ChEESE

www.cheese-coe-eu

Center of Excellence for Exascale in Solid Earth

LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

SeisSol training

today's LMU SeisSol training team

www.geophysik.uni-muenchen.de/Members/gabriel

Carsten Uphoff

Bo Li

Thomas Ulrich

Duo Li

Alice-Agnes Gabriel

Aniko Wirp

www.seissol.org

github.com/SeisSol

Oct. 19, 2021

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 823844

Training agenda

13:30 - 14:30 **General introduction** (Docker setup to run Jupyter notebooks)
& kinematic earthquake model example (Northridge)

14:30 - 15:30 **Dynamic rupture example 1** (normal fault with off-fault damage)

15:30 - 16:30 **Dynamic rupture example 2** (Palu Sulawesi)
& wrap up

As a **backup** we pre-calculated outputs for all three examples here:
/shared/cheese/seissol/outputs/

What is inside the seissol-training Docker container?

The Docker image contains a training environment based on JupyterLab. It comes with pre-installed tools for point source file and mesh creation (rconv/gmsh/pumgen).

You did install Docker, right?

On your desktop/laptop: Docker setup

Please open a terminal (Windows: PowerShell) and run (right now)

```
$ docker pull uphoffc/seissol-training
> docker pull uphoffc/seissol-training
```

Note: **$** Linux/Mac, **>** Windows, **$/>** all systems

On your desktop/laptop: Docker setup

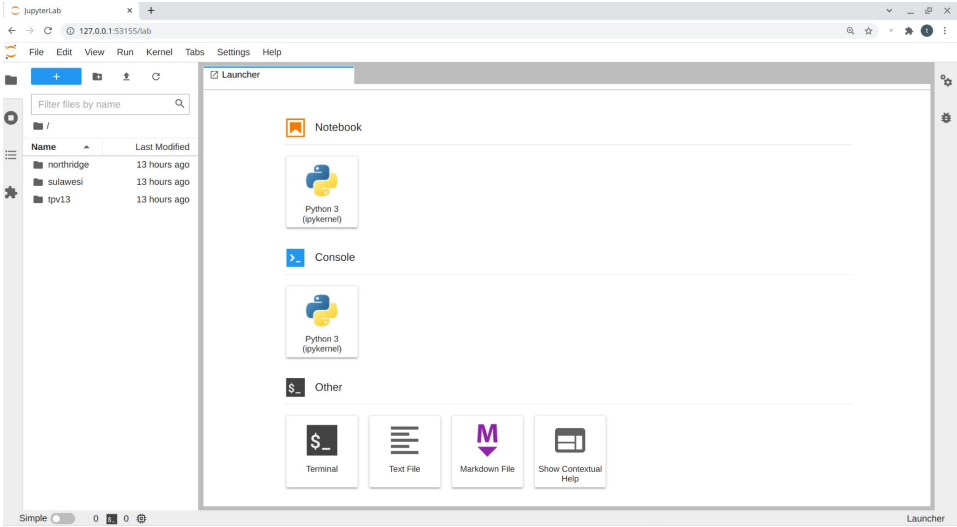
1. Run in a terminal (Windows: PowerShell)
   $/> docker run -p 53155:53155 uphoffc/seissol-training

2. After some time you should see:
   http://127.0.0.1:53155/lab?token=some5cryptic8hash123

Click on that link or enter the link in the address bar of your favourite web browser.

On your desktop/laptop: Docker setup

You should see the following:



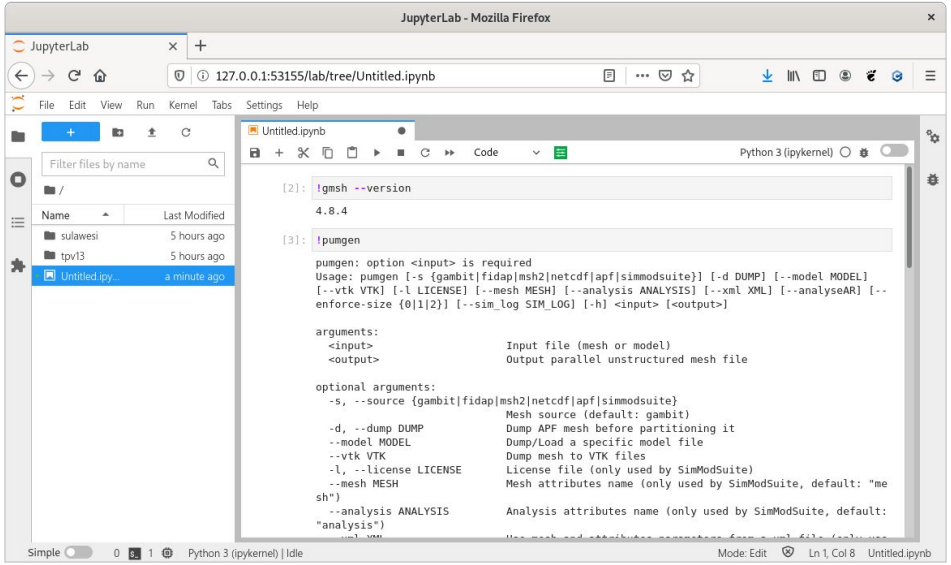
On your desktop/laptop: Docker setup

Create a new notebook.

The packaged tools and SeisSol can be run by prepending **!**

In-built commands:

- !gmsh
- !pumgen
- !rconv
- !SeisSol_Release_dhsw_4_elastic
- !SeisSol_Release_dhsw_4_viscoelastic2



Docker containers

Docker does not modify your local files. Instead files created within the training environment (JupyterLab) are saved in a container.

To list running containers enter

```
$/> docker container ls
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
5097c713b25f	uphoffc/seissol-training	"tini -g /entrypoint..."	28 minutes ago	Up 28 minutes	0.0.0.0:53155->53155/tcp, :::53155->53155/tcp	**sleepy_carson**

Copying files from Docker containers

Do you remember the name of your container? Replace sleepy_carson by the **name of your container**:

$/> docker cp **sleepy_carson**:/home/training/tpv13/tpv13.jpg .

Above command copies the image tpv13.jpg to your current working directory.

Docker containers

Stopped containers do not appear in the list. To see them enter

$/> docker container ls **-a**

Want to clean up? Enter

$/> docker system prune

Terminal efficiency

Enter Ctrl + R to browse your history.

E.g., “Ctrl + R” and “docker”
finds your last docker command!

(Works in PowerShell, too.)

Example 1 - ground motions from a kinematic rupture model of the 1994 Northridge earthquake

seismic wave propagation from many point sources, specifically moment tensor sources, describing a kinematic finite earthquake model

The 1994, M6.7 Northridge earthquake

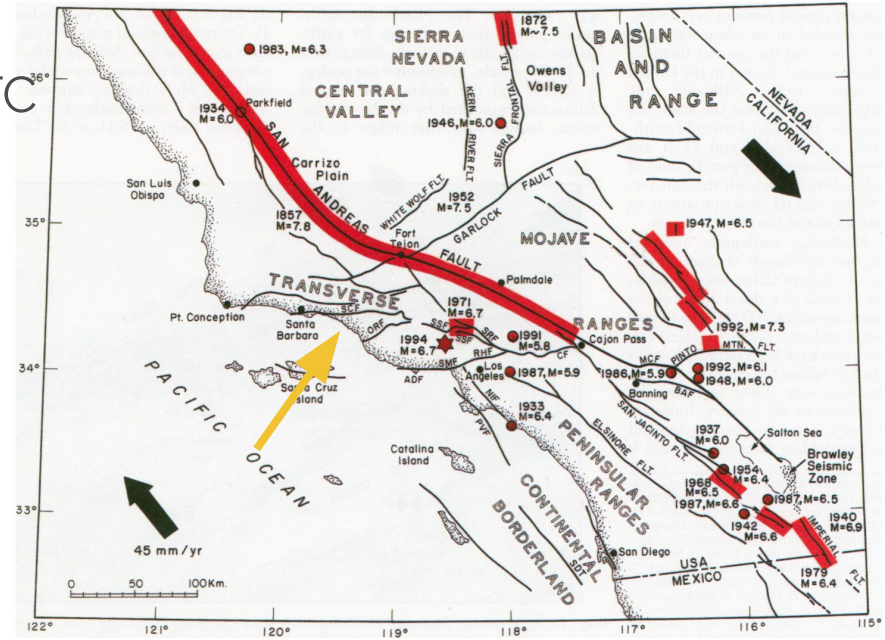
- "60 people were killed, more than 7,000 injured, 20,000 homeless and more than 40,000 buildings damaged in Los Angeles, Ventura, Orange and San Bernardino Counties." …
- "Fires caused additional damage." …
- "Damage are between 13 and 20 billion U.S. dollars"

(USGS)

→ **most costly American earthquake since 1906**

The 1994, M6.7 Northridge earthquake

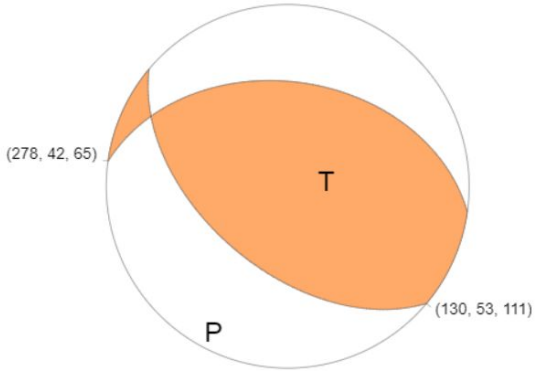
- Origin time: 1994-01-17 12:30:55.390 UTC
- Magnitude: 6.7
- Location: 34.213°N 118.537°W
- Depth: 18.2 km ± 0.4
- Duration: approx. 10-20 sec.



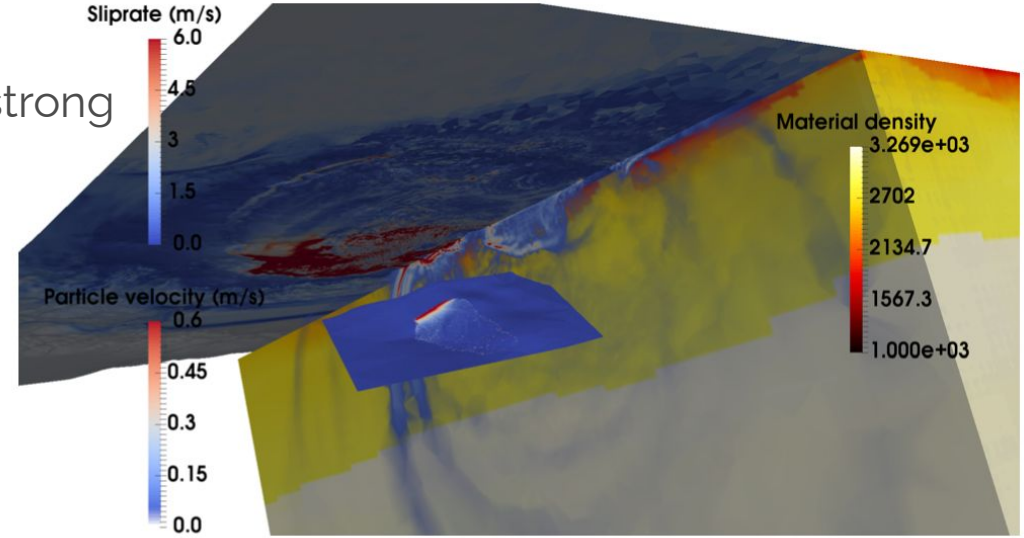
above: Map of southern california, with major faults and historic fault ruptures (red). USGS, SCEC, science (1994)

The 1994, M6.7 Northridge earthquake

- ‘blind’ thrusting earthquake on an unknown fault
- Located directly under the city, very strong shaking



(USGS)



Kinematic earthquake modeling

Represent source as surface distribution of body forces.

Far-field approximation: Source is concentrated at point.

Source-time function

Normal

$$f_p(\boldsymbol{x}, t) = -AS_i(t)c_{ijpq}n_j \frac{\partial}{\partial x_q}\delta(\boldsymbol{x} - \boldsymbol{x}_c).$$

Area

Stiffness tensor

"Centre"

(Aki & Richards 2002)

Kinematic earthquake modeling

- Kinematic rupture descriptions available for many earthquakes (e.g, SRCMOD data-base http://equake-rc.info/srcmod/, USGS earthquake catalog)
- Inferred from observations by solving the inverse problem
- But: Solutions of the inverse problem are strongly non-unique due to ill-conditioning (Mai 2016)
- Finite source by superposition
- No description of stresses and strength necessary

Hands on! (30 minutes)

We will go into breakout rooms to do the exercise in 2 groups. Later we will meet again in the main zoom room.

The trainers will help you during the exercise, you will use the Jupyter notebook.

Task 1

- Work through the Jupyter Notebook
- Generate the mesh for our Northridge example

For more information on the meshing:
https://seissol.readthedocs.io/en/latest/gmsh.html

Task 2

- Understand the parameter.par file

for help:
https://seissol.readthedocs.io/en/latest/parameter-file.html

parameters.par

https://seissol.readthedocs.io/en/latest/parameter-file.html

```
&equations
! yaml file defining spatial dependance of material properties
MaterialFileName = 'northridge_material.yaml'
/
&IniCondition
/
&Boundaries
BC_fs = 1    ! enable free surface boundaries
BC_dr = 0    ! enable fault boundaries
BC_of = 1    ! enable absorbing boundaries
/
&SourceType
Type = 42   ! 42: finite source type in netcdf format
FileName = 'northridge_resampled.nrf'   ! name of input file
/
```

parameters.par

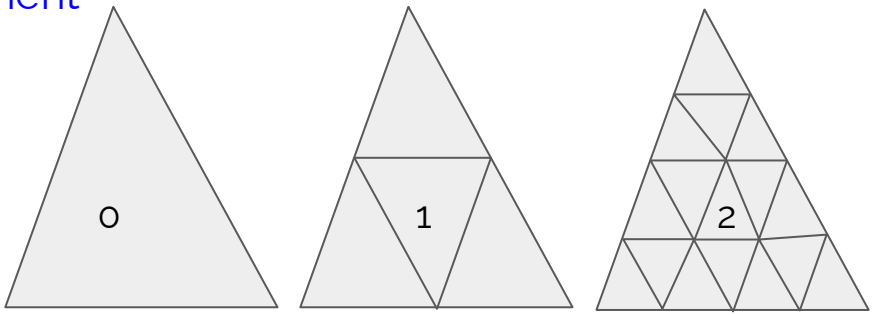
https://seissol.readthedocs.io/en/latest/parameter-file.html

```
&SpongeLayer
/
&MeshNml
MeshFile = 'mesh_northridge.puml.h5' ! hdf5 unstructured tetrahedral mesh
meshgenerator = 'PUML'               ! name of mesh generator
/
&Discretization
CFL = 0.5            ! The Courant-Friedrichs-Levy (CFL) Stability Criterion (<=1.0)
ClusteredLTS = 2
! 1 for global time-stepping, 2,3,5,...for local time-stepping
! Elements are clustered by their time-step in the time bins
! [t_{min} * ClusteredLTS^c, t_{min} * ClusteredLTS^{c+1})
! where t_{min} = min_{element} t_{element} and c in N_0 is chosen such that
! t_{min} * ClusteredLTS^c <= t_{element}
! Default choice: ClusteredLTS = 2
```

parameters.par

https://seissol.readthedocs.io/en/latest/parameter-file.html

```
&Output
FaultOutputFlag = 0                    ! write 2D dynamic rupture output across fault interfaces
OutputFile = 'output/test'             ! output files will be written to a folder named 'output' and
                                       all output files will have 'test' prefix

Format = 10                            ! Volume output format (10= no output, 6=hdf5 output)
!                    |stress    |vel
iOutputMask = 1 1 1 1 1 1 1 1 1
TimeInterval =  5.0      ! output time interval of volume output every 5 seconds
refinement = 1           ! use sub-element refinement

! free surface output parameters
SurfaceOutput = 1        ! activate surface output
SurfaceOutputRefinement = 1
SurfaceOutputInterval = 1.0
printIntervalCriterion = 2
! how output frequency is chosen, based on: 1=#timesteps, 2=time (s), 3=timesteps+time
```



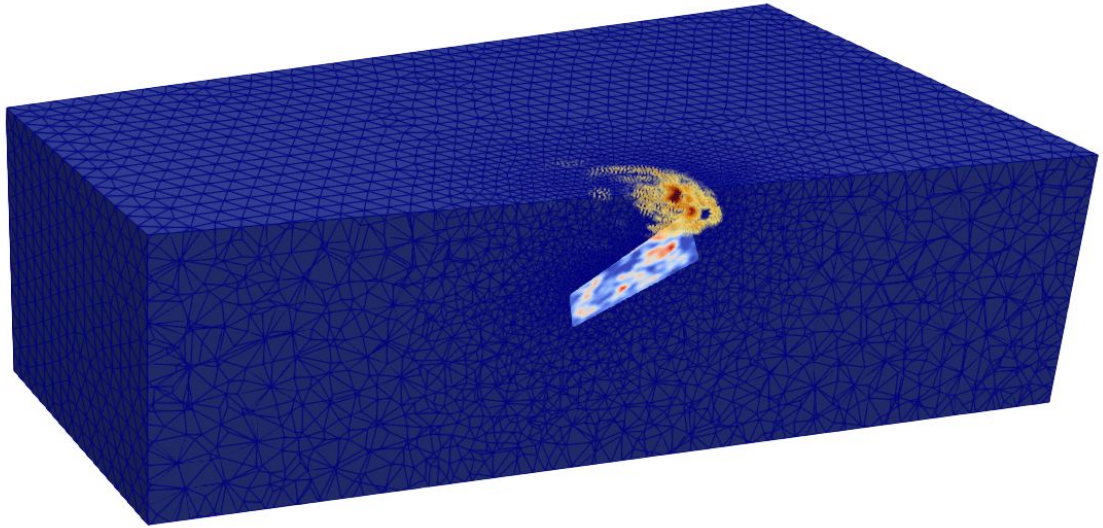
parameters.par

https://seissol.readthedocs.io/en/latest/parameter-file.html

```
pickdt = 0.005                    ! synthetic seismogram sampling of 0.005 s
pickDtType = 1                    ! Pickpoint Type (1 = time, 2 = timestep(s))
nRecordPoints = 6                 ! number of receivers
RFileName = 'offreceivers.dat'    ! receiver coordinates (ASCII file, x y z)

!xdmfWriterBackend = 'posix' ! (optional)
! The backend used in fault, wavefield, and free-surface output.
! The 'hdf5' backend is only supported when SeisSol is compiled with HDF5 support.
/
&AbortCriteria
EndTime = 10.0 ! the simulation will run for 10 seconds physical time
/
&Analysis
/
&Debugging
/
```

offreceivers.dat

x1 y1 z1
x2 y2 z2
x3 y3 z3
...



Locations of nRecordPoints seismometers recording during the simulation in (x,y,z) in meters

positive z is up

material.yaml → easi illustration

