SeisSol Training 1

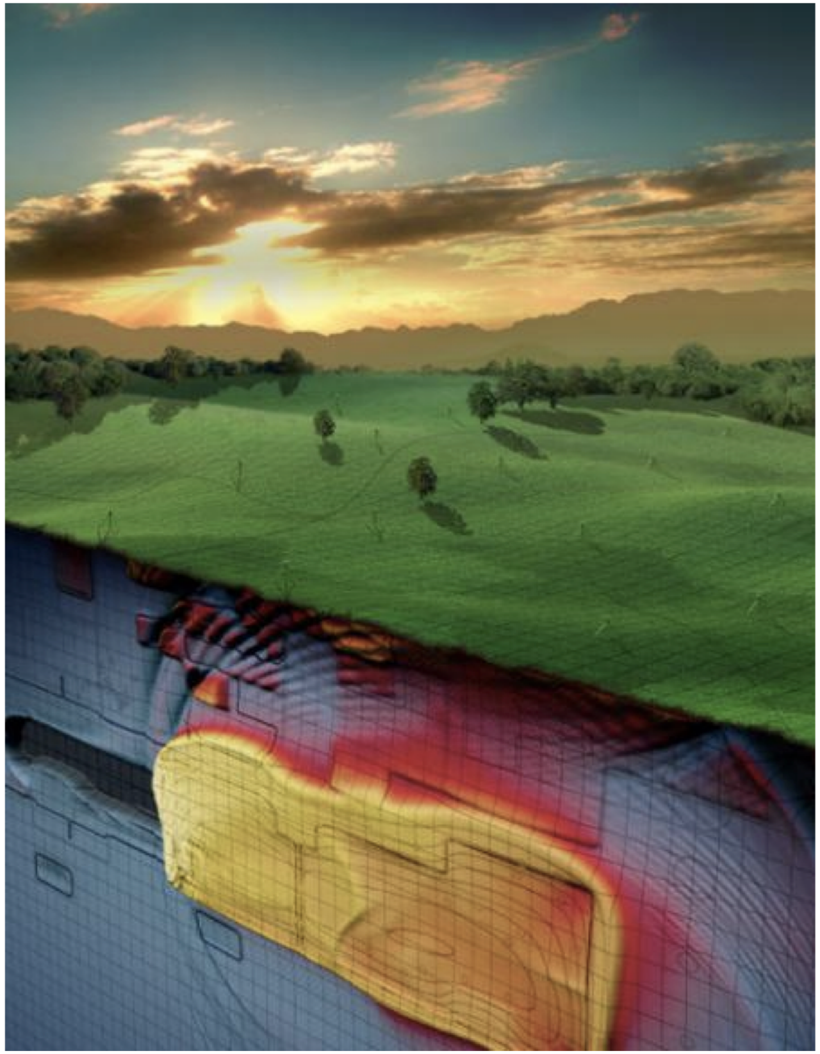
Duo Li and Alice Gabriel

Sept. 4, 2019

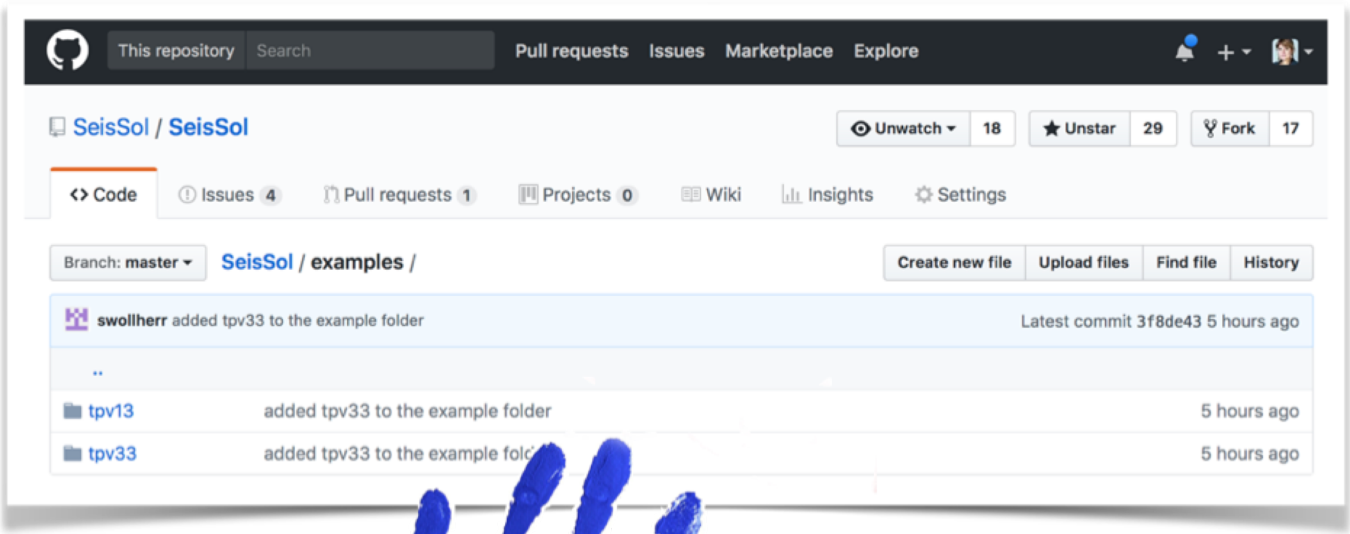
Advanced Workshop on Earthquake Fault Mechanics: Theory, Simulation and Observations

Outline

- SeisSol Introduction
- Problem description
- Parameter setup
- Results Visualization
- Hands-on exercise

