2016 CIG WORKSHOP
SPECFEM3D

Dmitry Borisov, Princeton University

June 24, UC DAVIS

Contents

1 Getting started

Connect to Peloton:

```
$ ssh -X <id>@peloton.cse.ucdavis.edu
```

Download the Specfem3D package:

```
$ git clone --recursive --branch devel https://github.com/geodynamics/specfem3d.git
```

Check what modules are loaded by default:

```
$ module list
```

Load python (not required on Peloton).

```
$ module load python/2.7
```

In the same way load appropriate Fortran (GNU/intel) and MPI compilers (not required on Peloton).

Configure the package for your system from the Specfem3D-root directory:

```
$ cd specfem3d
$ ./configure --with-openmp=no --with-mpi=yes --with-cuda=no --with-adios=no
make -j 4 all
```

Compile the source code:

```
$ make all
```

Check generated executable files:

```
$ ls ./bin
```

Take a quick look at the user manual, mainly as a reminder to come back to it for details.

```
$ evince doc/USER_MANUAL/manual_SPECFEM3D_Cartesian.pdf &
```

2 Example #1: Internal mesher

The purpose of this example is to step through all the key points of Specfem3D combined with internal mesher: (1) create the mesh, (2) generate databases, (3) run the solver, and check the output. The example is trivial, however the internal mesher can handle more complex settings.

Go to the **working directory** you have just created:

```
$ cd /home/<id>/2016_CIG_SPECFEM3D/ex1_internal
```

Copy the folder with Specfem3D binaries to the working directory and create **output folders**

```
$ cp -r ../specfem3d/bin ./
$ mkdir OUTPUT_FILES
$ mkdir OUTPUT_FILES/DATABASES_MPI
```

Check the content of the "DATA" folder and the **parameter file**:

```
$ vim ./DATA/Par_file
```

Input model is a text-file with a header (first four lines) and the rest 6-columns:
1) x; 2) y; 3) z; 4) Vp; 5) Vs; 6) Density.

```
$ vim ./DATA/tomo_files/tomography_model.xyz
```

Check the source information:

```
$ vim ./DATA/FORCESOLUTION
```

Check the sensors information:

```
$ vim ./DATA/STATIONS
```

(1) Create mesh

The first step in running a SEM simulation consists of constructing a high-quality mesh for the region under consideration. In this example we are going to use the provided, internal mesher - xmeshfem3D.

Before running the mesher, a number of parameters need to be set in the "Mesh_Par_file"

```
$ vim DATA/meshfem3D_files/Mesh_Par_file
```

Make sure that you create the mesh in a ".vtk" format for visualization in paraview:

```
CREATE_VTK_FILES = .true.
```

Add these tree lines in the submission scripts (e.g. submit16_mesh3d) to send mail when the process begins, and when it ends. Make sure you define your email address.

```
$ vim submit16_mesh3d
  #SBATCH --mail-type=begin
  #SBATCH --mail-type=end
  #SBATCH --mail-user=<your_email>
```

Submit a job for a mesh creation:

```
$ sbatch submit16_mesh3d
  Submitted batch job <job_id>
```

Check the status of the submitted job:

```
$ squeue
  JOBID PARTITION     NAME     USER ST       TIME  NODES NODELIST(REASON)
```

The job should take few seconds. When the mesher is finished, check the output information:

```
$ vim ./OUTPUT_FILES/output_meshfem3D.txt
  ...
  **************************
  Checking mesh quality
  **************************
  ...
  Maximum suggested time step for simulation =    0.00083095
  ...
  Max CFL stability condition of the time scheme = 0.4703999999999
  ...
  done
```

Make sure mesh quality is good, time step in Par_file is smaller then suggested one and the Courant-Friedrichs-Lewy (CFL) condition is valid.

Copy generated *mesh.vtk files to your machine:

```
$ scp <id>@peloton.cse.ucdavis:/home/<id>/ex1_internal/OUTPUT_FILES/
DATABASES_MPI/*mesh.vtk ./
```

Plot the mesh (*mesh.vtk files) in Paraview (fig. 1). In the properties, select "Surface with edges" as a representation for visualizing the mesh and check the box "show axis".

