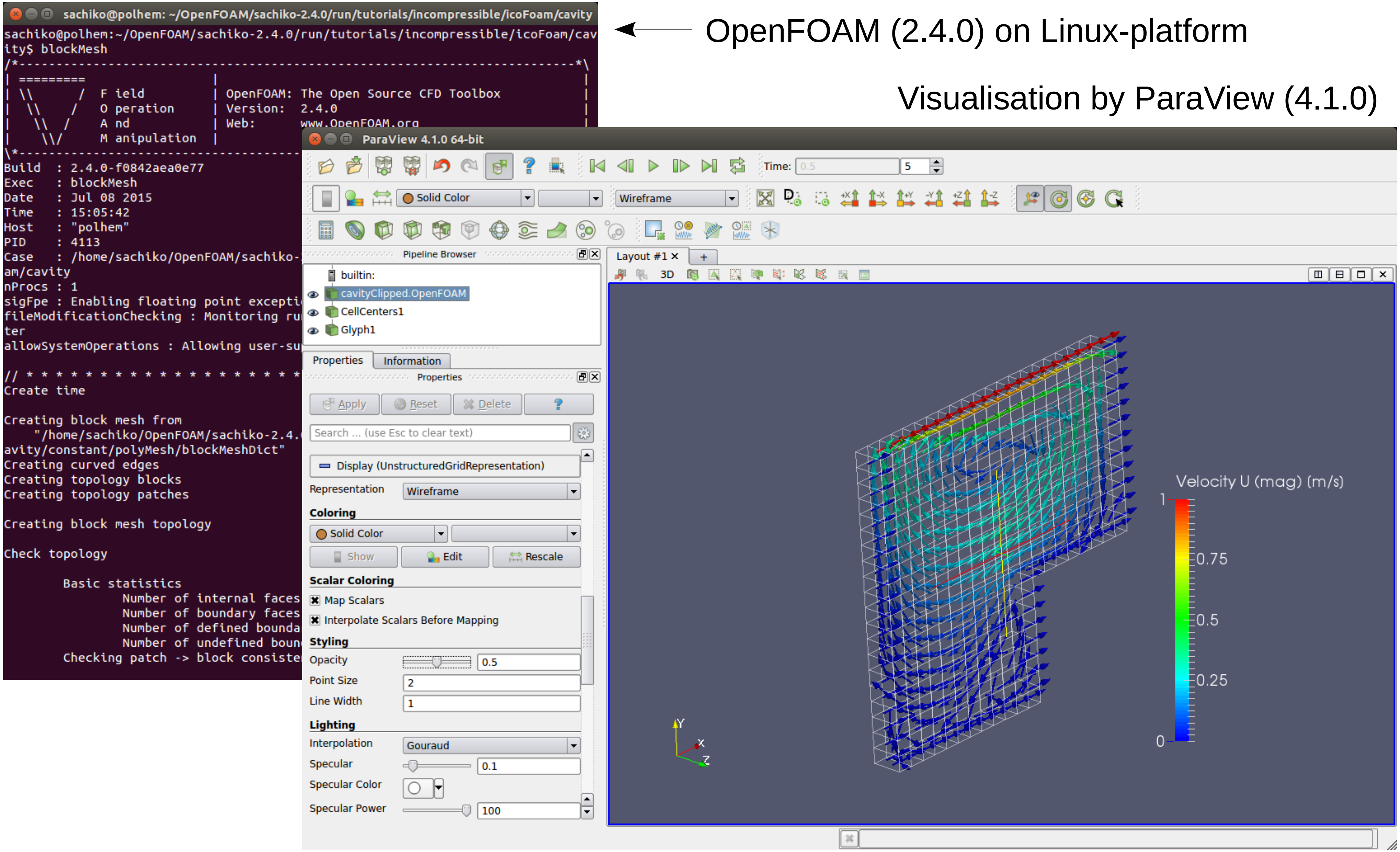
Simple CFD Simulations and Visualisation using OpenFOAM and ParaView

Sachiko Arvelius, PhD

Purpose of this presentation

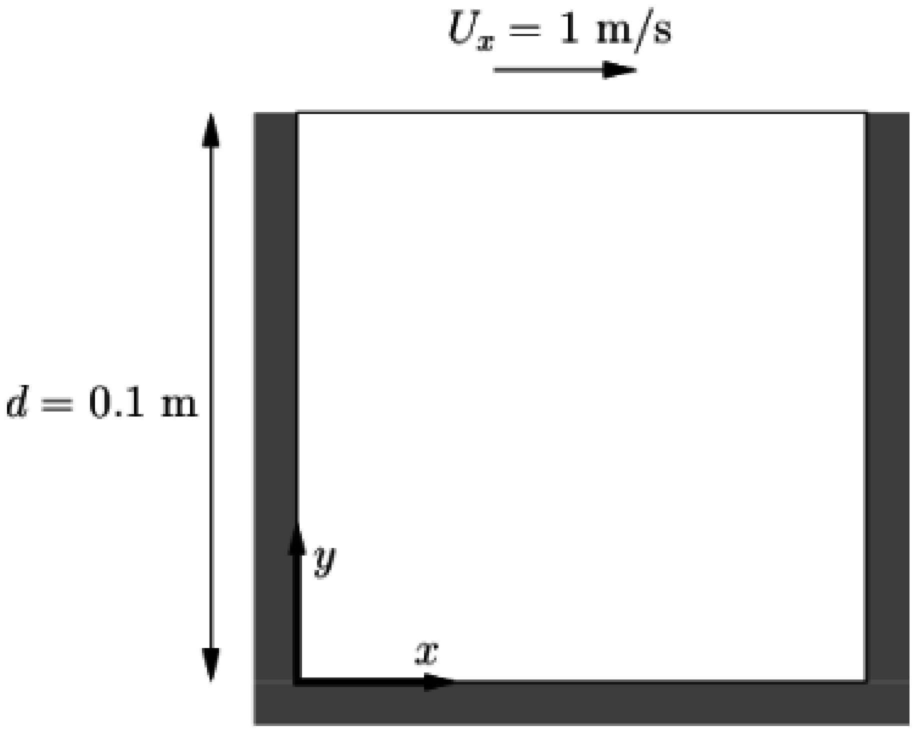
- To show my competence in CFD (Computational Fluid Dynamics) simulation and visualisation using **OpenFOAM** (http://www.openfoam.org/index.php) and **ParaView** (http://www.paraview.org/)
- To suggest potential hydrological applications by OpenFOAM