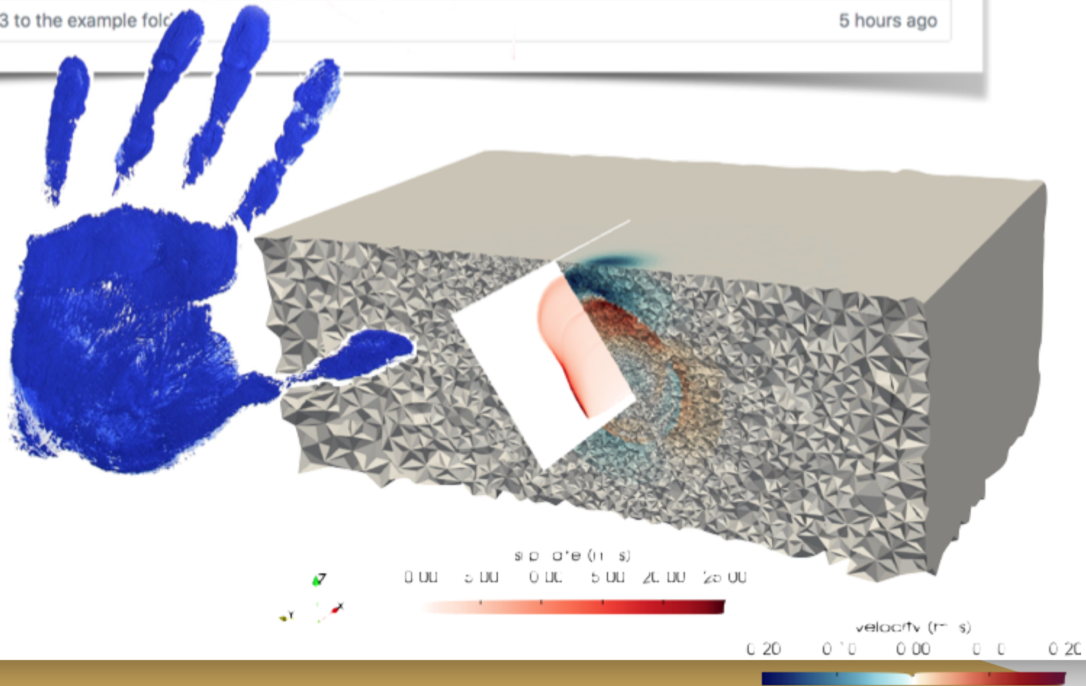


SeisSol Introduction



github.com/SeisSol

www.seissol.org



https://seissol.readthedocs.io/en/latest/index.html

SeisSol
latest

Search docs

INTRODUCTION
- History
- Compilation
- A first example
- Acknowledgements
- Related publications

STRUCTURAL MODELS
- CAD models
- Meshing with SimModeler
- Meshing with PUMGen
- Gmsh

Docs » SeisSol Edit on GitHub

SeisSol

SeisSol is a software package for simulating wave propagation and dynamic rupture based on the arbitrary high-order accurate derivative discontinuous Galerkin method (ADER-DG).

Characteristics of the SeisSol simulation software are:

- use of arbitrarily high approximation order in time and space
- use of tetrahedral meshes to approximate complex 3D model geometries (faults & topography) and rapid model generation
- use of elastic, viscoelastic and viscoplastic material to approximate realistic geological subsurface properties
- parallel geo-information input (ASAGI)
- to produce reliable and sufficiently accurate synthetic seismograms or other seismological data set

SCEC TPV29
SCEC TPV104
Point Source
Kinematic source example - 1994 Northridge earthquake
Copyrights

UNSORTED

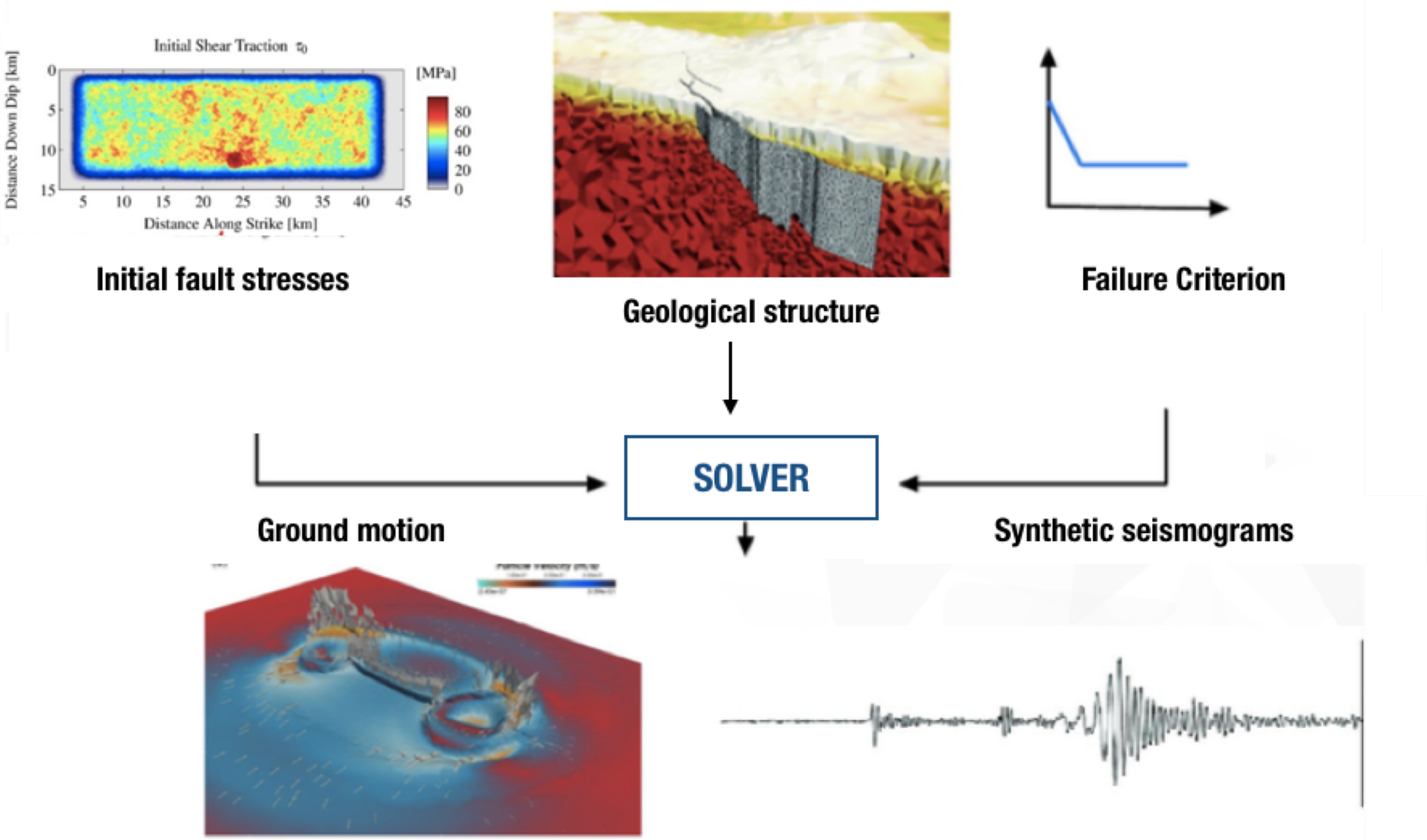
Building SeisSol on Stampede KNL test system
Left lateral, right lateral, normal, reverse
Optimization for non Intel architectures