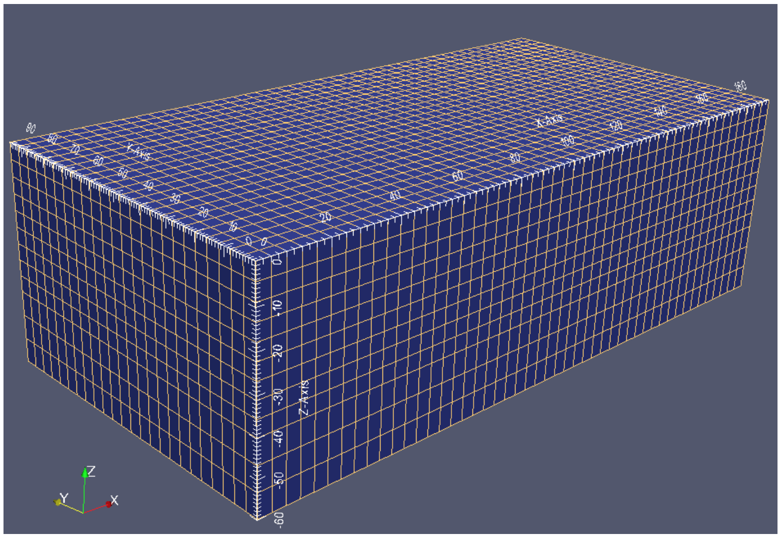

Figure 1: 3D mesh from Example 1.

(2) Generate the databases:

After the mesh was created using xmeshfem3D, the next step in the workflow is to run xgenerate_databases. This program is going to create all the missing information needed by the SEM solver.

```
$ sbatch submit24_generate
```

When the databases generation is finished (the job should take few seconds), check the content of the output folder ./OUTPUT_FILES/DATABASES_MPI/:

```
$ ls ./OUTPUT_FILES/DATABASES_MPI/
```

Check the output information:

```
$ vim ./OUTPUT_FILES/output_mesher.txt
  ...
  Maximum suggested time step = 2.84234178E-04
  ...
  total number of elements in entire mesh:        13824
  ...
  approximate total number of points in entire mesh :    960400
  ...
  done
```

Combine the model (e.g. Vp) from domain-decomposed pieces:

```
$ ./bin/xcombine_vol_data 0 15 vp ./OUTPUT_FILES/DATABASES_MPI/ OUTPUT_FILES/ 0
```

Copy created vtk-files to your machine and then use PARAVIEW to visualize the model (fig. 2).

(3) Run the solver

Now that you have successfully generated the databases, you are ready to run the solver.

```
$ sbatch submit16_specfem
```

Find why there is a following problem with the solver:

```
forrtl: error (72): floating overflow
```

When the solver is finished (should take about 1 minute), check the output information:

```
$ vim ./OUTPUT_FILES/output_solver.txt
  ...
  *********************************************
  *** Verification of simulation parameters ***
  *********************************************
  ...
  Time-Loop Complete. Timing info:
  Total elapsed time in seconds =    64.729827880859375
  Total elapsed time in hh:mm:ss =      0 h 01 m 04 s

  End of the simulation
```

Check the seismograms by running python script from "py" folder (fig. 3):

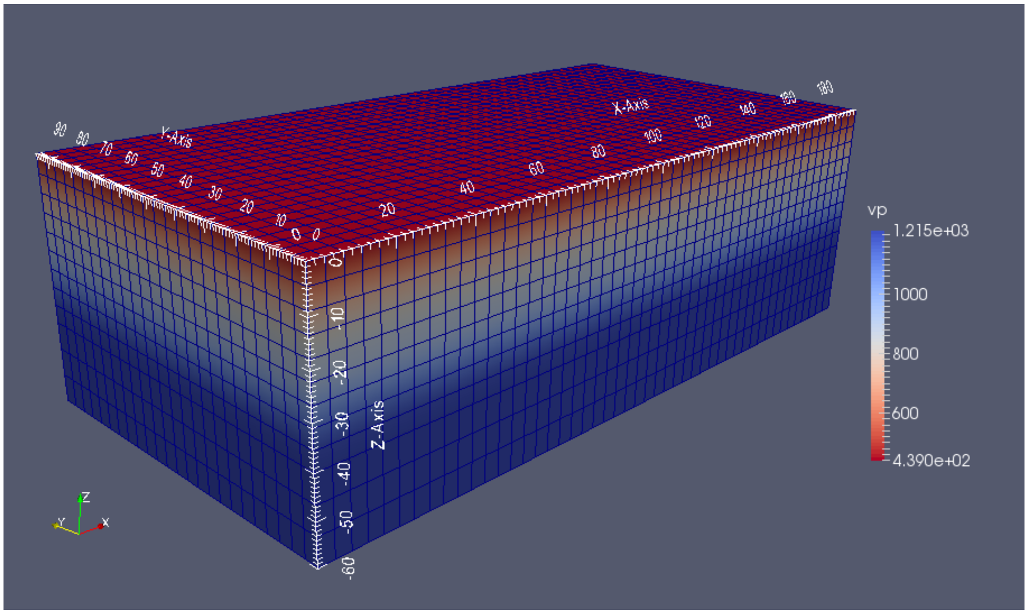

Figure 2: 3D P-wave velocity model overlaid with the mesh from Example 1

```
$ python plot_3traces.py
```

