package developed by François Pellegrini et al. from LaBRI and INRIA in Bordeaux, France, downloadable from the web page `https://gforge.inria.fr/projects/scotch/`. It is more recent than METIS, actively maintained and performs better in many cases. A recent version of its source code is provided in directory `scotch_5.1.11/`. In case no SCOTCH libraries can be found on the system, the configuration will bundle this version for compilation. The path to an existing SCOTCH installation can to be set explicitly with the option `--with-scotch-dir`. Just as an example:

  ```
  ./configure FC=ifort MPIFC=mpif90 --with-mpi --with-scotch-dir=/opt/scotch
  ```

  For further details about the installation of SCOTCH, go to subdirectory `scotch_5.1.11/` and read `INSTALL.txt`. You may want to download more recent versions of SCOTCH in the future from (`http://www.labri.fr/perso/pelegrin/scotch/scotch_en.html`) . Support for the METIS graph partitioner has been discontinued because SCOTCH is more recent and performs better.

- edit the `Makefile` for more specific modifications. Especially, there are several options available :
  `-DUSE_MPI` compiles with use of an MPI library.
  `-DUSE_SCOTCH` enables use of graph partitioner SCOTCH.

After these steps, go back to the main directory of SPECFEM2D/ and type

```
make
```

to create all executables.

Chapter 3

Mesh Generation

3.1 How to use SPECFEM2D

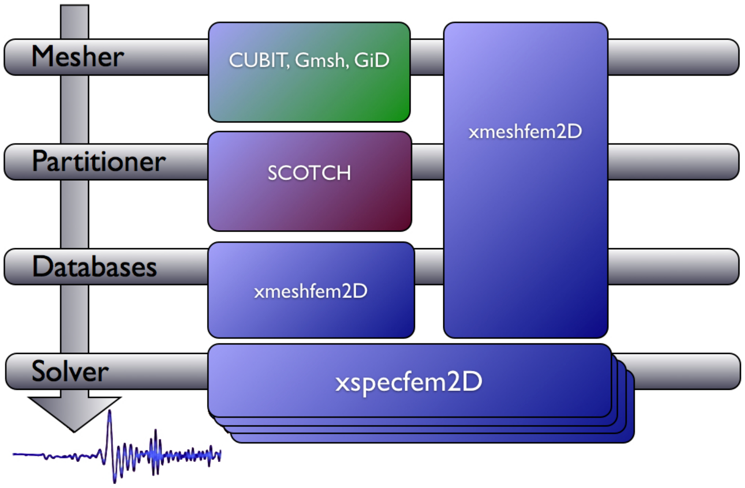


Figure 3.1: Schematic workflow for a SPECFEM2D simulation. The exectable `xmeshfem2D` creates the GLL mesh points and assigns specific model parameters. The executable `xspecfem2D` solves the seismic wave propagation.

To run the mesher, please follow these steps:

- edit the input file `DATA/Par_file` which describes the simulation. It contains comments and should be almost self-explanatory, if you need more details we do not have a manual for the 2D version but you can find useful information in the manuals of the 3D versions, since many parameters and the general philosophy is similar. They are available at (`http://geodynamics.org/wsvn/cig/seismo/3D`) in subdirectories `USER_MANUAL/`. To create acoustic (fluid) regions, just set the S wave speed to zero and the code will see that these elements are fluid and switch to the right equations there automatically, and automatically match them with the solid regions
- if you are using an external mesher (like GID or CUBIT), you should set `read_external_mesh` to `.true.`:

  `mesh_file` is the file describing the mesh : first line is the number of elements, then a list of 4 nodes (quadrilaterals only) forming each elements on each line.

  `nodes_coords_file` is the file containing the coordinates (x and z) of each nodes : number of nodes on the first line, then coordinates x and z on each line.

materials_file is the number of the material for every elements : an integer ranging from 1 to nbmodels on each line.

free_surface_file is the file describing the edges forming the acoustic free surface : number of edges on the first line, then on each line number of the element, number of nodes forming the free surface (1 for a point, 2 for an edge), the nodes forming the free surface for this element. If you do not want free surface, just put 0 on the first line.

absorbing_surface_file is the file describing the edges forming the absorbing boundaries : the format is the same as the `free_surface_file`.

tangential_detection_curve_file contains points describing the envelope, used for `source_normal_to_surfa` and `rec_normal_to_surface`. Should be fine grained, and ordained clockwise. Number of points on the first line, then (x,z) coordinates on each line.

- if you have compiled with MPI, you must specify the number of processes.