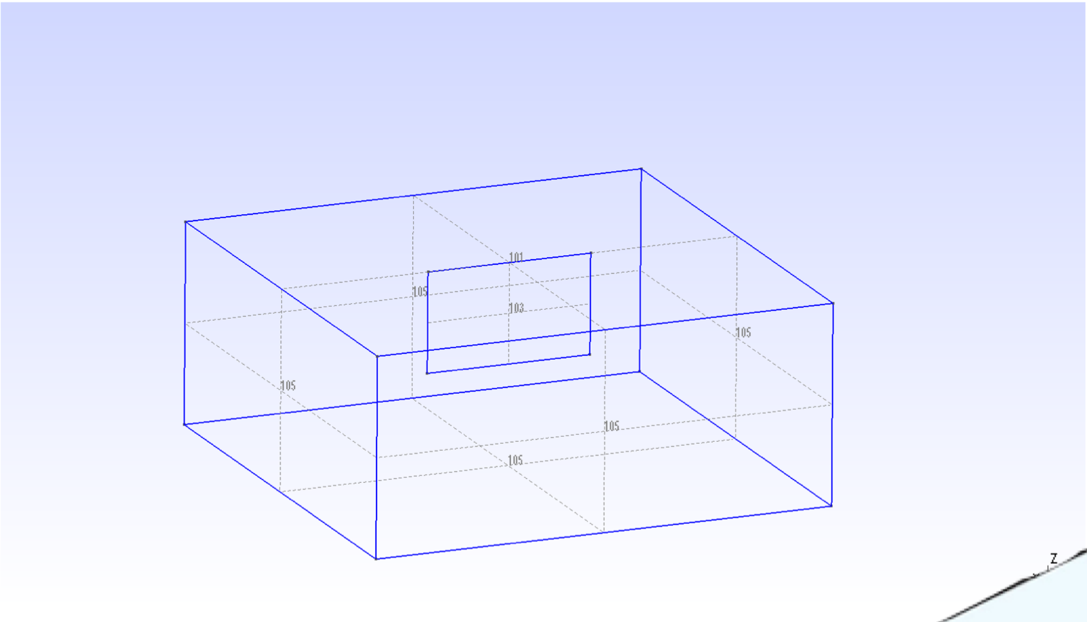
description, a section of *geometry, initial setups (stress, nucleation, friction, etc.)*, and *simulation results*.