OpenFOAM (2.4.0) on Linux-platform

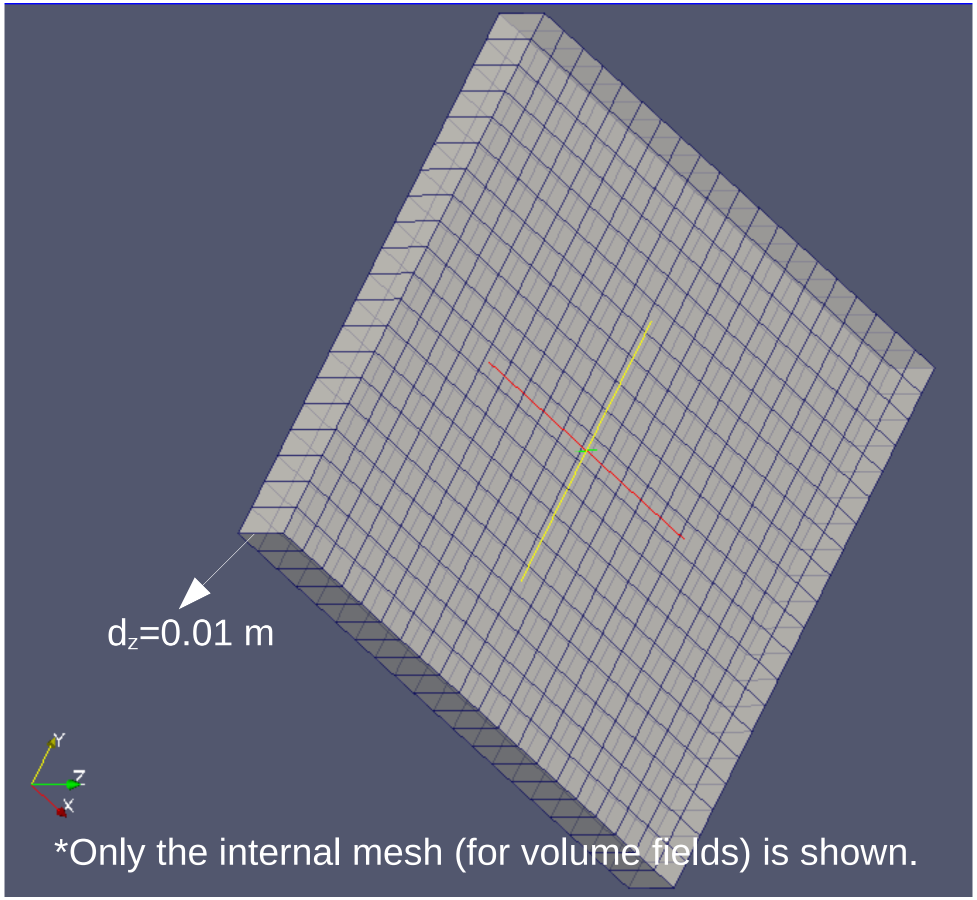
Visualisation by ParaView (4.1.0)