Then type

```
./xmeshfem2D
```

to create the mesh (which will be stored in directory `OUTPUT_FILES/`). `xmeshfem2D` is serial; it will output several files called `Database??????`, one for each process.

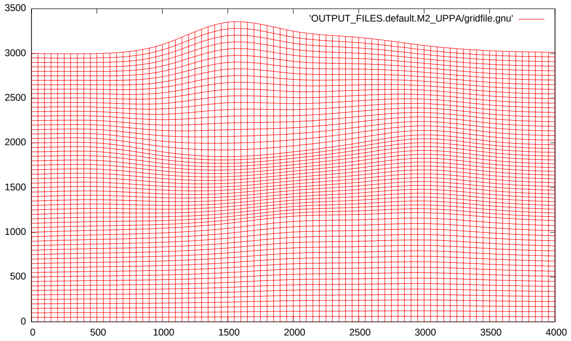


Figure 3.2: Example of a grid file generated by `xmeshfem2D` and visualized with gnuplot.

3.2 Controlling the quality of an external mesh

To examine the quality of the elements in your externally build mesh, type

```
./xcheck_quality_external_mesh
```

(and answer "3" to the first question asked). This code will tell you which element in the whole mesh has the worst quality (maximum skewness, i.e. maximum deformation of the element angles) and it should be enough to modify this element with the external software package used for the meshing, and to repeat the operation until the maximum skewness of the whole mesh is less or equal to about 0.75 (above is dangerous: from 0.75 to 0.80 could still work, but if there is a single element above 0.80 the mesh should be improved).

The code also shows a histogram of 20 classes of skewness which tells how many element are above the skewness = 0.75, and to which percentage of the total this amounts. To see this histogram, you could type:

```
gnuplot plot_mesh_quality_histogram.gnu
```

This tool is useful to estimate the mesh quality and to see it evolve along the successive corrections.

Chapter 4

Running the Solver xspecfem2D

To run the solver, type:

```
./xspecfem2D
```

to run the main solver (use `mpirun` or equivalent if you compiled with parallel support). This will output the seismograms and snapshots of the wave fronts at different time steps in directory `OUTPUT_FILES/`. To visualize them, type "`gs OUTPUT_FILES/vect*.ps`" to see the Postscript files (in which the wave field is represented with small arrows, fluid/solid matching interfaces with a thick pink line, and absorbing edges with a thick green line) and "`gimp OUTPUT_FILES/image*.gif`" to see the color snapshot showing a pixelized image of one of the two components of the wave field (or pressure, depending on what you have selected for the output in `DATA/Par_file`).

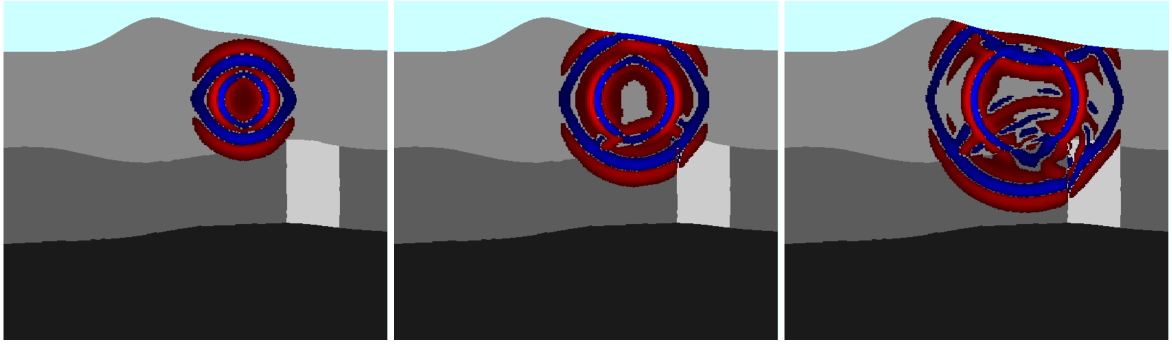


Figure 4.1: Wavefield snapshots of the default example generated by `xspecfem2D` when parameter `output_color_image` is set to `.true.`.