```yaml
!LayeredModel
map: !AffineMap
  matrix:
    depth: [0.0, 0.0, -1.0]
  translation:
    depth: 0
interpolation: linear
parameters: [rho, mu, lambda]
nodes:
  -100: [   2600 ,5667078183.27402 ,5667078183.27402]
  2000: [   2600 ,16693624487.2577 ,16693624487.2577]
  4000: [   2700 ,48290591620.5055 ,48290591620.5054]
  6000: [   2700 ,60447064950.5947 ,60447064950.5947]
  8000: [   3000 ,84158664883.6690 ,84158664883.6689]
  10000: [  3000 ,108603910096.339 ,108603910096.339]
  12000: [  3300 ,141703292308.016 ,141703292308.016]
  15000: [  3300 ,230555644357.842 ,230555644357.842]
  18000: [  3500 ,290670977957.982 ,290670977957.982]
  23000: [  3500 ,583499107063.831 ,583499107063.831]
  30000: [  3800 ,13004698061216.9 ,13004698061216.9]
```

1D model:
rho, mu and lambda are assigned for specific depths and linearly interpolated between the layers

‘Standard’ Rupture Format (SRF)

ASCII file format to store moment tensor sources (restricted to double-couple).

specification file:

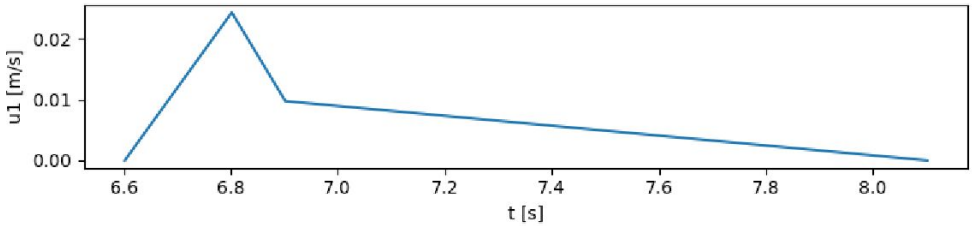
http://hypocenter.usc.edu/research/SRF/srf-v2.0_rev1.pdf

‘Standard’ Rupture Format (SRF)

LON	LAT	DEP	STK	DIP	AREA	TINIT	DT	(	VS	DEN)
RAKE	SLIP1	NT1	SLIP2	NT2	SLIP3	NT3				
SR1[1]	SR1[2]	SR1[3]	. . .	SR1[NT1]						
SR2[1]	SR2[2]	SR2[3]	. . .	SR2[NT3]						
SR3[1]	SR3[2]	SR3[3]	. . .	SR3[NT3]						

version 2.0 only

LON	longitude of subfault center
LAT	latitude of subfault center
DEP	depth of subfault center (km)
STK	strike
DIP	dip



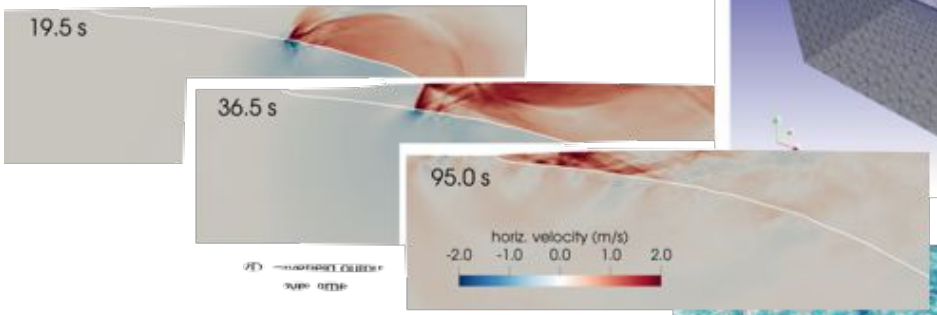
Task 3: Copying files from Docker containers

Do you remember the name of your container? Replace sleepy_carson by the **name of your container**:

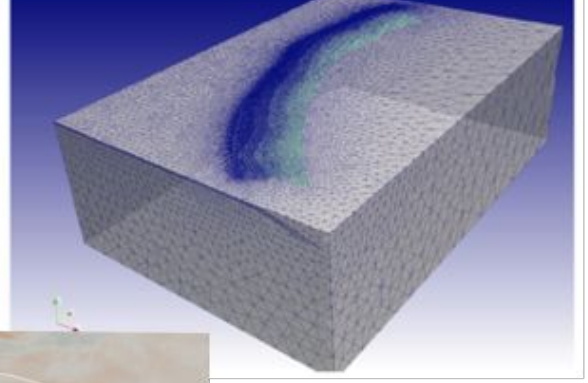
```
$/> docker cp sleepy_carson:/home/training/northridge/output .
```

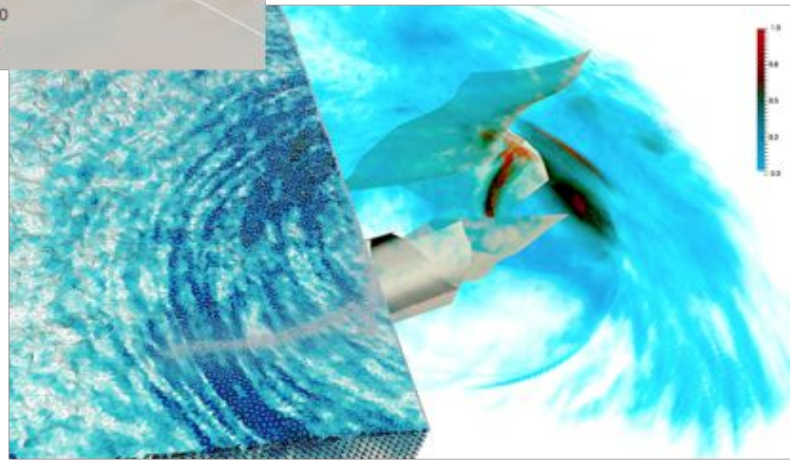
Above command copies the outputs of seissol to your current working directory.

ParaView for visualizing your results



2D SeisSol (Laptop)

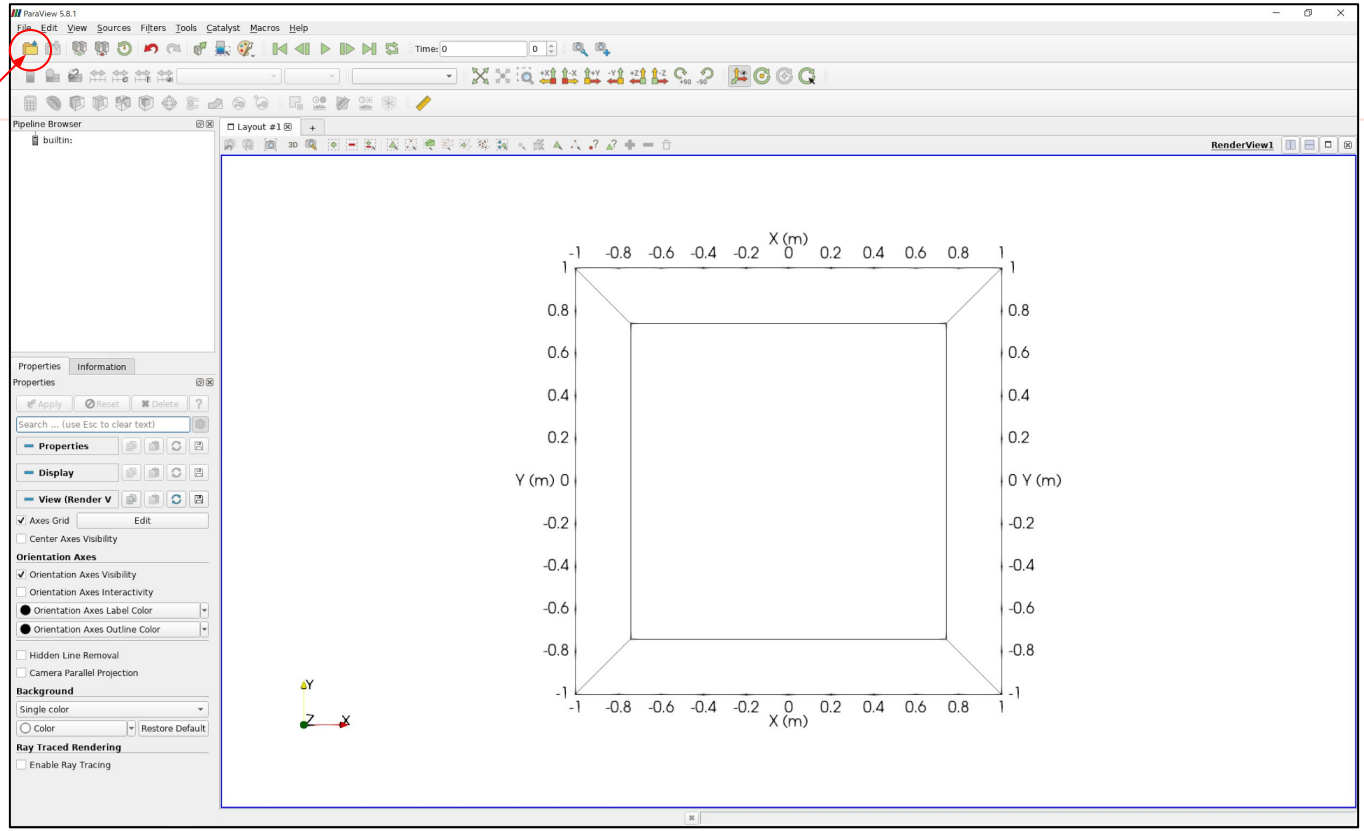




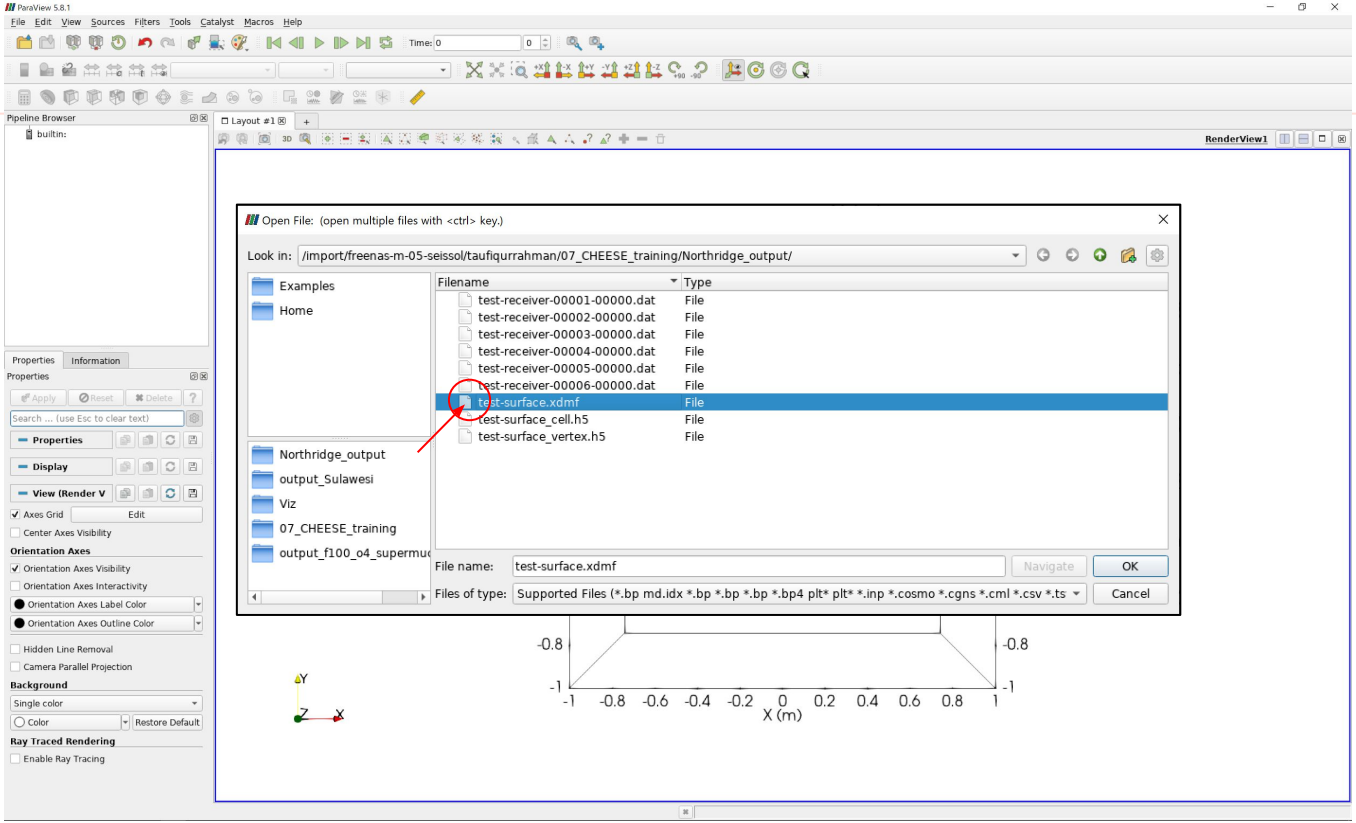
Kaikoura: 29 mio elements, 90 sec.,
2 hours on 3000 Sandy Bridge cores

- Time series (seismograms), 2D (fault, free surface) and 3D output fields
- Paraview filters for simple post-processing

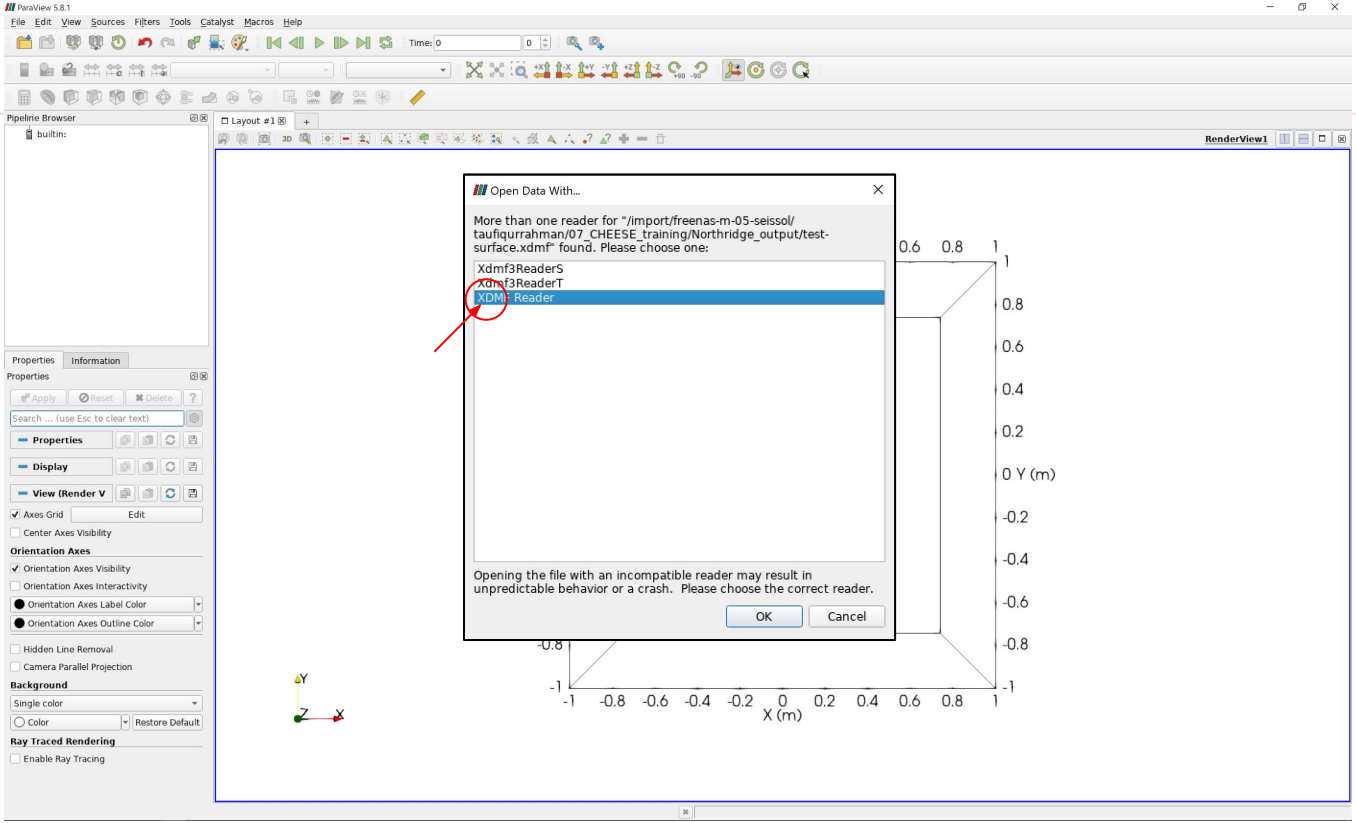
local desktop/laptop, ParaView: open file



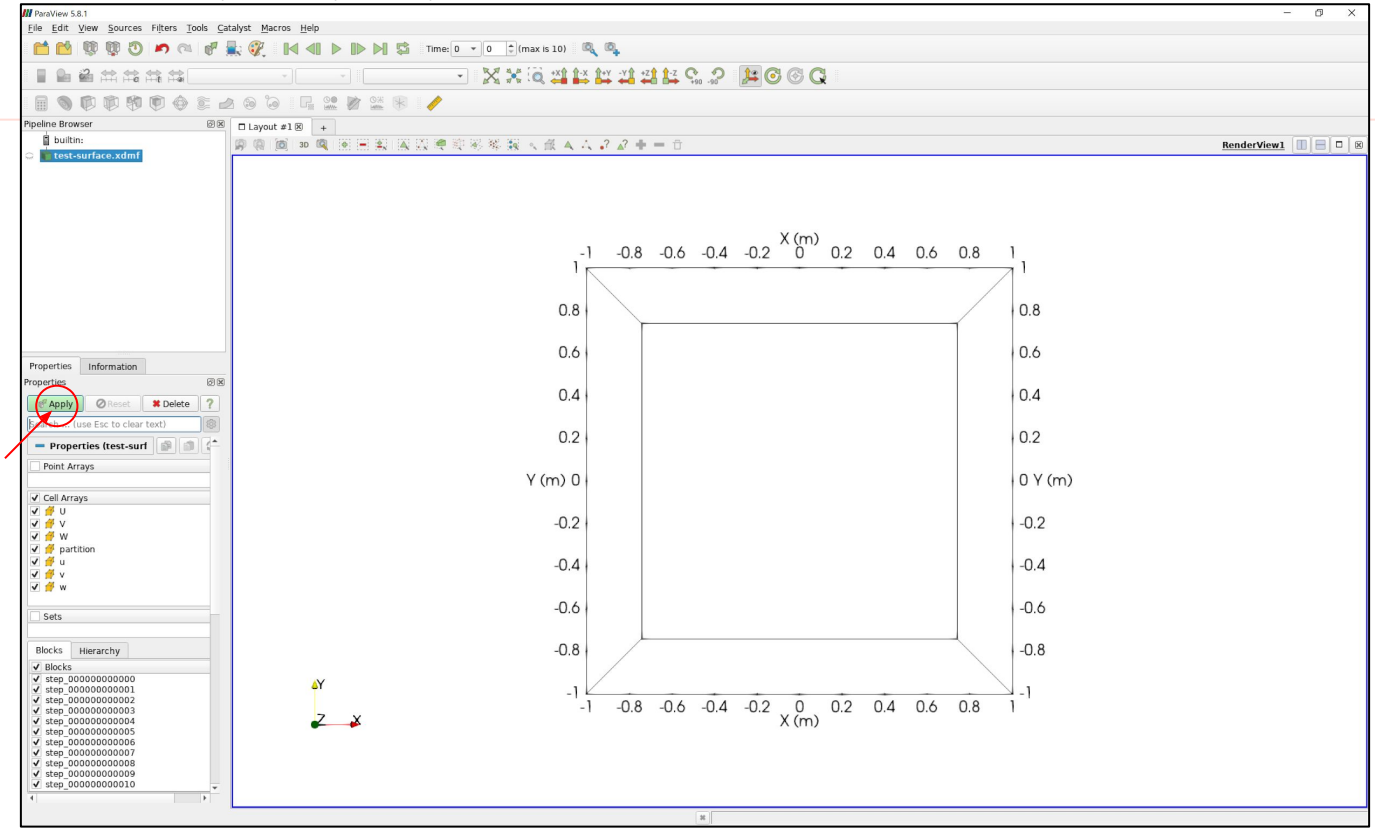
local desktop/laptop, ParaView: load free surface 2D output



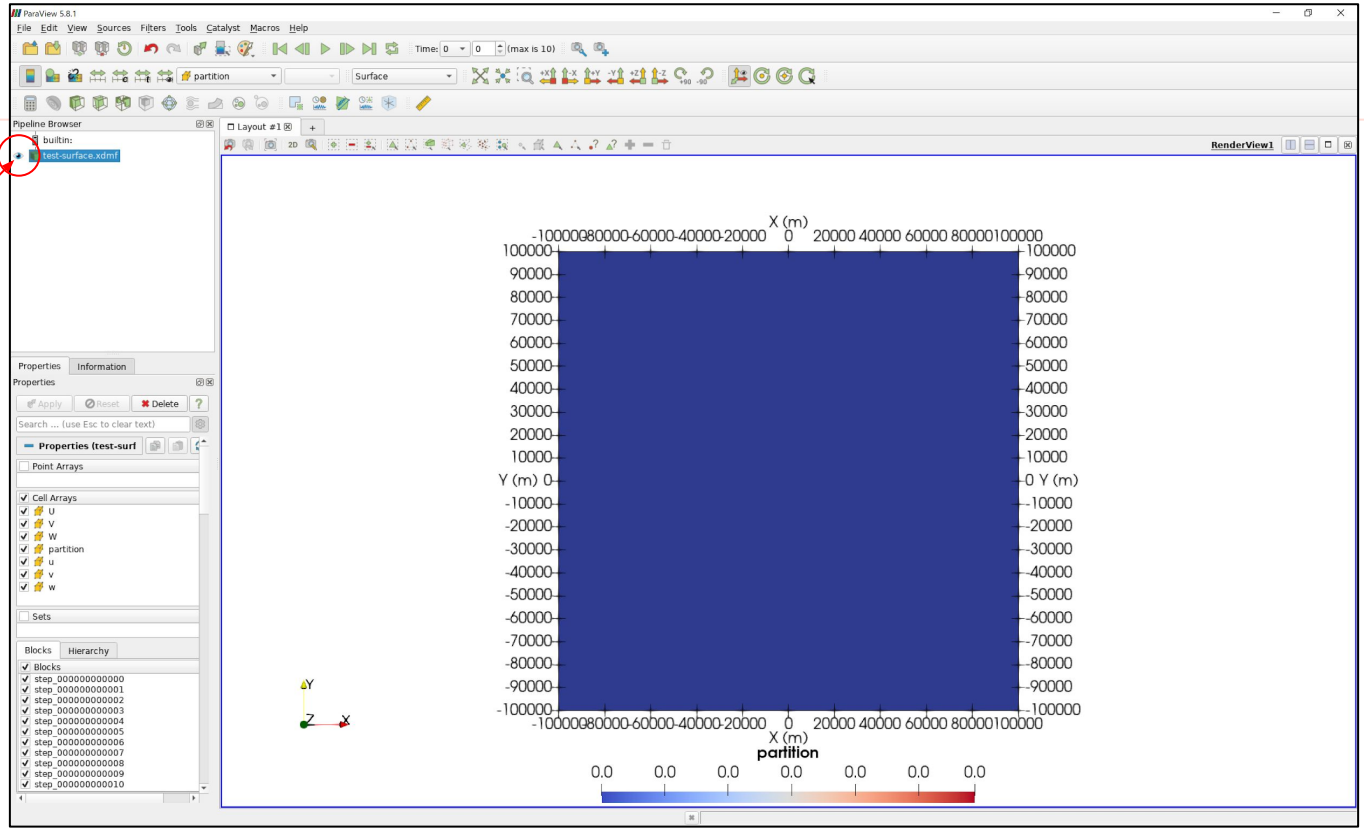
local desktop/laptop, ParaView: load free surface 2D output



local desktop/laptop, ParaView: load free surface 2D output

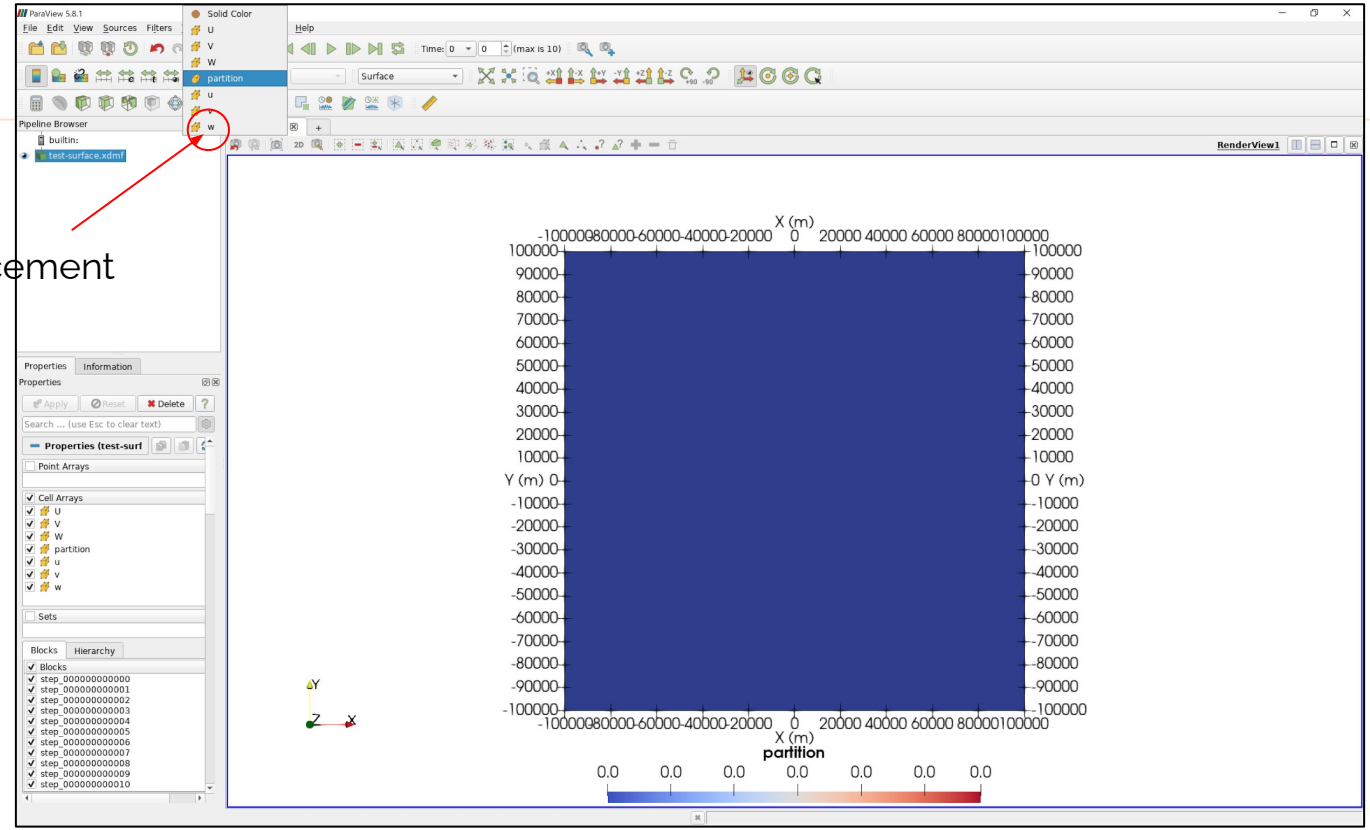


local desktop/laptop, ParaView: analyse free surface output

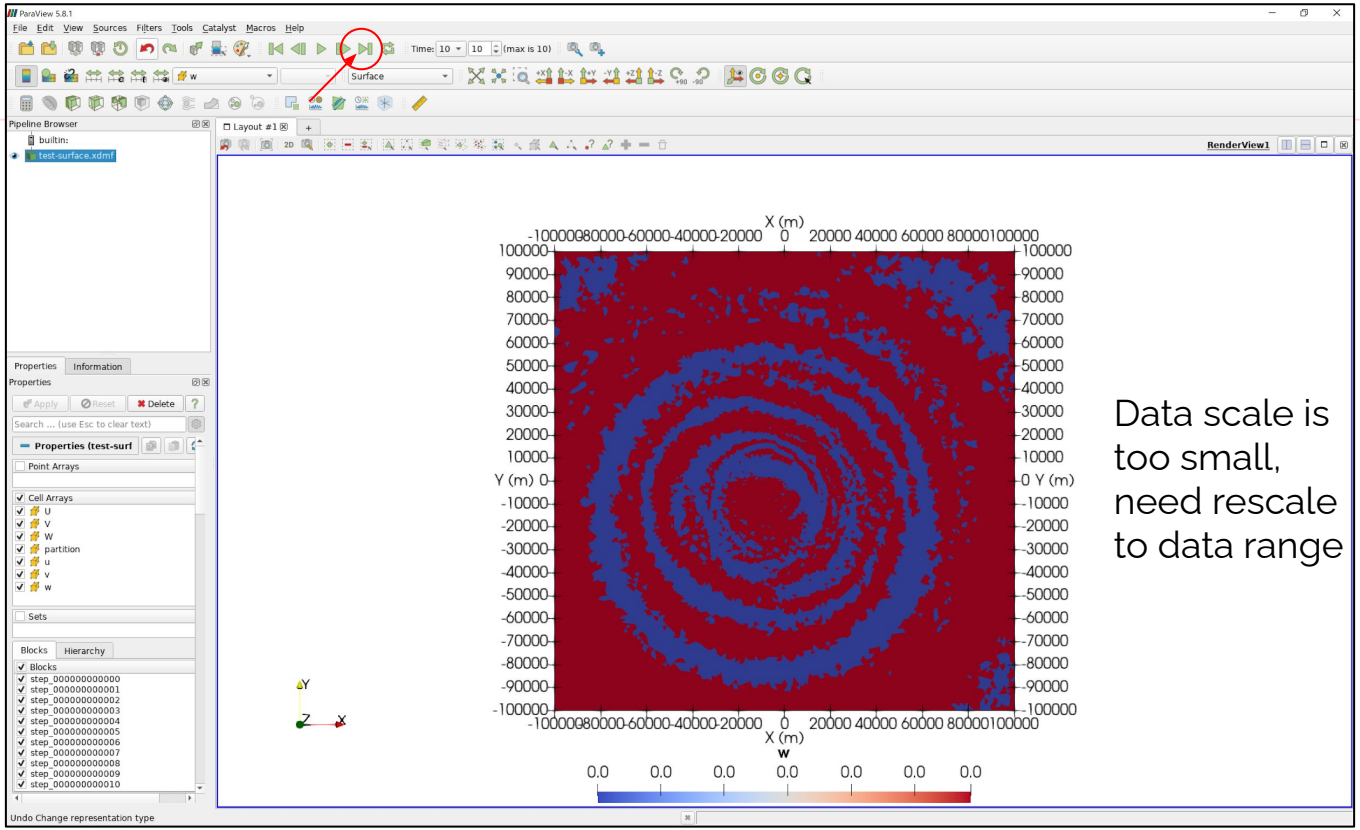


local desktop/laptop, ParaView: select an output attribute

u, v, w : velocity
U, V, W : displacement

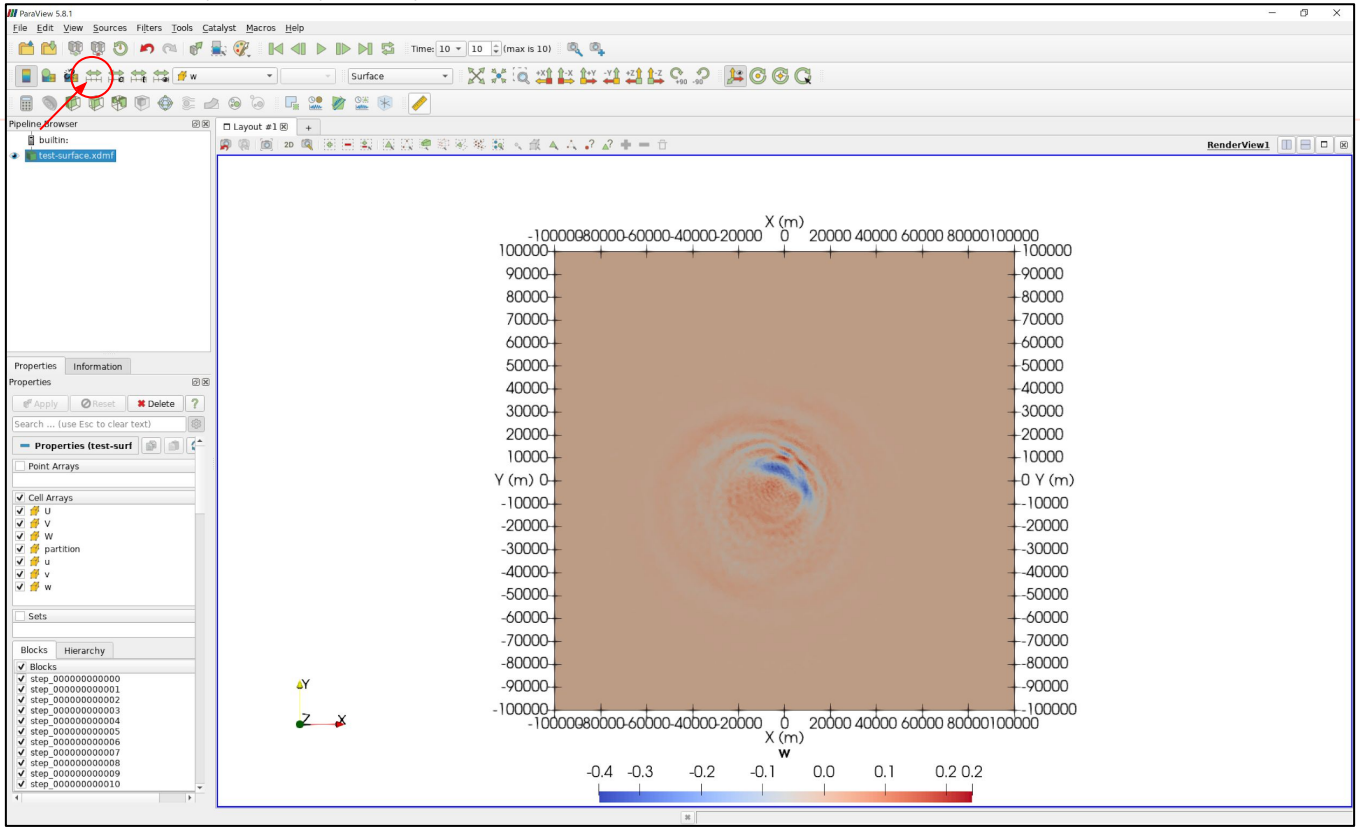


local desktop/laptop, ParaView: move to the last time step

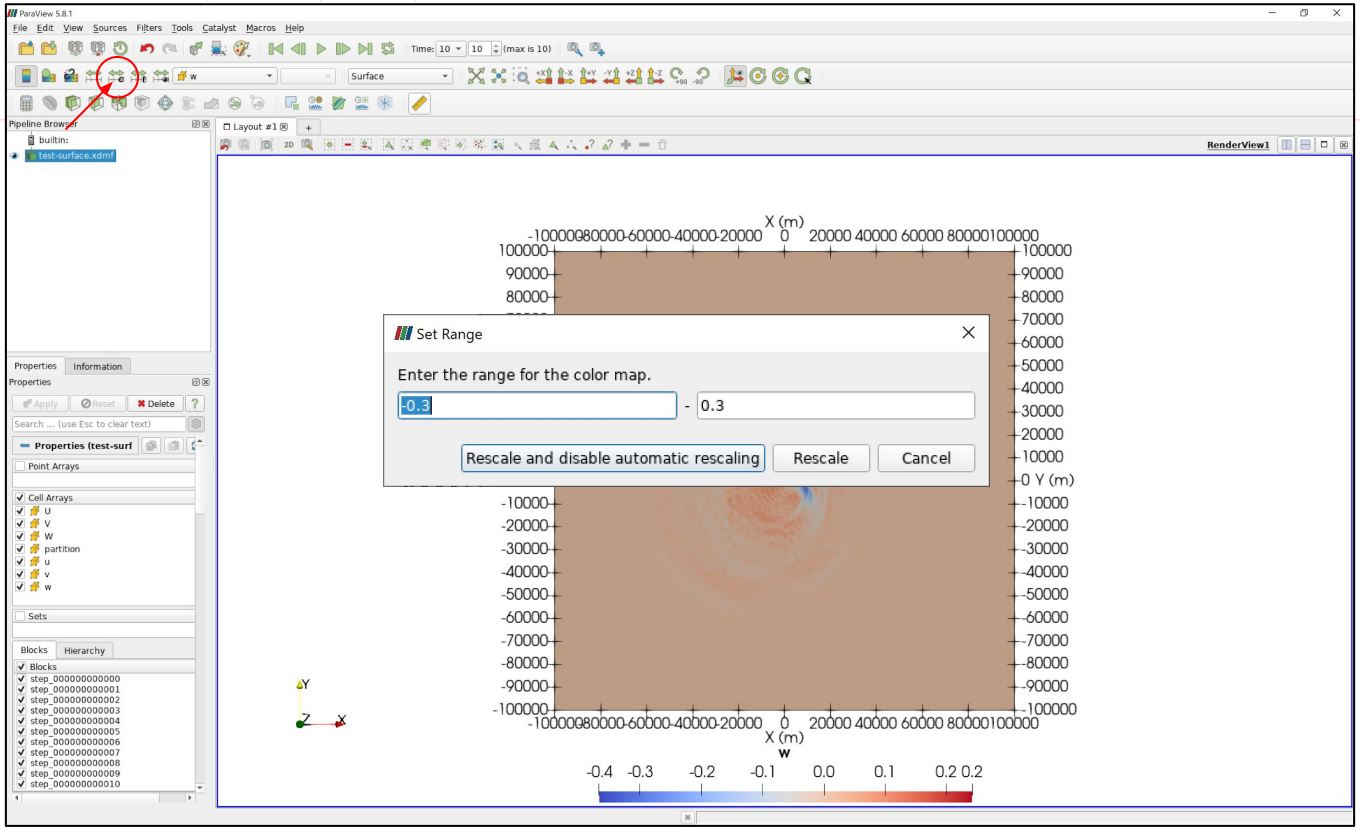


Data scale is too small, need rescale to data range

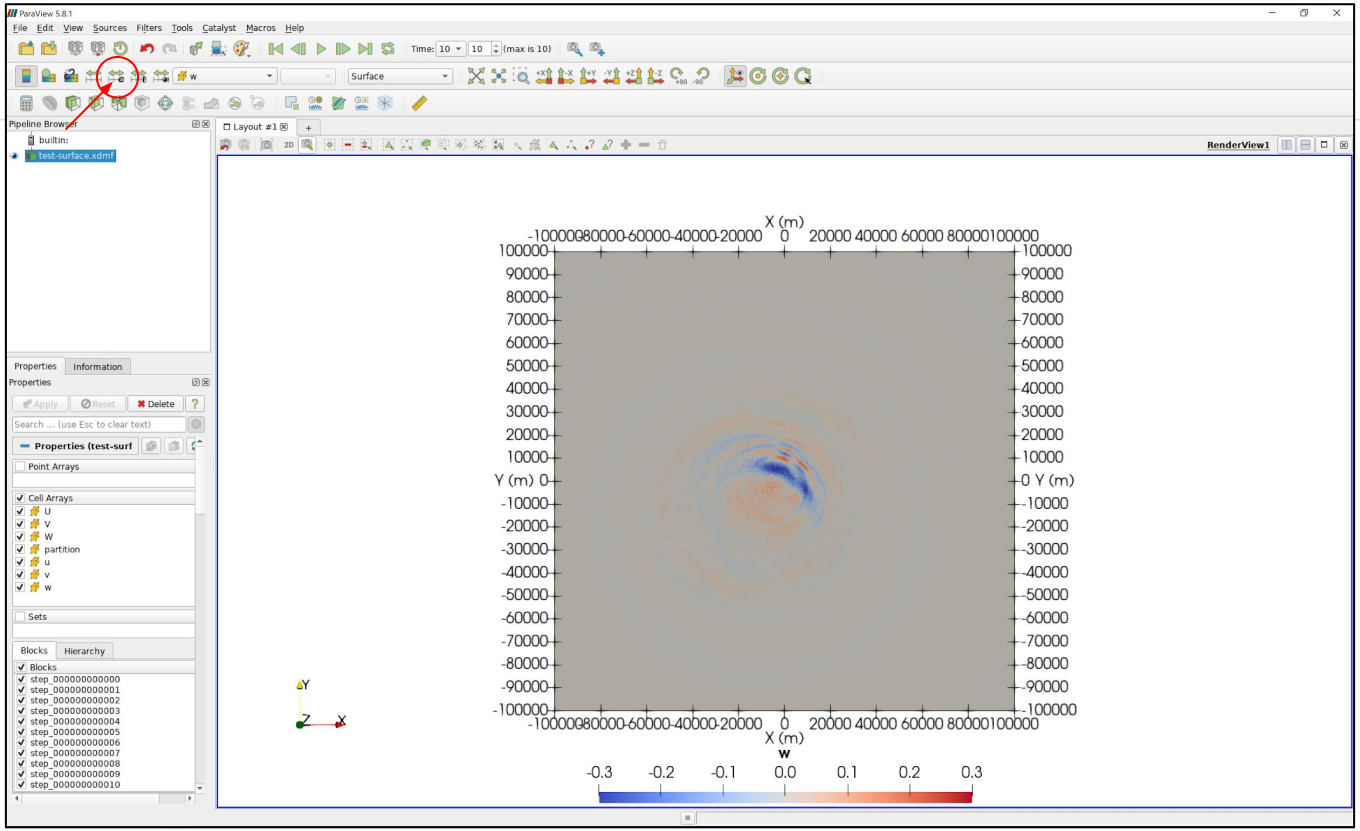
local desktop/laptop, ParaView: rescale to data range



local desktop/laptop, ParaView: rescale to custom data range

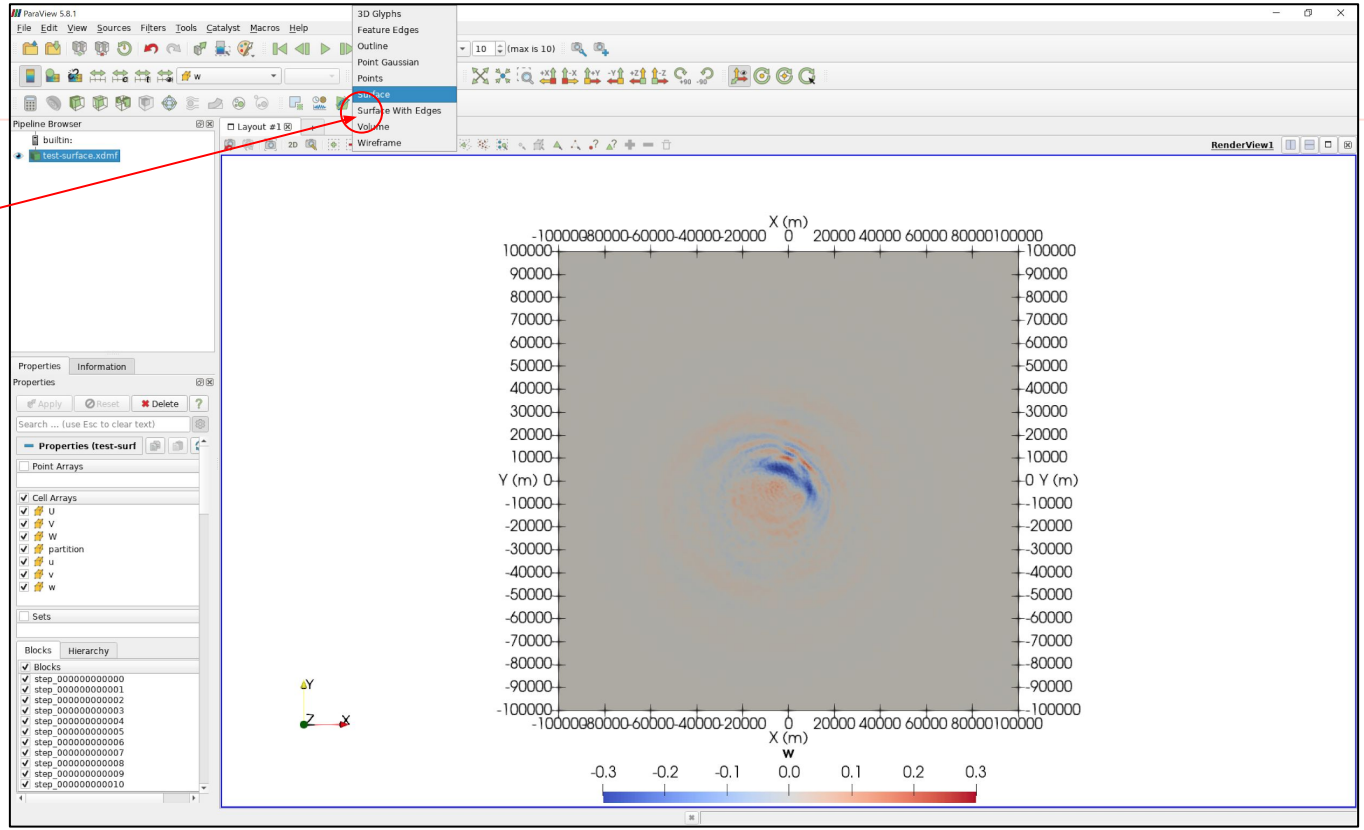


local desktop/laptop, ParaView: rescale to custom data range

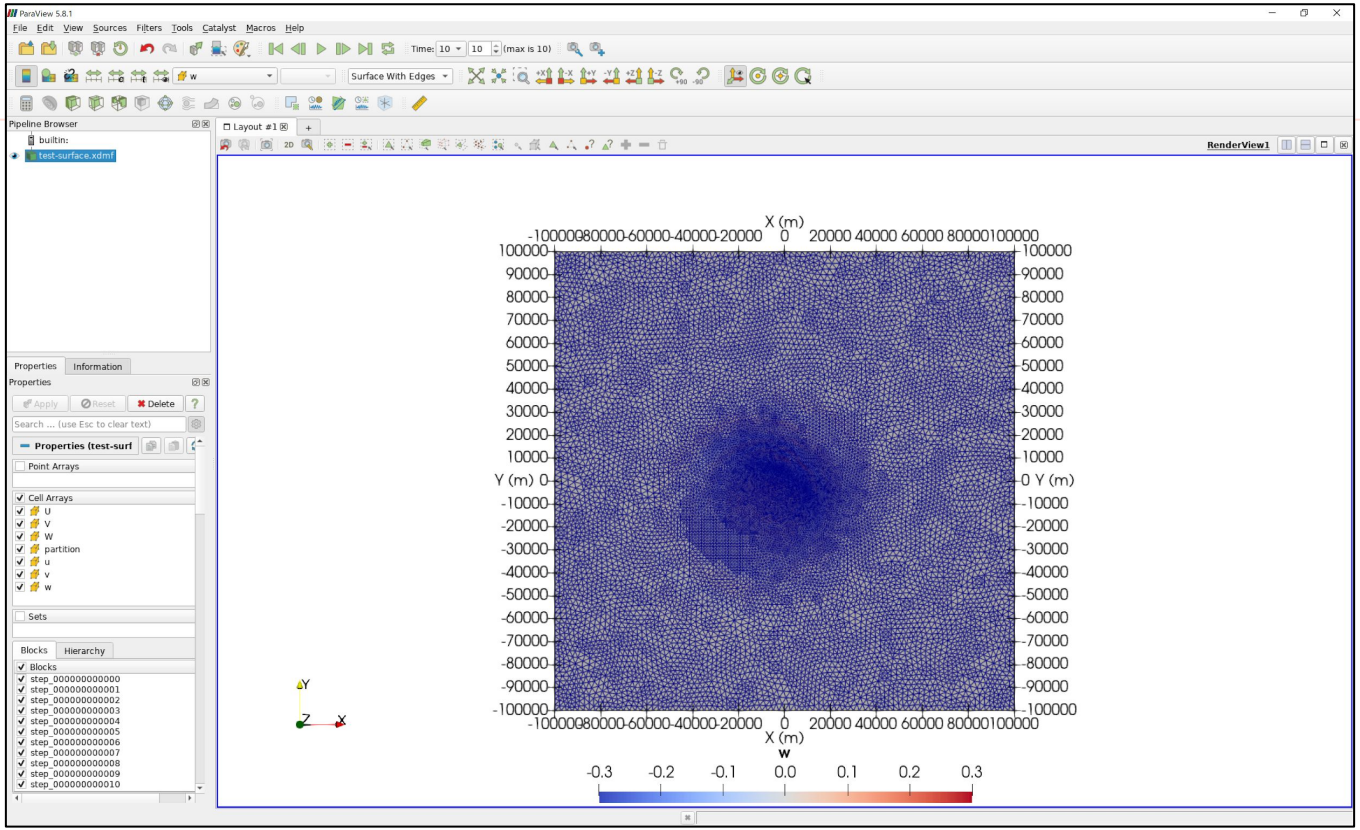


local desktop/laptop, ParaView: show element edges

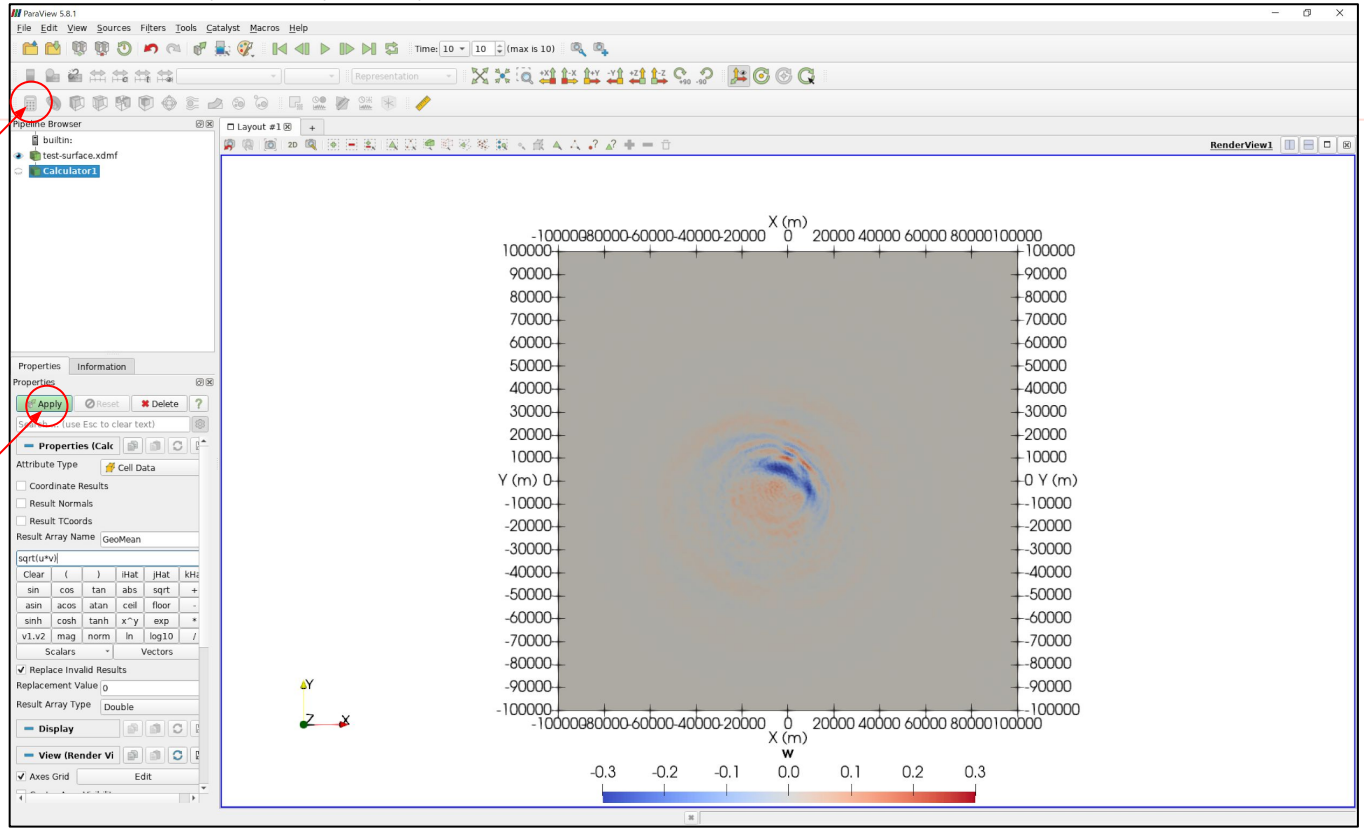
select
Surface With Edges



local desktop/laptop, ParaView: show element edges

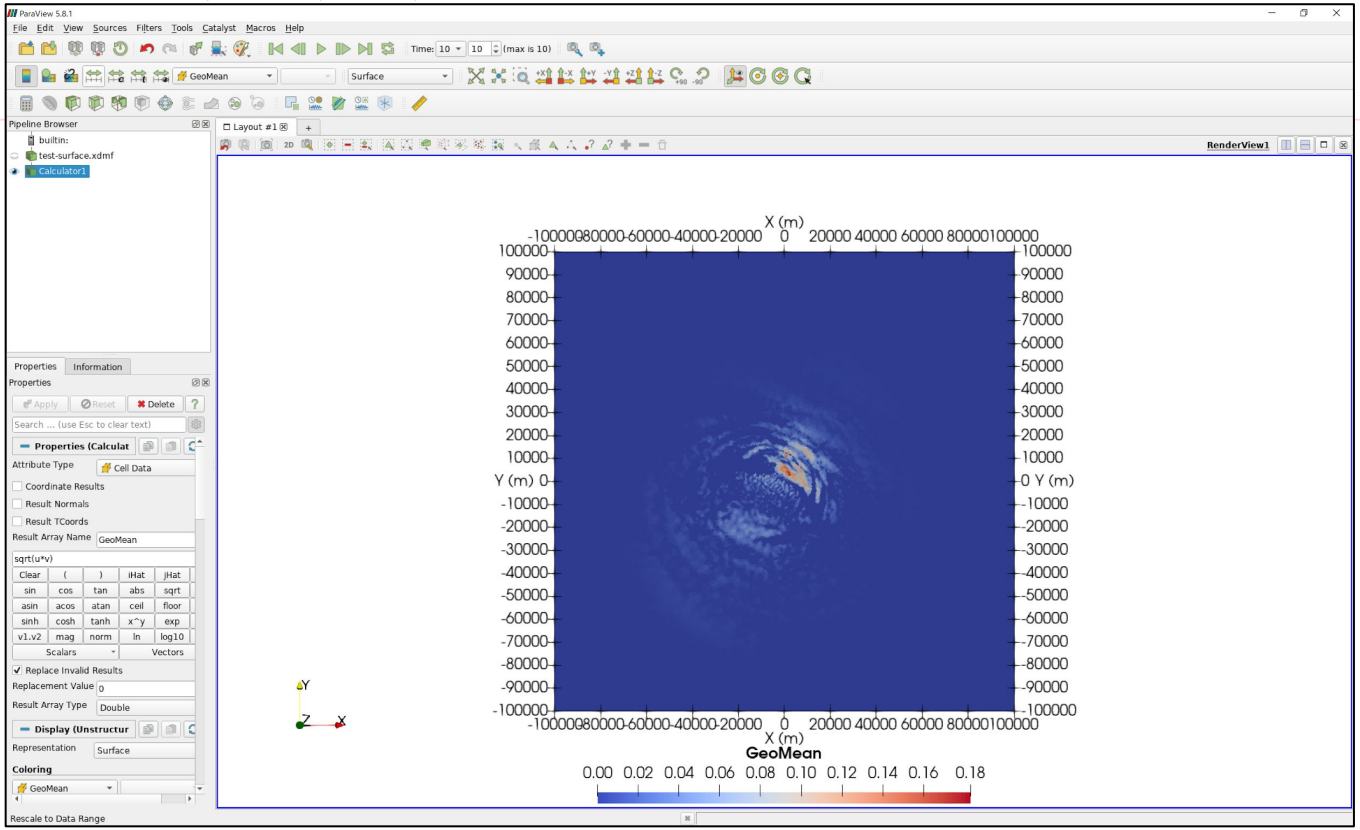


local desktop/laptop, ParaView: calculator

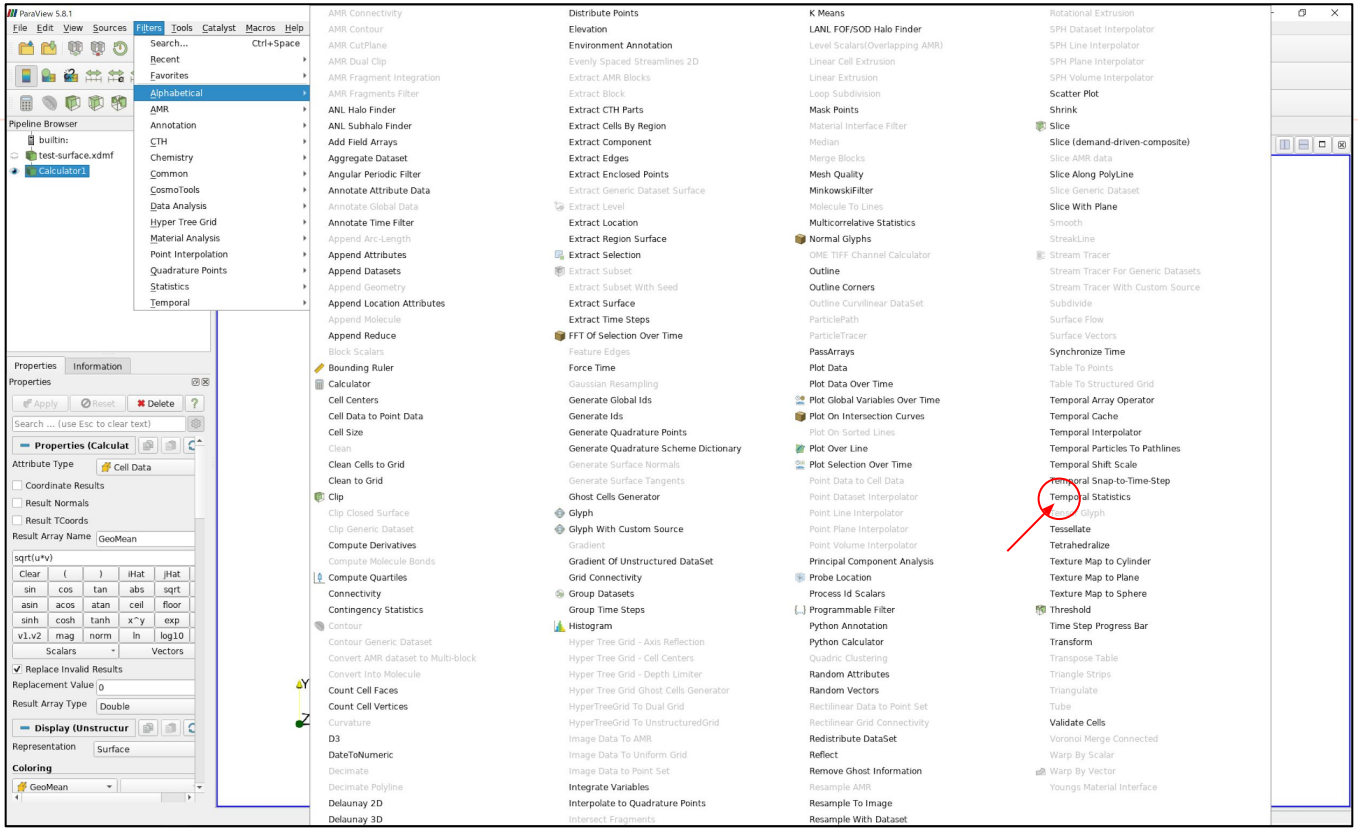


compute velocity geometrical mean

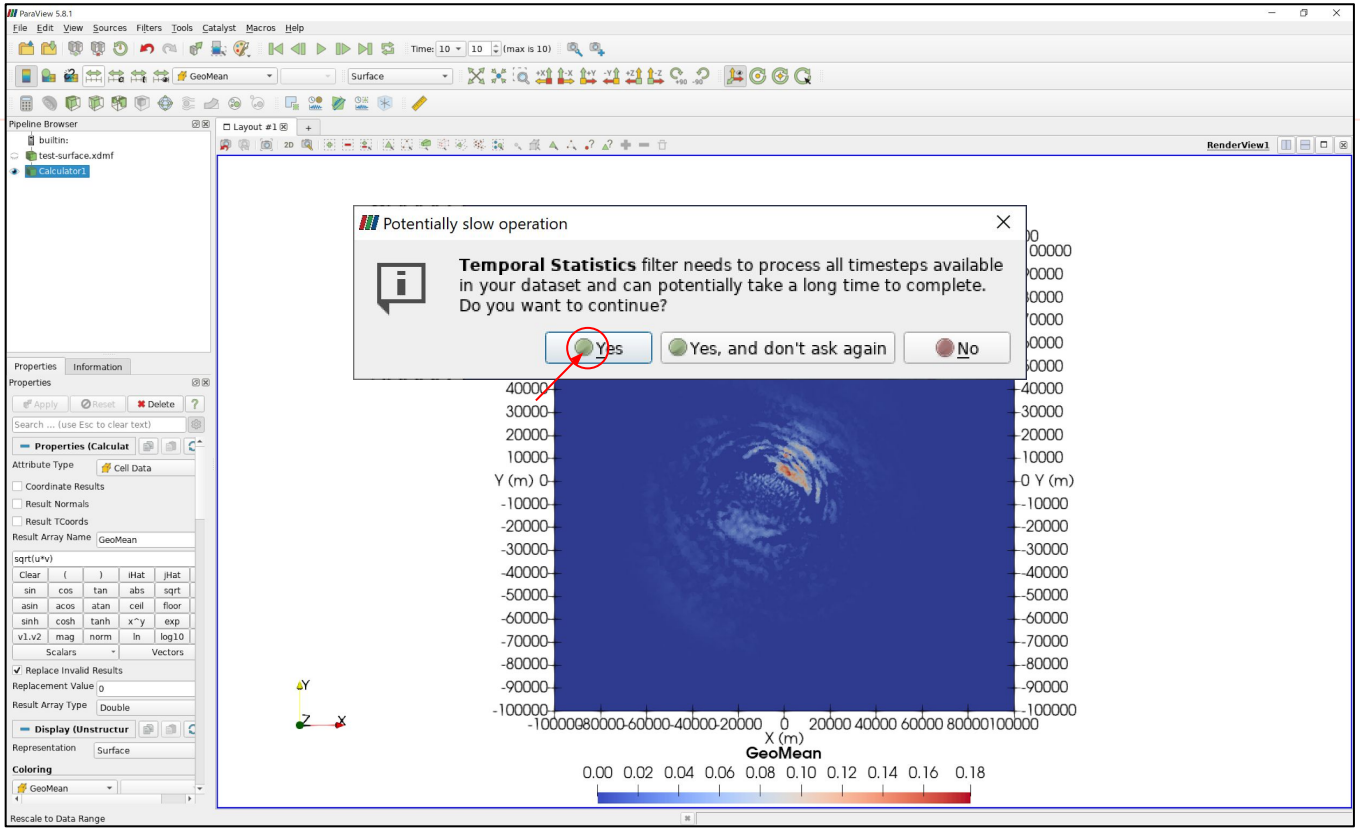
local desktop/laptop, ParaView: calculator



local desktop/laptop, ParaView: TemporalStatistics filter

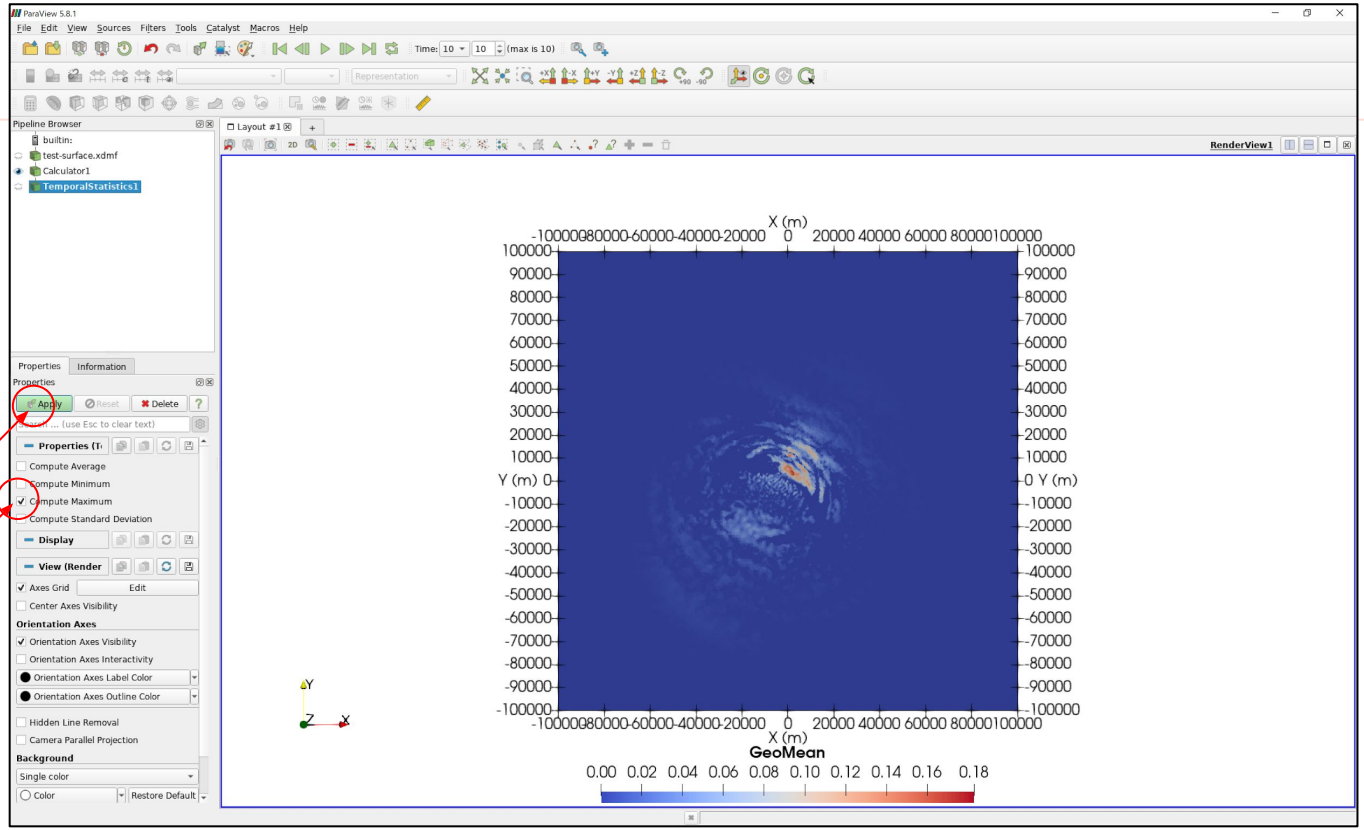


local desktop/laptop, ParaView: TemporalStatistics filter

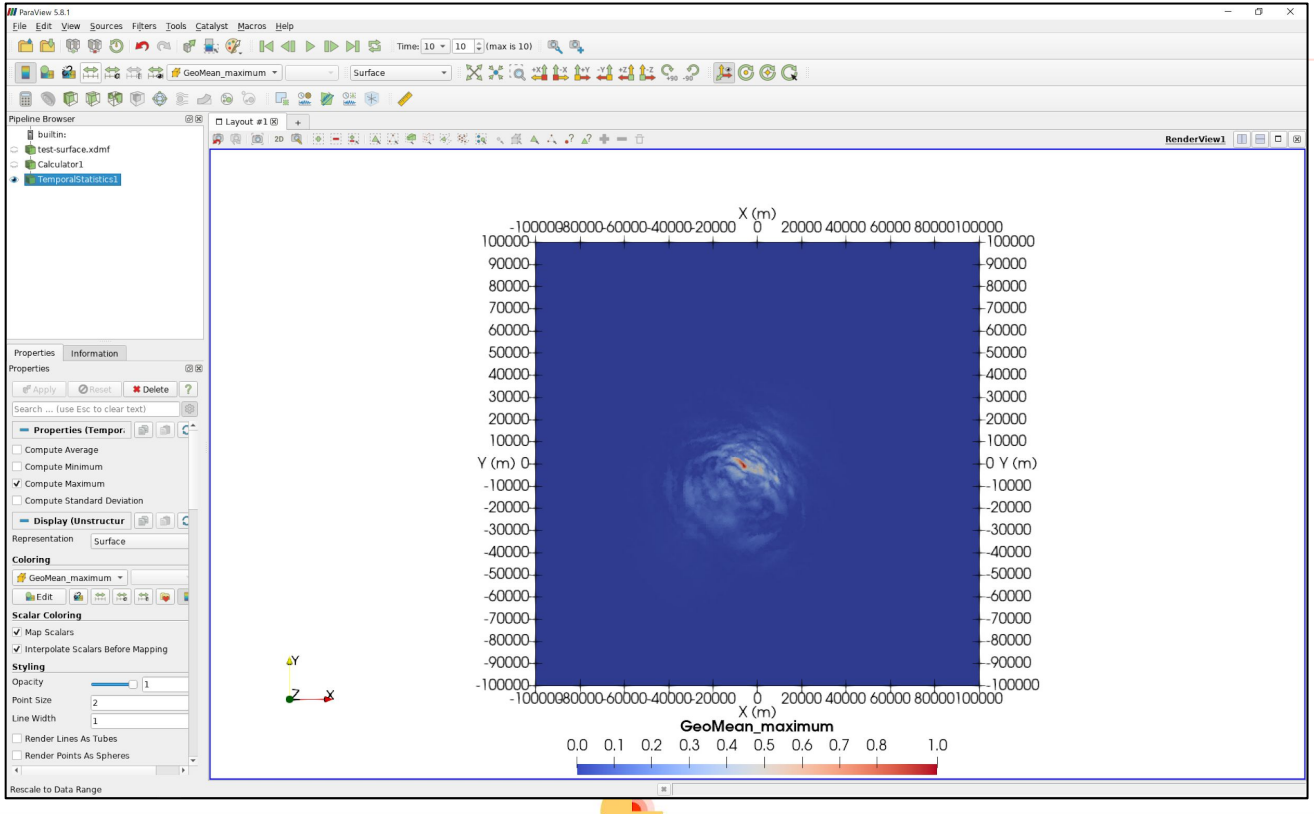


local desktop/laptop, ParaView: TemporalStatistics filter

Compute Maximum value of all timesteps then **Apply**



local desktop/laptop, ParaView: simple shake map maximum value of all timesteps (~peak ground velocity)



Done (for now)!

Let's get back to the main zoom room

Examples 2 and 3 - geometrically simple and complex 3D dynamic rupture modeling

seismic wave propagation non-linearly coupled to frictional failure following linear slip weakening and fast velocity weakening friction laws and to off-fault plastic deformation

Dynamic earthquake modeling

Goal: prediction of time histories of the ground motion
→ will be used for engineering purposes/hazard assessment

- Can model complex phenomena of large earthquakes which may originate from frictional and stress heterogeneities (physics based)
- Predescribed fault geometry
- Frictional constitutive law is needed
  (derived from frictional experiments)

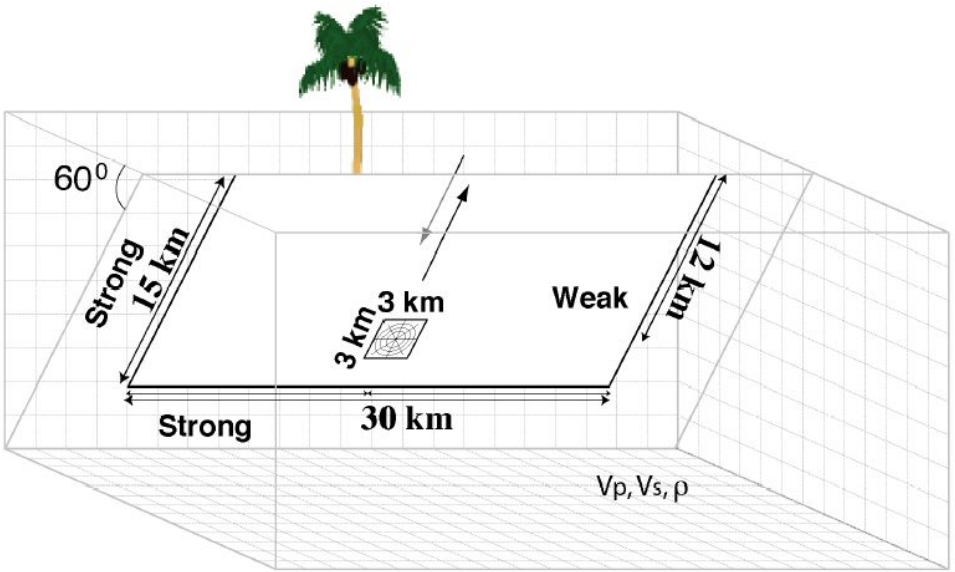
Challenge: dynamic source analysis is computationally expensive

Advantages of dynamic earthquake rupture modeling

- Capture high variability of source dynamics in a physics-based manner
- Spontaneously generate a variability of rupture styles (cracks/pulses)
- Capture dynamic fault interaction, dynamic triggering, slip-reactivation
- Capture supershear propagation, multiple rupture fronts and co-seismic off-fault deformation

Community benchmark “TPV 13”

- SCEC/USGS led effort to verify many codes for dynamic rupture (and seismic cycling) problems
- TPV13: a 3D dipping fault allowing for continuous plastic deformation off the fault



https://strike.scec.org/cvws/tpv12_13docs.html

Description of TPV13

- Spontaneous rupture on a **60-degree dipping normal fault**
- Geometry: 30 by 15 km planar fault interface
- **Homogeneous half-space**: Vp=5716 m/s, Vs=3300 m/s, rho=2700 km/m^3
- Non-associative Drucker-Prager plasticity
- Linear-slip weakening friction (LSW)
- Initial stress conditions are **depth-dependent**
- **Prescribed nucleation** zone: square, 1.5 km size
- **Supershear** initial conditions (ratio of fault strength to initial loading)

parameters.par