Simple case for CFD simulation – **Lid-driven cavity flow**



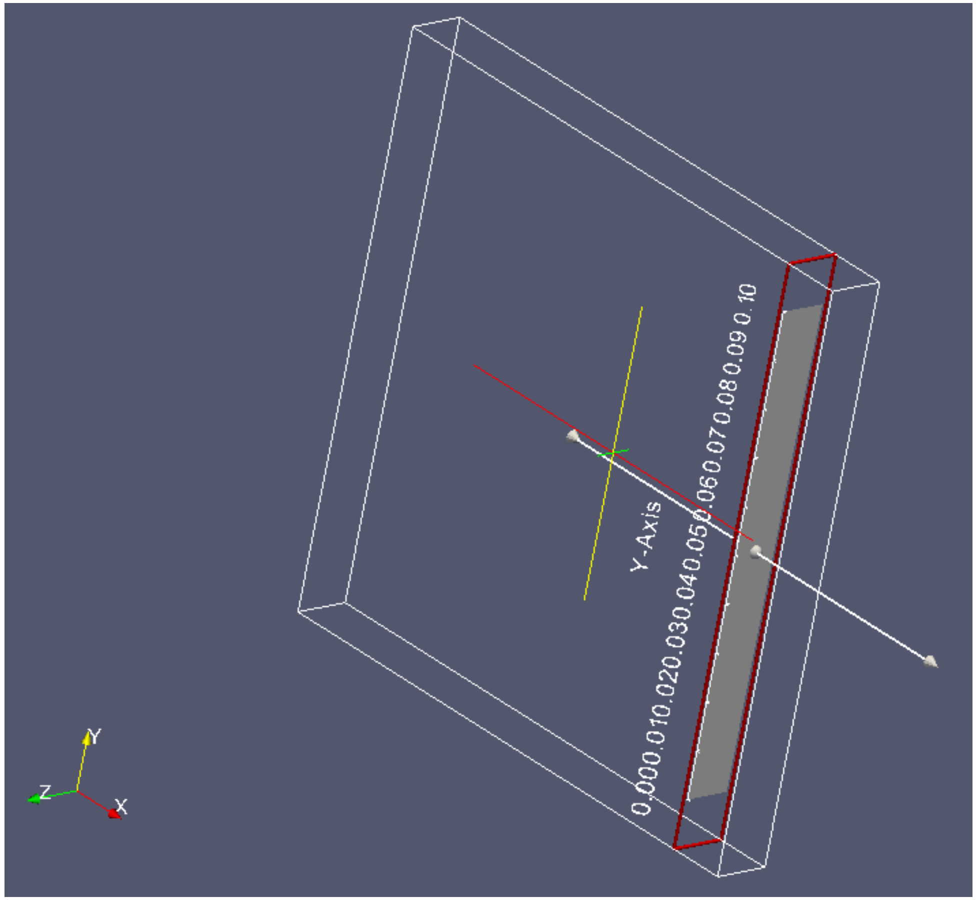
- … in a **two-dimensional square** domain.
- All the boundaries of the square are **walls**.
- The top wall moves in the x-direction at a speed of 1 m/s while the other 3 are stationary.
- The flow to be simulated is laminar (Re=10 – 10^2) to turbulent (Re=10^4), but isothermal and incompressible.

Set-ups for Lid-driven cavity flow



- Number of cells: 20x20x1 (so that the size of one cell is 0.005(m)x0.005(m)x0.01(m)).
- Mesh grading towards the wall is **not necessary** when using the standard *k-ε* model for turbulent flow (e.g. Re=10^4). Therefore, all the cases use the same geometry.
- Using Navier-Stokes equation **with only kinematic quantities**, i.e. no external force such as gravity (g).
- Changing Re (Reynolds Number) from **10 to 10^4** logarithmically.
- Using the **laminar model for Re≤10^3**.

Simulation Results

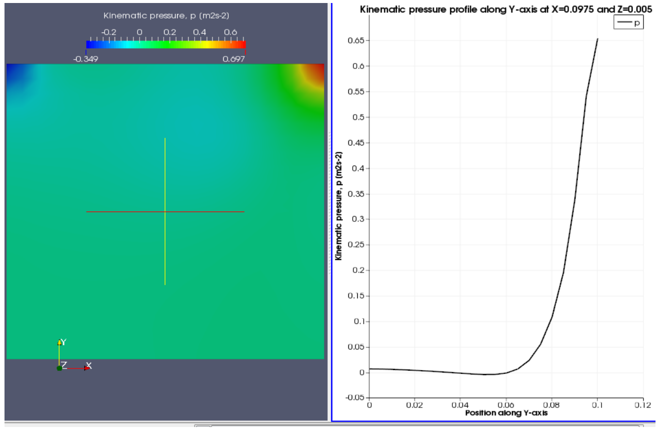


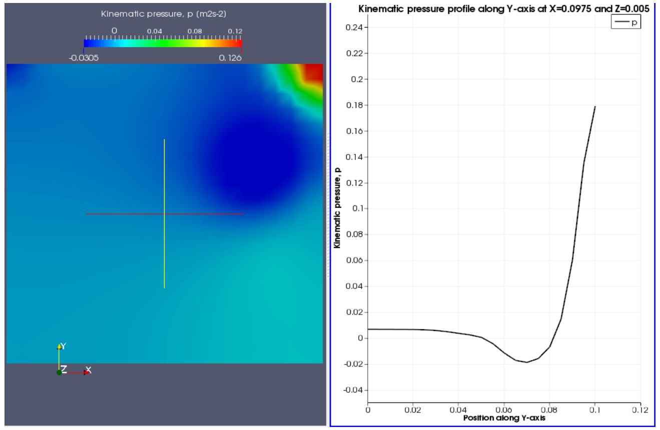
- Shown by the **distribution** and the **profile** (along Y-axis in a slice normal to X-direction, See the figure on the left side) of kinematic pressure (*$p=P/\rho_0$* and the unit is m^2s^{-2}).
- … because the velocity field converges faster than the pressure field.