Please consider these following points, when running the solver:

- the `DATA/Par_file` given with the code works fine, you can use it without any modification to test the code
- the seismograms `OUTPUT_FILES/*.sem*` are simple ASCII files with two columns: time in the first colum and amplitude in the second, therefore they can be visualized with any tool you like, for instance "`gnuplot`"
- if you set flag `assign_external_model` to `.true.` in `DATA/Par_file`, the velocity and density model that is given at the end of `DATA/Par_file` is then ignored and overwritten by the external velocity and density model that you define yourself in `define_external_model.f90`
- when compiling with Intel ifort, use "`-assume byterecl`" option to create binary PNM images displaying the wave field
- we do not have PML absorbing conditions implemented in the fluid/solid code yet. We use (older and less efficient) paraxial Clayton-Engquist or Sommerfeld equations instead. This is only by lack of time, we have

a developer who is currently implementing PML but the code is not fully ready. For now, since the paraxial conditions are less efficient, please use a larger model

- there are a few useful scripts and Fortran routines in directory `UTILS/`.
- if you find bugs (or if you have comments or suggestions) please send an email to cig-seismo AT geodynamics.org and the developers will try to fix them and send you an updated version
- you can find a Fortran code to compute the analytical solution for simple media that we use as a reference in benchmarks in many of our articles at (`http://www.spice-rtn.org/library/software/EX2DDIR`). That code is described in: Berg et al. [1994]

The `SOURCE` file located in the `DATA/` directory should be edited in the following way:

source_surf Set this flag to `.true.` to force the source to be located at the surface of the model, otherwise the source will be placed inside the medium

xs source location x in meters

zs source location z in meters

source_type Set this value equal to `1` for elastic forces or acoustic pressure, set this to `2` for moment tensor sources.

time_function_type Choose a source-time function: set this value to `1` to use a Ricker, `2` the first derivative, `3` a Gaussian, `4` a Dirac or `5` a Heaviside source-time function.

f0 Set this to the dominant frequency of the source. For point-source simulations using a Heaviside source-time function (time_function_type "5"), we recommend setting the source frequency parameter `f0` equal to a high value, which corresponds to simulating a step source-time function, i.e., a moment-rate function that is a delta function.

The `half duration` of a source is obtained by `1/f0`. If the code will use a Gaussian source-time function (time_function_type "3") (i.e., a signal with a shape similar to a ‘smoothed triangle’, as explained in Komatitsch and Tromp [2002] and shown in Fig 4.2), the source-time function uses a half-width of `half duration`. We prefer to run the solver with `half duration` set to zero and convolve the resulting synthetic seismograms in post-processing after the run, because this way it is easy to use a variety of source-time functions. Komatitsch and Tromp [2002] determined that the noise generated in the simulation by using a step source time function may be safely filtered out afterward based upon a convolution with the desired source time function and/or low-pass filtering. Use the serial code `convolve_source_timefunction.f90` and the script `convolve_source_timefunction.csh` for this purpose, or alternatively use signal-processing software packages such as SAC (`www.llnl.gov/sac`). Type

```
make convolve_source_timefunction
```

to compile the code and then set the parameter `hdur` in `convolve_source_timefunction.csh` to the desired half-duration.

t0 For single sources, set the time shift parameter `t0` equal to 0.0 (the solver will not run otherwise.) The time shift parameter would simply apply an overall time shift to the synthetics, something that can be done in the post-processing. This time shift parameter can be non-zero when using multiple sources.

angleforce Angle of the source (for a force only)

Mxx,Mzz,Mxz Moment tensor components (valid only for moment tensor sources, source_type "2"). Note that the units for the components of a moment tensor source are different in SPECFEM2D and in SPECFEM3D:

SPECFEM3D: in SPECFEM3D the moment tensor components are in dyne*cm

SPECFEM2D: in SPECFEM2D the moment tensor components are in N*m

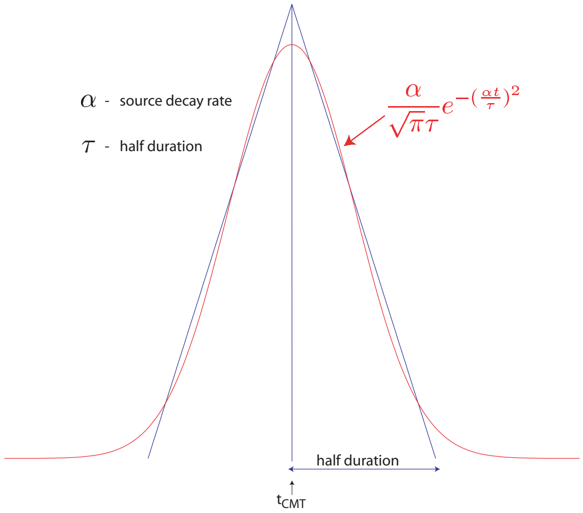

α - source decay rate
τ - half duration
$\frac{\alpha}{\sqrt{\pi}\tau}e^{-(\frac{\alpha t}{\tau})^2}$
half duration
t_{CMT}

Figure 4.2: Comparison of the shape of a triangle and the Gaussian function actually used.

factor amplification factor

Note, the zero time of the simulation corresponds to the center of the triangle/Gaussian, or the centroid time of the earthquake. The start time of the simulation is $t = -1.2 *$ `half duration` $+$ `t0` (the factor 1.2 is to make sure the moment rate function is very close to zero when starting the simulation; Heaviside functions use a factor 2.0), the half duration is obtained by 1/f0.