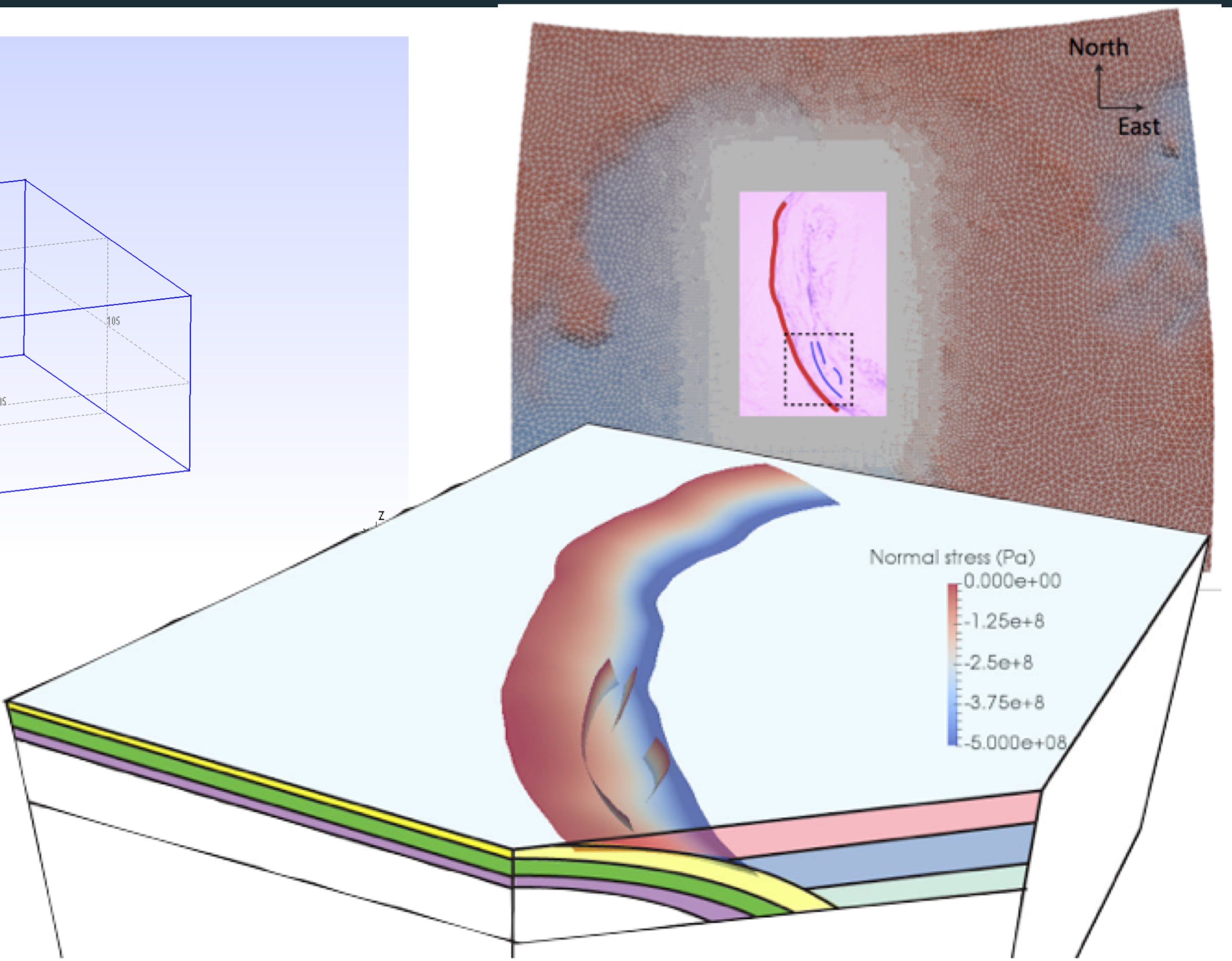
Please note that examples used here are only for demonstration purpose. For detailed benchmark tests please refer to SCEC benchmark center.

No.	Fault type	Difficulty	Description
TPV5	strike-slip	beginner	slip-weakening and heterogeneous initial stres
TPV6	strike-slip	beginner	bi-material fault and, slip-weakening and heter
TPV12	normal fault	beginner	linear elastic and initial stress conditions are de
TPV13	normal fault	beginner	non-associative Drucker-Prager plastic with yi
TPV16	strike-slip	intermediate	randomly-generated heterogeneous initial stre
TPV24	branching strike-slip	intermediate	a rightward branch forming a 30 degree angle.
TPV29	strike-slip	difficult	stochastic roughness. Linear elastic material pr
TPV104	strike-slip	difficult	Rate-state friction, using a slip law with strong
Point Source	strike-slip	intermediate	benchmark of SISMOWINE WP2_LOH1.
Kinematic	reverse fault	intermediate	Kinematic source of 1994 Mw6.7 Northridge e

Table: Overall of examples suites.


Initial Shear Traction τ_0
[MPa]
80
60
40
20
0
Distance Down Dip [km]
0
5
10
15
5 10 15 20 25 30 35 40 45
Distance Along Strike [km]
Initial fault stresses
Geological structure
Failure Criterion
SOLVER
Ground motion
Synthetic seismograms