Create a movie showing wavefield propagation at the surface:

```
$ ./bin/xcreate_movie_shakemap_AVS_DX_GMT
```

Use the following option values during that process:

```
2,1,2001,1,1
```

Copy AVS-files to your machine

```
$ scp <id>@peloton.cse.ucdavis:/home/<id>/ex1_internal/OUTPUT_FILES/AVS* ./
```

Open AVS-files in PARAVIEW and play (enjoy) the movie.

To include attenuation, modify the Par_file:

```
$ vim ./DATA/Par_file
  ATTENUATION = .true.
  USE_OLSEN_ATTENUATION = .true.
```

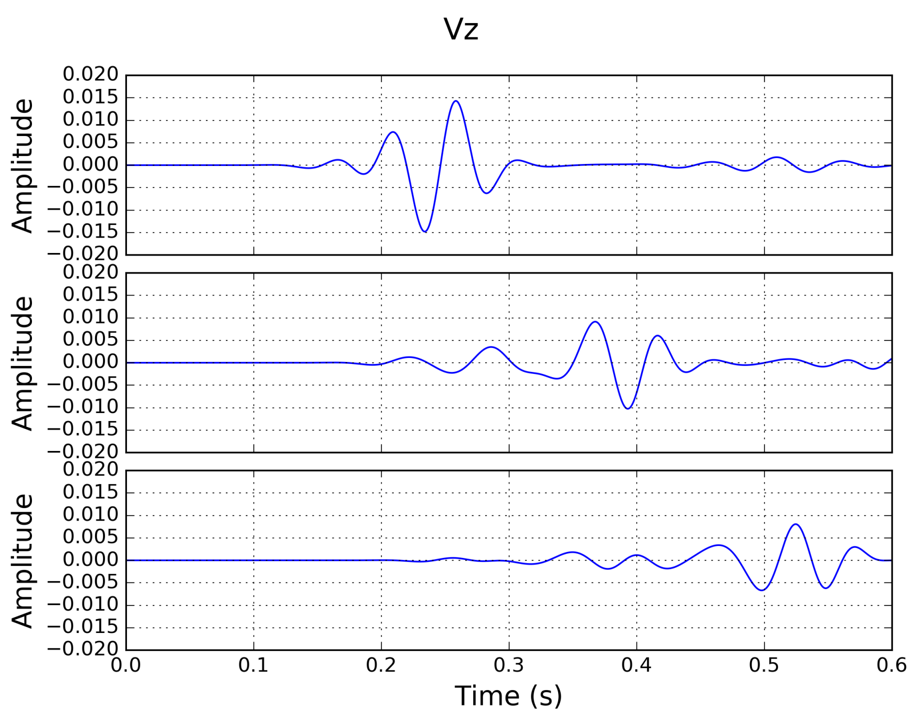

Figure 3: Seismic records (vertical component of particle velocity) from Example 1. Three traces from near (top), middle (mid) and far (bottom) offset are presented.

3 Example #2: External mesher

The purpose of this example is to step through all the key points of Specfem3D combined with external mesher: (1) export the mesh from Trelis (Cubit), (2) mesh decomposition, (3) generate databases, (4) run the solver, and check the output. This approach is useful for simulations in models with a complex geometry.

Go to the second example **working directory**:

```
$ cd /home/<id>/2016_CIG_SPECFEM3D/ex2_external
```

Copy the folder with Specfem3D binaries to the working directory and create **output folders**

```
$ cp -r ../specfem3d/bin ./
$ mkdir OUTPUT_FILES
$ mkdir OUTPUT_FILES/DATABASES_MPI
```

Go to the MESH-folder in your working directory:

```
$ cd ./MESH
```

Load required software (not required on Peloton):

```
$ module load netcdf/gcc/hdf5-1.8.16/4.4.0
$ module load hdf5/gcc/1.8.16
```

(1) Mesh format conversion

Compile and run c-code, required to convert the mesh from Trelis (Cubit) to SPECFEM3D format

```
$ gcc trelis2specfem3d.c -o trelis2specfem3d
$ ./trelis2specfem3d mountain_mesh.e -bin=1
```

Check the elastic properties of the model:

```
$ vim nummaterial_velocity_file
```

Based on the Example1, define your own sensors and sources in the DATA folder.