Caution

See file `todo_list_please_dont_remove.txt` for a list of known bugs, problems, or missing options.

Coupled Simulations

The code supports acoustic/elastic, acoustic/poroelastic, elastic/poroelastic, and acoustic,elastic/poroelastic simulations.

Elastic/poroelastic coupling supports anisotropy, but not attenuation for the elastic material.

4.1 How to run P-SV or SH (membrane) wave simulations

For elastic materials, you have these additional options:

P-SV: To run a P-SV waves calculation propagating in the x-z plane, set `p_sv = .true.` in the `Par_file`.

SH: To run a SH (membrane) waves calculation traveling in the x-z plane with a y-component of motion, set `p_sv = .false.`

This feature is only implemented for elastic materials and sensitivity kernels can be calculated (see Tape et al. [2007] for details on membrane surface waves).

A useful Python script called `SEM_save_dir.py`, written by Paul Cristini from Laboratoire de Mecanique et d'Acoustique, CNRS, Marseille, France, is provided. It allows one to automatically save all the parameters and results of a given simulation.

4.2 How to use Poroelasticity

Check the following new inputs in `Par_file`:

In section **"# geometry of model and mesh description"**:
`TURN_VISCATTENUATION_ON`, `Q0`, and `FREQ0` deal with viscous damping in a poroelastic medium. `Q0` is the quality factor set at the central frequency `FREQ0`. For more details see Morency and Tromp [2008].

In section **"# time step parameters"**:
`SIMULATION_TYPE` defines the type of simulations
(1) forward simulation
(2) adjoint method and kernels calculation

In section **"# source parameters"**:
The code now support multi sources. `NSOURCE` is the number of source. Parameters of the sources are displayed in the file `SOURCE`, which must be in the directory `DATA/`. The components of a moment tensor source must be given in N.m, not in dyne.cm as in the `DATA/CMTSOLUTION` source file of the 3D version of the code.

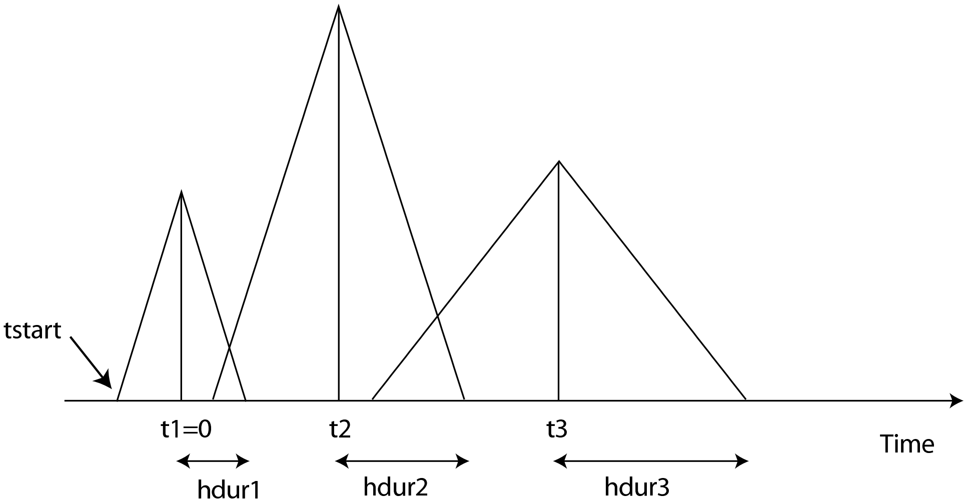


Figure 4.3: Example of timing for three sources. The center of the first source triangle is defined to be time zero. Note that this is NOT in general the hypocentral time, or the start time of the source (marked as tstart). The time shift parameter `t0` in the `SOURCE` file would be t1(=0), t2, t3 in this case, and the half-duration parameter, resp. `f0`, would be hdur1=1/f0_1, hdur2=1/f0_2, hdur3=1/f0_3 for the sources 1, 2, 3 respectively.