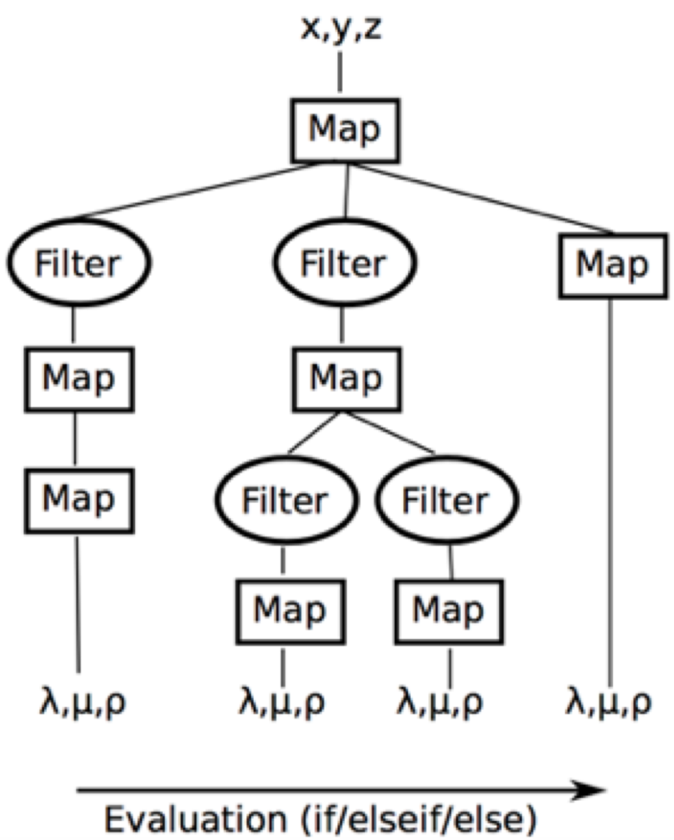




Mesh software

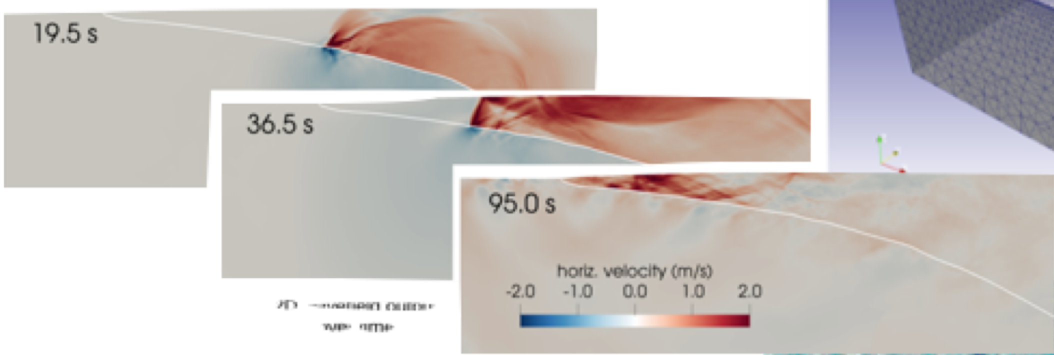
- Gmsh
- Simmodeler

easi composition principle

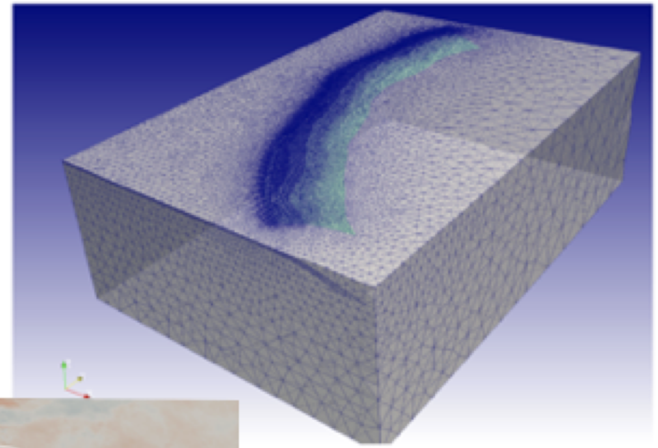


- Easy Initialization of parameters
- Written in YAML language
- Different types of Map and Filter

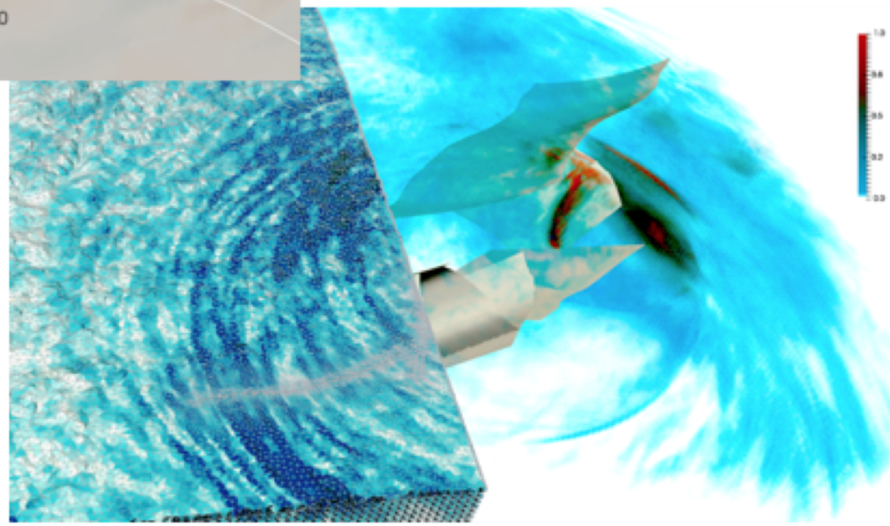
Paraview - hdf5



2D SeisSol (Laptop)