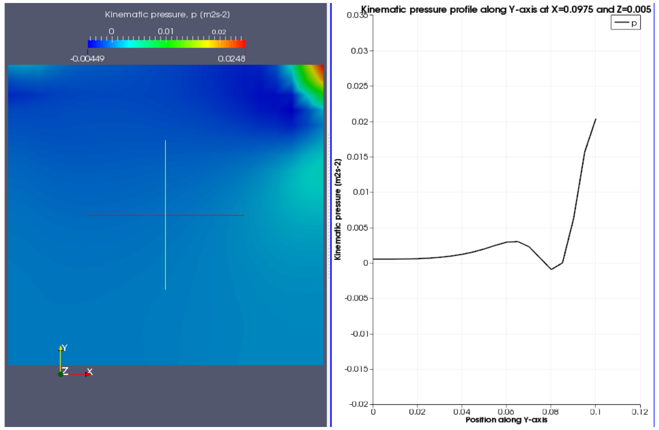
Kinematic pressure distribution and profile for the lid-driven cavity flow, laminar ($Re=10^2$) to turbulent ($Re=10^4$)

$Re=10^2$	$Re=10^3$	$Re=10^4$

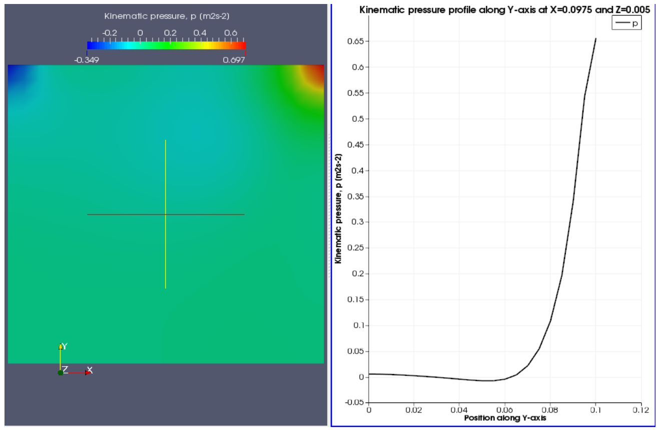
Time=0.5 s

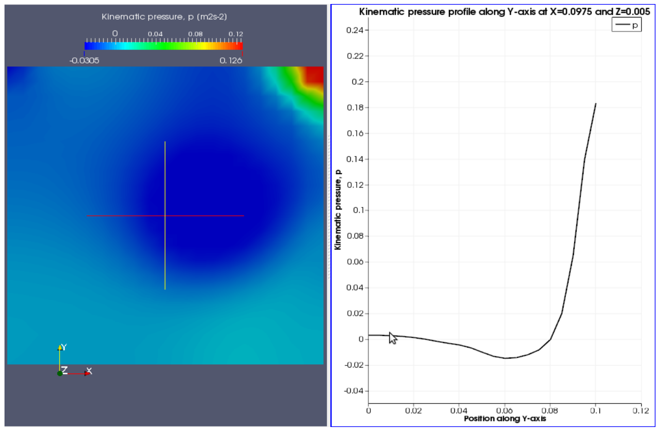


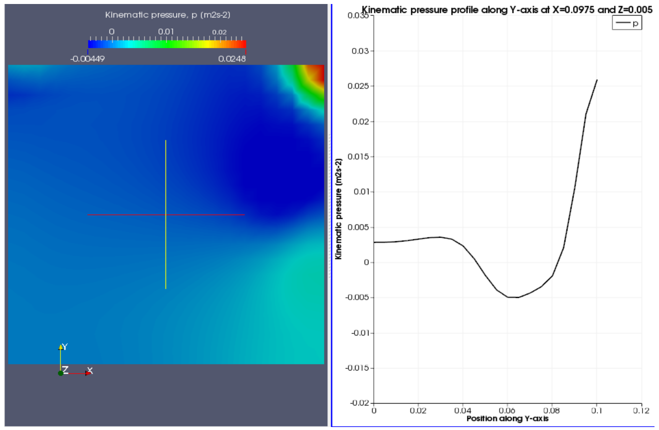




Time=1.0 s

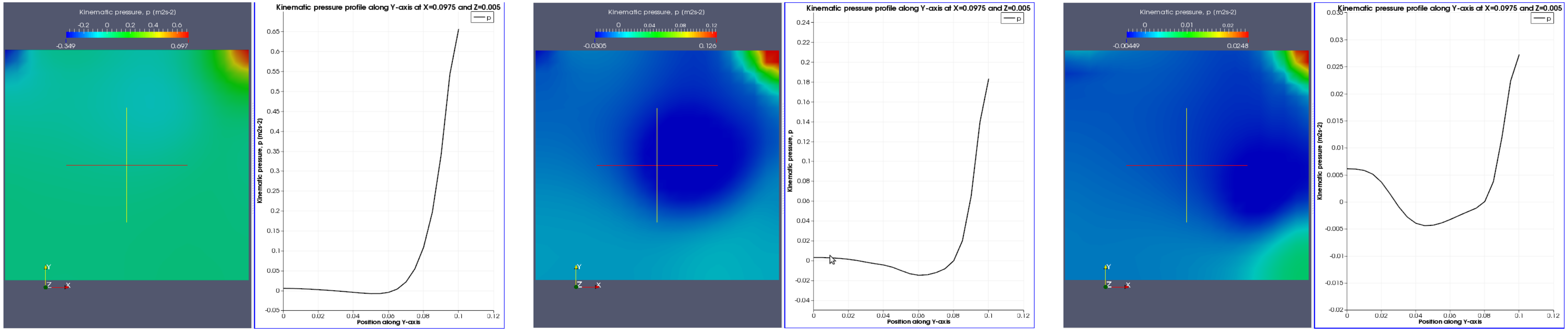




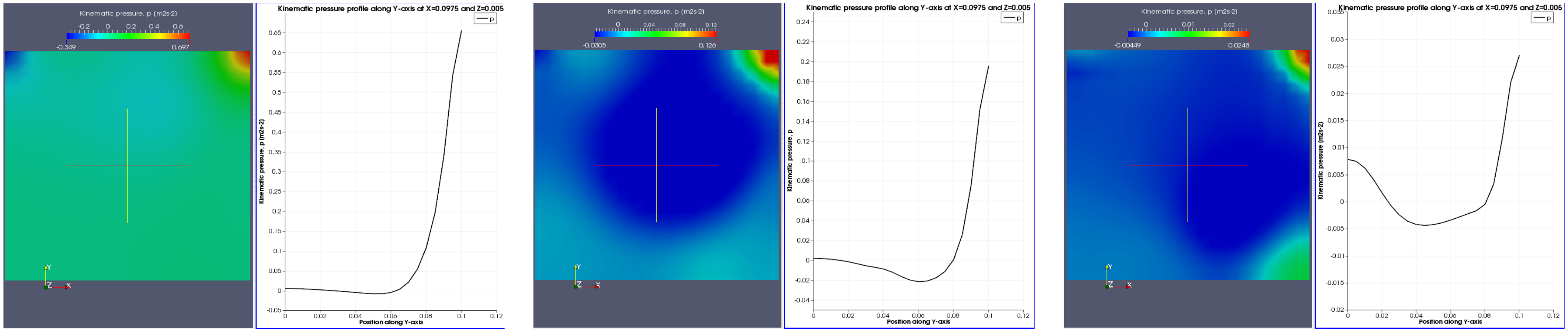


Re=10^2 Re=10^3 Re=10^4