```
&equations
MaterialFileName = 'tpv12_13_material.yaml'        ! include material parameter.
Plasticity = 1                                     ! off-fault plasticity, purely elastic = 0
Tv = 0.03                                          ! viscoplastic relaxation time
/

&Boundaries
BC_fs = 1                                          ! free surface boundary condition
BC_dr = 1                                          ! dynamic rupture fault boundary condition
BC_of = 1                                          ! absorbing boundary condition
/
```

for more details look at: https://seissol.readthedocs.io/en/latest/tpv13.html

parameters.par

<https://seissol.readthedocs.io/en/latest/parameter-file.html?highlight=reference%20point#reference-point>

```
&DynamicRupture
FL = 2                                  ! FL=2: use linear-slip weakening friction law
ModelFileName = 'tpv12_13_fault.yaml'


XRef = 0.0                              ! Reference point
YRef = -1.0
ZRef = 0.0
OutputPointType = 5
! Type (0 : no output; 3 : ASCII fault receivers; 4 : paraview file; 5 : both)
/
```

```
RF_output_on = 0                    ! Rupture front ascii output
magnitude_output_on = 1             ! Moment magnitude output
energy_rate_output_on = 0           ! Moment rate output
OutputPointType = 4
/

&Elementwise                        ! Paraview fault output
printIntervalCriterion = 2          ! 1=iteration, 2=time
printtimeinterval_sec = 1.0         ! Time interval at which output will be written
OutputMask = 1 1 1 0 1 1 1 1 1 1 1  ! output 1/ yes, 0/ no - position: (further details next slide)
refinement_strategy = 2             ! triangles are split into 4 triangles
refinement = 1                      ! subdivides each triangle
/
```

OutputMask

https://seissol.readthedocs.io/en/latest/fault-output.html

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. **Tso,Tdo,Pno**: total stress, including initial stress
6. **Sls and Sld**: fault slip in strike and dip direction
7. **Vr**: rupture speed, computed from the spatial derivatives of the rupture time
8. **ASl**: accumulated slip
9. **PSR**: peak slip rate
10. **RT**: rupture time
11. **DS**: only with linear slip-weakening friction, time at which ASl>D_c
12. **P_f** and **Tmp**: pore fluid pressure and temperature for thermal pressurization

parameters.par

```
&MeshNml
MeshFile = 'tpv13_training.puml.h5'        ! Name of mesh file in HDF5 format
meshgenerator = 'PUML'                     ! Name of mesh generator
/

&Discretization
CFL = 0.5                                  ! The Courant-Friedrichs-Levy (CFL) Stability Criterion (<=1.0)
ClusteredLTS=2
! 1 for Global time stepping, 2,3,5,... Local time stepping (advised value 2)
! ClusteredLTS defines the multi-rate for the time steps of the clusters 2 for Local time stepping
/
```

```
&Output
FaultOutputFlag = 1                    ! DR output (add this line only if DR is active)
OutputFile = 'output/tpv13'
Format = 10                            ! Format (10=no output, 6=hdf5 output)
!          |stress     |vel
iOutputMask = 1 1 1 1 1 1 1 1 1
TimeInterval =  5.                     ! Index of printed info at time
refinement = 1

! Free surface output
SurfaceOutput = 1
SurfaceOutputRefinement = 1
SurfaceOutputInterval = 5.0

printIntervalCriterion = 2        ! Criterion for index of printed info: 1=timesteps,2=time,3=timesteps+time

pickdt = 0.005                         ! Pickpoint Sampling
pickDtType = 1                         ! Pickpoint Type
nRecordPoints = 0                      ! number of Record points which are read from file
RFileName = 'tpv12_13_receivers.dat'      ! Record Points in extra file

&AbortCriteria
EndTime = 8.0
/
```

tpv13_material.yaml

```
!Switch
[rho, mu, lambda, plastCo, bulkFriction]: !ConstantMap
  map:
    rho:                    2700
    mu:               2.9403e+010
    lambda:            2.941e+010
    plastCo:             1.0e+06
    bulkFriction:           0.60
[s_xx, s_yy, s_zz, s_xy, s_yz, s_xz]: !Include tpv13_initial_stress.yaml
```

tpv13_initial_stress.yaml

```
!Switch
[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:
    s_xx: return -s_max_minus_Pf * depth;
    s_yy: return -s_max_minus_Pf * depth;
    s_zz: return -s_max_minus_Pf * depth;
```

< https://easyinit.readthedocs.io/en/latest/maps.html#functionmap >

< https://easyinit.readthedocs.io/en/latest/filters.html >

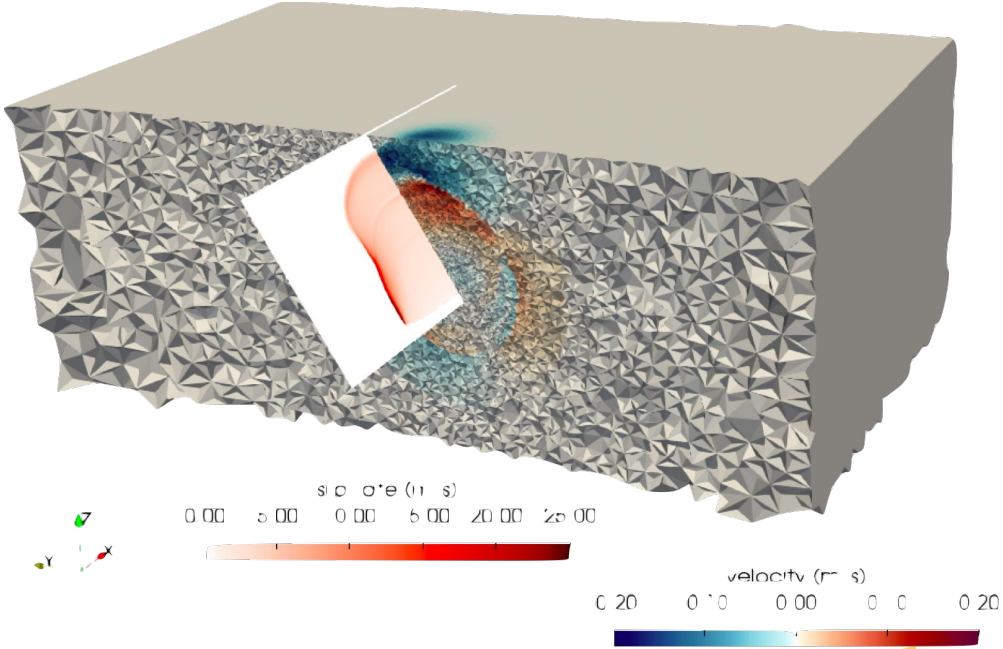
Parameter	Value
above 11951.15 m	
σ_1	26460 Pa/m $\times$ H
σ_3	15624.3 Pa/m $\times$ H
σ_2	$(\sigma_1 + \sigma_3)/2$
P_f	$1000\mathrm{kg/m^3} \times 9.8\mathrm{m/s^2} \times H$
below 11951.15 m	
$\sigma_1, \sigma_2, \sigma_3$	$2700\mathrm{kg/m^3} \times 9.8\mathrm{m/s^2} \times H$

<https://strike.scec.org/cvws/tpv12_13docs.html>

tpv13_fault.yaml

```yaml
!Switch
[s_xx, s_yy, s_zz, s_xy, s_yz, s_xz]: !Include tpv13_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:      0.54
        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
```

Hands on! (30 minutes)

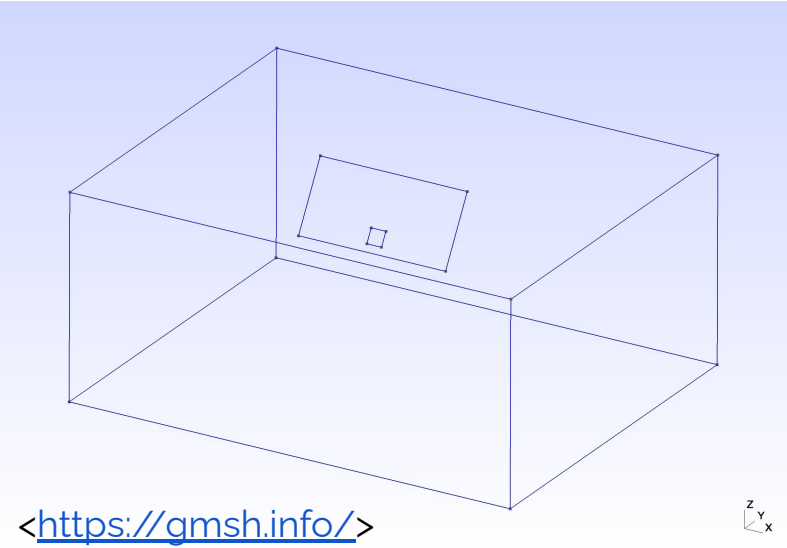


Task 1

- Look at the tpv13_training.geo file and try to understand the meshing instructions
- Create a finer resolved mesh by adjusting the h_fault parameter
  (nucleation zone: 300 m, remaining fault:600 m high-order element edge length)

tpv13_training .geo file: geometry

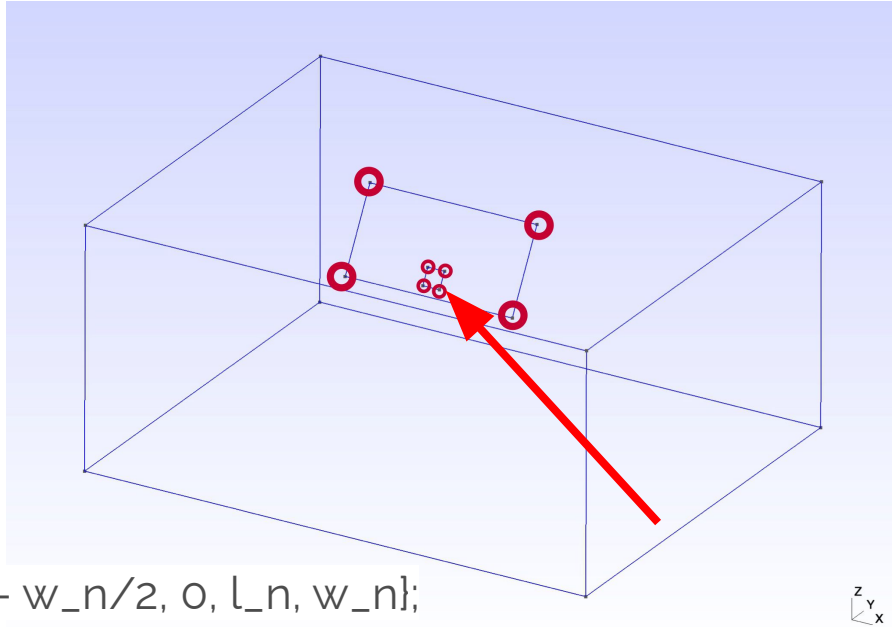
```
DefineConstant[ h_domain = {5e3, Min 0, Max 500e3, Name "Mesh spacing in domain" } ];
DefineConstant[ h_fault = {1000.0, Min 0, Max 30e3, Name "Mesh spacing on fault" } ];
DefineConstant[ h_nucl = {500.0, Min 0, Max 3e3, Name "Mesh spacing on nucleation patch" } ];
DefineConstant[ dip = {60.0, Min 0, Max 90, Name "Fault dip" } ];

SetFactory("OpenCASCADE");

l_f = 30e3;                 //Fault length
w_f = 15e3;                 //Fault width
dip_rad = dip*Pi/180.;      //Fault dip
```



<https://gmsh.info/>

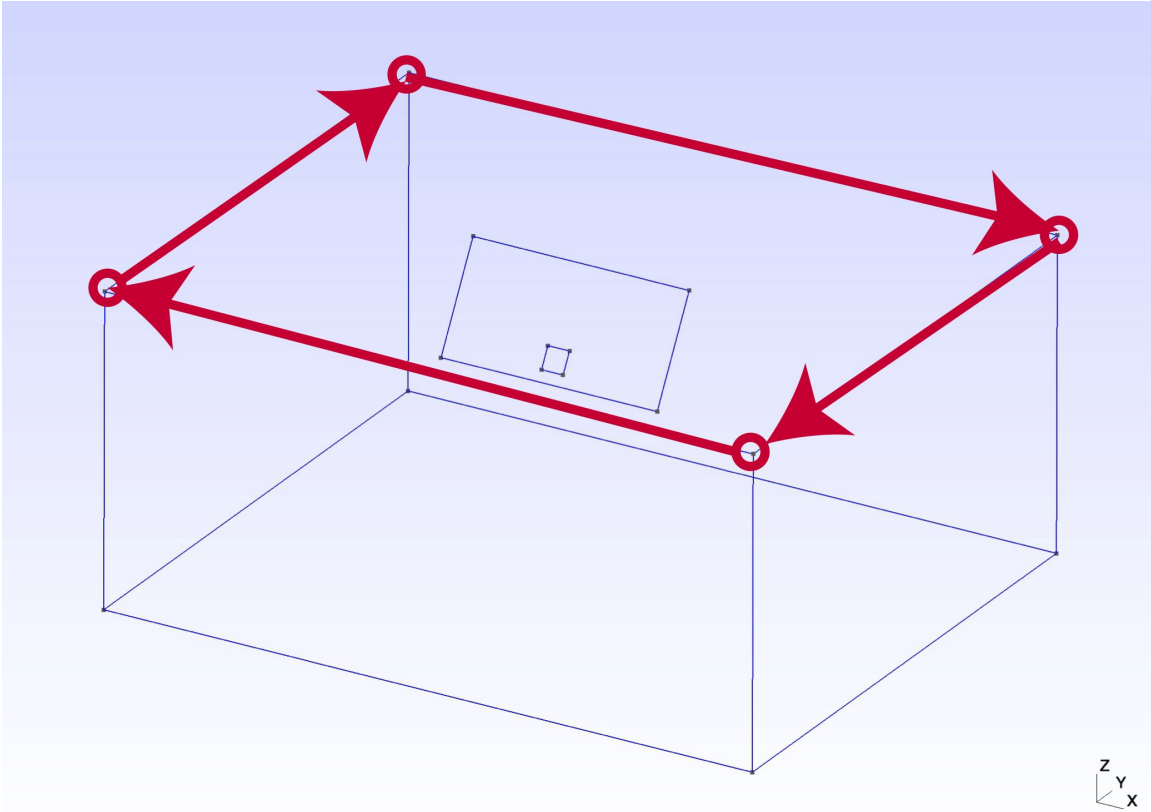
GMSH .geo file

```
// Square nucleation patch in local coordinates
x_n = 0e3;
z_n = -12e3;
l_n = 3e3;
w_n = 3e3;

// Create nucleation patch
nucl = news; Rectangle(nucl) = {x_n - l_n/2, z_n - w_n/2, 0, l_n, w_n};
```



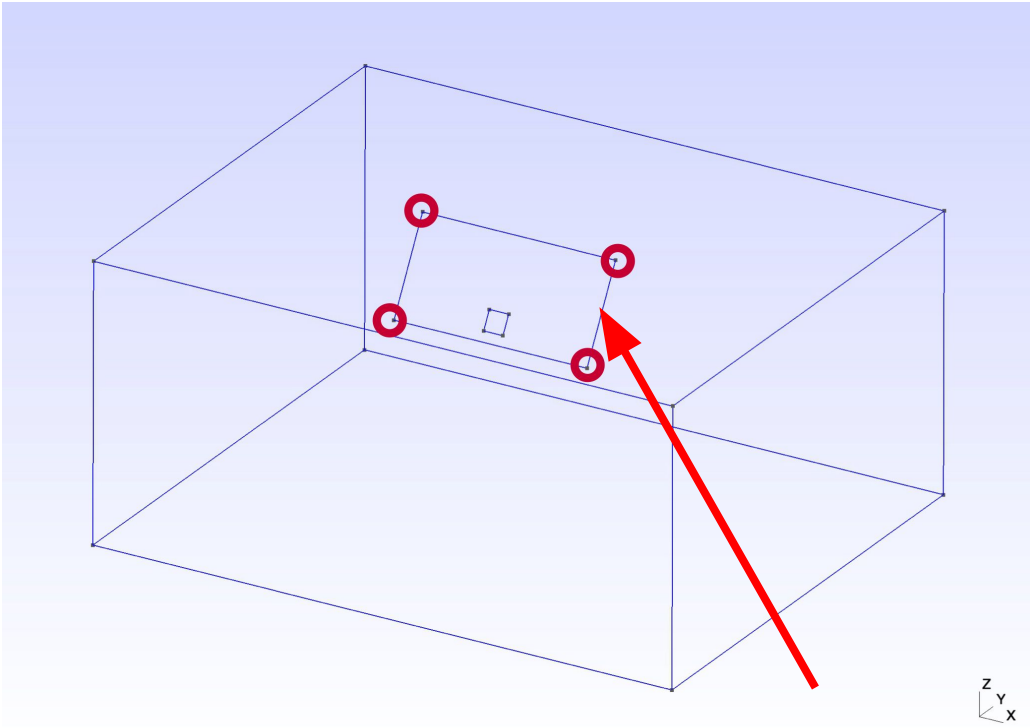
GMSH .geo file