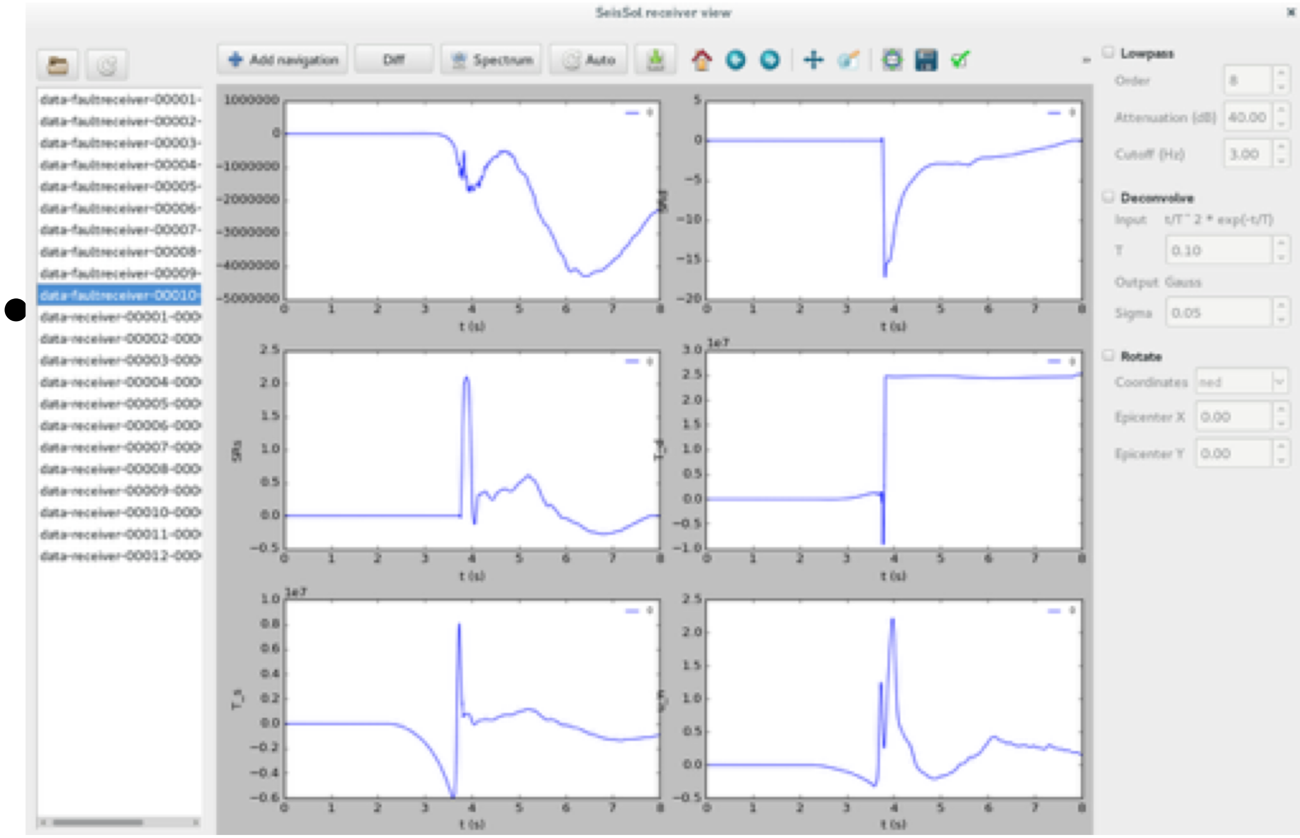


- Snapshots
- Movies
- Calculation of variables



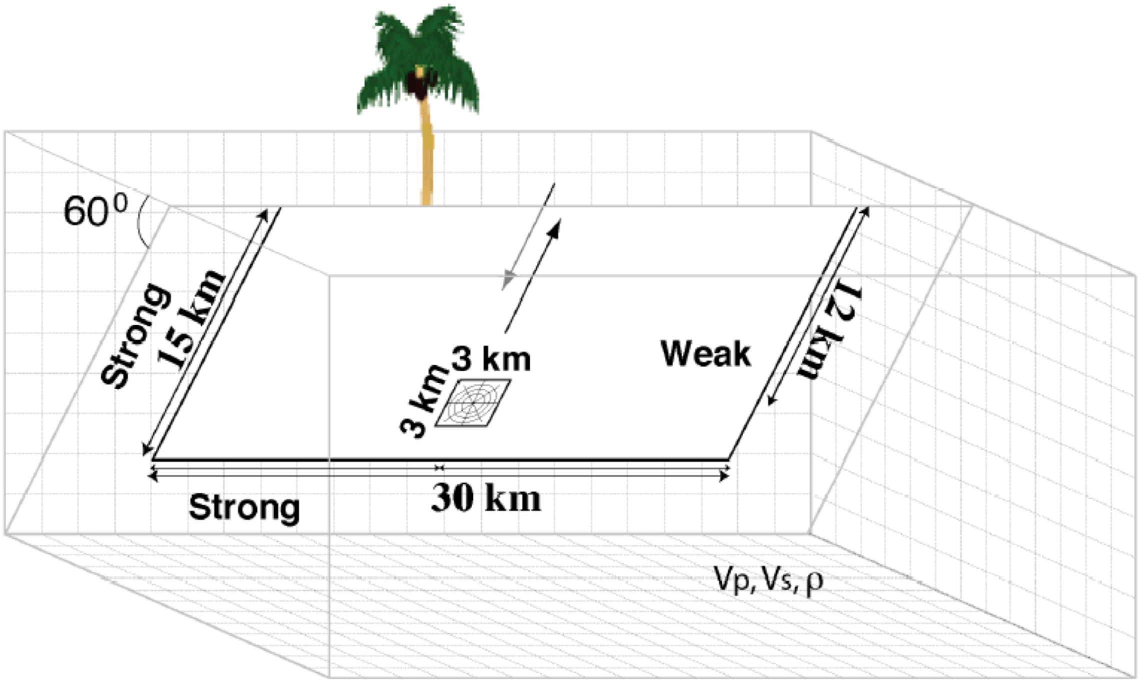
Kaikoura: 29 mio elements, 90 sec.,
2 hours on 3000 Sandy Bridge cores