Time=1.5 s

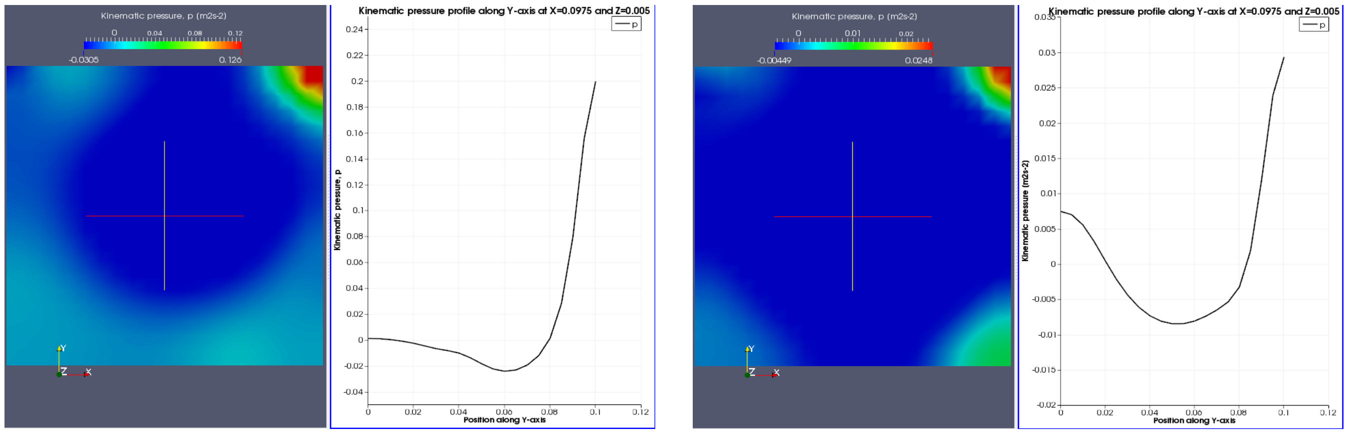


Time=2.0 s



The pressure field for Re=10^2 converged before Time=2.0 s. Therefore the run is terminated at Time=2.0 s.

Time=4.0 s

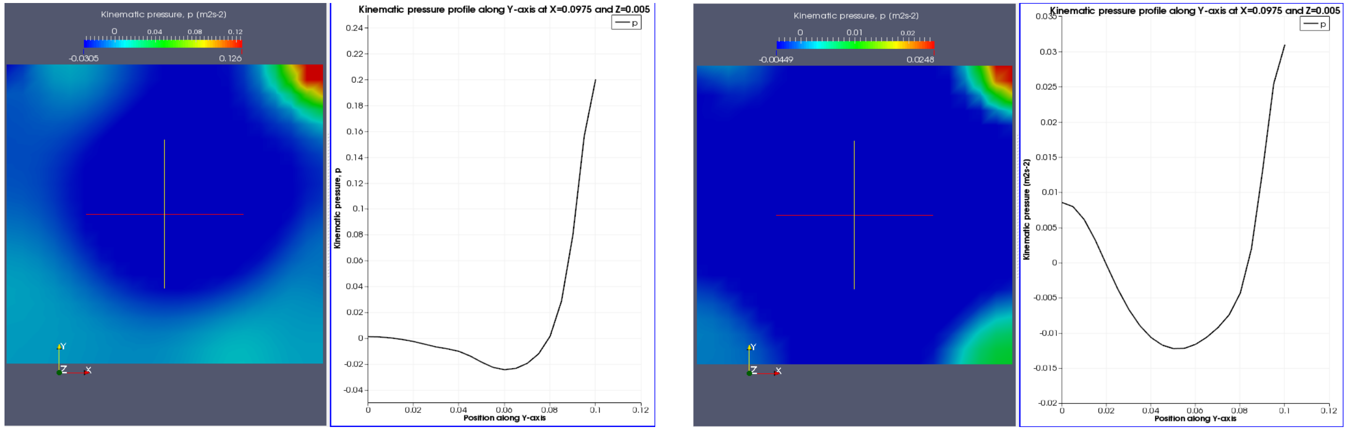


For the case of Re=10^3, a cavity started to develop nearby the top-right corner and shifted to the centre as time passes. The pressure field is fluctuating and never converges and thus the run is terminated at Time=8 s.