(2) Decompose mesh

The spectral-element mesh created with Trelis (CUBIT) needs to be distributed on the processors. This partitioning is executed once and for all prior to the execution of the solver so it is referred to as a static mapping.

To decompose the mesh run the program from the working directory:

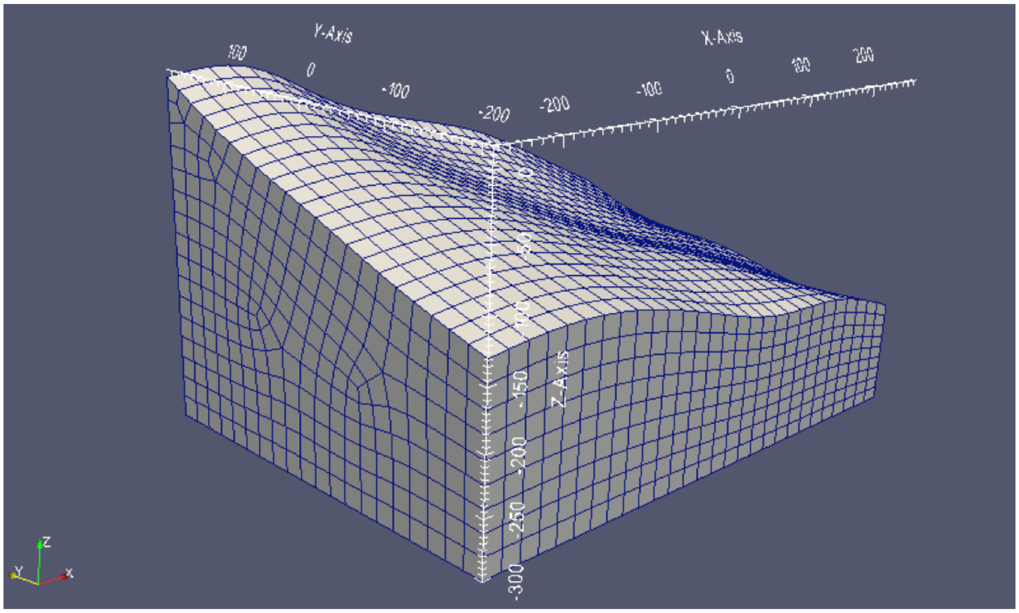

Figure 4: 3D mesh from Example 2.

```
$ cd ..
$ ./bin/xdecompose_mesh 16 ./MESH/ ./OUTPUT_FILES/DATABASES_MPI/
  total number of nodes:
  nnodes =         17453
  ...
  Databases files in directory: ./OUTPUT_FILES/DATABASES_MPI/
  finished successfully
```

Check the output folder, which now should contain proc0000*_Database files:

```
$ ls ./OUTPUT_FILES/DATABASES_MPI/
```

Plot the mesh (mountain_mesh.e file) in Paraview (fig. 4). In the properties, select "Surface with edges" as a representation for visualizing the mesh and check the box "show axis".

(3) Generate the databases

After the mesh was created using xmeshfem3D, the next step in the workflow is to run xgenerate_databases. This program is going to create all the missing information needed by the SEM solver.

```
$ sbatch submit16_generate
```

Similar to the first example, combine the model using xcombine_vol_data program and plot the result (*.vtk files) in Paraview (fig. 5).