In section **"# receiver line parameters for seismograms"**:
`SAVE_FORWARD` determines if the last frame of a forward simulation is saved (`.true.`) or not (`.false`)

In section **"# define models...."**:
There are three possible types of models:
`I`: (model_number 1 rho Vp Vs 0 0 Qp Qs 0 0 0 0 0 0) or
`II`: (model_number 2 rho c11 c13 c15 c33 c35 c55 0 0 0 0 0 0) or
`III`: (model_number 3 rhos rhof phi c kxx kxz kzz Ks Kf Kfr etaf mufr Qs).

For istropic elastic/acoustic material use `I` and set Vs to zero to make a given model acoustic, for anisotropic elastic use `II`, and for isotropic poroelastic material use `III`. The mesh can contain acoustic, elastic, and poroelastic models simultaneously. rho_s = solid density
rho_f = fluid density
phi = porosity
tort = tortuosity

permxx = xx component of permeability tensor
permxz = xz,zx components of permeability tensor
permzz = zz component of permeability tensor
kappa_s = solid bulk modulus
kappa_f= fluid bulk modulus
kappa_fr= frame bulk modulus
eta_f = fluid viscosity
mu_fr = frame shear modulus
Qs = shear quality factor

Note: for the poroelastic case, mu_s is irrelevant. For details on the poroelastic theory see Morency and Tromp [2008].

`get_poroelastic_velocities.f90` allows to compute cpI, cpII, and cs function of the source dominant frequency. Notice that for this calculation we use permxx and the dominant frequency of the first source , f0(1). Caution if you use several sources with different frequencies and if you consider anistropic permeability.

Chapter 5

Adjoint Simulations

5.1 How to obtain Finite Sensitivity Kernels

1. Run a forward simulation:
   => `SIMULATION_TYPE = 1`
   => `SAVE_FORWARD = .true.`
   => `seismotype = 1` (we need to save the displacement fields to later on derive the adjoint source. Note: if the user forgets it, the program corrects it when reading the proper `SIMULATION_TYPE` and `SAVE_FORWARD` combination and a warning message appears in the ouput file)

   Important output files (for example, for the elastic case, P-SV waves):
   `absorb_elastic_bottom*****.bin`
   `absorb_elastic_left*****.bin`
   `absorb_elastic_right*****.bin`
   `absorb_elastic_top*****.bin`
   `lastframe_elastic*****.bin`
   `S****.AA.BHX.semd`
   `S****.AA.BHZ.semd`

2. Define the adjoint source:
   Use `adj_seismogram.f90`
   Edit to update `NSTEP`, `nrec`, `t0`, `deltat`, and the position of the cut to pic any given phase if needed (`tstart,tend`), add the right number of stations, and put one component of the source to zero if needed. The ouput files of `adj_seismogram.f90` are `S****.AA.BHX.adj` and `S****.AA.BHZ.adj`, for P-SV waves (and `S****.AA.BHY.adj`, for SH (membrane) waves). Note that you will need these three files (`S****.AA.BHX.adj`, `S****.AA.BHY.adj` and `S****.AA.BHZ.adj`) to be present in the `OUTPUT_FILES/` directory together with the `absorb_elastic_****.bin` and `lastframe_elastic.bin` files to be read when running the adjoint simulation.

3. Run the adjoint simulation:
   Make sure that the adjoint source files absorbing boundaries and last frame files are in the `OUTPUT_FILES/` directory.
   => `SIMULATION_TYPE = 2`
   => `SAVE_FORWARD = .false.`

   Output files (for example for the elastic case):
   `snapshot_rho_kappa_mu*****`
   `snapshot_rhop_alpha_beta*****`

which are the primary moduli kernels and the phase velocities kernels respectively, in ascii format and at the local level, that is as "kernels(i,j,ispec)".

Caution

Please note that

- at the moment, adjoint simulations do not support anisotropy, attenuation, and viscous damping.
- you will need `S****.AA.BHX.adj`, `S****.AA.BHY.adj` and `S****.AA.BHZ.adj` to be present in directory `OUTPUT_FILES/` even if you are just running an acoustic or poroelastic adjoint simulation.
  `S****.AA.BHX.adj` is the only relevant component for an acoustic case.
  `S****.AA.BHX.adj` and `S****.AA.BHZ.adj` are the only relevant components for a poroelastic case.

Acknowledgments

We thank Paul Cristini from Laboratoire de Mecanique et d'Acoustique of Marseille, France, for very carefully testing version 6.1 of the package and helping us locate and fix several important bugs, as well as introducing support for the Gmsh mesher in the package.