Seismograph - Ascii

- 

Problem description

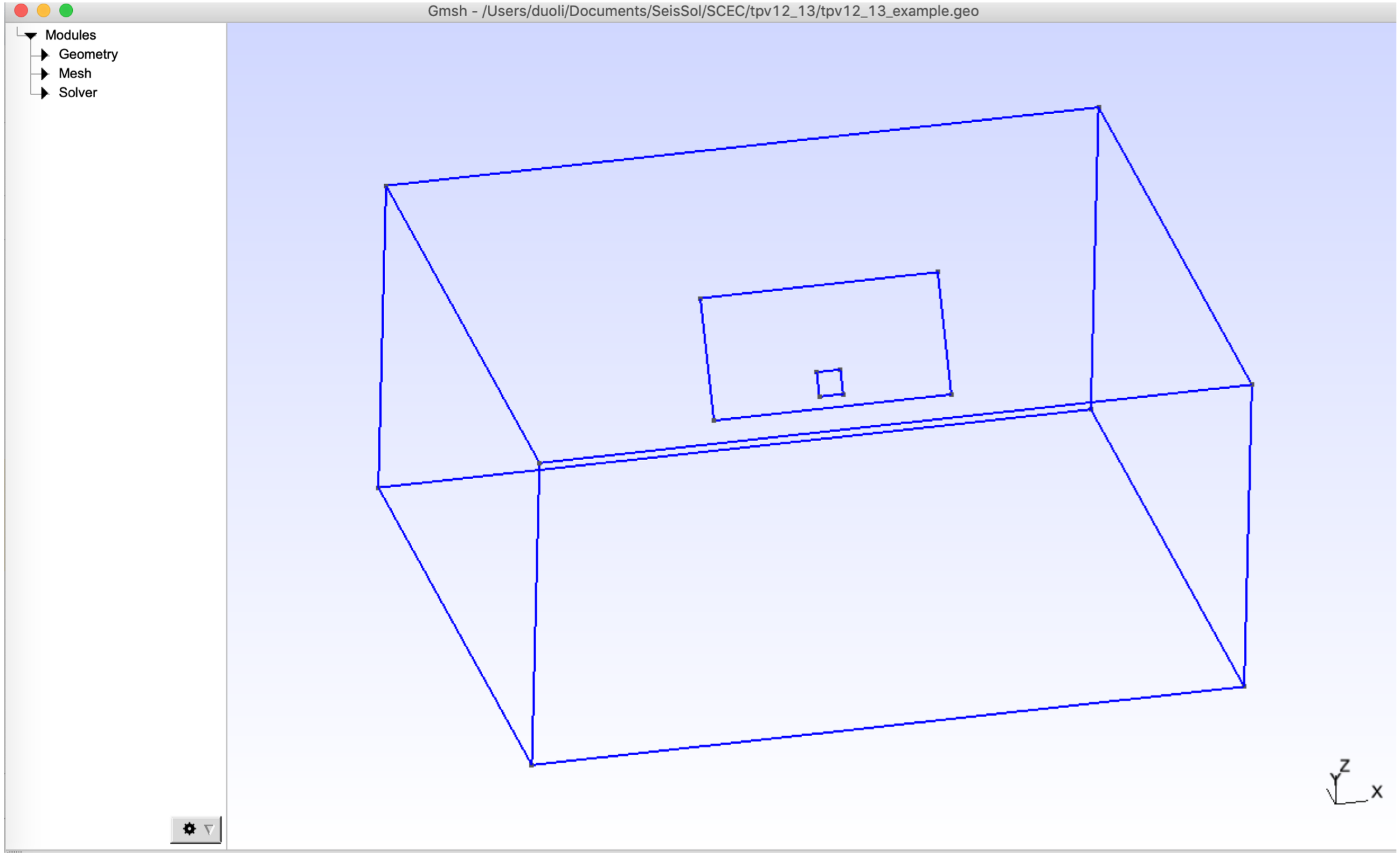
TPV13 - a dipping fault with off-fault plasticity



TPV13 - a dipping fault with off-fault plasticity

- TPV 13 describes spontaneous rupture on a **60-degree dipping normal fault** in a homogeneous half-space. Material properties are linear **elastic**.
- Initial stress conditions are dependent on depth. Strongly super-shear rupture conditions.
- TPV13 is using non-associative Drucker-Prager plasticity with yielding in shear.

Parameter setup


Gmsh - /Users/duoli/Documents/SeisSol/SCEC/tpv12_13/tpv12_13_example.geo
Modules
Geometry
Mesh
Solver
Z
Y
X

```
&equations
MaterialFileName = 'tpv12_13_material.yaml'
Plasticity = 1
Tv = 0.03
/

&Boundaries
BC_fs = 1                          ! free surface
BC_dr = 1                          ! Fault boundaries
BC_of = 1                          ! Absorbing boundaries
/
```

```yaml
!Switch
[rho, mu, lambda, plastCo, bulkFriction]: !ConstantMap
 map:
   rho:            2700
   mu:        2.9403e+010
   lambda:     2.941e+010
   plastCo:      5.0e+06
   bulkFriction:    0.85
[s_xx, s_yy, s_zz, s_xy, s_yz, s_xz]: !Include tpv12_13_initial_stress.yaml
```

```
&DynamicRupture
FL = 16                    ! variation of friction law
ModelFileName = 'tpv12_13_fault.yaml'

XRef = 0.0                 ! Reference point
YRef = -3.0e5
ZRef = 7.0e4
RF_output_on = 0           ! RF on
OutputPointType = 5        ! Type (0 : no output; 3 : ASCII fault receivers; 4 :
paraview file; 5 : both)
/
```

```yaml
!Switch
[s_xx, s_yy, s_zz, s_xy, s_yz, s_xz]: !Include tpv12_13_initial_stress.yaml
[mu_s, mu_d, d_c, cohesion]: !IdentityMap
 components:
  # Inside nucleation patch
  - !AxisAlignedCuboidalDomainFilter
    limits:
      x: [-1500, 1500]
      y: [-.inf, .inf]
      z: [-11691.34295108992, -9093.266739736605]
    components: !ConstantMap
      map:
        mu_s:       0.4
        mu_d:       0.10
        d_c:        0.50
        cohesion: -200000
  # Outside nucleation patch
  - !ConstantMap
    map:
      mu_s:       0.70
      mu_d:       0.10
      d_c:        0.50
      cohesion: -200000
```

```
!Switch
[s_xy, s_yz, s_xz]: !ConstantMap
 map:
   s_xy: 0
   s_yz: 0
   s_xz: 0
[s_xx, s_yy, s_zz]: !FunctionMap
 map:
   depth: return abs(z);
   s_max_minus_Pf: return 9.8 * (2700.0 - 1000.0);
 components:
   # Upper region (includes fault)
   - !AxisAlignedCuboidalDomainFilter
     limits:
       depth:          [0, 11951.15]
       s_max_minus_Pf: [-.inf, .inf]
     components: !FunctionMap
       map:
         # Round to two significant digits as in benchmark description
         s_xx: return -0.01 * round(100.0 * (0.5 * (1.0 + 0.3496) * s_max_minus_Pf)) * depth;
         s_yy: return -0.01 * round(100.0 *             (0.3496 * s_max_minus_Pf)) * depth;
         s_zz: return -s_max_minus_Pf * depth;
   # Lower region (excludes fault)
   - !FunctionMap
     map:
```

```
&Elementwise
printIntervalCriterion = 2          ! 1=iteration, 2=time
printtimeinterval_sec = 0.5          ! Time interval at which output will be written
OutputMask = 1 1 1 0 1 1 1 1 1 0 0      ! output 1/ yes, 0/ no - position:
refinement_strategy = 1
refinement = 1
/
&Pickpoint
printtimeinterval = 1       ! Index of printed info at timesteps
OutputMask = 1 1 1 0      ! output 1/ yes, 0/ no - position: 1/ slip rate 2/ stress 3/ normal velocity 4/ in case
of rate and state output friction and state variable
nOutpoints = 10
PPFileName = 'tpv13_faultreceivers.dat'
/
```