```
// Domain size: 3-4 times of the fault's dimension
X0 = -45e3;
X1 = -X0;
Y0 = -36e3;
Y1 = -Y0;
Z0 = -42e3;

// Create the domain
domain = newv; Box(domain) = {X0, Y0, Z0, X1-X0, Y1-Y0, -Z0};
```



GMSH .geo file

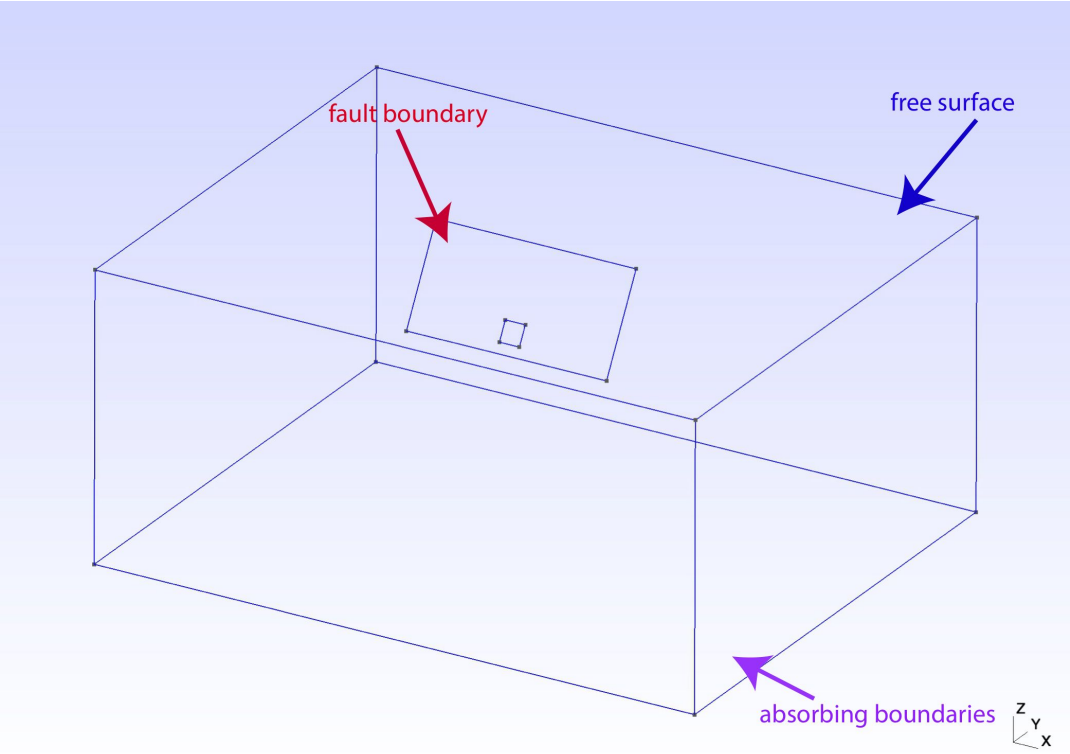
```
//create the fault
fault = news; Rectangle(fault) = {-l_f/2,
-w_f, 0, l_f, w_f};

// Rotate the fault
Rotate{ {1, 0, 0}, {0, 0, 0}, dip_rad } {
Surface{fault, nucl}; }
```



GMSH .geo file

https://seissol.readthedocs.io/en/latest/gmsh.html#boundary-conditions

```
// Define boundary conditions
// free surface
Physical Surface(1) = {top[]};
// fault boundaries
Physical Surface(3) = {fault_final[]};
// absorbing boundaries
Physical Surface(5) = {other[]};
```



Task 2

- Understand the “BooleanFragments” operation in the .geo file
- Run the 3D DR earthquake scenario
- Adjust the OMP_NUM_THREADS variable to match the number of cores on your PC or laptop
- Visualize the fault output

Boolean operations

- Can be applied on curves, surfaces and volumes
- We use it to **intersect the domain with the fault**

General syntax:

```
BooleanFragments { boolean-list } { boolean-list }
```

Computes all the fragments resulting from the **intersection of the entities** in the object and in the tool, **making all interfaces conformal.**

When applied to entities of different dimensions, **the lower dimensional entities will be automatically embedded in the higher dimensional entities** if they are not on their boundary

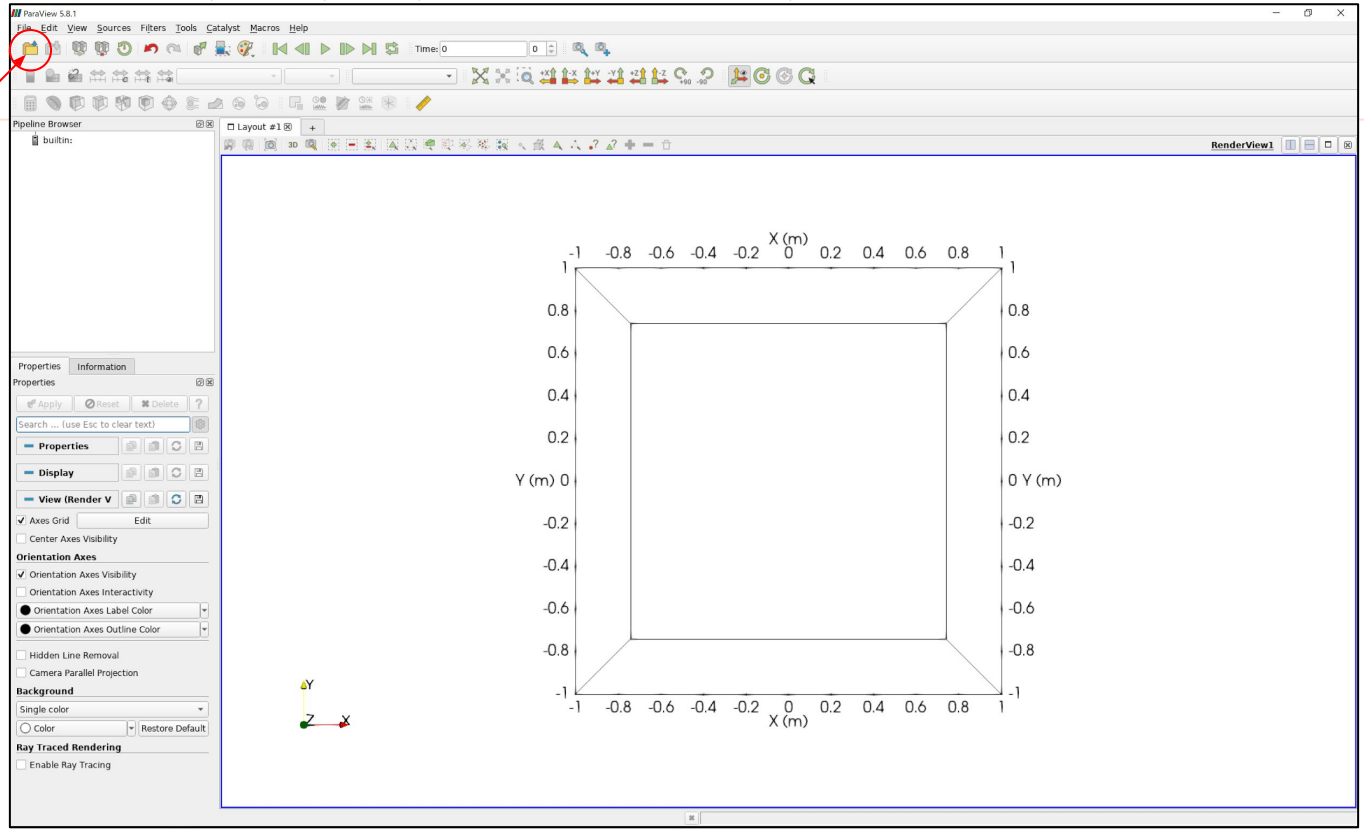
Task 3

- Run SeisSol once with rate-2 local time-stepping (LTS) and once with global time-stepping (GTS)
- Compute the speed-up and compare it to the theoretical speed-up due to LTS

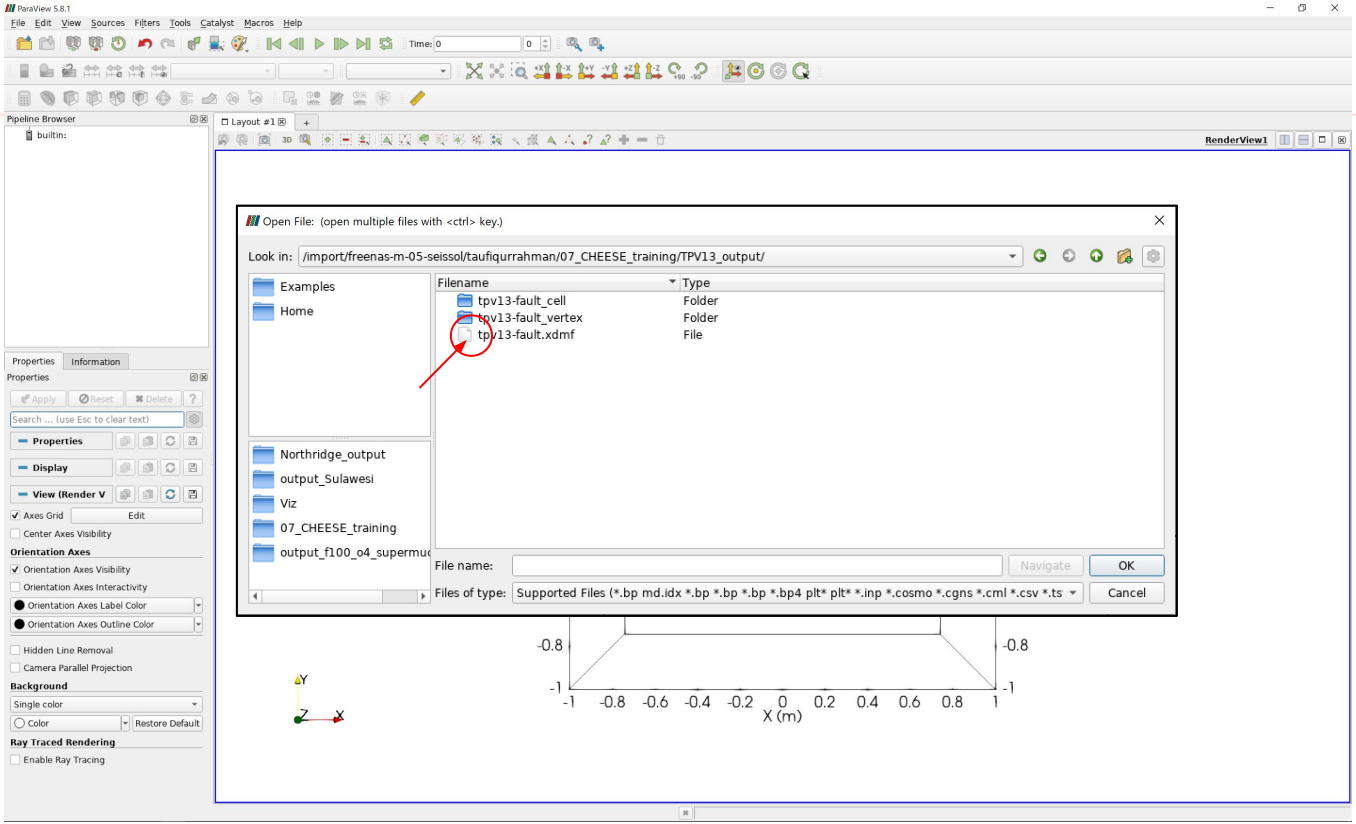
Task 4

- Turn off plasticity
- Analyse differences between the 2 models

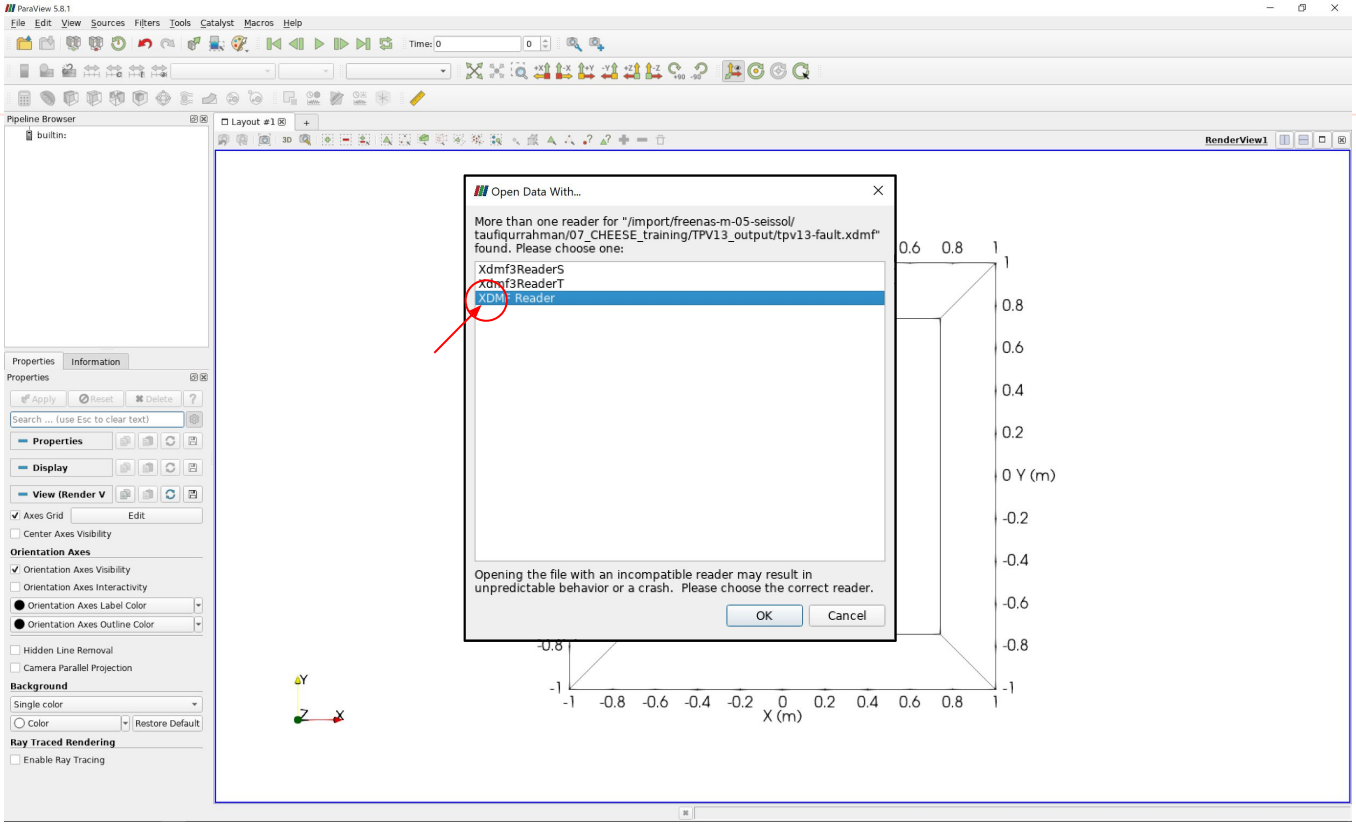
local desktop/laptop, Paraview: open file



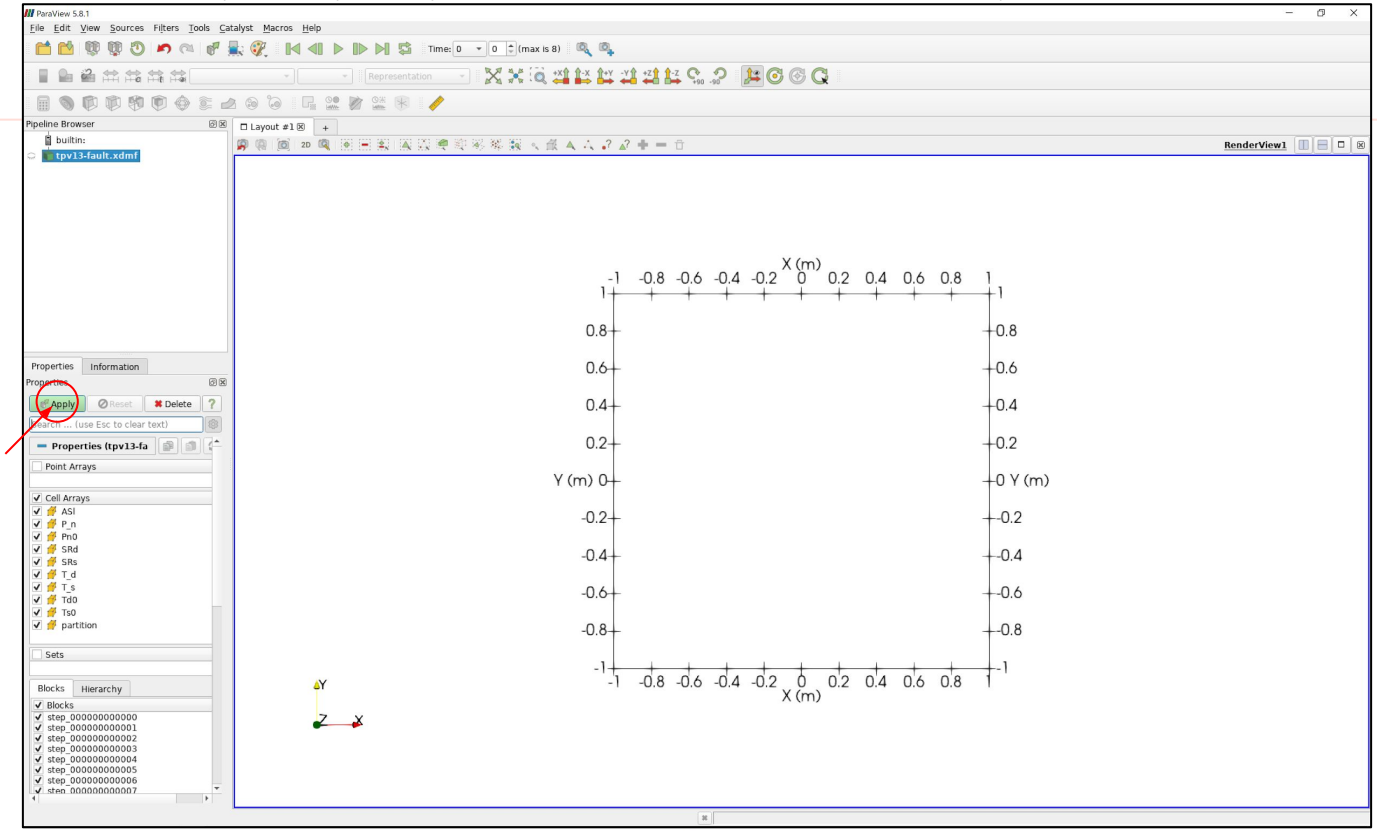
local desktop/laptop, Paraview: load output file



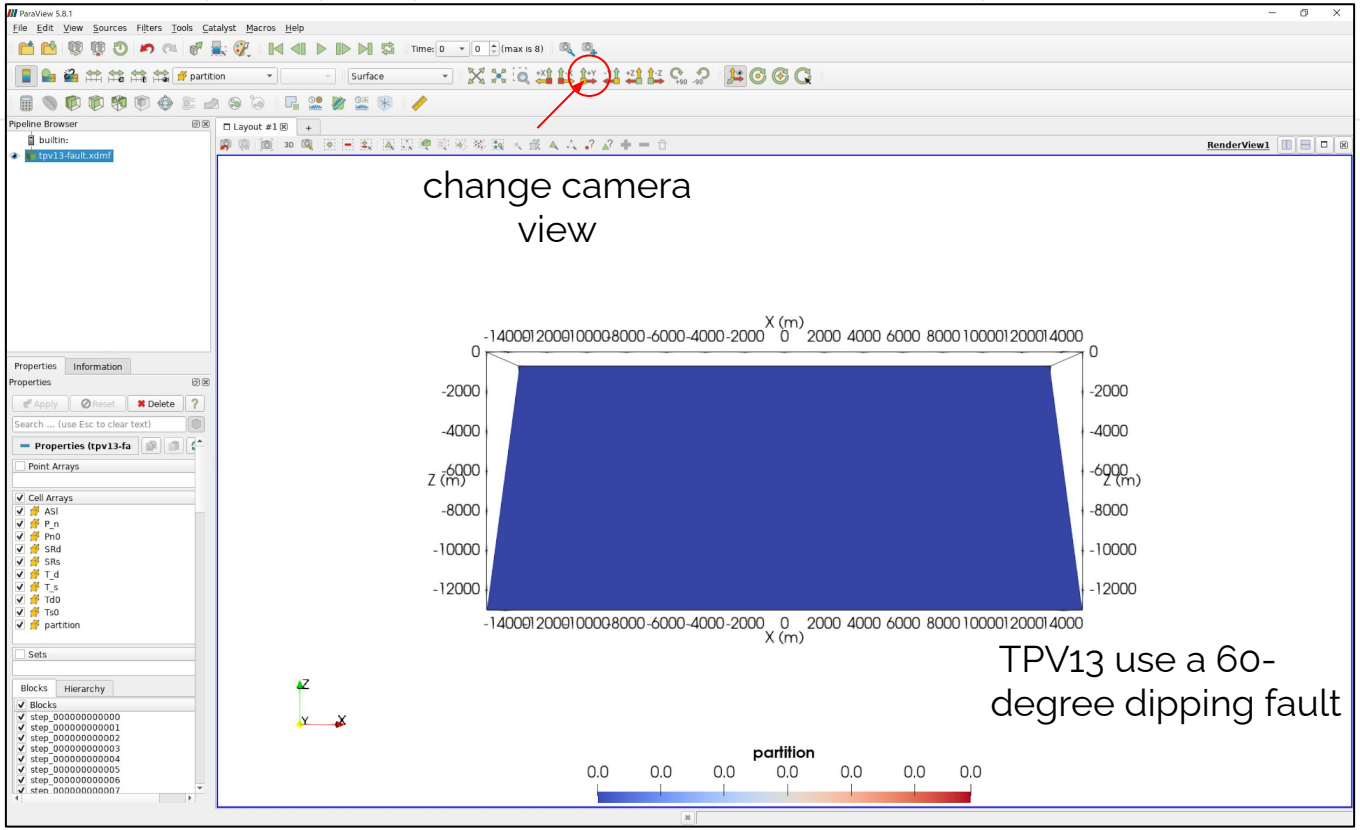
local desktop/laptop, Paraview: load output file



local desktop/laptop, Paraview: load output file

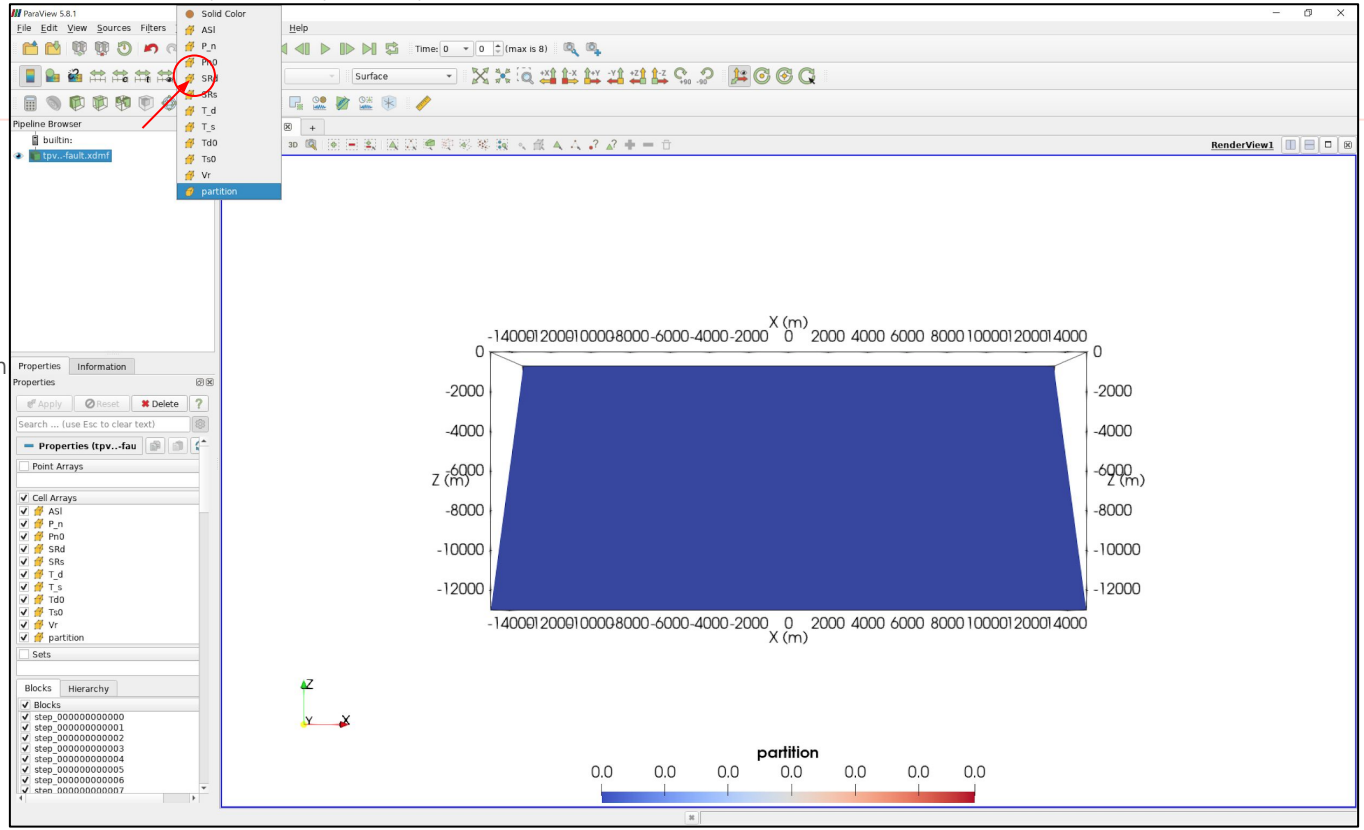


local desktop/laptop, Paraview: load output file



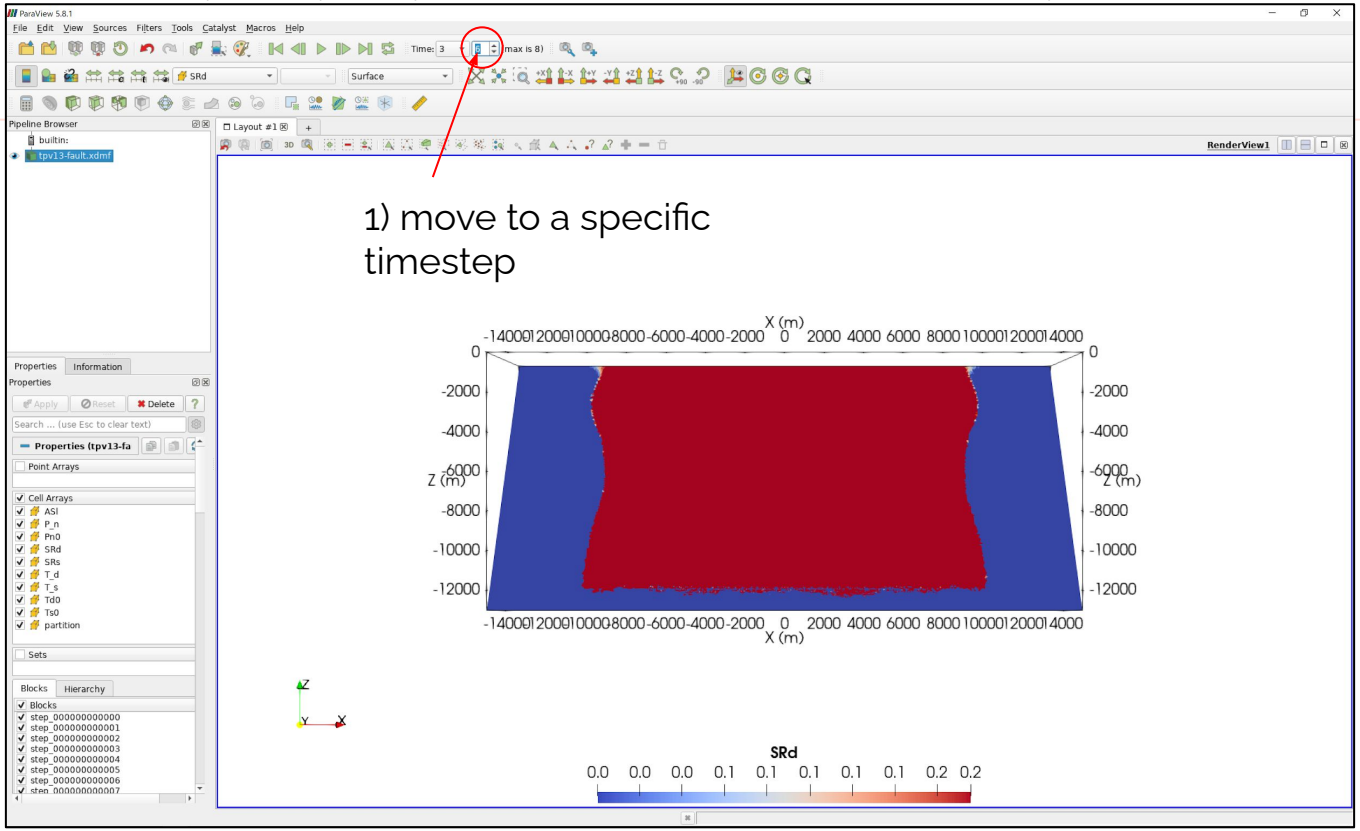
local desktop/laptop, Paraview: select a fault attribute

ASl: absolute slip
P_n: transient normal stress
Pno: total stress normal direction
SRs and SRd: slip rates in strike and dip direction
T_s, T_d: transient shear stress in strike and dip direction
Tso,Tdo: total stress in strike and dip direction
Sls and Sld: slip in strike and dip direction
Vr: rupture velocity, computed from the spatial derivatives of the rupture time

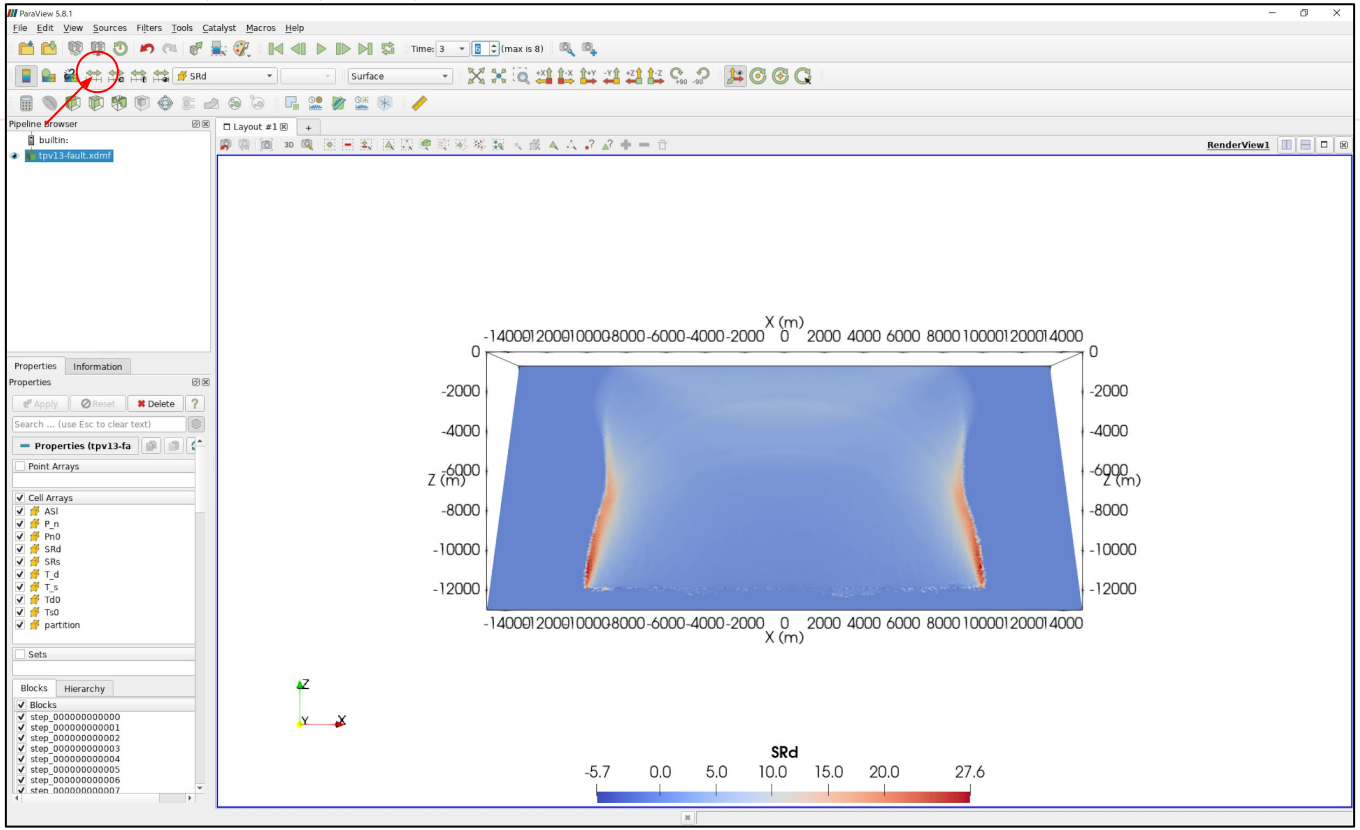


local desktop/laptop, Paraview: choose a specific time step

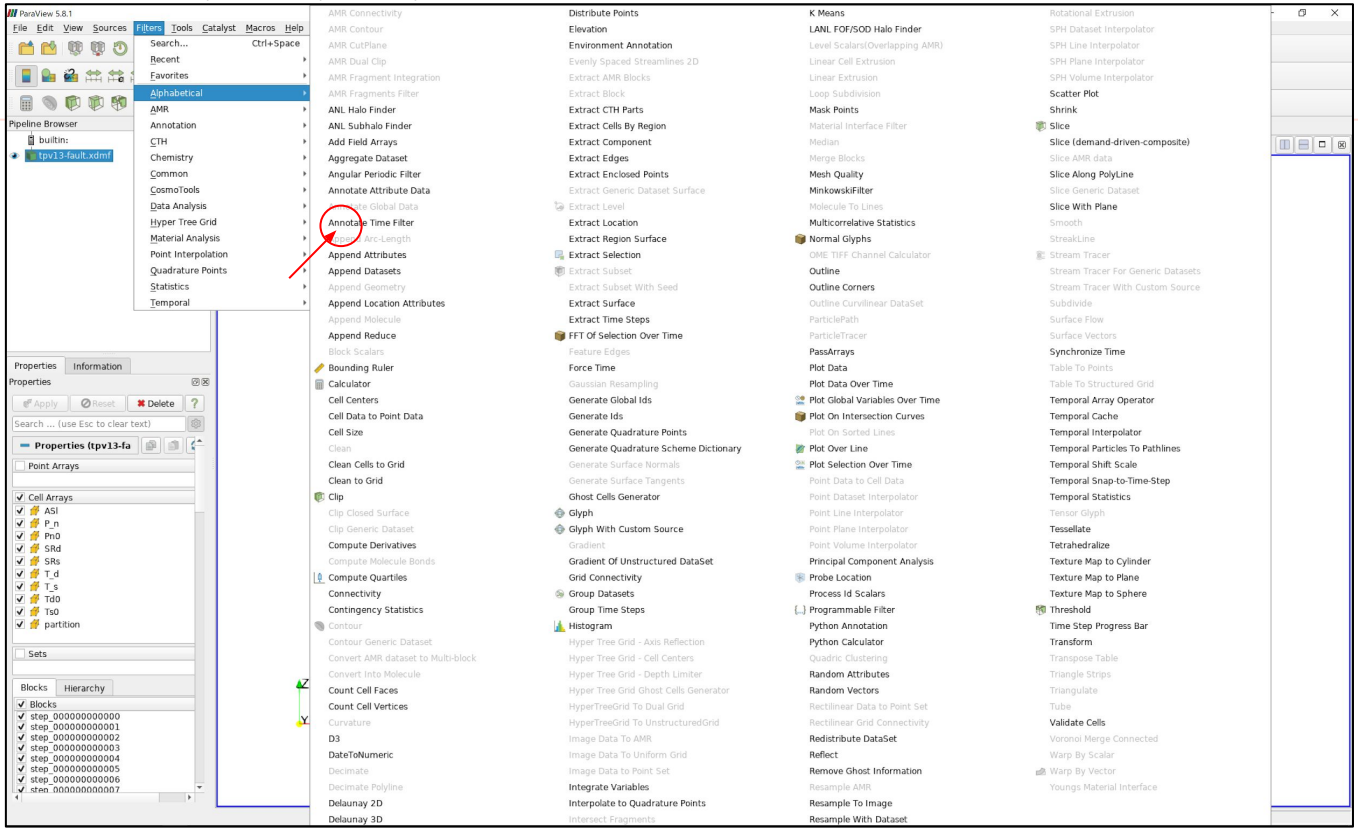
2) Data scale is too small, need rescale to data range



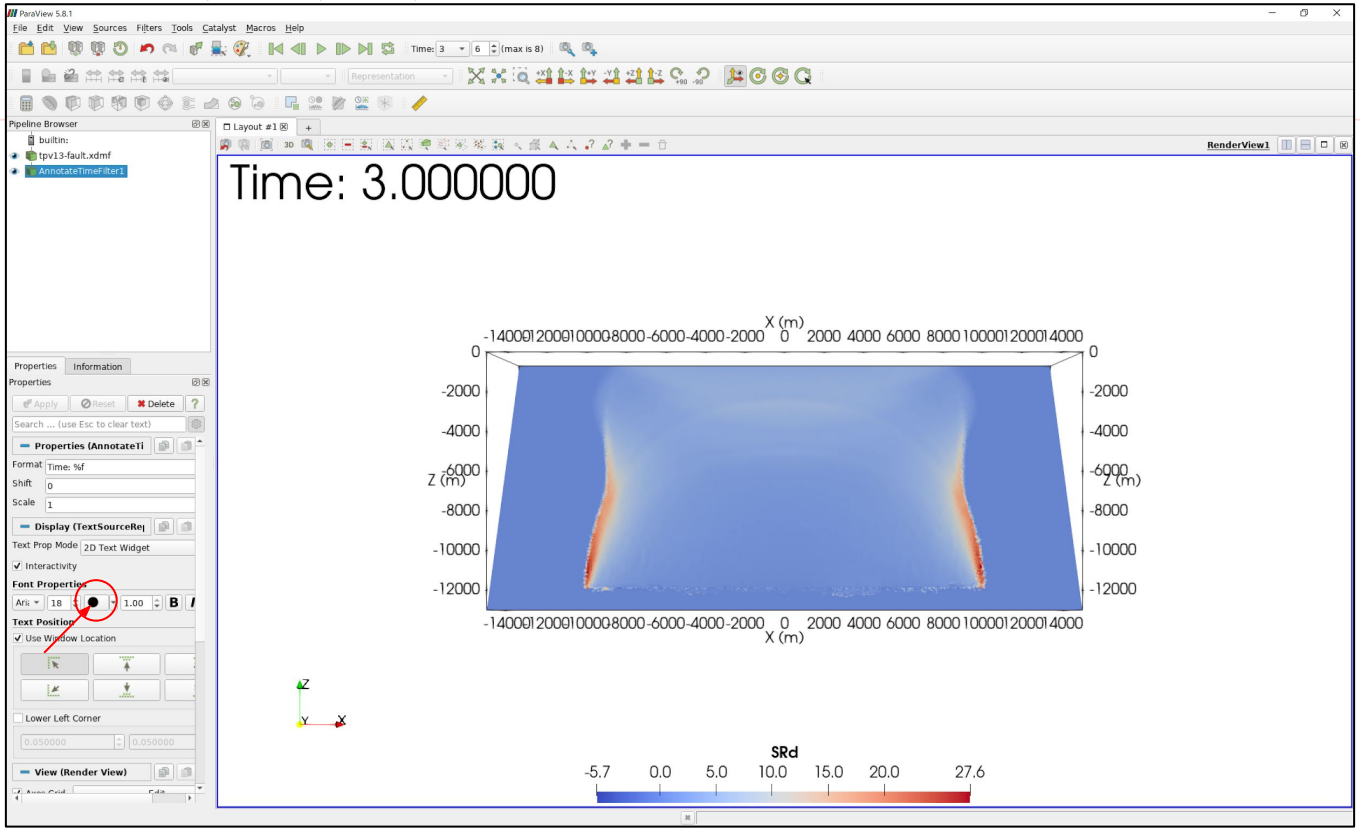
local desktop/laptop, Paraview: rescale to data range



local desktop/laptop, Paraview: add time annotation



local desktop/laptop, Paraview: add time annotation

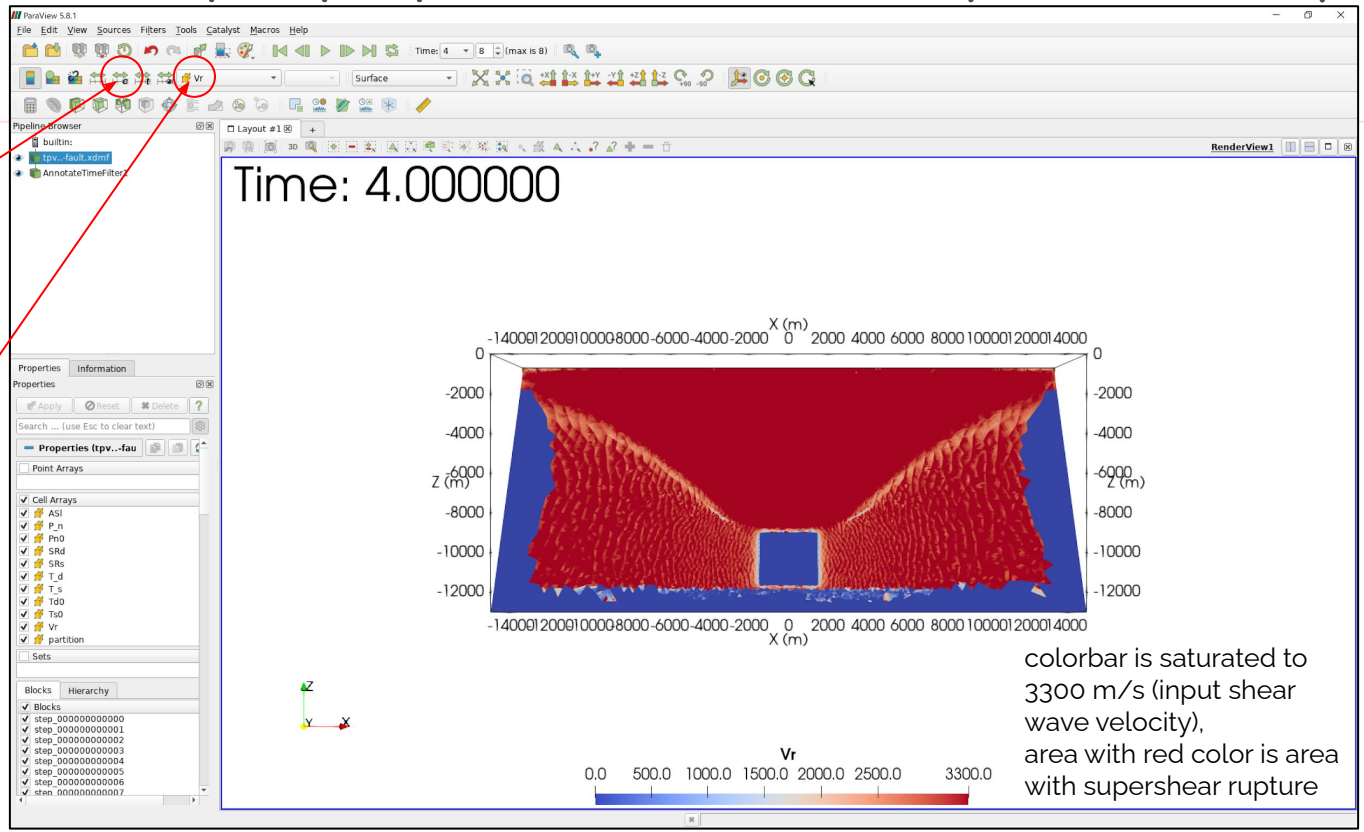


change time annotation font color

local desktop/laptop, Paraview: show rupture velocity

2) manually adjust data range from 0-3300 m/s

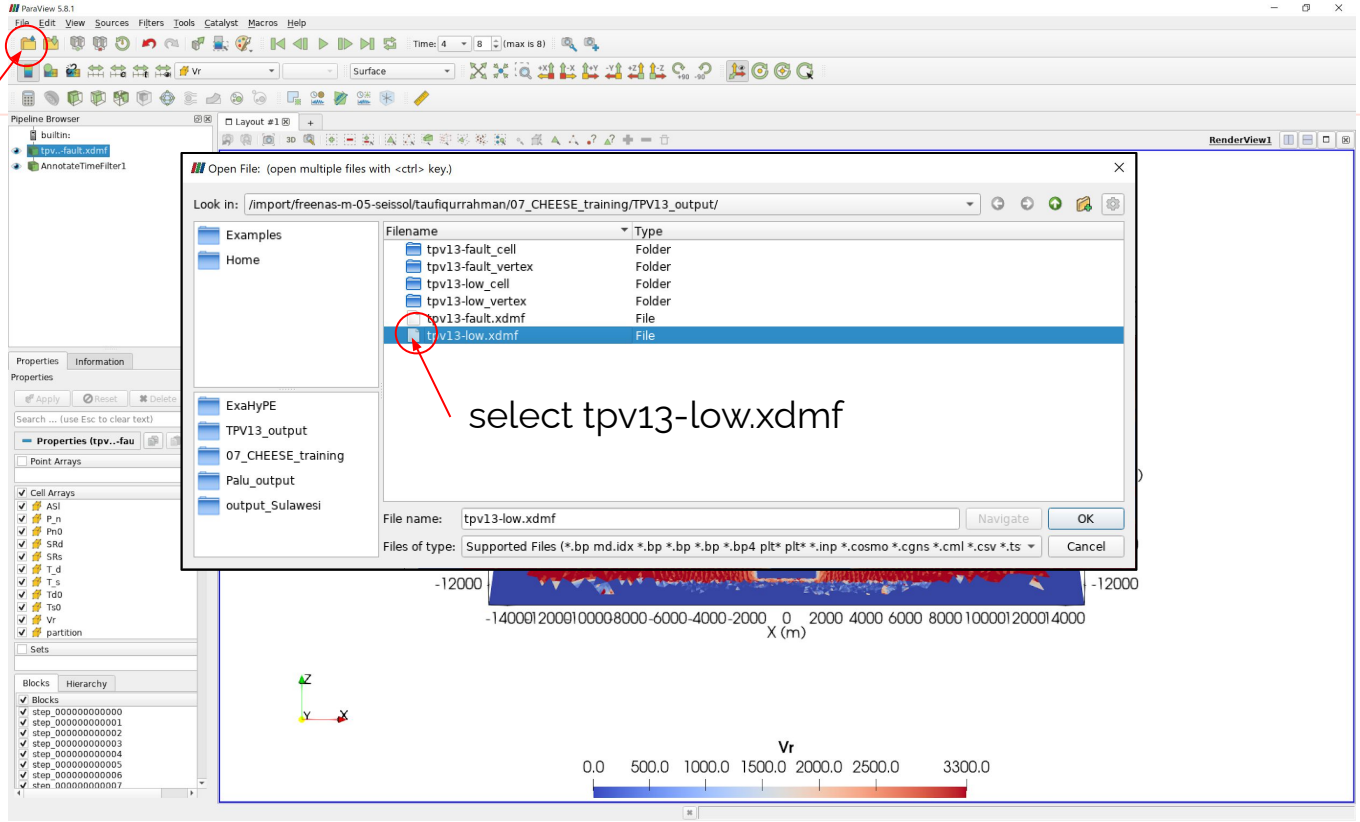
1) select attribute **Vr**



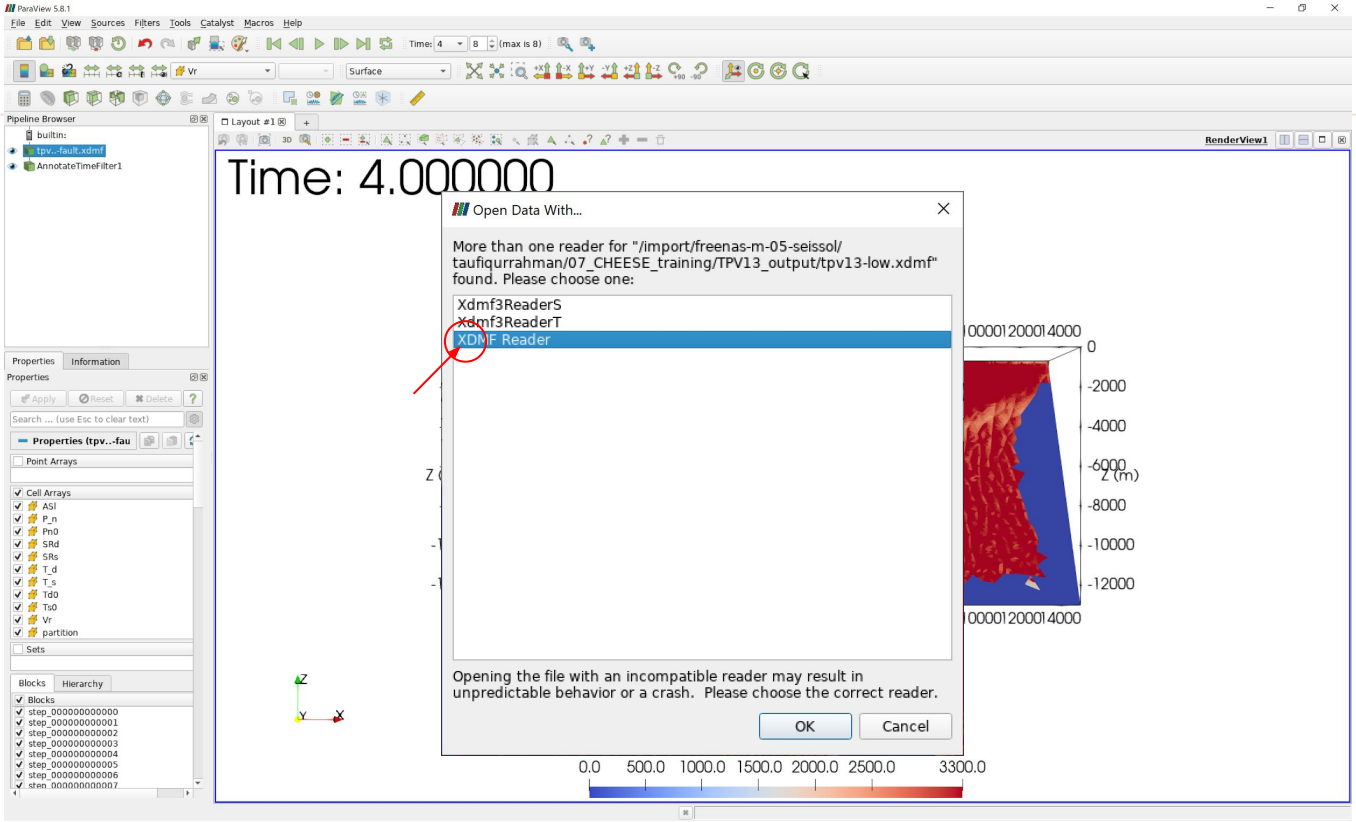
colorbar is saturated to 3300 m/s (input shear wave velocity),
area with red color is area with supershear rupture

local desktop/laptop, Paraview: add plastic strain output

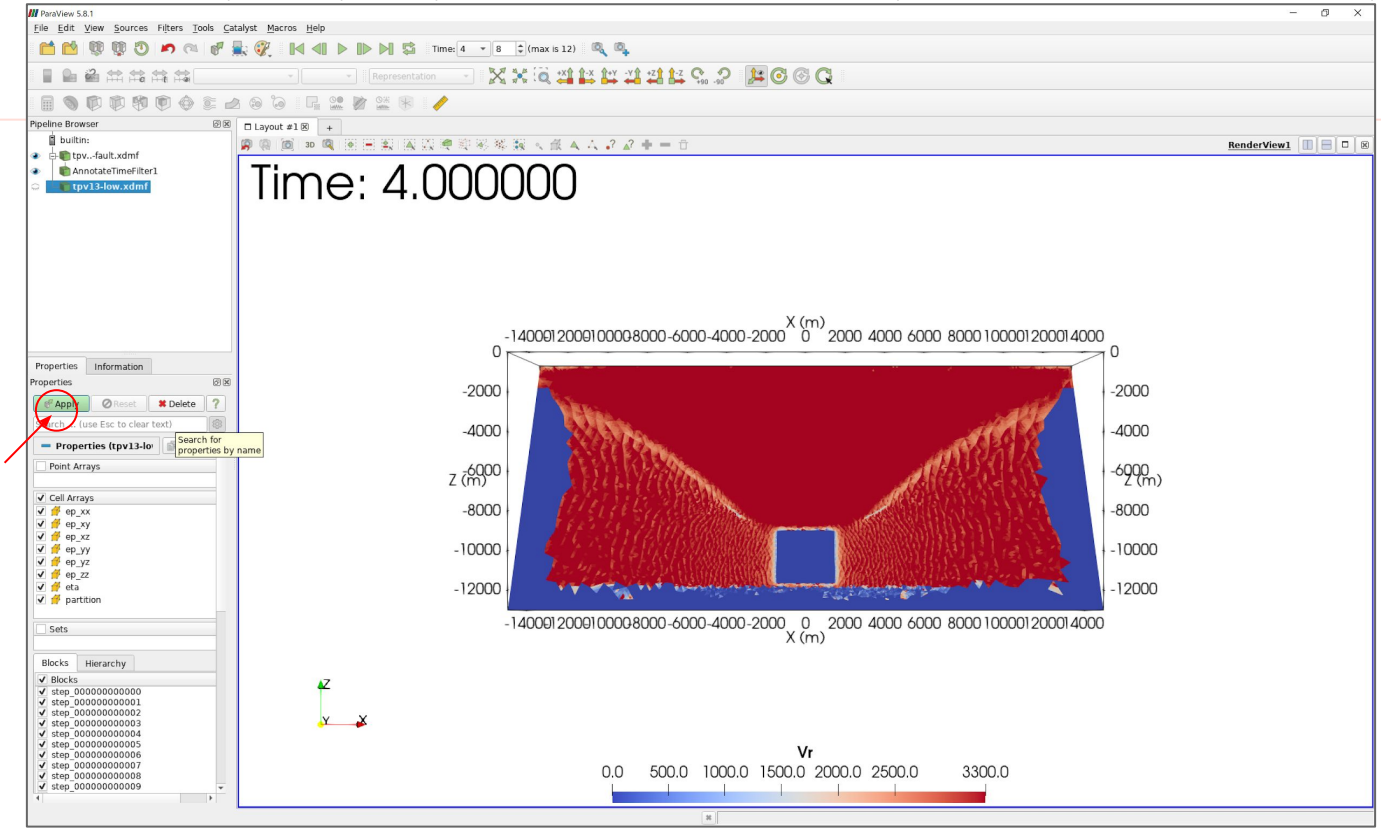
open file



local desktop/laptop, Paraview: add plastic strain output



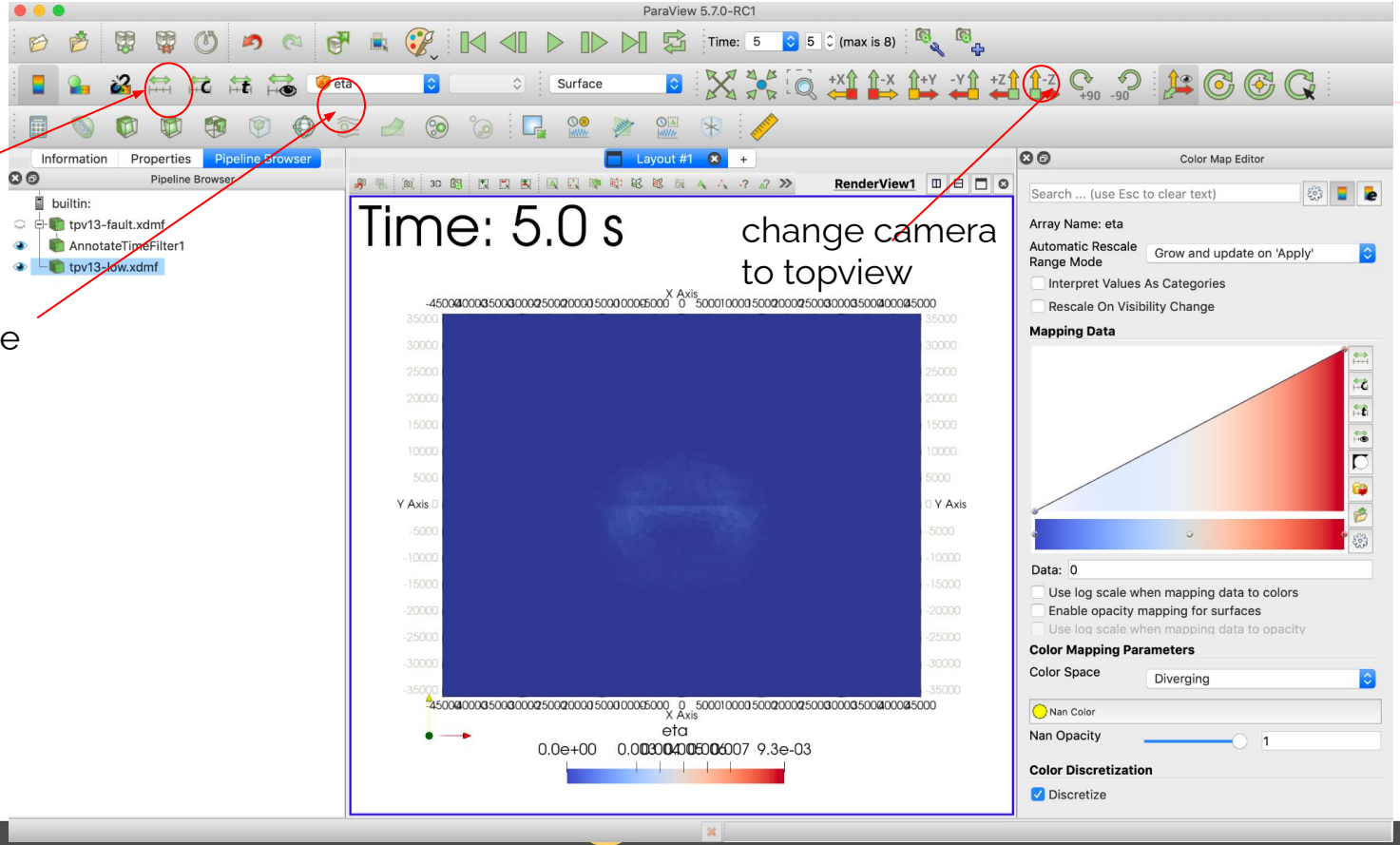
local desktop/laptop, Paraview: add plastic strain output



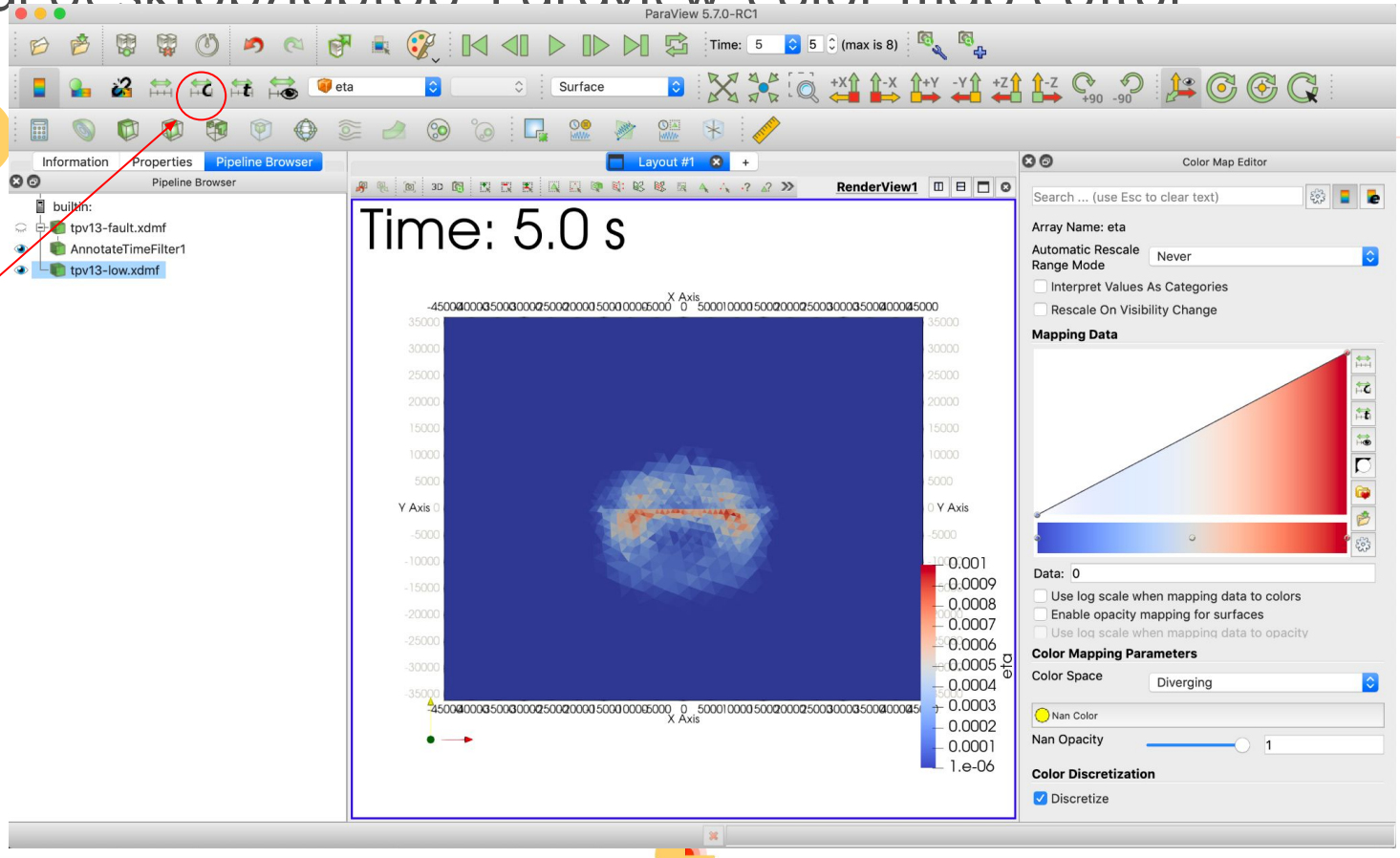
local desktop/laptop, Paraview: add plastic strain output

Automatic scale data

select attribute eta

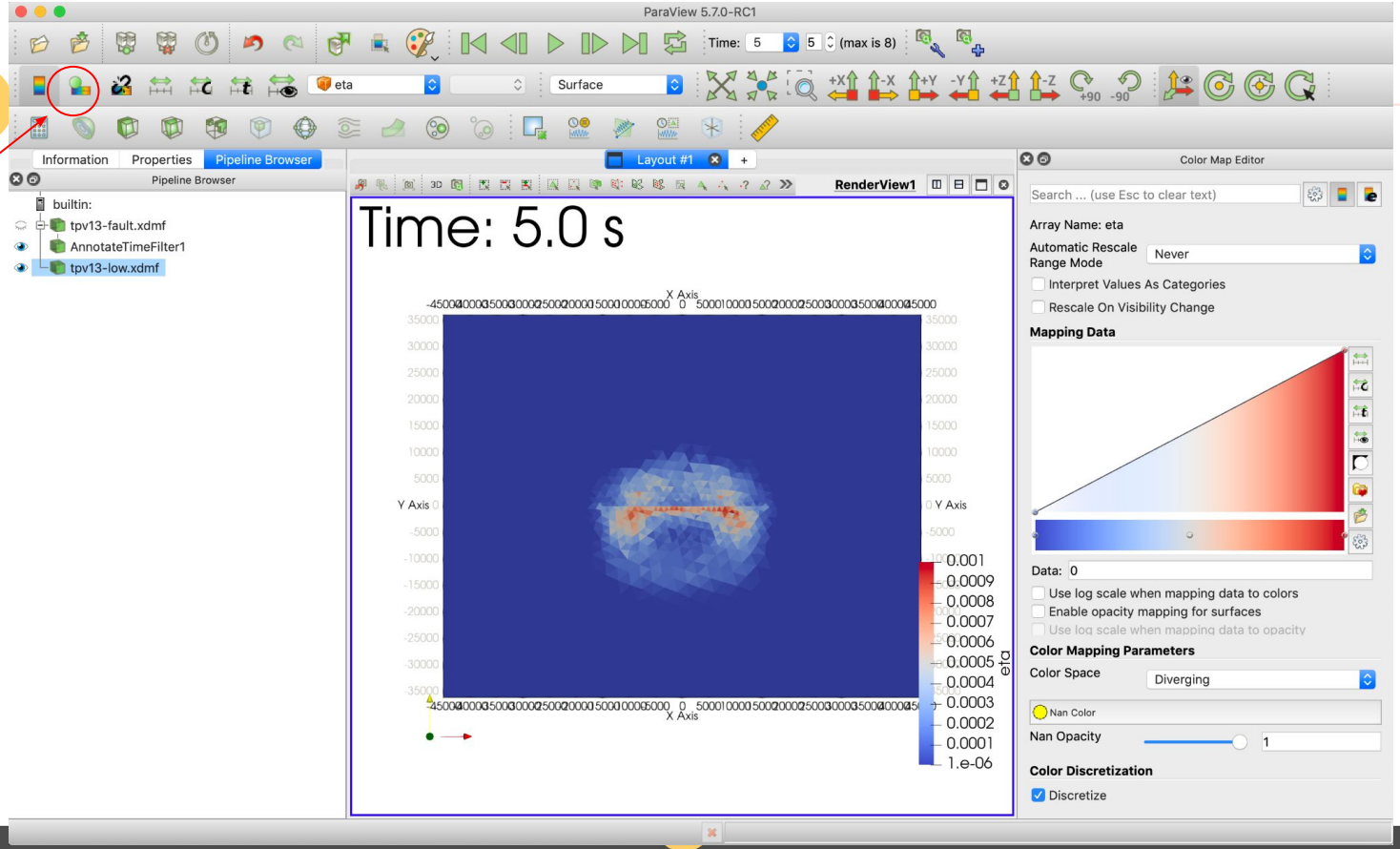


local desktop/laptop Paraview: color map editor

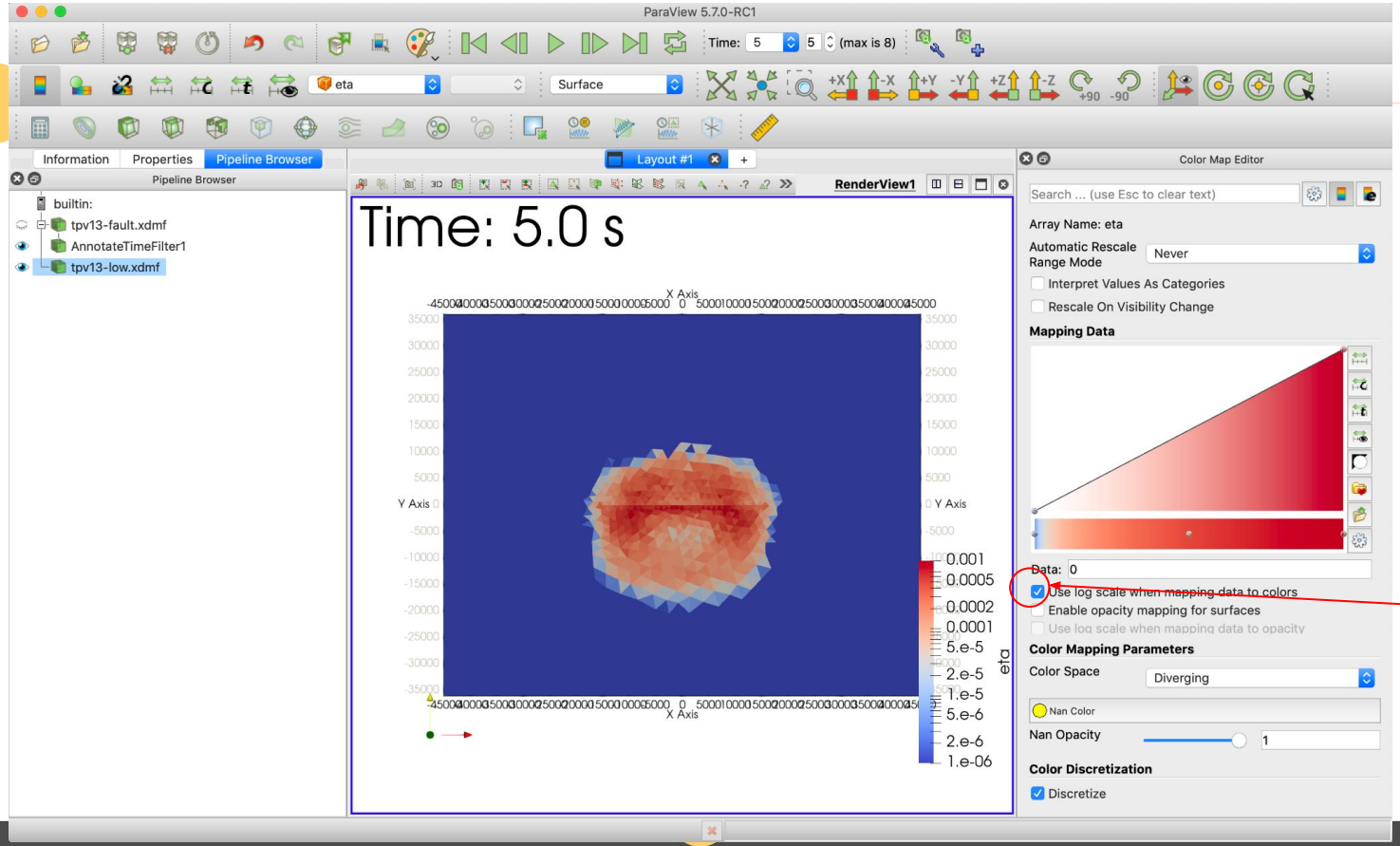


set data range 1e-6 to 1e-3

local desktop/laptop, Paraview: color map editor

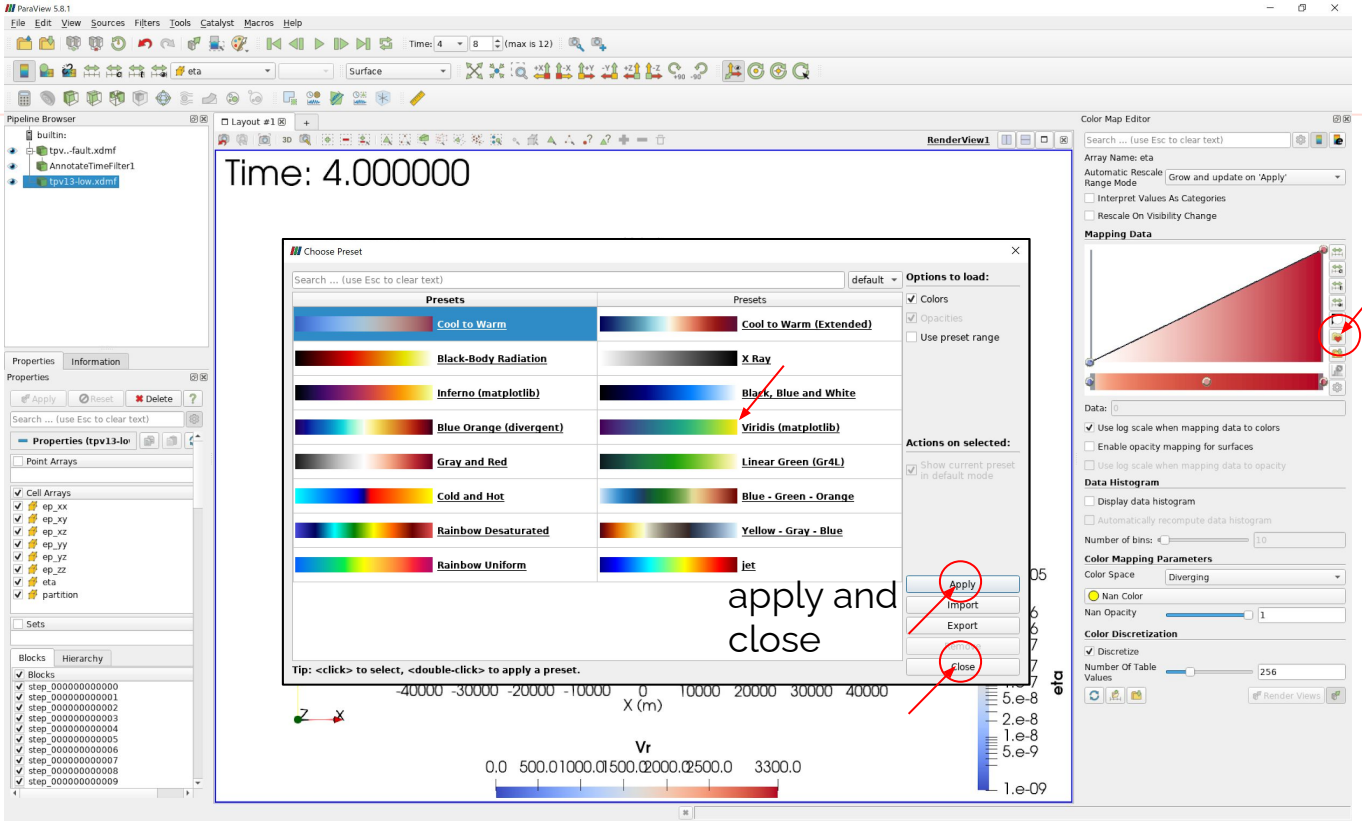


local desktop/laptop, Paraview: edit scalebar



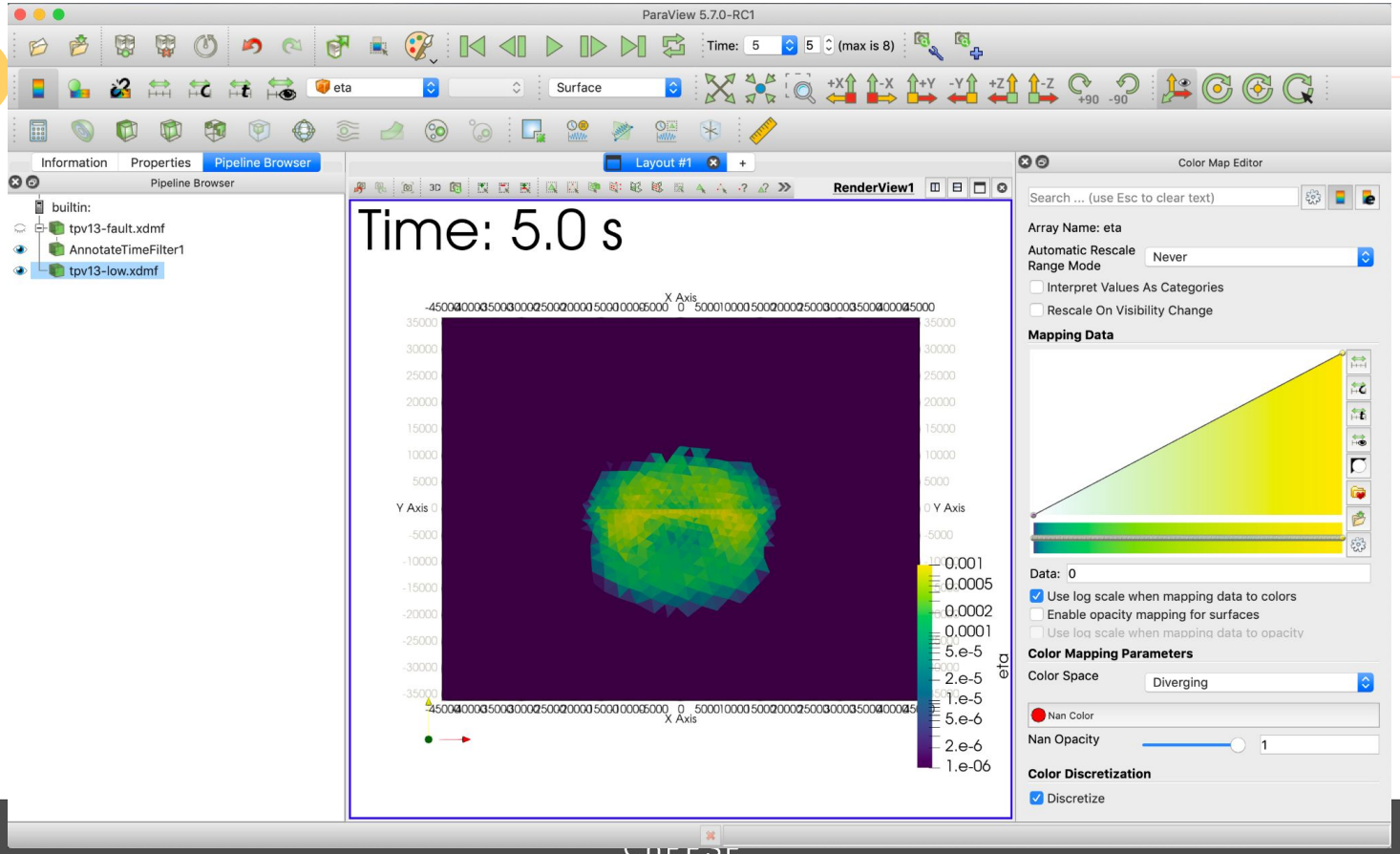
change to log scale

local desktop/laptop, Paraview: change colormap



choose preset colormap

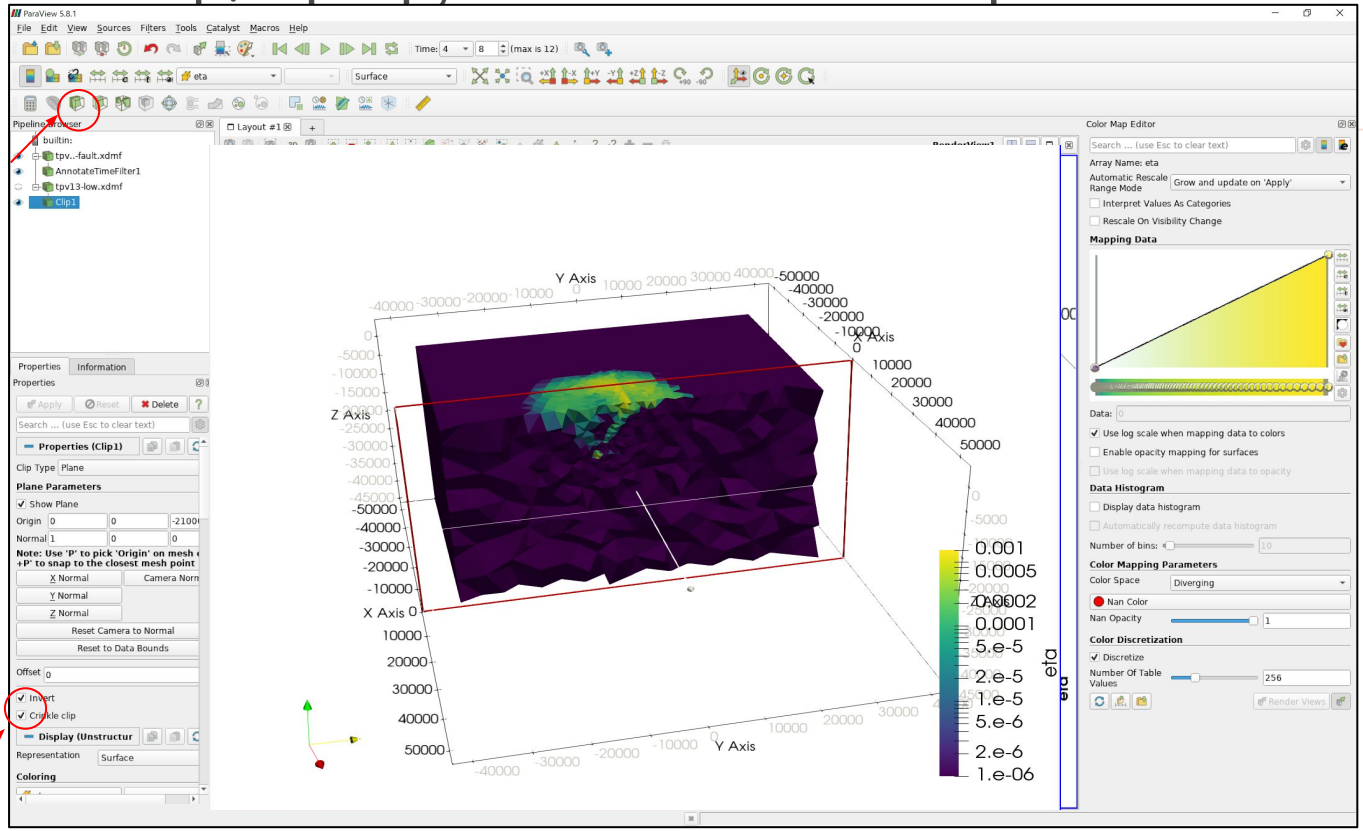
local desktop/laptop, Paraview: change colormap



local desktop/laptop, Paraview: model clip

select clip

crinkle clip



Done ! ☺️

Let's go back to the main Zoom room

Example 3
The Palu Sulawesi earthquake

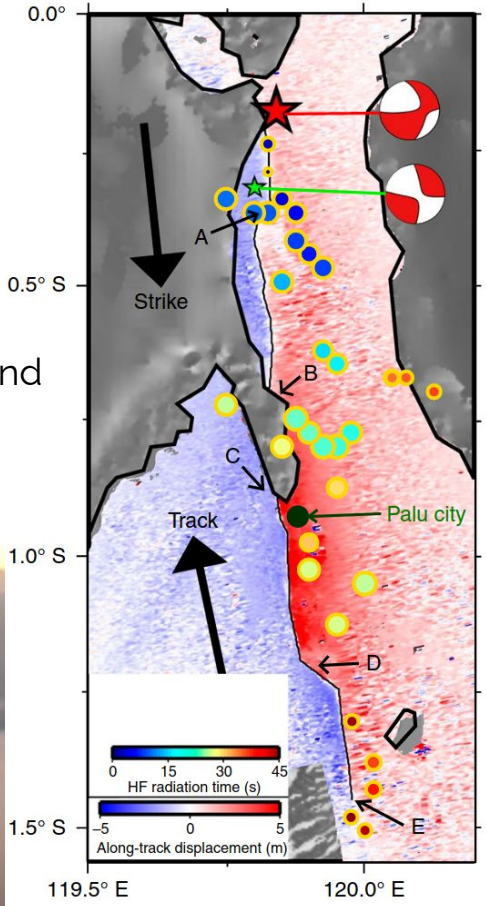
Example 3 (dynamic): 2018 Mw7.5 Palu, Sulawesi earthquake

- Left-lateral, dominantly strike-slip faulting on overall straight and connected fault segments
- Powerful localized tsunami
- 4,340 fatalities,

deadliest earthquake of 2018

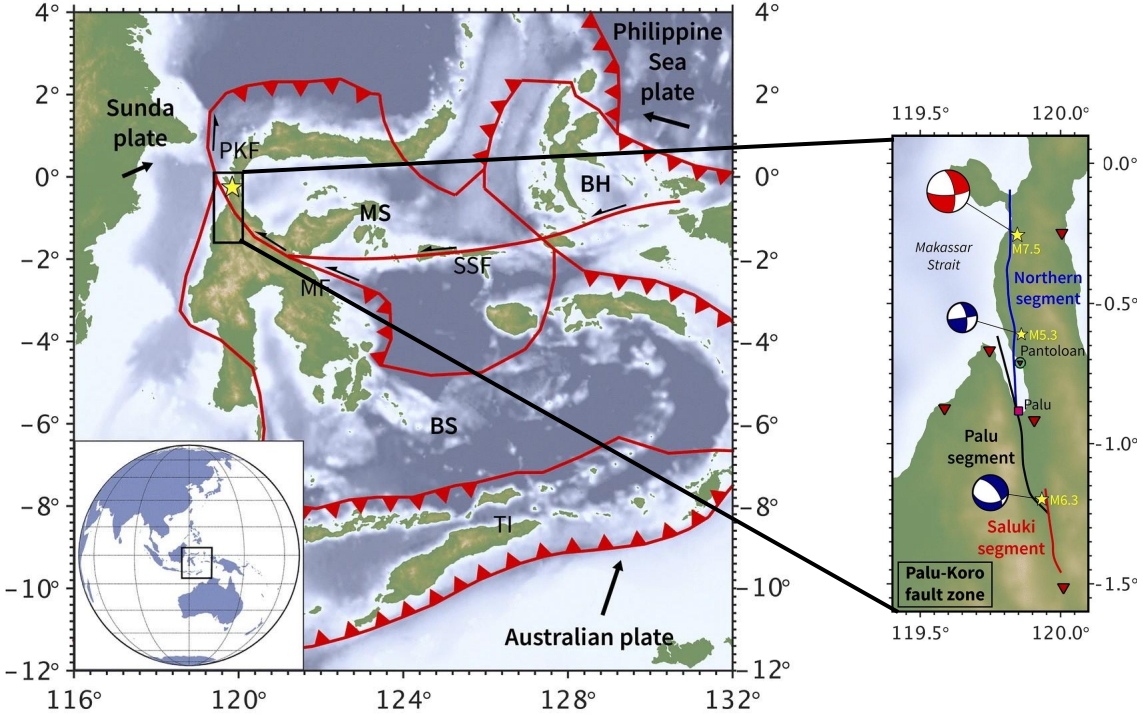
- Supershear rupture

→ Back projection and INSar analysis of the Palu EQ, [Bao et al. (2019)]



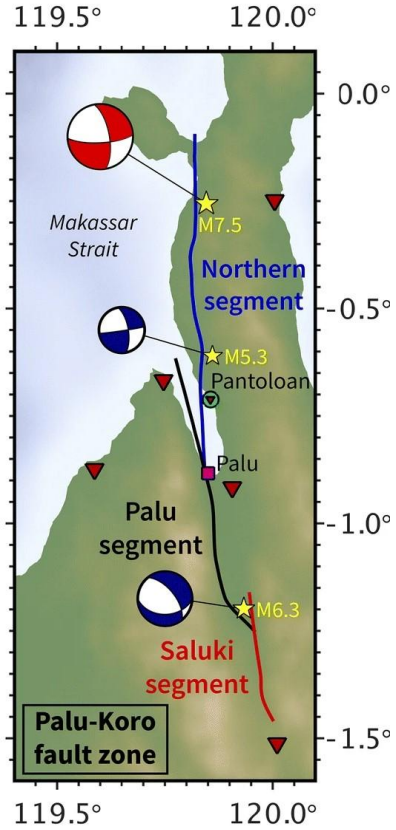
2018 Mw7.5 Palu, Sulawesi earthquake

- Tectonic loading
- Strike-slip fault system
- Shallow depths
- Followed by an unexpected localized tsunami within Palu Bay



Tectonic setting of the September 28, 2018 Mw 7.5 Sulawesi earthquake

Sulawesi earthquake - fault system

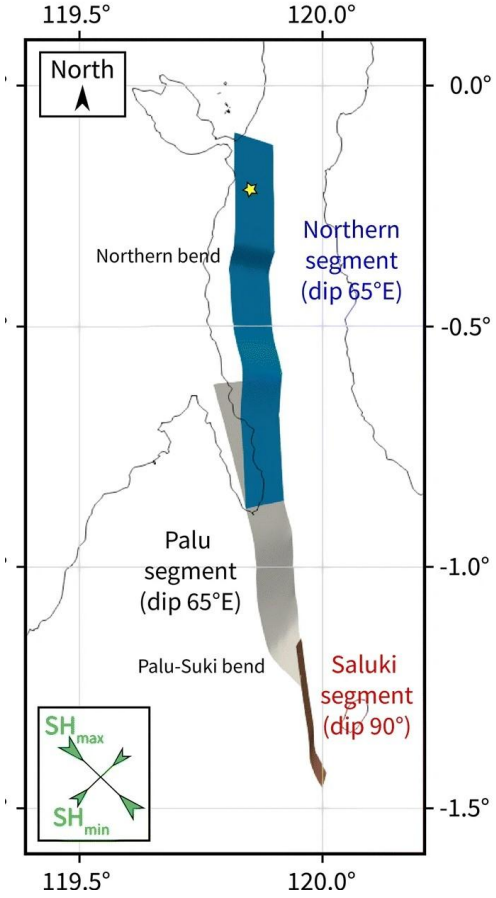


- The later two events provide constraints on the dip angles of individual segments of the fault network
- A complex 3D fault system can be created

left: Focal mechanisms and epicenters of the September 28, 2018 Palu earthquake (USGS (2018a), top), October 1, 2018 Palu aftershock (middle), and January 23, 2005 Sulawesi earthquake (bottom)

right: 3D model of the fault network viewed from top

(Ulrich et al., 2019)



Parameters setup

- A parameter file: **parameters.par**
- Several dependent files: **yaml :**

*__fault.yaml__ contains fault parameters

*__initial_stess.yaml__ contains initial stresses in 3D

*__material_3d.yaml__ assigns the initial stresses and Lamé parameters

*__nucleation_stress.yaml__ forced nucleation within predefined patch

parameters.par