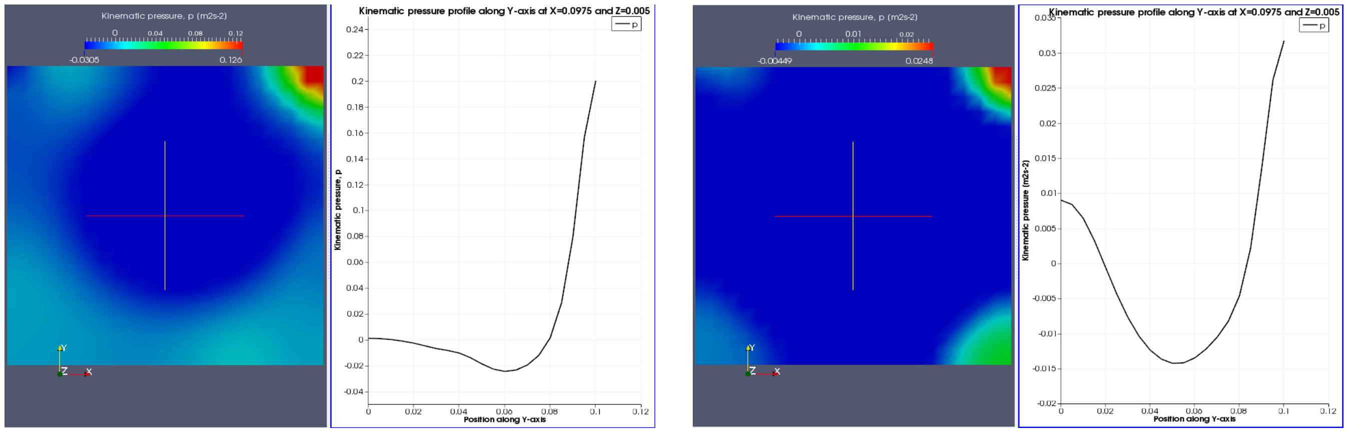
For the case of Re=10^4, a cavity is developing continuously because the shear energy (by moving lid) which is converted into turbulent kinetic energy is supplied continuously.