1. SRs and SRd: slip rates in strike and dip direction
2. T_s, T_d: transient shear stress in strike and dip direction, P_n: transient normal stress
3. U_n*: normal velocity (note that there is no fault opening in SeisSol)
4. Mud: current friction, StV: state variable in case of RS friction
5. Ts0,Td0,Pn0: total stress, including initial stress
6. Sls and Sld: slip in strike and dip direction
7. Vr: rupture velocity, computed from the spatial derivatives of the rupture time
8. ASl: absolute slip
9. PSR: peak slip rate
10. RT: rupture time
11. DS: only with LSW, time at which ASl>D_c

```
&MeshNml
MeshFile = 'tpv13_mesh.h5'          ! Name of mesh file
meshgenerator = 'PUML'              ! Name of mesh generator (format)
/


&Discretization
CFL = 0.5                           ! CFL number (<=1.0)
FixTimeStep = 5                     ! Manualy chosen minimum time
ClusteredLTS=2                      ! This enables local time stepping
/
```

```
&Output
OutputFile ='./output/data'
iOutputMask = 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1      ! Variables output
! x y z stress (6) vel (3) rho, cp,cs
Format = 6                                      ! Format (10= no output, 6=hdf5 output)
Refinement =1                    ! Format (0=IDL, 1=TECPLOT, 2=IBM DX, 4=GiD))
TimeInterval = 0.5               ! Index of printed info at time
printIntervalCriterion = 2       ! Criterion for index of printed info: 1=timesteps,2=time,3=timesteps+time
SurfaceOutput = 1
SurfaceOutputRefinement = 1
SurfaceOutputInterval = 0.05
pickdt = 0.005                   ! Pickpoint Sampling
pickDtType = 1                   ! Pickpoint Type
FaultOutputFlag = 1              ! DR output (add this line only if DR is active)
nRecordPoints = 12               ! number of Record points which are read from file
RFileName = 'tpv13_receivers.dat'        ! Record Points in extra file
```

&AbortCriteria

EndTime = 8.0

/

1.Download SeisSol

git clone --recursive https://github.com/SeisSol/SeisSol.git

git submodule update --init

2.Compile SeisSol

```
$ cd SeisSol_dir
$ scons buildVariablesFile=compilation.py
$ cp build/SeisSol_excution_code working_dir/
$ echo SeisSol_dir/Maple/ > working_dir/DGPATH
```

3.Download TPV13

$ Git clone https://github.com/daisy20170101/SeisSol_Cookbook.git

Under the main directory, you will find tpv12_13

More information about SCEC dynamic simulation validation project:
http://scecdata.usc.edu/cvws/benchmark_descriptions.html

Hands-on exercise

Group work

As we don’t have enough HPC resources for everyone, we can only 7 volunteers to work now but others can work later!

The 7 volunteers can help the others in the rest of days!

Login in Argo

First, *login* to ICTP desktop

Then, *ssh argo.ictp.it -l account_name* to Argo HPC cluster

Load SeisSol in Argo

$ cp -rf /home/dli/seissol_exer1 your_working_dir

$ source bash_seissol

$ bash interactivte_script.sh

This will take a while to allocate the nodes that are asked.

When you see something like:

Srun: your resources has been invoked

It means that you successfully get allocated.

Then you can submit by:

$ mpirun ./SeisSole_xxxx paramters_tpv12_13.par

Load SeisSol in Argo

```
$ sinfo

$ sbatch submission.sh

$ squeue -u your_account
```

Results Visualization

Gmsh

Gmsh mesh generation

$ Gmsh -3 -optimize tpv12_13.geo -o tpv12_13.msh

Gmsh mesh generation

$ scp tpv12_13.msh account@argo.ictp.it:~

$ ssh argo.ictp.it -l account

$ cp /home/dli/.bashrc .

$ source .bashrc

$ gmsh2gambit -i tpv12_13.msh -o tpv12_13.neu

Gmsh mesh generation

```
$ /home/dli/PUMGen/build2/pumgen tpv12_13.neu tpv12_13
```

Gmsh mesh generation

Some explanations:

- .geo is Gmsh geometry file
- Gmsh2gambit: SeisSol_main/preprocessing/meshing/gmsh2gambit. Compile follow the instruction
- PUMGen: https://github.com/SeisSol/PUMGen/wiki/How-to-compile-PUMGen

Download results

$ cp -rf /home/netapp/clima-scratch/dli/tpv13_output your_own_dir

Gmsh mesh generation

$ Gmsh -3 -optimize tpv12_13.geo -o tpv12_13.msh

$ Gmsh2gambit -i tpv12_13.msh -o tpv12_13.neu

$ /home/dli/PUMGen/build2/pumgen tpv12_13.neu tpv12_13

Some explanations:

- .geo is Gmsh geometry file
- Gmsh2gambit: SeisSol_main/preprocessing/meshing/gmsh2gambit. Compile follow the instruction
- PUMGen: https://github.com/SeisSol/PUMGen/wiki/How-to-compile-PUMGen

Thanks!