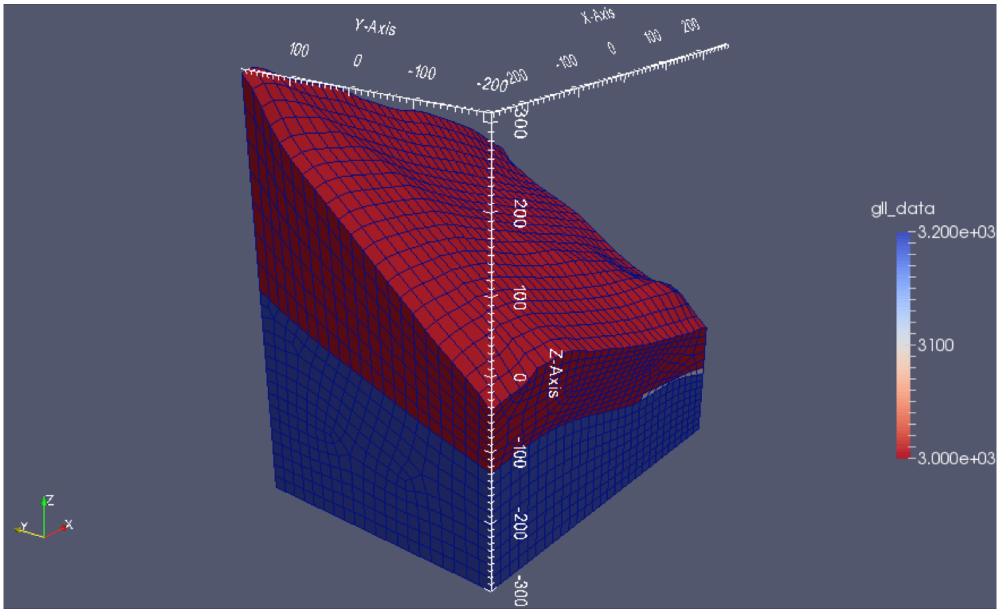

Figure 5: 3D P-wave velocity model overlaid with the mesh from Example 2.

(4) Run the solver

Now after you have successfully generated the databases, you are ready to run the solver.

```
$ sbatch submit16_specfem
```

When the solver is finished (should take about 1 minute), check the output information:

```
$ vim ./OUTPUT_FILES/output_solver.txt
  **********************************************
  **** Specfem 3-D Solver - MPI version f90 ****
  **********************************************
  ...
  Time-Loop Complete. Timing info:
  Total elapsed time in seconds =    81.148675203323364
  Total elapsed time in hh:mm:ss =      0 h 01 m 21 s

  End of the simulation
```

Similar to Example1, create a movie of wavefield propagation at the surface in the AVS-format.
Copy AVS-files to your machine.
Open AVS-files in PARAVIEW and play (enjoy) the movie.

4 Example #3: SeisFlows - 2D checkerboard

SeisFlows is an open source seismic inversion package that

- delivers a complete, customizable waveform inversion workflow
- provides a framework for research in regional, global, and exploration seismology

For detailed information about SeisFlows go to

```
http://seisflows.readthedocs.io/en/latest/
```

Step-by-step instructions:

```
http://seisflows.readthedocs.io/en/latest/instructions_remote.html
```

1. Download SeisFlows

```
$ mkdir ~/packages
$ cd ~/packages
$ git clone https://github.com/PrincetonUniversity/seisflows.git
```

2. Set environment variables

```
export PATH=$PATH:~/packages/seisflows/scripts
export PYTHONPATH=~/packages/seisflows
```

3. Run "system" test to make sure everything is working. Open run.py

```
$ cd ~/packages/seisflows/tests/test_system
$ vim ./run.py
```

Modify run.py for Peloton system by replacing the first line

```
#!/bin/env python
```

by

```
#!/usr/bin/env python
```

Run the system test

```
./clean.py; ./run.py
```

If a "hello" message is displayed, the test was successful.

4. Run nonlinear optimization test (prior to this modify run.py as in step 3)

```
$ cd ~/packages/seisflows/tests/test_optimize
$ ./clean.py; ./run.py
```

If the optimization problem is solved in 50 iterations or fewer, the test was successful.

5. Configure and compile SPECFEM2D

```
$ cd ~/packages
$ git clone --recursive --branch devel
https://github.com/geodynamics/specfem2d.git specfem2d-d745c542

$ cd specfem2d-d745c542
$ git checkout d745c542
```

For now, it is important to work with the exact version specified above (d745c542). This is necessary because, unlike SPECFEM3D and 3D_GLOBE, SPECFEM2D development is sometimes a bit haphazard, with frequent interface changes.

Next, configure and compile SPECFEM2D using ifort (preferred) or gfortran:

```
$ cd ~/packages/specfem2d-d745c542
$ ./configure
$ make all
```

6. Set up checkerboard test
Download the starting model and other input files required for the waveform inversion checkerboard test. Let's assume the checkerboard working directory will be placed in
∼/test/
If you prefer a different location, then modify the following commands accordingly:

```
$ mkdir ~/test
$ cd ~/test
$ wget --recursive --no-parent --no-host-directories --cut-dirs=2
  --reject "index.html*" http://tigress-web.princeton.edu/~rmodrak/2dAcoustic/
```

A directory ∼/tests/checkers is now being created. Among other files, parameters.py and paths.py are being downloaded.
After the download completes, make sure that all paths specified in paths.py are correct. For example, if you compiled SPECFEM2D somewhere other than ∼/packages/specfem2d-d745c542, you will need to modify the SPECFEM2D_BIN entry accordingly.

Next, take a minute to view the parameters.py file and note the close similarity between the first set of parameters and the directory structure of the SeisFlows repository.

7. Run checkerboard test in serial
To run the checkerboard test type type within ∼/tests/checkers:

```
sfclean ; sfrun
```