```
&equations
MaterialFileName = 'Sulawesi_material3d.yaml'
!off-fault plasticity parameter (ignored in Plasticity=0)
Plasticity=1
Tv=0.05
!Attenuation parameters (ignored if not compiled with attenuation)
FreqCentral=0.3
FreqRatio=100
/
&IniCondition
/
&Boundaries
BC_fs = 1                    ! enable free surface boundaries
BC_dr = 1                    ! enable fault boundaries
BC_of = 1                    ! enable absorbing boundaries
/
```

parameters.par

```
&DynamicRupture
FL = 103                    ! Friction law rate and state friction with strong
velocity weakening
!yaml file defining spatial dependance of fault properties
ModelFileName = 'Sulawesi_fault.yaml'

!non spatially dependent fault parameters
RS_f0 = 0.6
RS_sr0 = 1d-6
RS_b = 0.014
Mu_W=0.1
RS_iniSlipRate1 = 1d-16
RS_iniSlipRate2 = 0d0
t_0 = 0.5
```

parameters.par

```
!reference vector for defining strike and dip direction
XRef = -0.1                          ! Reference point
YRef = 0.0
ZRef = -1.0
refPointMethod = 1

RF_output_on = 0                     ! Rupture front ascii output
magnitude_output_on =0               ! Moment magnitude output
energy_rate_output_on =1             ! Moment rate output
OutputPointType = 5                  ! Type (5: ascii file and paraview file)
/
```

parameters.par

```
!see: https://seissol.readthedocs.io/en/latest/fault-output.html
! parameterize paraview file output
&Elementwise
printIntervalCriterion = 2                  ! 1=iteration, 2=time
printtimeinterval_sec = 0.5                 ! time interval at which output will be written
OutputMask = 1 1 1 1 1 1 1 1 1 1 1          ! turn on and off fault outputs
refinement_strategy = 2
refinement = 1
/

! parameterize ascii fault file outputs
&Pickpoint
printtimeinterval = 1                       ! Index of printed info at timesteps
OutputMask = 1 1 1 1 1 1 1 1 1 1 1           ! turn on and off fault outputs
nOutpoints = 1
PPFileName = 'faultreceiver_hypocenter.dat'
/
```

parameters.par

```
&MeshNml
MeshFile = 'Sulawesi_65dip_straightBay_ShortNorth_micro'
meshgenerator = 'PUML'                    ! Name of meshgenerator (PUML)
/

&Discretization
CFL = 0.5                                 ! CFL number (<=1.0)
ClusteredLTS = 2                          ! 2: Local time stepping
!ClusteredLTS defines the multi-rate for the time steps of the clusters 2 for Local
time stepping
/
```

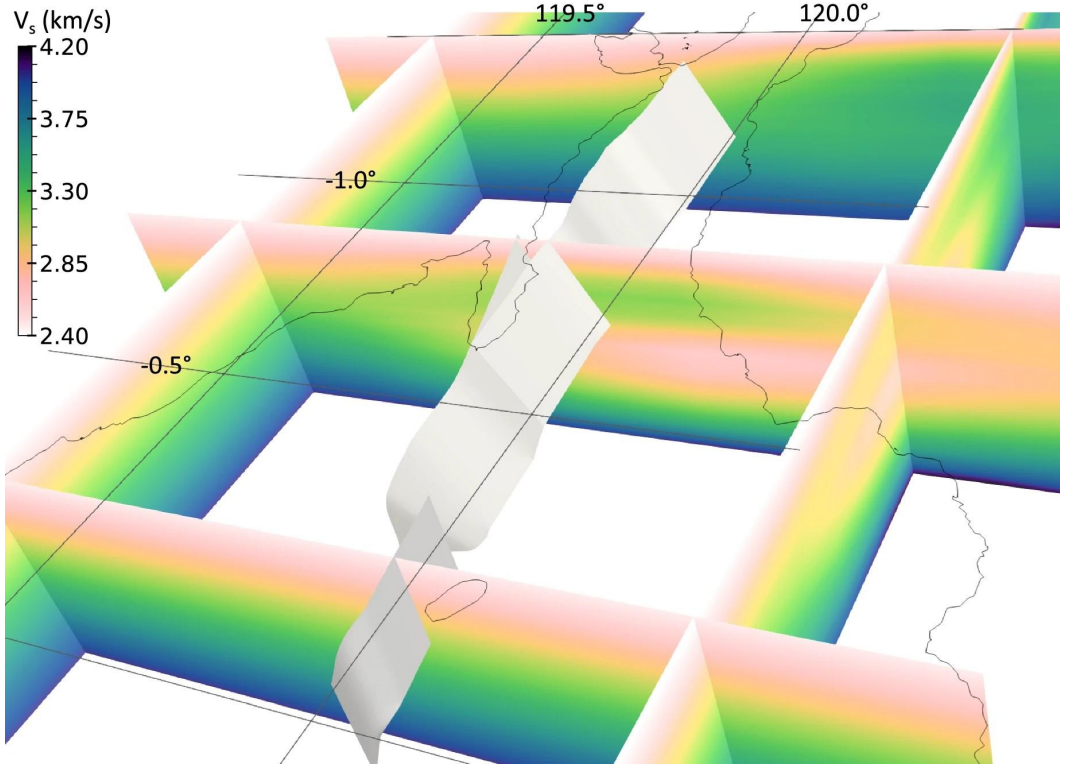
parameters.par

```
FaultOutputFlag = 1                    ! Dynamic Rupture output (add this line
only if active)
OutputFile = 'output/Sulawesi'
Format = 10                            ! Format (10= no output, 6=hdf5 output)

!                    |stress     |vel
iOutputMask = 0 0 0 0 0 0 1 1 1
printIntervalCriterion = 2             ! Criteria for index of printed info (2=time)
TimeInterval = 5.                      ! Index of printed info at time
refinement = 2
```

parameters.par

```
! off-fault ascii receivers
nRecordPoints = 33                          ! number of Record points which are read from file
RFileName = 'GPSfiltered_proj_200m_8e6.dat'    ! Record Points in extra file
pickdt = 0.01                               ! Pickpoint Sampling
pickDtType = 1                              ! Pickpoint Type
! (Optional) Synchronization point for receivers.
!If omitted, receivers are written at the end of the simulation.
ReceiverOutputInterval = 1.0

! Free surface output
SurfaceOutput = 1
SurfaceOutputRefinement = 1
SurfaceOutputInterval = 0.25
!ReceiverOutputInterval = 0.05

xdmfWriterBackend = 'posix' ! (optional)
```

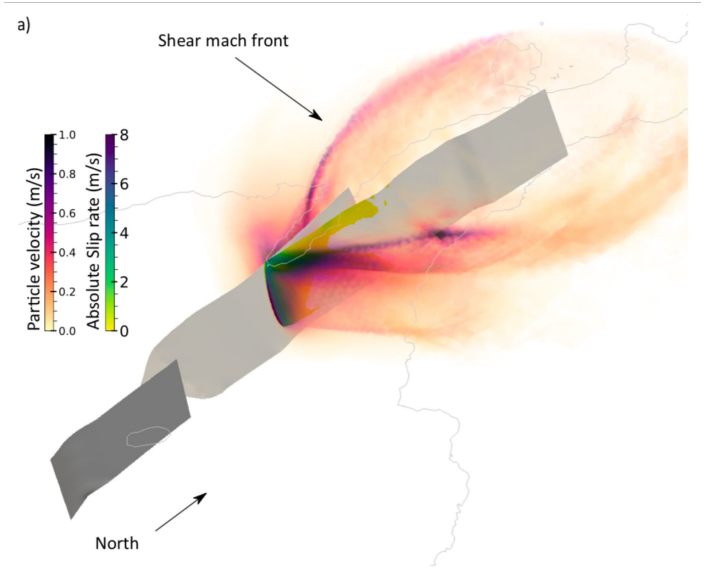
Sulawesi earthquake - 3D material properties



S-wave speeds (Vs) on five cross-sections of the 3D subsurface structure of Awaliah et al. (2018), incorporated into the model

(Ulrich et al., 2019)

Hands on! (30 minutes)



Task 1

- Follow the Jupyter notebook to see how the mesh of the Palu Sulawesi earthquake scenario was generated

Task 2: login to training cluster & copy the generated mesh to the cluster

$ ssh sca50XXX@training.hlrs.de

$ SULAWESI=$(ws_allocate sulawesi 10)

$ cd $SULAWESI

<u>on your local machine:</u>

$ docker cp **sleepy_carson**:/home/training/sulawesi/ .

$ `scp -r sulawesi scaXXXXX@training.hlrs.de:/path/to/workspace/`

training cluster:
What is a job script?

```
#!/bin/bash
#PBS -N seissol
# name of the job
#PBS -l select=1:node_type=sb:ncpus=8:mpiprocs=1:mem=14gb,walltime=02:00:00
# 1 node (sandbridge architecture), 8 CPU, 1 rank, 14Gb of Ram, 2h walltime# ….

cd $PBS_O_WORKDIR
# directory from which the job was submitted (same as where the input files are)
module load compiler/intel
module load mpi/impi

# OpenMP (multi-threading) settings
OMP_NUM_THREADS=8
OMP_PROC_BIND="spread"
OMP_PLACES="cores"
# Run SeisSol with MPI
mpirun -n 1 /shared/cheese/seissol/env/bin/SeisSol_Release_dsnb_4_elastic parametersRS.par
```

training cluster:
run SeisSol

$ **cd SULAWESI**

(before submit job, please check if parameter files, mesh file, etc, exists in current folder !)

$ **qsub submit.sh**

To check job status:

$ **qstat**

To delete a job:
$ **qdel job_identifier**

desktop/laptop: post-processing

- Copy output files from remote (HLRS) to local (desktop/laptop)

  In your terminal:

  ```
  $ scp -r scaXXXXX@training.hlrs.de:/path/to/workspace/ .
  ```

- We will use Paraview to visualize the surface and fault output
- Back up output link:
  `/shared/cheese/seissol/outputs/sulawesi/RS_supershear`

Task 3

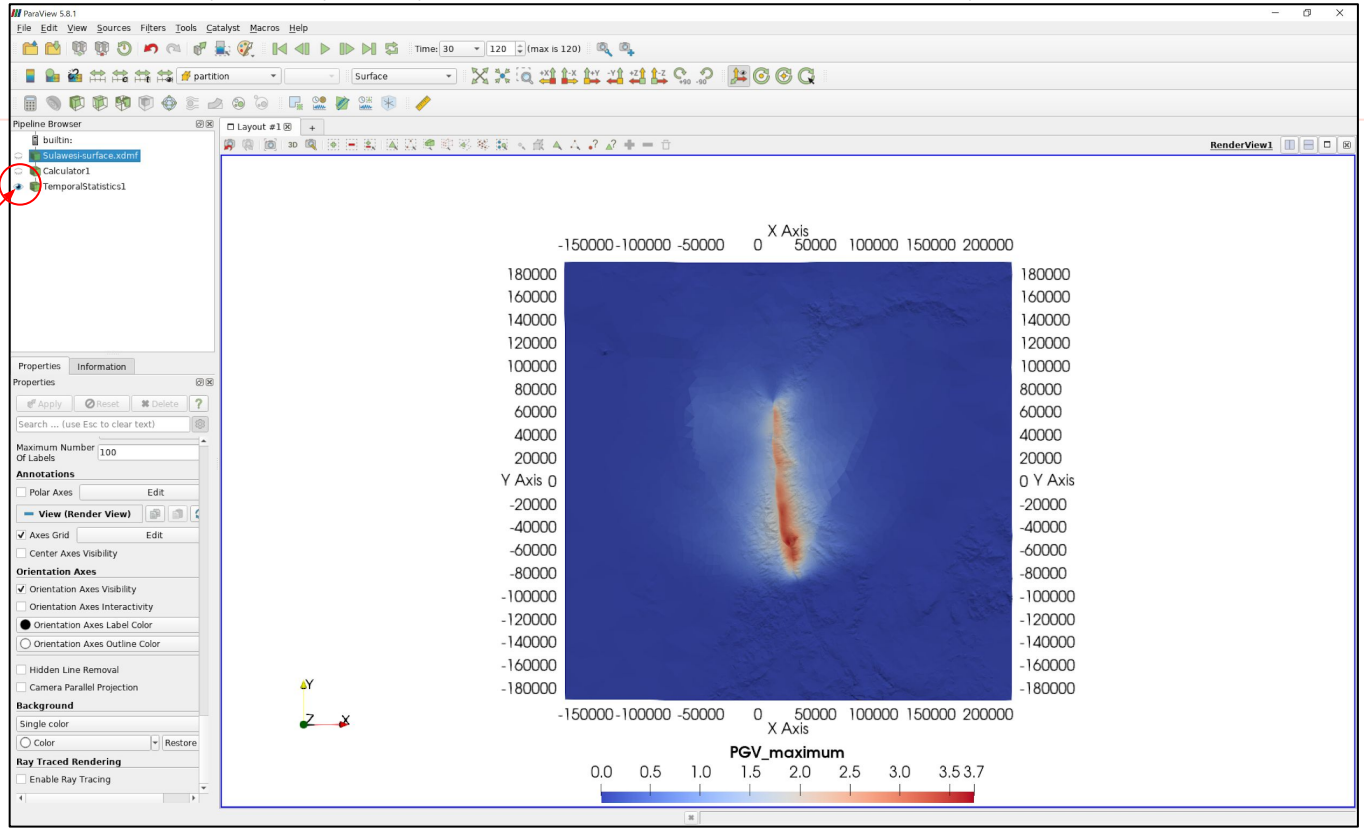
- Proceed with the Jupyter notebook and run the linear slip weakening dynamic rupture model (which one is faster :-D)

Task 4

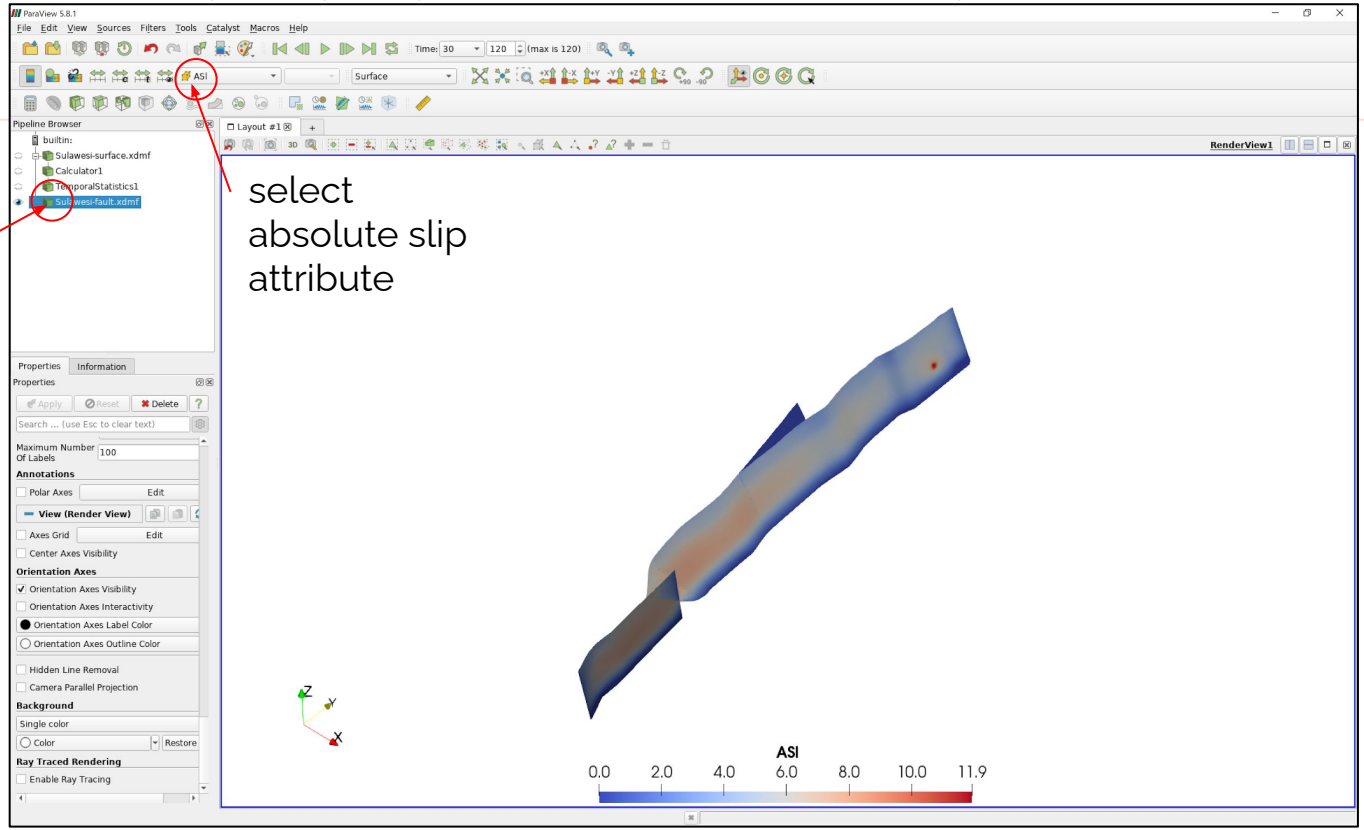
- Visualize the fault output for both, the Jupyter and the cluster example

local desktop/laptop, Paraview: PGV map

follow previous exercise to generate PGV map



local desktop/laptop, Paraview: fault slip

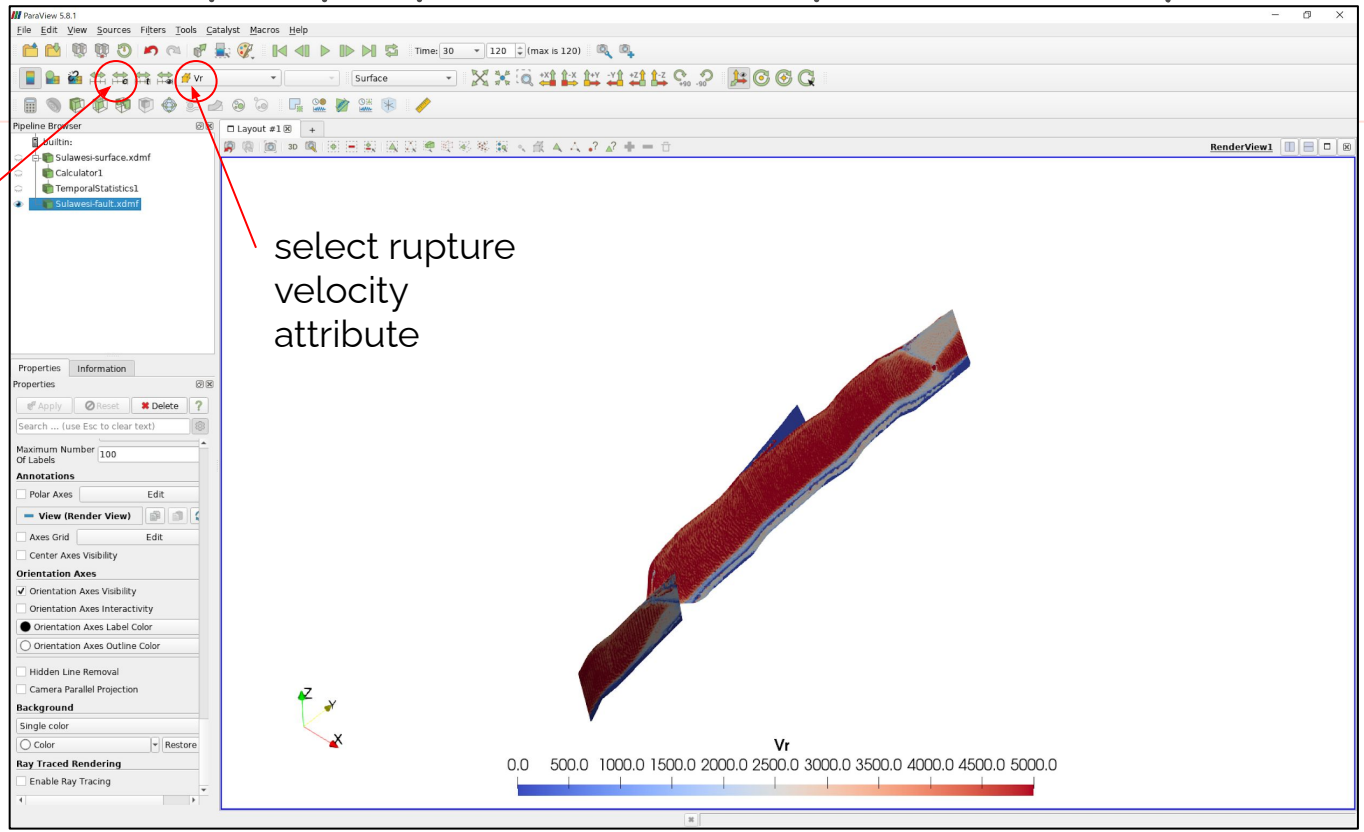


load
fault output

select
absolute slip
attribute

local desktop/laptop, Paraview: rupture velocity

manually rescale data to range 0-5000 m/s



select rupture velocity attribute

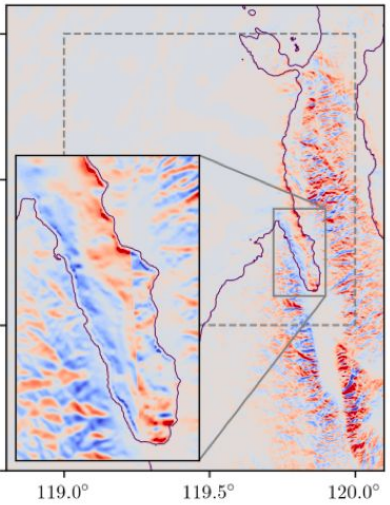
Done (for now)!

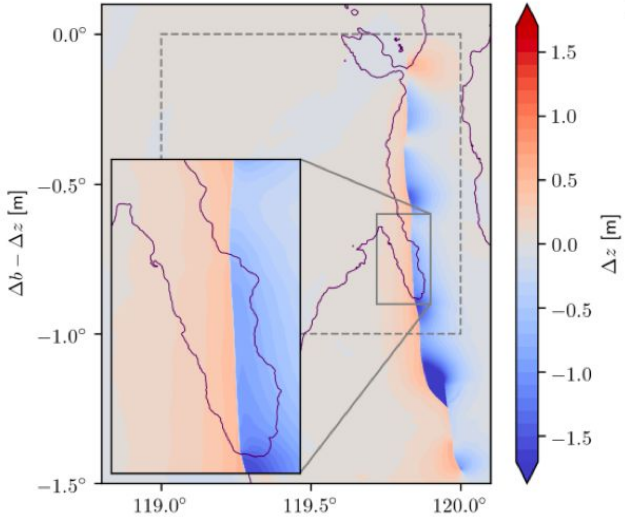
Let's get back to the main zoom room

Rapid earthquake/tsunami modeling - The 2018, Palu-Sulawesi Event

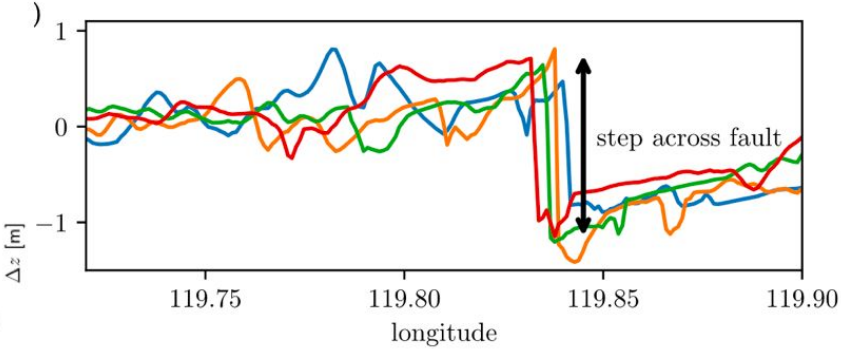
Ulrich et al., PAGEOPH 2019

Vertical displacements + Contribution of horizontal displacements Tanioka and Satake (1996)





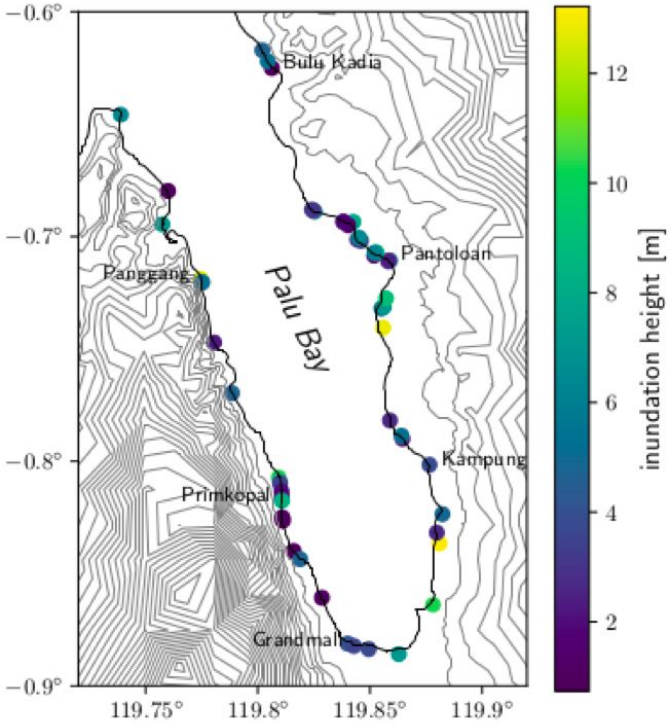
Step of ~1.5m across fault due to normal faulting component and horizontal movements beneath bathtub-like bathymetry



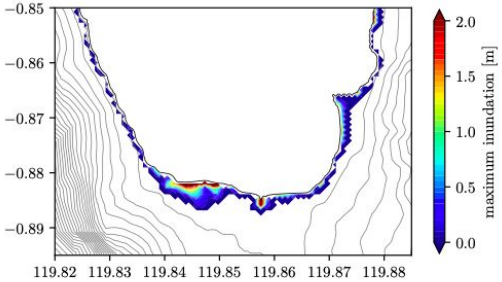
Modelled spontaneous coseismic oblique-slip on a dipping strike-slip fault is sufficient to produce reasonable tsunami amplitude and wave run-up.

Rapid earthquake/tsunami modeling - The 2018, Palu-Sulawesi Event

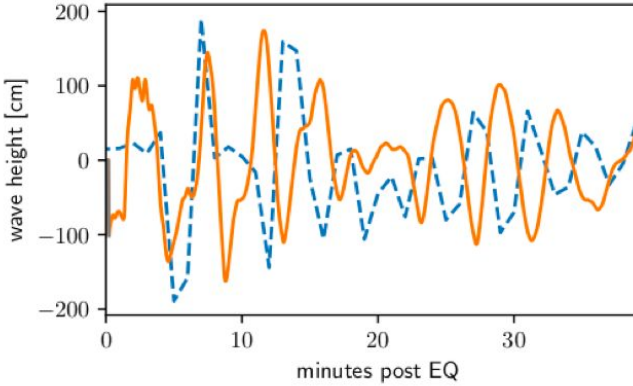
Ulrich et al., PAGEOPH 2019



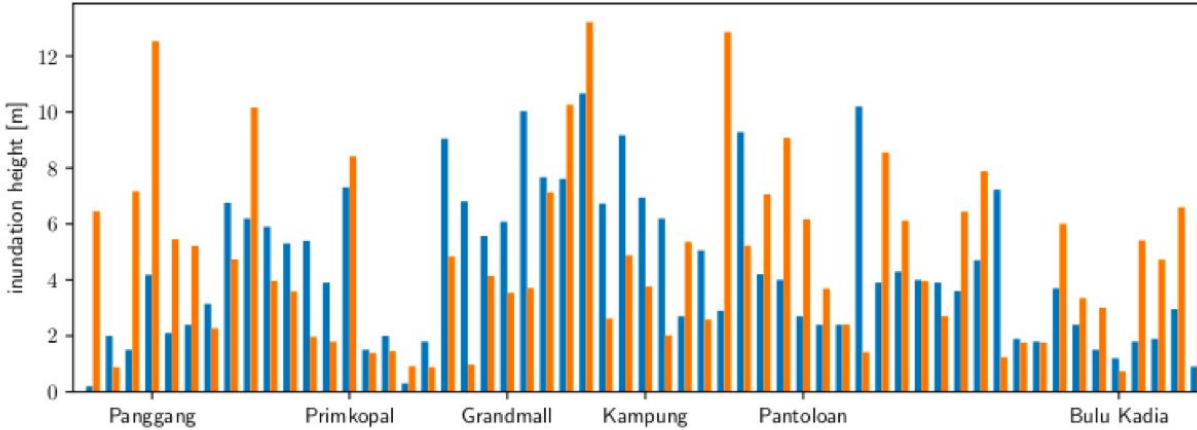
Model maximum inundation map



Model results compared with data at Pantaloan wave gauge

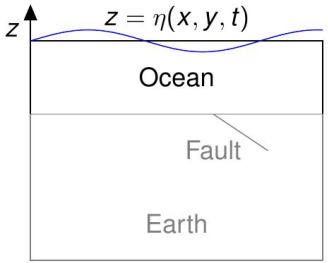


synthetics, observation



3D Acoustic-Elastic Coupling with Gravity: The Dynamics of the 2018 Palu, Sulawesi Earthquake and Tsunami

Krenz et al., 2021 https://arxiv.org/pdf/2107.06640.pdf



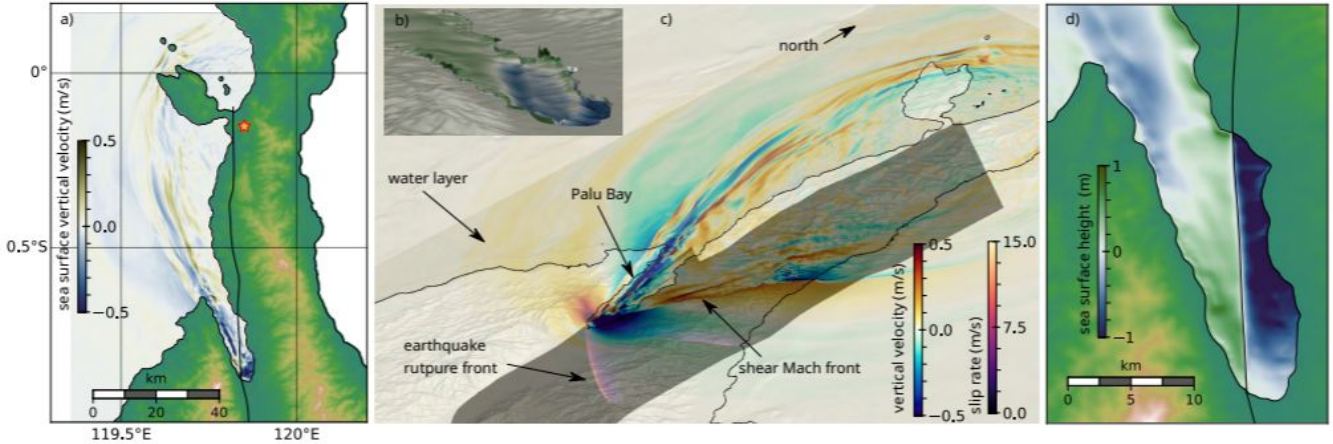


Figure 1: 3D fully coupled acoustic-elastic model with gravity of the 2018 Sulawesi supershear earthquake and local tsunami in Palu Bay. (a) Map view of the vertical sea surface velocity at 15 s simulation time. The star indicates the earthquake epicenter. Black lines mark the traces of a system of complex faults. (b) Sea surface height (m, same color map as in (d)) at 60 s (warped vertical displacements in the bay, amplified ×2000). (c) 3D view of earthquake and tsunami model in Palu Bay. Particle velocity (slip rate) across the faults, vertical sea-surface and Earth ground velocity at 15 s. Dynamic earthquake rupture, propagating faster than the shear-wave velocity of the surrounding rocks, generates a clearly visible Mach cone altering the acoustic response of the ocean. (d) Sea surface height (vertical displacements in Palu Bay) at 15 s.

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

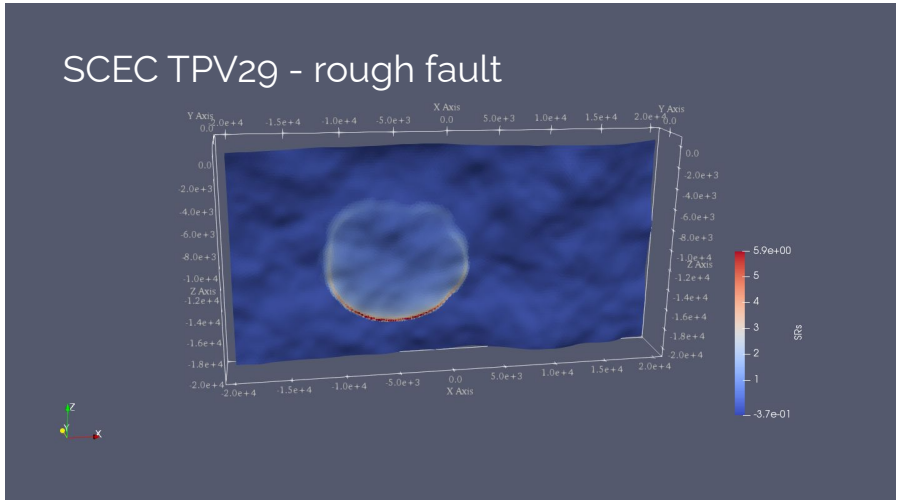
SeisSol

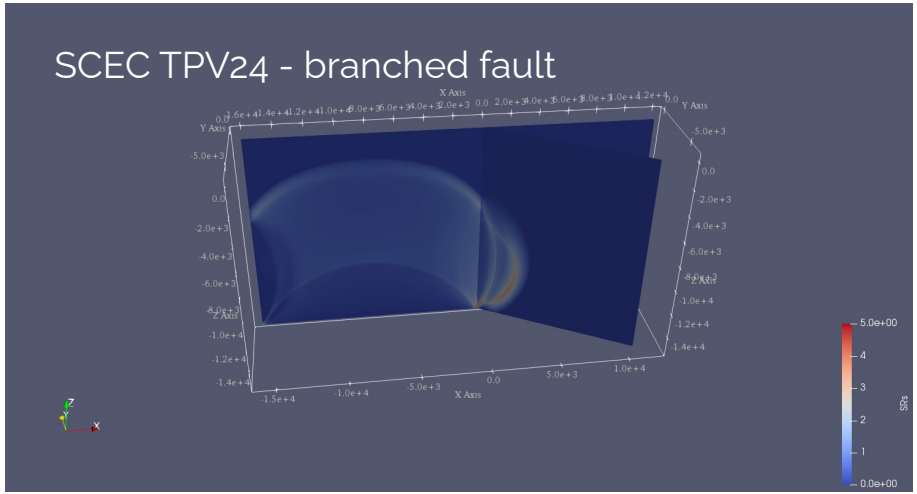
SeisSol is a software package for simulating wave propagation and dynamic rupture based on the arbitrary high-order accurate derivative discontinuous Galerkin method (ADER-DG).

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

SeisSol further examples, references and available reproducible models





more examples at SeisSol documentation
<https://seissol.readthedocs.io/en/latest/cookbook_overview.html>

SeisSol further examples, references and available reproducible models

Palgunadi, Kadek Hendrawan, Alice-Agnes Gabriel, Thomas Ulrich, José Ángel Lopéz-Comino, and Paul Martin Mai (2020), Dynamic Fault Interaction during a Fluid-Injection-Induced Earthquake: The 2017 Mw 5.5 Pohang Event, BSSA, doi:10.1785/0120200106, open access version available at https://eartharxiv.org/w5b9s/.

Ulrich, Thomas, Alice-Agnes Gabriel, Jean-Paul Ampuero, and Wenbin Xu (2019), Dynamic viability of the 2016 Mw 7.8 Kaikōura earthquake cascade on weak crustal faults, Nature Communications, 10(1213), doi:10.1038/s41467-019-09125-w.

Ulrich, Thomas, Stefan Vater, Elizabeth H. Madden, Joern Behrens, Ylona van Dinther, Iris van Zelst, Eric J. Fielding, Cunren Liang, and Alice-Agnes Gabriel (2019), Coupled, Physics-based Modelling Reveals Earthquake Displacements are Critical to the 2018 Palu, Sulawesi Tsunami, Pure and Applied Geophysics

Wollherr, Stephanie, Alice-Agnes Gabriel, and Paul Martin Mai (2019), Landers 1992 ”reloaded”: an integrative dynamic earthquake rupture model, Journal of Geophysical Research - Solid Earth, 124, doi:10.1029/2018JB016355,

Wollherr, Stephanie, Alice-Agnes Gabriel, and Carsten Uphoff (2018), Off-fault plasticity in three-dimensional dynamic rupture simulations using a modal Discontinuous Galerkin method on unstructured meshes: Implementation, verification, and application, Geoph.J.Int., 214(3), 1556–1584, doi:10.1093/gji/ggy213.

Uphoff, Carsten, Sebastian Rettenberger, Michael Bader, Elizabeth H. Madden, Thomas Ulrich, Stephanie Wollherr, and Alice-Agnes Gabriel (2017), Extreme Scale Multi-Physics Simulations of the Tsunamigenic 2004 Sumatra Megathrust Earthquake, Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis SC17, doi:10.1145/3126908.3126948.

Rettenberger, Sebastian, Oliver Meister, Michael Bader, and Alice-Agnes Gabriel (2016), ASAGI - A Parallel Server for Adapative Geoinformation, in 2016 Exascale Applications and Software Conference (EASC2016) Proceedings, doi:10.1145/2938615.2938618.

Breuer, Alexander, Alexander Heinecke, Sebastian Rettenberger, Michael Bader, Alice-Agnes Gabriel, and Christian Pelties (2014), Sustained Petascale Performance of Seismic Simulations with SeisSol on SuperMUC, in International Supercomputing Conference (ISC) Proceedings, Lecture Notes in Computer Science, vol. 8488, pp. 1-18, Springer International Publishing, doi:10.1007/978-3-319-07518-1_1, PRACE ISC Award Winning Paper.

Heinecke, A., et al. (2014), Petascale High Order Dynamic Rupture Earthquake Simulations on Heterogeneous Supercomputers, in SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 3-14, doi:10.1109/SC.2014.6, Gordon Bell Prize Finalist 2014.

SeisSol further examples, references and available reproducible models

Pelties, Christian, Alice-Agnes Gabriel, and Jean-Paul Ampuero (2014), Verification of an ADER-DG method for complex dynamic rupture problems, Geoscientific Model Development, 7(3), 847-866, doi:10.5194/gmd-7-847-2014.

Wenk, Stefan, Christian Pelties, Heiner Igel, and Martin Käser (2013), Regional wave propagation using the discontinuous Galerkin method, Solid Earth, 43-57, doi:10.5194/se-4-43-2013.

Pelties, Christian, Josep De la Puente, Jean-Paul Ampuero, Gilbert B. Brietzke, and Martin Käser (2012), Three-Dimensional Dynamic Rupture Simulation with a High-order Discontinuous Galerkin Method on Unstructured Tetrahedral Meshes, J. Geophys. Res. - Solid Earth, doi:10.1029/2011JB008857.

Chaljub, Emmanuel, Peter Moczo, Seiji Tsuno, Pierre-Yves Bard, Jozef Kristek, Martin Käser, Marco Stupazzini, and Miriam Kristekova (2010), Quantitative Comparison of Four Numerical Predictions of 3D Ground Motion in the Grenoble Valley, France,

Bull. Seis. Soc. Am., 100(4), 1427-1455, doi:10.1785/0120090052.

Käser, Martin, Hugues Djikpesse, and Michael Prange (2010), Numerical Modeling of Borehole-guided Waves and Reservoir Formation Reflections, in SEG Expanded Abstracts, vol. 29, pp. 573-577, doi:10.1190/1.3513849.

De la Puente, Josep, Michael Dumbser, Martin Käser, and Heiner Igel (2008), Discontinuous Galerkin Methods for Wave Propagation in Poroelastic Media, Geophysics, 73(5), T77-T97, doi:10.1190/1.2965027.

Käser, Martin, Verena Hermann, and Josep de la Puente (2008), Quantitative Accuracy Analysis of the Discontinuous Galerkin Method for Seismic Wave Propagation, GJI, 173(3), 990-999, doi:10.1111/j.1365-246X.2008.03781.x.

De la Puente, J., M. Käser, M. Dumbser, and H. Igel (2007), An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes IV: Anisotropy, Geophysical Journal International, 169(3), 1210-1228, doi:10.1111/j.1365-246X.2007.03381.x.

Dumbser, Michael, Martin Käser, and Eleuterio Toro (2007), An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes V: Local Time Stepping and p-Adaptivity, Geophysical Journal International, 171(2), 695-717, doi:10.1111/j.1365-246X.2007.03427.x.

Käser, M., M. Dumbser, J. de la Puente, and H. Igel (2007), An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes III: Viscoelastic Attenuation, Geophysical Journal International, 168(1), 224-242, doi:10.1111/j.1365-246X.2006.03193.x.

Dumbser, M., and M. Käser (2006), An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes II: The Three-Dimensional Isotropic Case, Geophysical Journal International, 167(1), 319-336, doi:10.1111/j.1365-246X.2006.03120.x.

Käser, M., and M. Dumbser (2006), An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes I: The Two-Dimensional Isotropic Case with External Source Terms, Geophysical Journal International, 166(2), 855-877, doi:10.1111/j.1365-246X.2006.03051.x.

SeisSol further examples, references and available reproducible models

https://zenodo.org/record/2538024

zenodo | Search | Upload | Communities | Log in | Sign up

January 11, 2019 — Dataset — Open Access

SeisSol model setup of the Kaikōura's dynamic rupture model of Ulrich et al.

Ulrich, Thomas; Gabriel, Alice-Agnes; Ampuero, Jean-Paul; Xu, Wenbin

All data required to run the dynamic rupture model of the Kaikōura earthquake presented in Ulrich, T., Gabriel, A. A., Ampuero, J. P., & Xu, W. (2018). Dynamic viability of the 2016 Mw 7.8 Kaikōura earthquake cascade on weak crustal faults. A detailed readme file summarizing the data and data formats is also provided.

Files (2.8 GB)

Name	Size	
NZ_asagi.nc md5:5b99f8d6191923e12c04826d0fd2a55f	75.7 MB	Download
NZ_asagi_att.nc md5:1ecc27c8174b8eba0c97eaf234828fce	50.5 MB	Download
NZ_fault.yaml md5:8ee52ea5327e7fb47cebb4f0d82c6ed9	1.1 kB	Download
NZ_initial_stress.yaml md5:9c79f0e6fd37ae7562ed308da23a184a	3.8 kB	Download
NZ_material.yaml md5:1b7fb0c8dc4d9c064ceff49d951cf2e2	860 Bytes	Download
NZ_nucleation_stress.yaml md5:59df05e8cbbf8cbd08ac7c3d8440ed62	1.2 kB	Download

148 views — 243 downloads — See more details...

Indexed in OpenAIRE

Publication date: January 11, 2019

DOI: DOI 10.5281/zenodo.2538024

License (for files): Creative Commons Attribution 4.0 International

Versions

Version 1 — Jan 11, 2019 — 10.5281/zenodo.2538024

Cite all versions? You can cite all versions by using the DOI 10.5281/zenodo.2538023. This DOI represents all versions, and will always resolve to the latest one. Read more.

https://zenodo.org/record/3234664

zenodo | Search | Upload | Communities | Log in | Sign up

May 29, 2019 — Dataset — Open Access

SeisSol dynamic rupture model setup of the Mw 7.5 Palu earthquake scenario published in Ulrich et al. (2019)

Thomas Ulrich; Stefan Vater; Elizabeth H. Madden; Joern Behrens; Ylona van Dinther; Iris van Zelst; Eric J. Fielding; Cunren Liang; Alice-Agnes Gabriel

All data required to run the dynamic rupture model of the Palu earthquake presented in:

Ulrich, T., Vater, S., Madden, E. H., Behrens, J., van Dinther, Y., van Zelst, I., Fielding, E. J., Liang, C. & Gabriel, A. A. (2019). Coupled, Physics-based Modeling Reveals Earthquake Displacements are Critical to the 2018 Palu, Sulawesi Tsunami. doi: 10.31223/osf.io/3bwqa.

A detailed readme file summarizing the data and data formats is also provided.

Files (341.8 MB)

Name	Size	
3dvel_Sulawesi.nc md5:f6c1b69b8695eb8b8f9791ba52958ae6	1.2 MB	Download
3dvel_Sulawesi_BMKG.nc md5:431ed396ebf3d4717dfb324ad13d6ab2	219.0 kB	Download
faultreceiver_hypocenter.dat md5:15cb10af1a45f365939b2a80c19c7f43	41 Bytes	Download
GPSfiltered_proj_200m_8e6.dat md5:28419db0f5550f0802d711ba5209fdd0	1.7 kB	Download
parameters.par	4.1 kB	Download

95 views — 188 downloads — See more details...

Indexed in OpenAIRE

Publication date: May 29, 2019

DOI: DOI 10.5281/zenodo.3234664

License (for files): Creative Commons Attribution 4.0 International

Versions

Version 1 — May 29, 2019 — 10.5281/zenodo.3234664

Cite all versions? You can cite all versions by using the DOI 10.5281/zenodo.3234663. This DOI represents all versions, and will always resolve to the latest one. Read more.

Add-on (if time allows) building a 3D model with complicated geometries using SimModeler

Geometry

We will build a 3D model with complicated geometries using **SimModeler**:

1. Creating a high resolution topography and bathymetry free surface and merge it with a simple box model

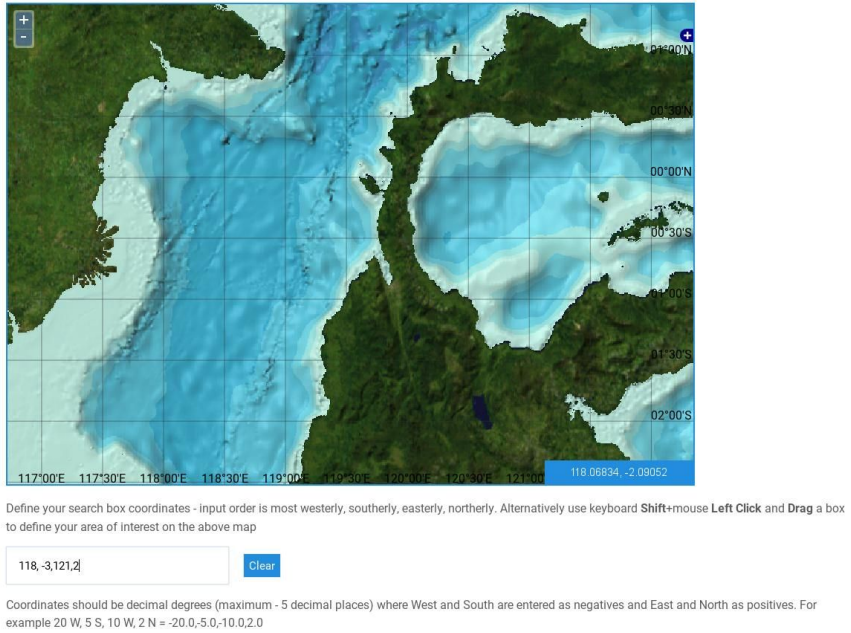
2. Creating complex fault networks constrained by fault traces and dip

3. Volume meshing with unstructured tetrahedral meshing

Creating the topographic layer

- We create the topography from a NetCDF file downloaded from GEBCO https://www.gebco.net/.

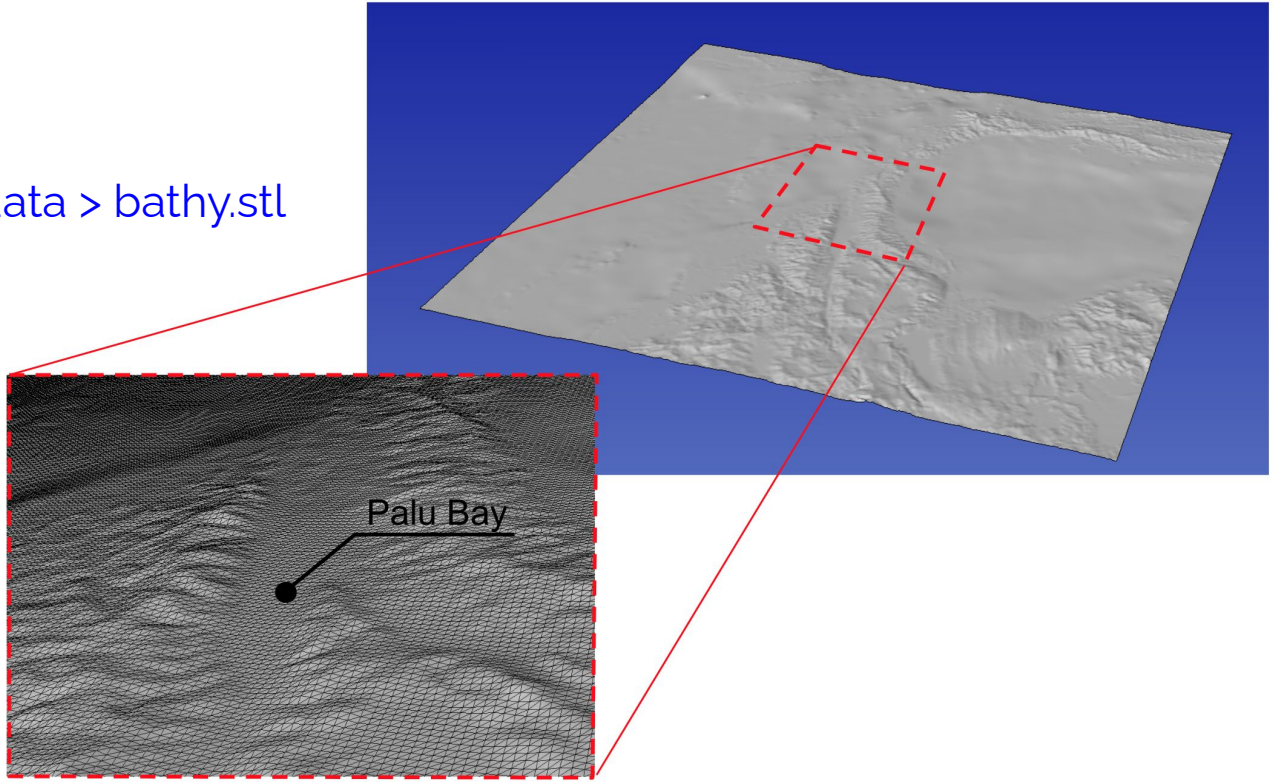
- We then project the data, triangulate it, and export it as stl (list of triangles):



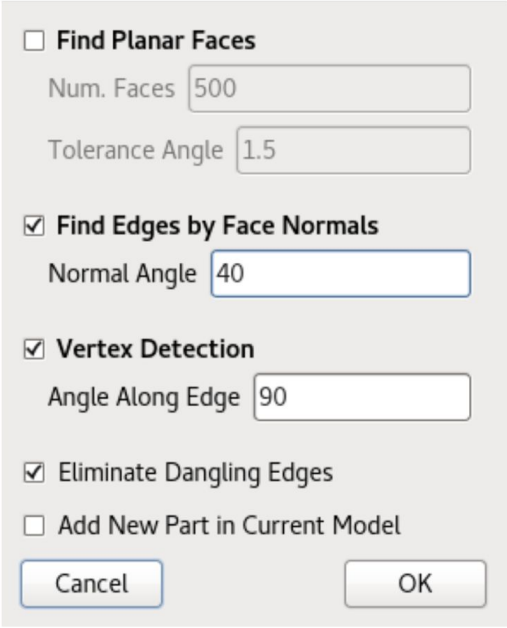

```
python createGOCADTSurfNXNY_netcdf.py --proj +init=EPSG:23839
data/GEBCO_2014_2D_118.1904_-2.4353_121.6855_1.0113.nc bathy.stl
```

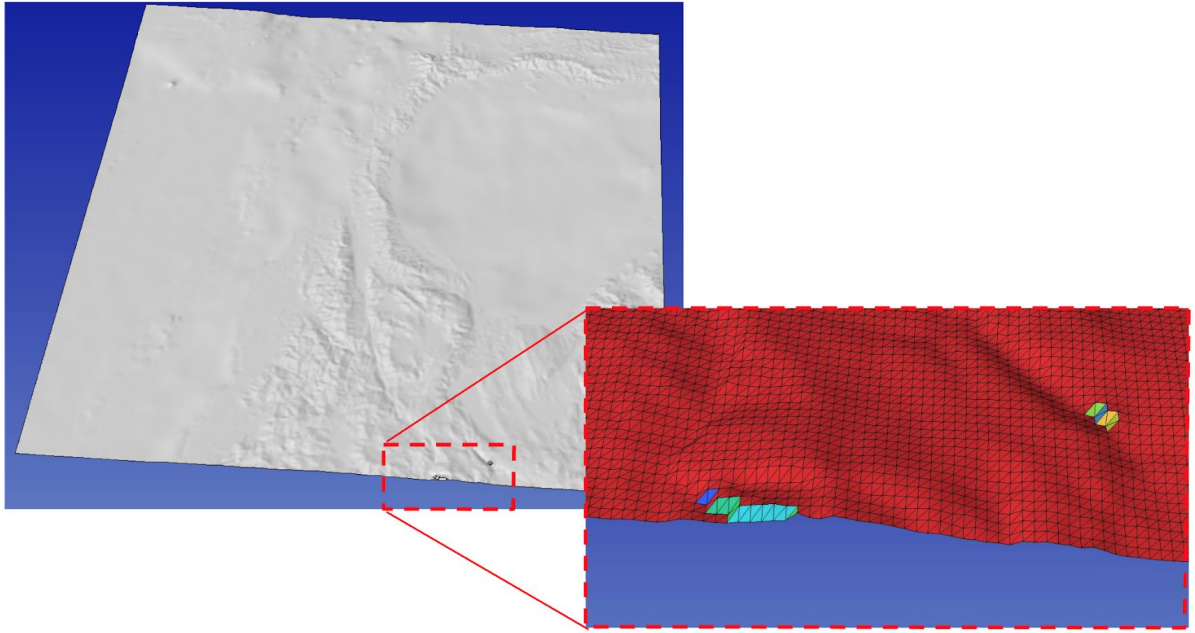
- We then load the stl file into **SimModeler**:

File > Import discrete data > bathy.stl



Note that in SimModeler import options, importing with a small angle isolate tiny faces from the main surface, that then needs to be accounted for by the mesh.





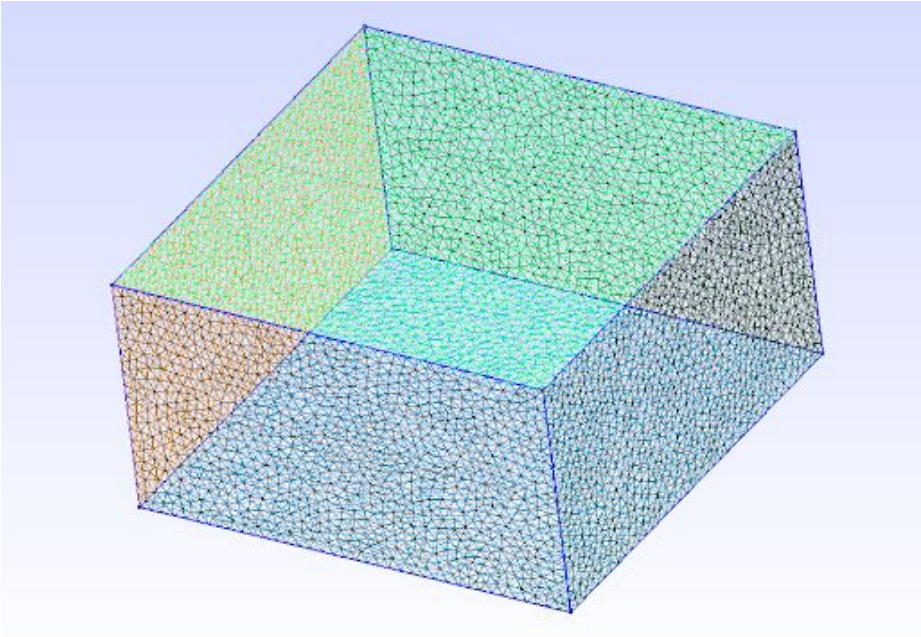
Creating the domain box

We generate a simple box with **Gmsh**:

```
gmsh -2 create_box.geo -format stl
```

The box dimensions are such as the topography is slightly wider than the box. The mesh size is chosen small enough to facilitate intersection with topography and large enough to limit the number of elements.

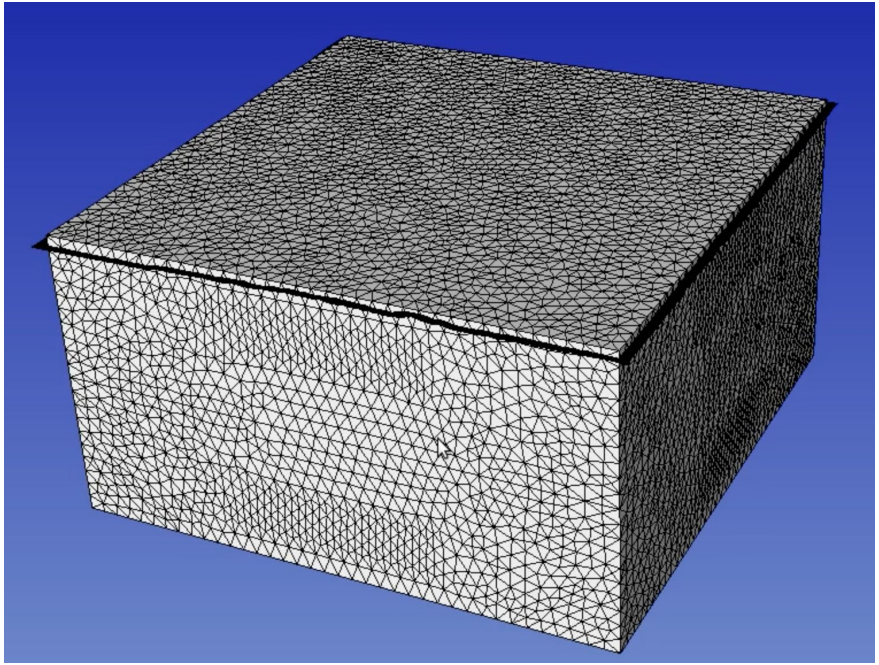
```
mesh_size = 10e3;
Xmax = -160e3; Xmin = 215e3;
Ymin = 1235e3; Ymax = 1605e3;
Zmin = -200e3; Zmax = 5e3;
```



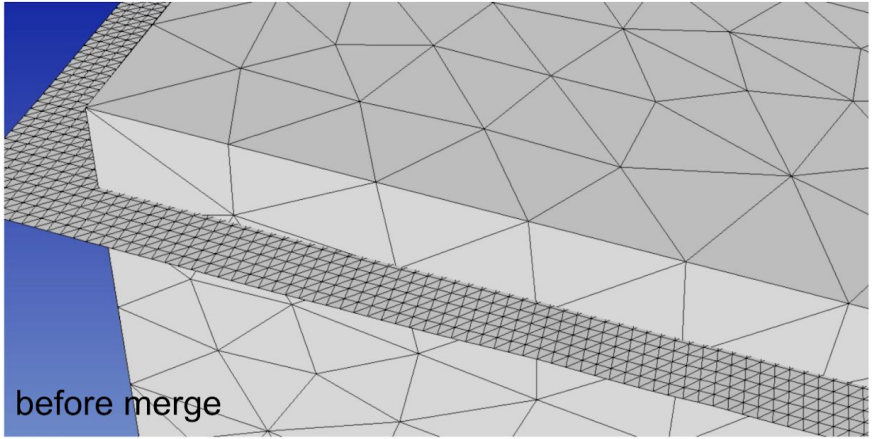
Merging topographic layer with box model

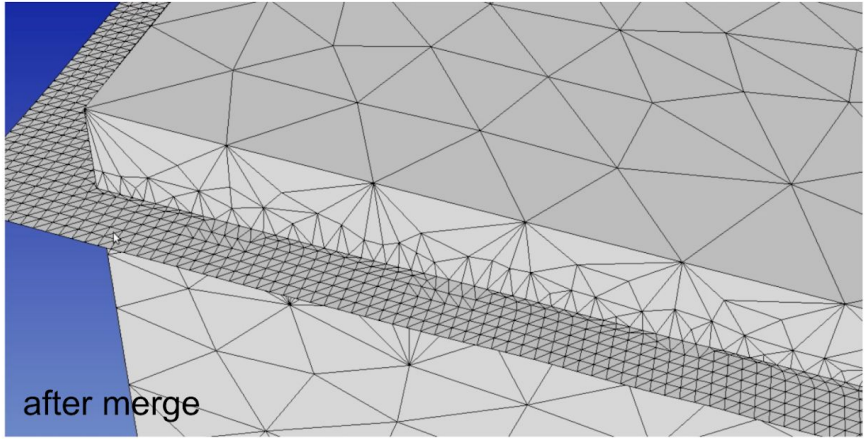
We first import `create_box.stl` with import options:

- activating **'Edges by Face Normals'** with normal angle 60◦,
- activating **'Vertex Detection'** with angle along edge 90◦, and
- eliminating **'Dangling Edges'.**
- Then we import the topographic layer bathy.stl with increasing **'Edges by Face Normals'** normal angle to 80◦, and
- activating **'Add New Part in Current Model'**.

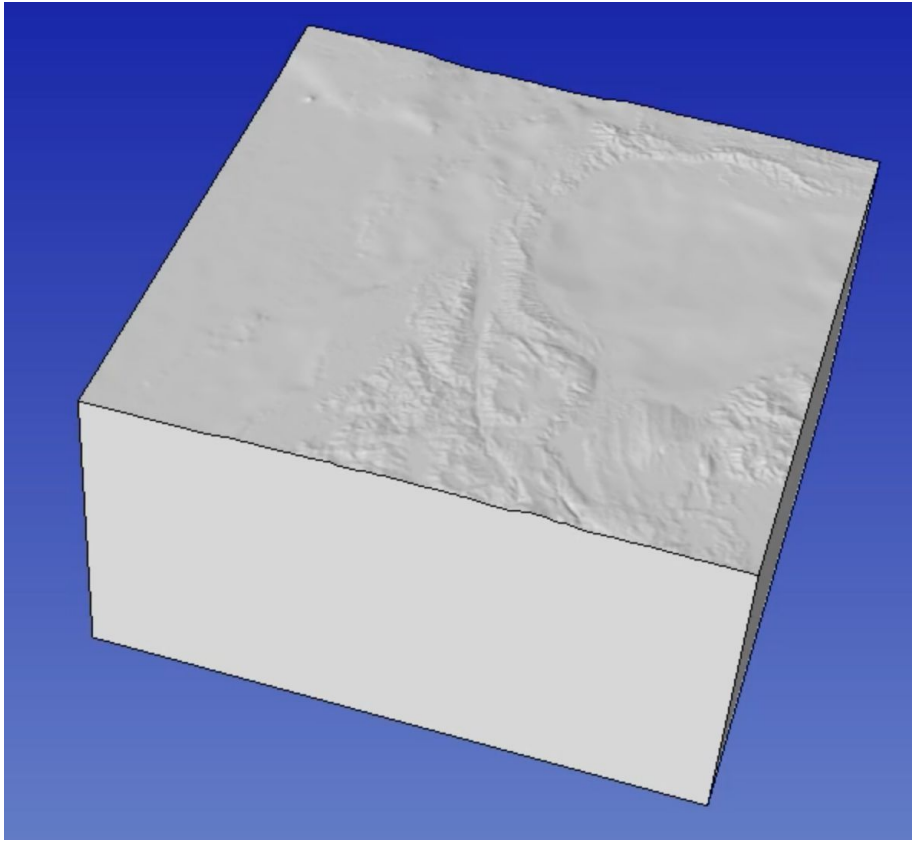


We then merge both models by using **'Union Parts'** in the **'Discrete'** tab and apply with tolerance 0.1.



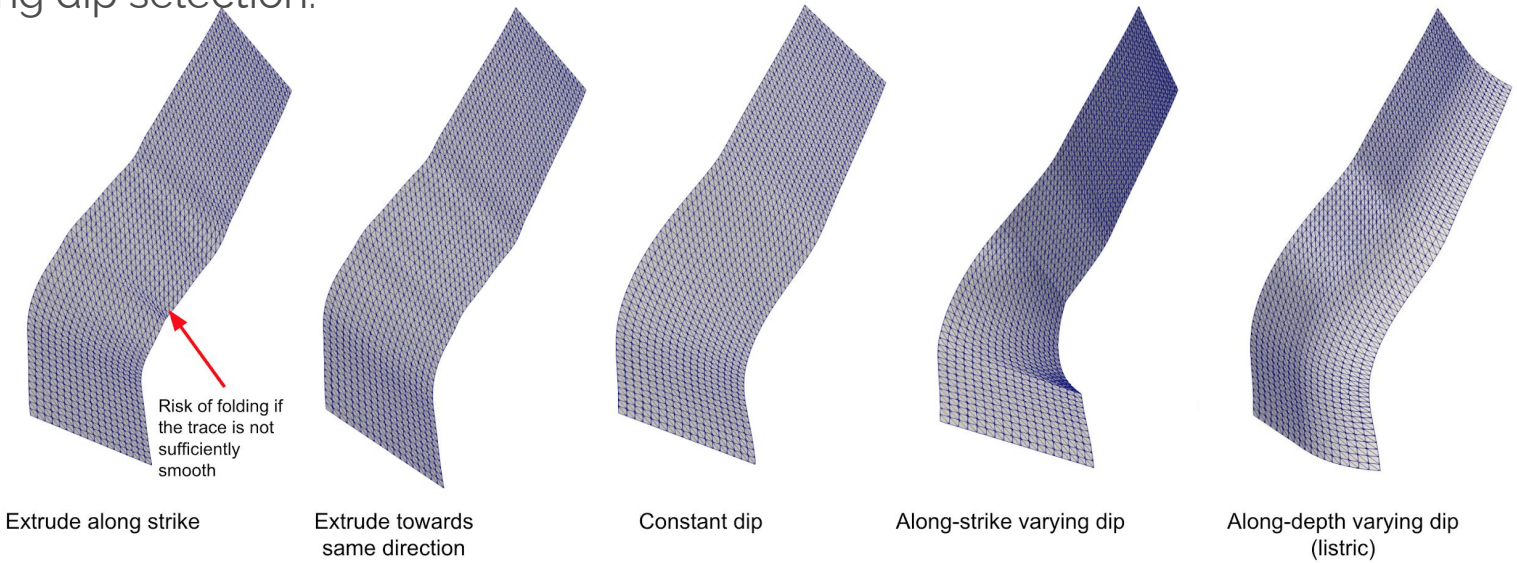


We then remove excess from both models using **‘Delete’** in the **‘Discrete’** tab.



Building faults from trace and dip

We generate the fault plane using `createFaultFromCurve.py`. Note that during building a fault model we need: smooth and resample fault trace, sweep trace towards z positive (because the topography can have positive z) and negative z, and constant or varying dip selection.



Application to Palu

The **'data'** folder contains (x,y,z) ASCII files describing fault traces and dip variation along-strike. The faults then are straightforwardly created using:

```
dx=0.5e3

python ~/SeisSol/Meshing/GocadRelatedScripts/createFaultFromCurve.py
data/segmentSouth_d90_long.dat 0 90 --dd $dx --maxdepth 16e3 --extend
4e3

python ~/SeisSol/Meshing/GocadRelatedScripts/createFaultFromCurve.py
data/smootherNorthBend.dat 0 65 --dd $dx --maxdepth 16e3 --extend 4e3

python ~/SeisSol/Meshing/GocadRelatedScripts/createFaultFromCurve.py
data/segmentBayAndConnectingFault.dat 2
data/segmentBayAndConnectingFaultDip.dat --dd $dx --maxdepth 16e3
--extend 4e3
```

Merging box and faults

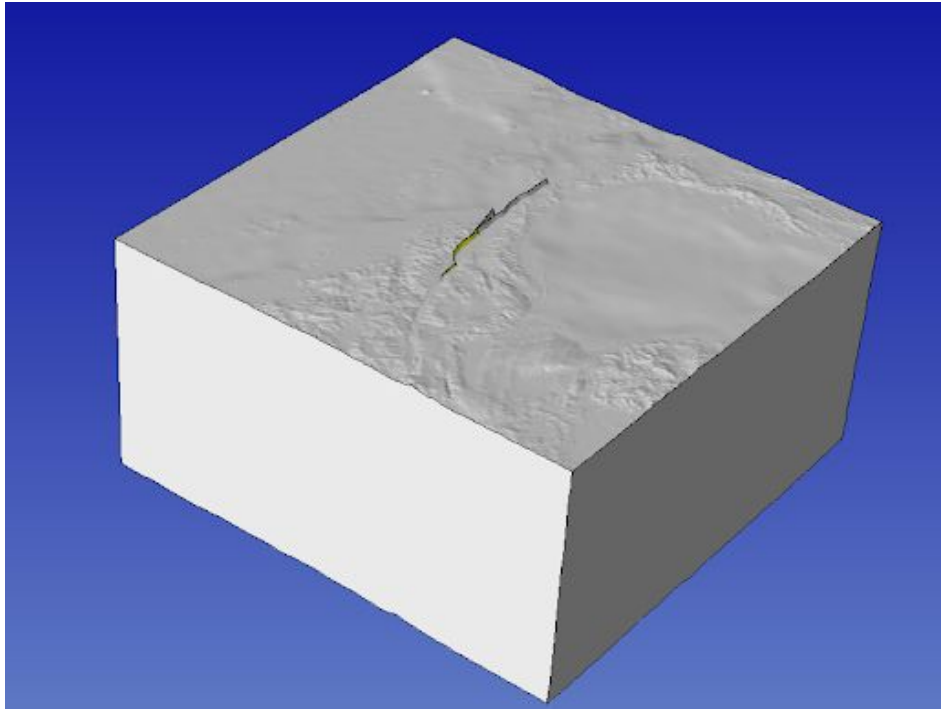
Let suppose that we know have 2 smd file, one with the intersected faults, the other with the box.

We can open one of them by:

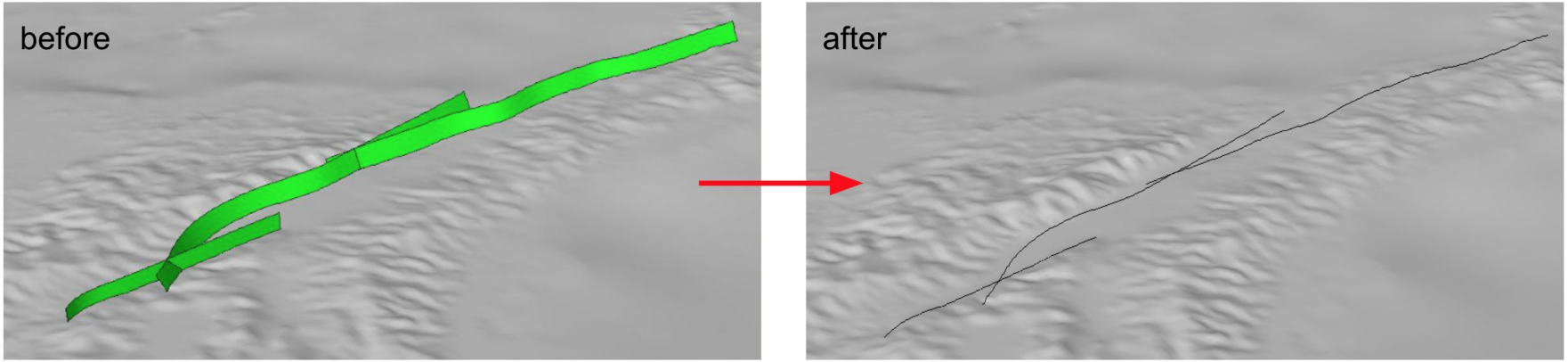
Discrete > Add parts > select the other.

Discrete > Union Parts > select both parts.

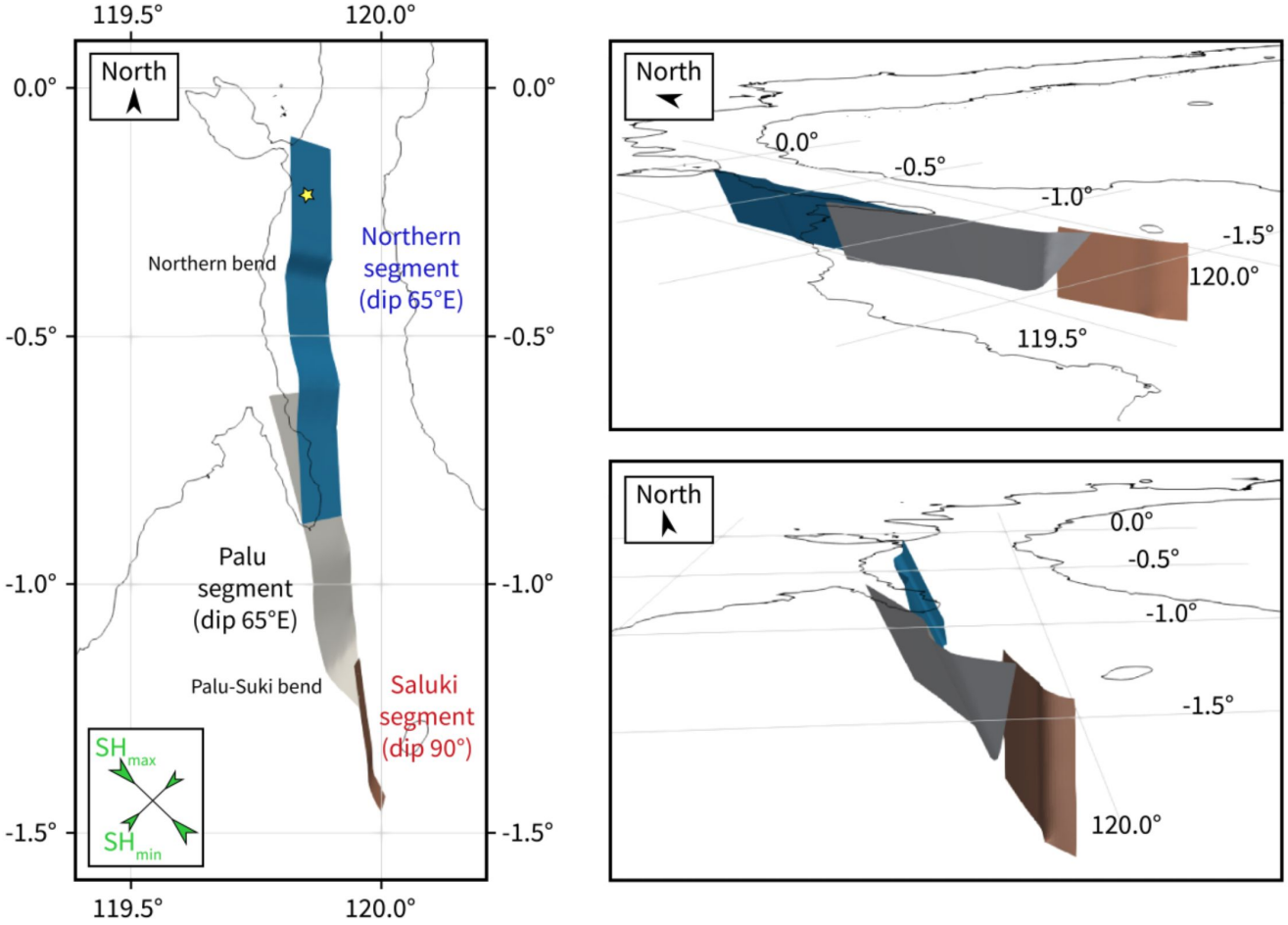
This intersects the faults with the topography.



Now we just have to delete the faults parts above the topography.

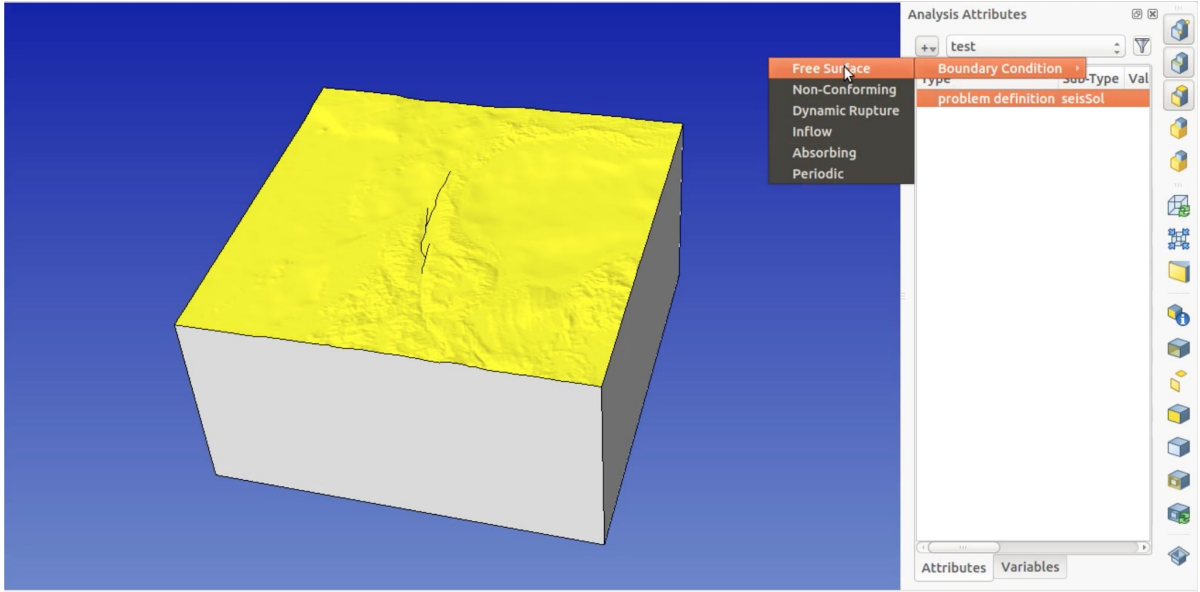


and the model is finished.

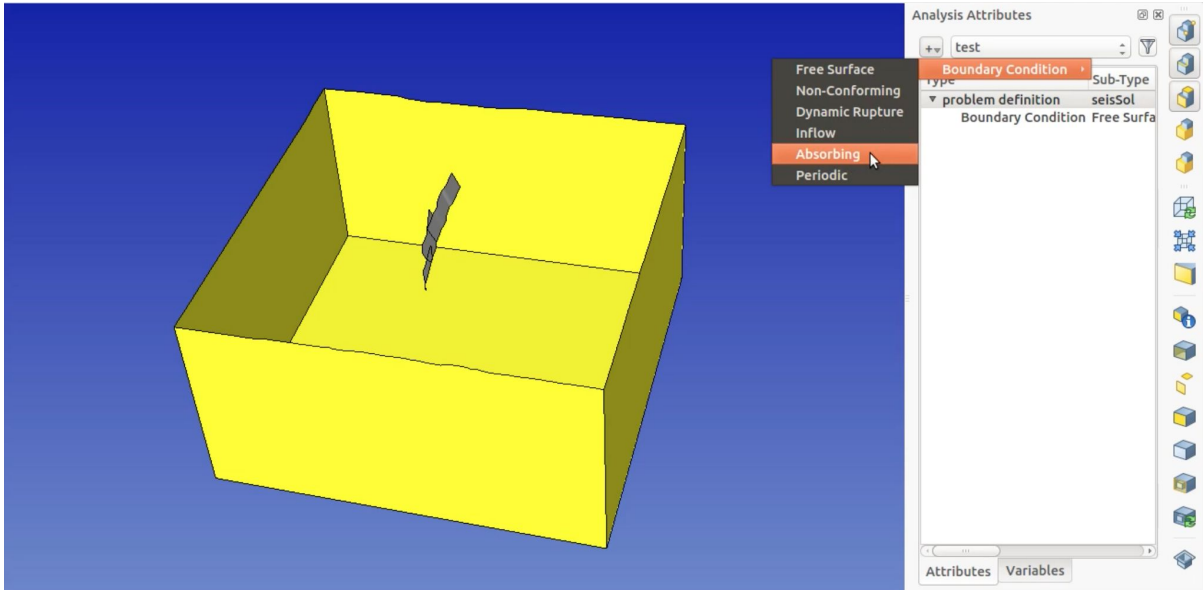

119.5°
120.0°
0.0°
-0.5°
-1.0°
-1.5°
North
Northern bend
Northern segment (dip 65°E)
Palu segment (dip 65°E)
Palu-Suki bend
Saluki segment (dip 90°)
SH max
SH min
North
0.0°
-0.5°
-1.0°
-1.5°
120.0°
119.5°
North
0.0°
-0.5°
-1.0°
-1.5°
120.0°

Assigning boundary conditions

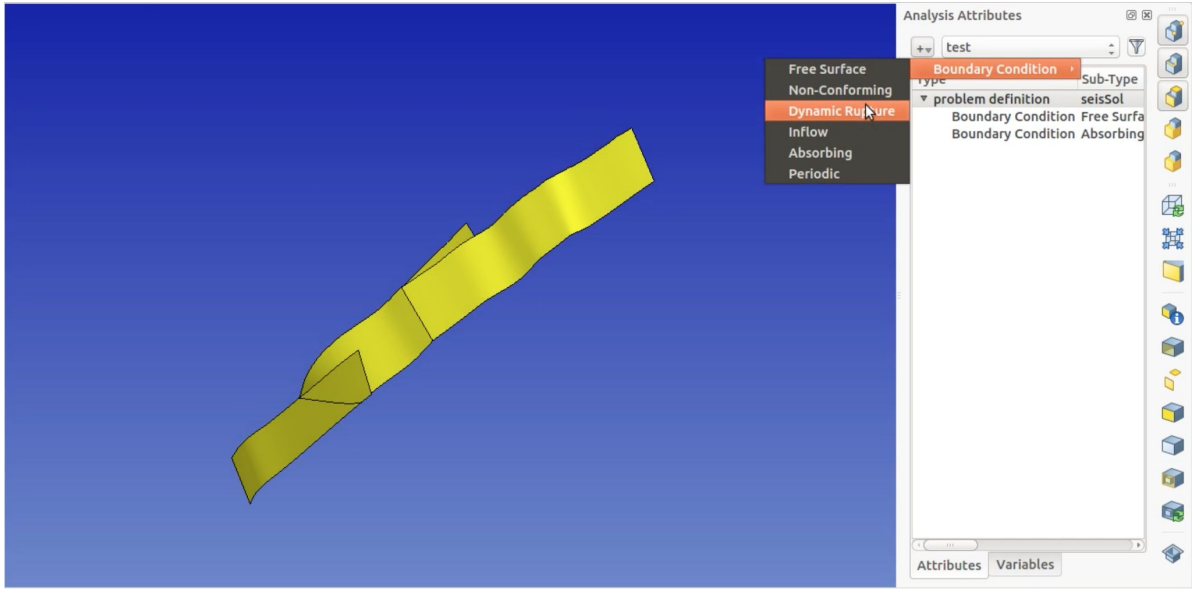
In SimModeler, select **'Analysis'** tab then select **'Analysis Attributes'.** We need to name a **'New Case …'** and select with **'SeisSol'** as the solver. Then select the top surface and assign it with **'Free Surface'** boundary condition.



We hide the top surface, then select all sides and bottom surfaces and assign them with **'Absorbing'** boundary condition.



We hide all sides and bottom surfaces, then select all fault segments and assign them with **‘Dynamic Rupture’** boundary condition.



Setting meshing parameters

We then Select **'Meshing'** tab and select **'Mesh Size'** to absolute 300 m within the fault segments.

Next we set **'Mesh Size'** to absolute 20000 m at the far side of our domain.

We also set **'Gradation Rate'** to 0.3.

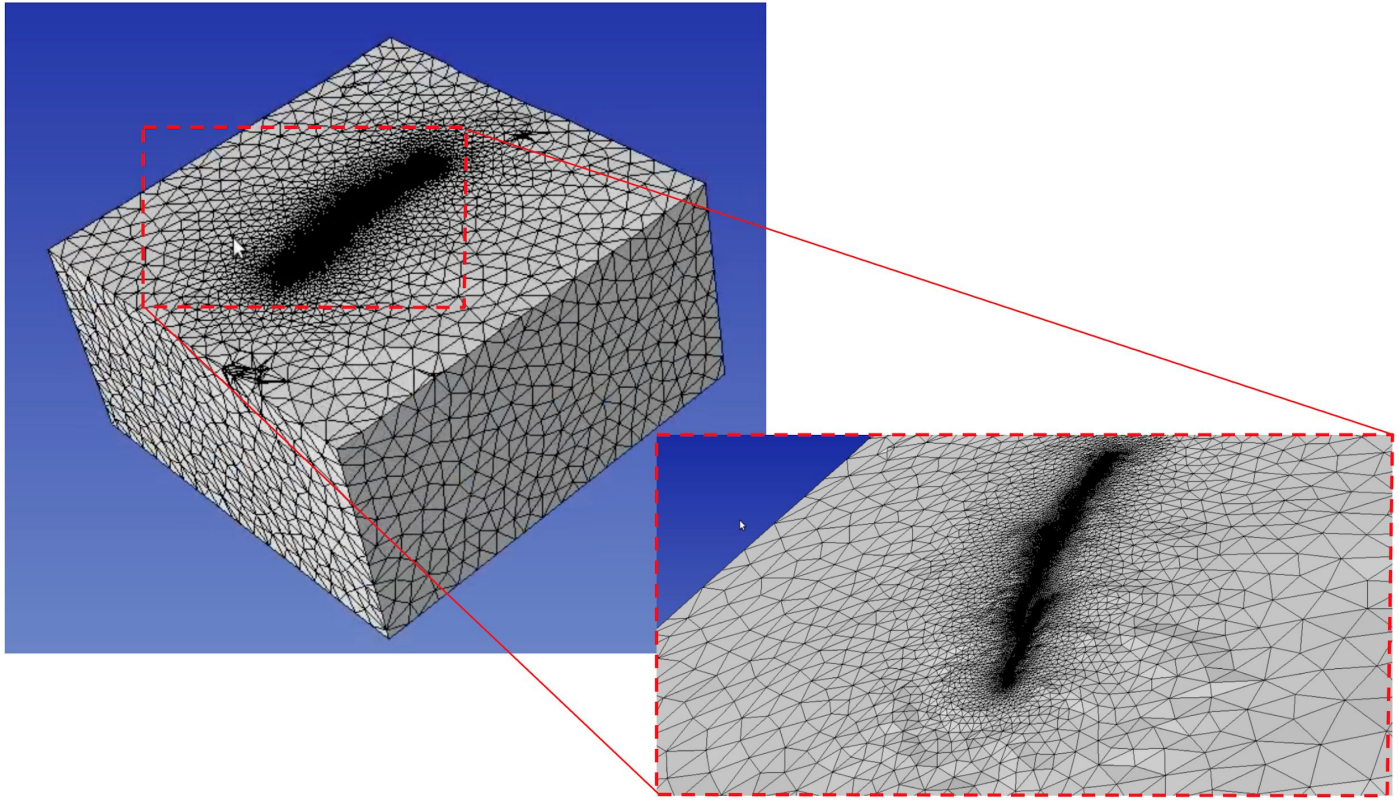
Then we set **'Surface Shape Metric'** with limiting aspect ratio to 6.0 and set **'Volume Shape Metric'** with limiting aspect ratio to 12.0.

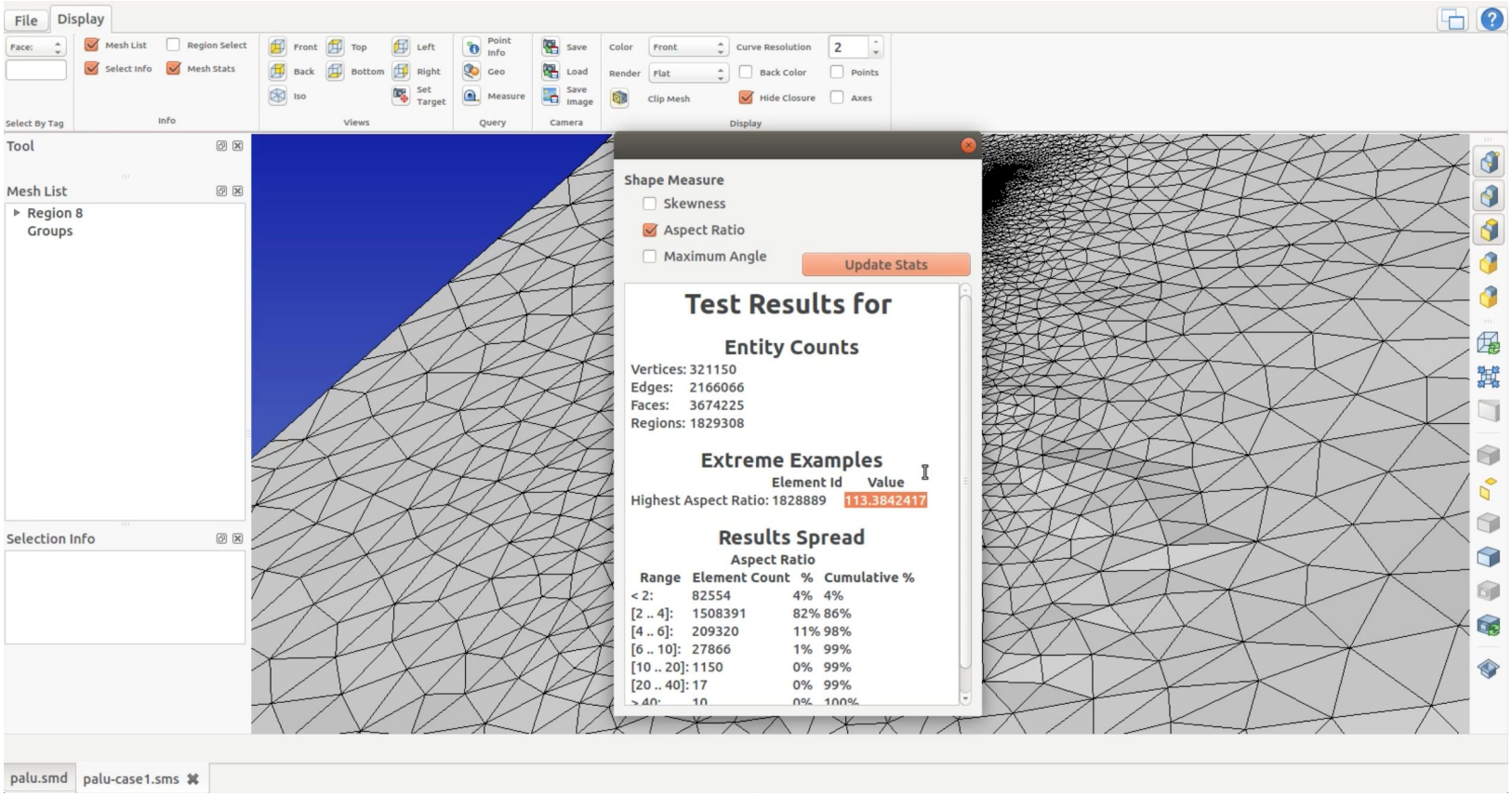
Then save (.smd) file and select **'Generate Mesh'** and start meshing.

Evaluating the mesh

After the meshing is completed, select **'Show Mesh'.**

We can see the mesh statistics by selecting **'Mesh Stats'.**




File
Display
Face:
Mesh List
Region Select
Select Info
Mesh Stats
Select By Tag
Info
Front
Top
Left
Back
Bottom
Right
Iso
Set Target
Views
Point Info
Geo
Measure
Query
Save
Load
Save image
Camera
Color
Front
Curve Resolution
2
Render
Flat
Back Color
Points
Clip Mesh
Hide Closure
Axes
Display
Tool
Mesh List
Region 8
Groups
Selection Info
Shape Measure
Skewness
Aspect Ratio
Maximum Angle
Update Stats
Test Results for
Entity Counts
Vertices: 321150
Edges: 2166066
Faces: 3674225
Regions: 1829308
Extreme Examples
Element Id Value
Highest Aspect Ratio: 1828889 113.3842417
Results Spread
Aspect Ratio
Range Element Count % Cumulative %
< 2: 82554 4% 4%
[2 .. 4]: 1508391 82% 86%
[4 .. 6]: 209320 11% 98%
[6 .. 10]: 27866 1% 99%
[10 .. 20]: 1150 0% 99%
[20 .. 40]: 17 0% 99%
palu.smd
palu-case1.sms

and then we save the generated mesh file (.neu).

Note that we recommend meshing using **PUMGen** for creating and exporting meshes in an efficient format (**PUML**). We can use the mesh analysis file (.smd) as the input or convert it from the ASCII mesh (.neu) file.