Re=10^3 | Re=10^4

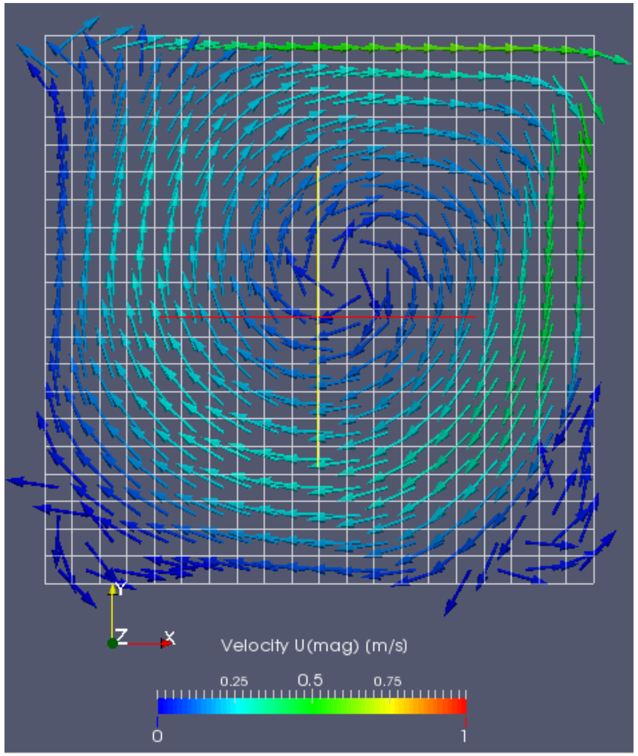
Time=6.0 s

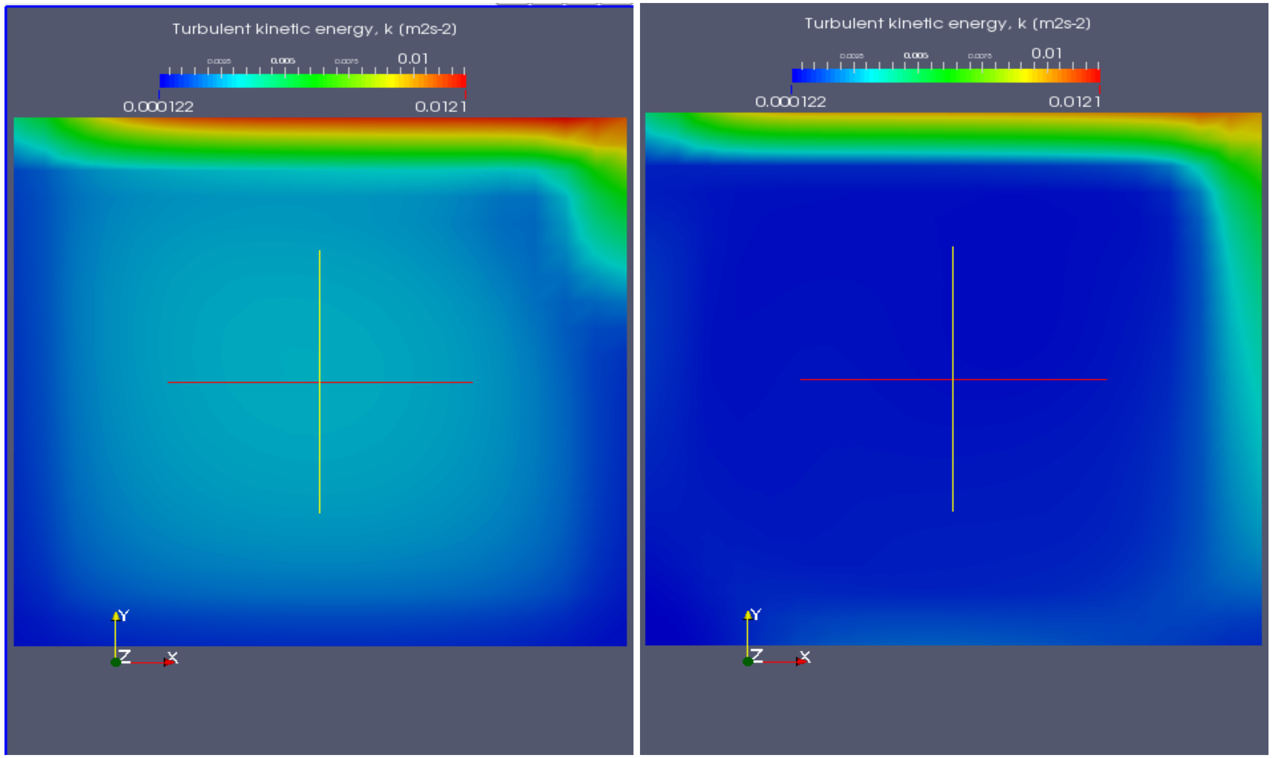


Time=8.0 s



Simulation Results

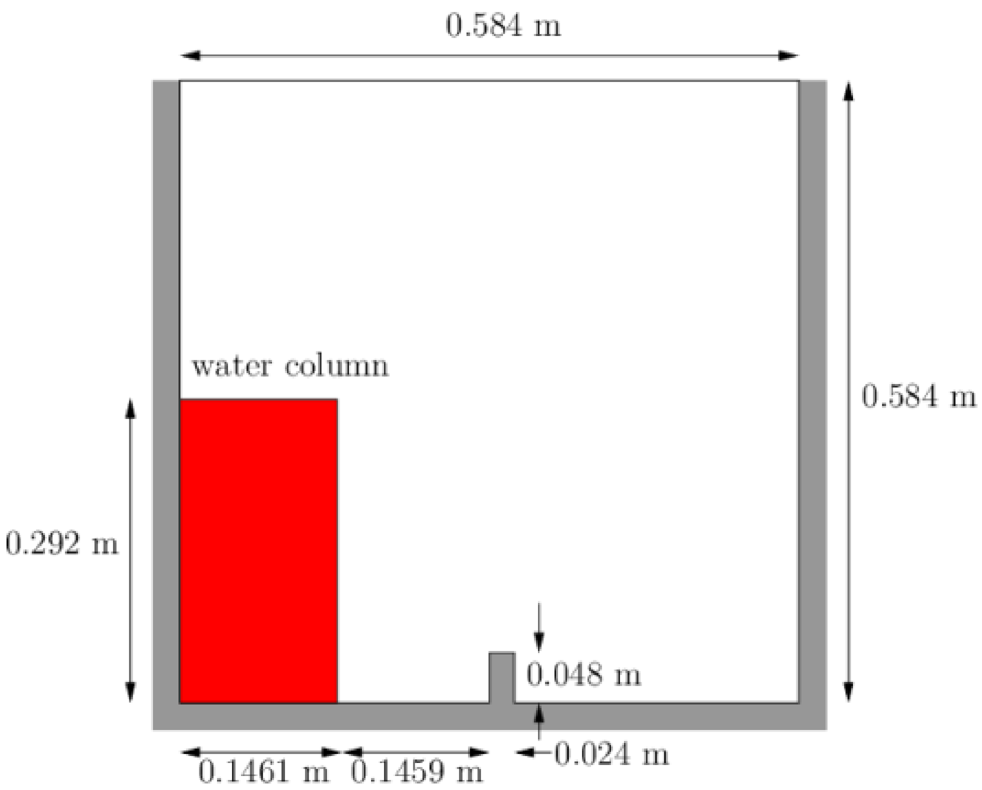




Converged velocity field for the case of Re=10^4. The velocity field converged at around Time=4.9 s.