The Gauss-Lobatto-Legendre subroutines in `gll_library.f90` are based in part on software libraries from the Massachusetts Institute of Technology, Department of Mechanical Engineering (Cambridge, Massachusetts, USA). The non-structured global numbering software was provided by Paul F. Fischer (Brown University, Providence, Rhode Island, USA, now at Argonne National Laboratory, USA).

Please e-mail your feedback, questions, comments, and suggestions to Jeroen Tromp (`jtromp-AT-princeton.edu`) or to the CIG Computational Seismology Mailing List (`cig-seismo@geodynamics.org`).

Copyright

Main authors: Dimitri Komatitsch and Jeroen Tromp
Princeton University, USA, and University of Pau / CNRS / INRIA, France

Bibliography

C. A. Acosta Minolia and D. A. Kopriva. Discontinuous Galerkin spectral element approximations on moving meshes. *J. Comput. Phys.*, 230(5):1876–1902, 2011. doi: 10.1016/j.jcp.2010.11.038.

M. Ainsworth, P. Monk, and W. Muniz. Dispersive and dissipative properties of discontinuous Galerkin finite element methods for the second-order wave equation. *Journal of Scientific Computing*, 27(1):5–40, 2006. doi: 10.1007/s10915-005-9044-x.

D. N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM Journal on Numerical Analysis*, 19(4):742–760, 1982. doi: 10.1137/0719052.

P. Berg, F. If, P. Nielsen, and O. Skovegaard. Analytic reference solutions. In K. Helbig, editor, *Modeling the Earth for oil exploration, Final report of the CEC's GEOSCIENCE I Program 1990-1993*, pages 421–427. Pergamon Press, Oxford, United Kingdom, 1994.

M. Bernacki, S. Lanteri, and S. Piperno. Time-domain parallel simulation of heterogeneous wave propagation on unstructured grids using explicit, nondiffusive, discontinuous Galerkin methods. *J. Comput. Acoust.*, 14(1):57–81, 2006.

C. Bernardi, Y. Maday, and A. T. Patera. A new nonconforming approach to domain decomposition: the Mortar element method. In H. Brezis and J. L. Lions, editors, *Nonlinear partial differential equations and their applications*, Séminaires du Collège de France, pages 13–51, Paris, 1994. Pitman.

E. Chaljub. *Modélisation numérique de la propagation d'ondes sismiques en géométrie sphérique : application à la sismologie globale (Numerical modeling of the propagation of seismic waves in spherical geometry: application to global seismology).* PhD thesis, Université Paris VII Denis Diderot, Paris, France, 2000.

E. Chaljub, Y. Capdeville, and J. P. Vilotte. Solving elastodynamics in a fluid-solid heterogeneous sphere: a parallel spectral-element approximation on non-conforming grids. *J. Comput. Phys.*, 187(2):457–491, 2003.

E. Chaljub, D. Komatitsch, J. P. Vilotte, Y. Capdeville, B. Valette, and G. Festa. Spectral element analysis in seismology. In R.-S. Wu and V. Maupin, editors, *Advances in wave propagation in heterogeneous media*, volume 48 of *Advances in Geophysics*, pages 365–419. Elsevier - Academic Press, London, UK, 2007.

B. Cockburn, G. E. Karniadakis, and C.-W. Shu. *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Springer, Heidelberg, Germany, 2000.

J. D. De Basabe and M. K. Sen. Stability of the high-order finite elements for acoustic or elastic wave propagation with high-order time stepping. *Geophys. J. Int.*, 181(1):577–590, 2010. doi: 10.1111/j.1365-246X.2010.04536.x.

J. D. De Basabe, M. K. Sen, and M. F. Wheeler. The interior penalty discontinuous Galerkin method for elastic wave propagation: grid dispersion. *Geophys. J. Int.*, 175(1):83–93, 2008. doi: 10.1111/j.1365-246X.2008.03915.x.

J. de la Puente, J. P. Ampuero, and M. Käser. Dynamic rupture modeling on unstructured meshes using a discontinuous Galerkin method. *J. Geophys. Res.*, 114:B10302, 2009. doi: 10.1029/2008JB006271.

M. Dumbser and M. Käser. An arbitrary high-order discontinuous Galerkin method for elastic waves on unstructured meshes-II. The three-dimensional isotropic case. *Geophys. J. Int.*, 167(1):319–336, 2006. doi: 10.1111/j.1365-246X.2006.03120.x.

V. Étienne, E. Chaljub, J. Virieux, and N. Glinsky. An hp-adaptive discontinuous Galerkin finite-element method for 3-D elastic wave modelling. *Geophys. J. Int.*, 183(2):941–962, 2010. doi: 10.1111/j.1365-246X.2010.04764.x.

R. S. Falk and G. R. Richter. Explicit finite element methods for symmetric hyperbolic equations. *SIAM Journal on Numerical Analysis*, 36(3):935–952, 1999. doi: 10.1137/S0036142997329463.

F. X. Giraldo, J. S. Hesthaven, and T. Warburton. Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations. *J. Comput. Phys.*, 181(2):499–525, 2002. doi: 10.1006/jcph.2002.7139.

M. J. Grote, A. Schneebeli, and D. Schötzau. Discontinuous Galerkin finite element method for the wave equation. *SIAM Journal on Numerical Analysis*, 44(6):2408–2431, 2006. doi: 10.1137/05063194X.

F. Q. Hu, M. Y. Hussaini, and P. Rasetarinera. An analysis of the discontinuous Galerkin method for wave propagation problems. *J. Comput. Phys.*, 151(2):921–946, 1999. doi: 10.1006/jcph.1999.6227.

D. Komatitsch and J. Tromp. Spectral-element simulations of global seismic wave propagation-I. Validation. *Geophys. J. Int.*, 149(2):390–412, 2002. doi: 10.1046/j.1365-246X.2002.01653.x.

D. Komatitsch and J. Tromp. Introduction to the spectral-element method for 3-D seismic wave propagation. *Geophys. J. Int.*, 139(3):806–822, 1999. doi: 10.1046/j.1365-246x.1999.00967.x.

D. Komatitsch and J. P. Vilotte. The spectral-element method: an efficient tool to simulate the seismic response of 2D and 3D geological structures. *Bull. Seismol. Soc. Am.*, 88(2):368–392, 1998.

D. Komatitsch, Q. Liu, J. Tromp, P. Süss, C. Stidham, and J. H. Shaw. Simulations of ground motion in the Los Angeles basin based upon the spectral-element method. *Bull. Seismol. Soc. Am.*, 94(1):187–206, 2004. doi: 10.1785/0120030077.

D. Komatitsch, D. Michéa, and G. Erlebacher. Porting a high-order finite-element earthquake modeling application to NVIDIA graphics cards using CUDA. *Journal of Parallel and Distributed Computing*, 69(5):451–460, 2009. doi: 10.1016/j.jpdc.2009.01.006.

D. Komatitsch, G. Erlebacher, D. Göddeke, and D. Michéa. High-order finite-element seismic wave propagation modeling with MPI on a large GPU cluster. *J. Comput. Phys.*, 229(20):7692–7714, 2010a. doi: 10.1016/j.jcp.2010.06.024.

D. Komatitsch, D. Göddeke, G. Erlebacher, and D. Michéa. Modeling the propagation of elastic waves using spectral elements on a cluster of 192 GPUs. *Computer Science Research and Development*, 25(1-2):75–82, 2010b. doi: 10.1007/s00450-010-0109-1.

D. A. Kopriva. Metric identities and the discontinuous spectral element method on curvilinear meshes. *Journal of Scientific Computing*, 26(3):301–327, 2006. doi: 10.1007/s10915-005-9070-8.

D. A. Kopriva, S. L. Woodruff, and M. Y. Hussaini. Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method. *Int. J. Numer. Meth. Eng.*, 53(1):105–122, 2002. doi: 10.1002/nme.394.

S. J. Lee, H. W. Chen, Q. Liu, D. Komatitsch, B. S. Huang, and J. Tromp. Three-dimensional simulations of seismic wave propagation in the Taipei basin with realistic topography based upon the spectral-element method. *Bull. Seismol. Soc. Am.*, 98(1):253–264, 2008. doi: 10.1785/0120070033.

S. J. Lee, Y. C. Chan, D. Komatitsch, B. S. Huang, and J. Tromp. Effects of realistic surface topography on seismic ground motion in the Yangminshan region of Taiwan based upon the spectral-element method and LiDAR DTM. *Bull. Seismol. Soc. Am.*, 99(2A):681–693, 2009a. doi: 10.1785/0120080264.

S. J. Lee, D. Komatitsch, B. S. Huang, and J. Tromp. Effects of topography on seismic wave propagation: An example from northern Taiwan. *Bull. Seismol. Soc. Am.*, 99(1):314–325, 2009b. doi: 10.1785/0120080020.

A. Legay, H. W. Wang, and T. Belytschko. Strong and weak arbitrary discontinuities in spectral finite elements. *Int. J. Numer. Meth. Eng.*, 64(8):991–1008, 2005. doi: 10.1002/nme.1388.

Q. Liu and J. Tromp. Finite-frequency kernels based on adjoint methods. *Bull. Seismol. Soc. Am.*, 96(6):2383–2397, 2006. doi: 10.1785/0120060041.

Q. Liu, J. Polet, D. Komatitsch, and J. Tromp. Spectral-element moment tensor inversions for earthquakes in Southern California. *Bull. Seismol. Soc. Am.*, 94(5):1748–1761, 2004. doi: 10.1785/012004038.

R. Martin, D. Komatitsch, C. Blitz, and N. Le Goff. Simulation of seismic wave propagation in an asteroid based upon an unstructured MPI spectral-element method: blocking and non-blocking communication strategies. *Lecture Notes in Computer Science*, 5336:350–363, 2008.

P. Monk and G. R. Richter. A discontinuous Galerkin method for linear symmetric hyperbolic systems in inhomogeneous media. *Journal of Scientific Computing*, 22-23(1-3):443–477, 2005. doi: 10.1007/s10915-004-4132-5.

C. Morency and J. Tromp. Spectral-element simulations of wave propagation in poroelastic media. *Geophys. J. Int.*, 175:301–345, 2008.

C. Morency, Y. Luo, and J. Tromp. Finite-frequency kernels for wave propagation in porous media based upon adjoint methods. *Geophys. J. Int.*, 2009. doi: 10.1111/j.1365-246X.2009.04332.

S. P. Oliveira and G. Seriani. Effect of element distortion on the numerical dispersion of spectral element methods. *Communications in Computational Physics*, 9(4):937–958, 2011.

F. Pellegrini and J. Roman. SCOTCH: A software package for static mapping by dual recursive bipartitioning of process and architecture graphs. *Lecture Notes in Computer Science*, 1067:493–498, 1996.

W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, USA, 1973.

B. Rivière and M. F. Wheeler. Discontinuous finite element methods for acoustic and elastic wave problems. *Contemporary Mathematics*, 329:271–282, 2003.

C. Tape, Q. Liu, and J. Tromp. Finite-frequency tomography using adjoint methods - Methodology and examples using membrane surface waves. *Geophys. J. Int.*, 168(3):1105–1129, 2007. doi: 10.1111/j.1365-246X.2006.03191.x.

J. Tromp, D. Komatitsch, and Q. Liu. Spectral-element and adjoint methods in seismology. *Communications in Computational Physics*, 3(1):1–32, 2008.

R. Vai, J. M. Castillo-Covarrubias, F. J. Sánchez-Sesma, D. Komatitsch, and J. P. Vilotte. Elastic wave propagation in an irregularly layered medium. *Soil Dynamics and Earthquake Engineering*, 18(1):11–18, 1999. doi: 10.1016/S0267-7261(98)00027-X.

L. C. Wilcox, G. Stadler, C. Burstedde, and O. Ghattas. A high-order discontinuous Galerkin method for wave propagation through coupled elastic-acoustic media. *J. Comput. Phys.*, 229(24):9373–9396, 2010. doi: 10.1016/j.jcp.2010.09.008.

Appendix A

Troubleshooting

FAQ

Regarding the structure of some of the database files :

> **Question:** Can anyone tell me what the columns of the SPECFEM2D boundary condition files in `SPECFEM2D/DATA/Mesh_canyon` are?
>
> `SPECFEM2D/DATA/Mesh_canyon/canyon_absorbing_surface_file`
> `SPECFEM2D/DATA/Mesh_canyon/canyon_free_surface_file`
>
> **Answer:** `canyon_absorbing_surface_file` refers to parameters related to the absorbing conditions: The first number (180) is the number of absorbing elements (nelemabs in the code). Then the columns are:
> column 1 = the element number
> column 2 = the number of nodes of this element that form the absorbing surface
> column 3 = the first node
> column 4 = the second node
>
> `canyon_free_surface_file` refers to the elements of the free surface (relevant for enforcing free surface condition for acoustic media): The first number (160) is the number of elements of the free surface. Then the columns are (similar to the absorbing case):
> column 1 = the element number
> column 2 = the number of nodes of this element that form the absorbing surface
> column 3 = the first node
> column 4 = the second node
>
> Concerning the free surface description file, nodes/edges pertaining to elastic elements are discarded when the file is read (if for whatever reason it was simpler to include all the nodes/edges on one side of a studied area and that there are among them some elements that are elastic elements, only the nodes/edges of acoustic elements are kept).
>
> These files are opened and read in `meshfem2D.F90` using subroutines `read_abs_surface()` and `read_acoustic_surface()`, which are in `part_unstruct.F90`