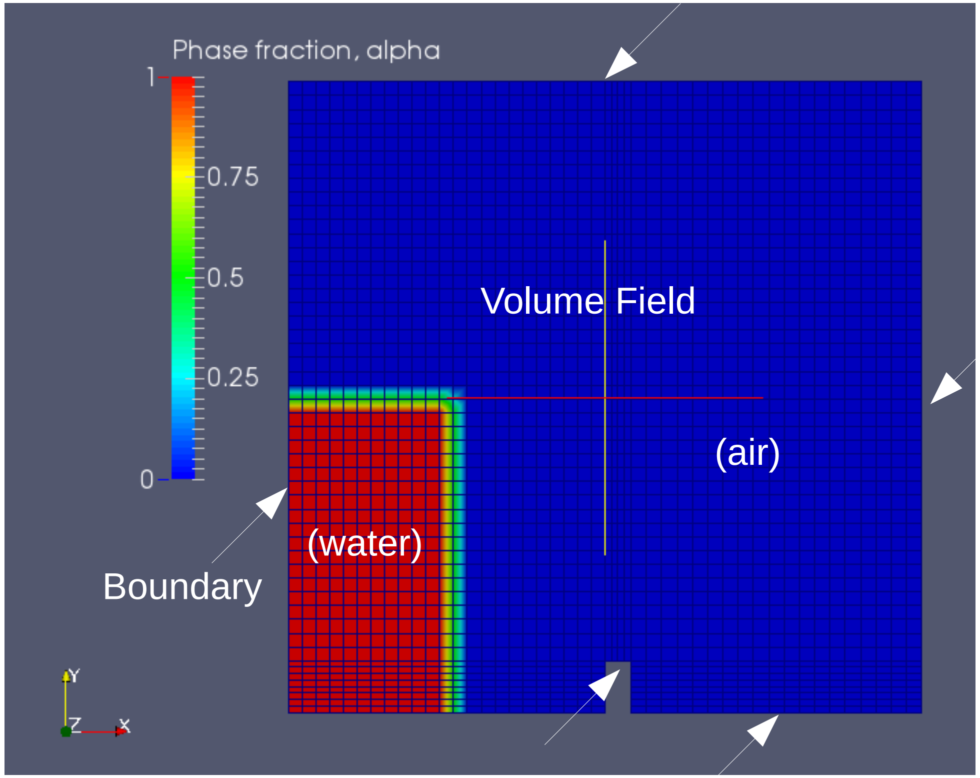
The distribution of the turbulent kinetic energy (k [m^2s^{-2}]) at Time=0.5 s (left) and Time=4.0 s (right). As seen in the figures, The energy generation takes place continuously nearby the right-hand side wall (close to X=0.1).

VOF and **Parallel Processing** for CFD simulation – **Dam break**



- A transient flow of **two fluids** (**water** and **air**) separated by a sharp interface.
- **VOF** is the volume of fluid method in which a specie transport equation is used to determine **the relative volume fraction of the two phases**, or phase fraction (α), **in each computational cell**.

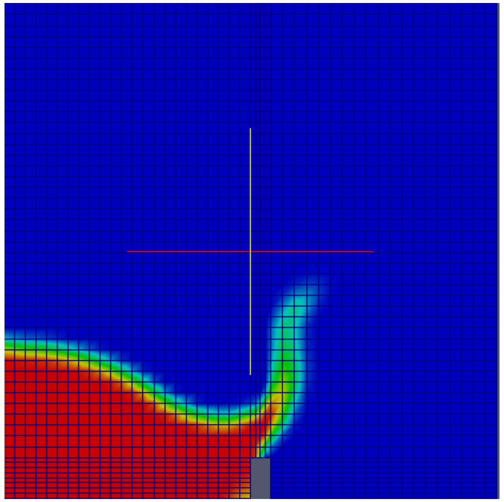
Phase fraction (α) in VOF



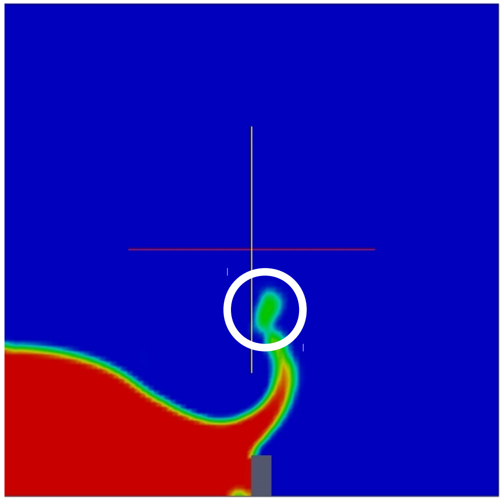
- The computational **volume field** is initially set as **non-uniform**, i.e. **water** and **air**, and the interface between them is separated sharply by phase fraction (α).
- The phase fraction can have any value between 0 and 1, and **0 means air and 1 does water in this case**.
- An interface between the species is not explicitly computed by the VOF method. Therefore **α=0.65 means that water occupies the volume of the cell by 65%**.
- The transitional colours (i.e. colours for 0<α<1, neither red nor blue) are due to the interpolation.

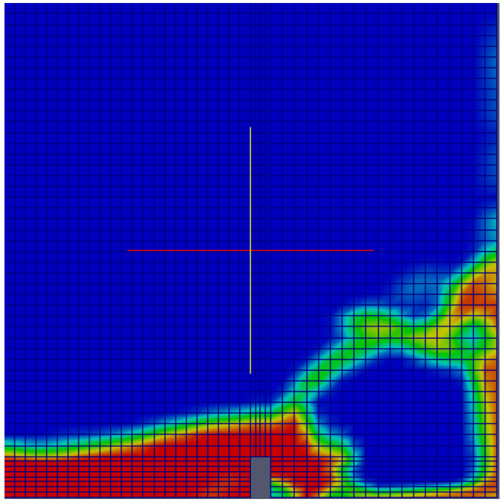
Comparison in terms of **mesh resolutions**, of a transient flow of two fluids

finer resolution

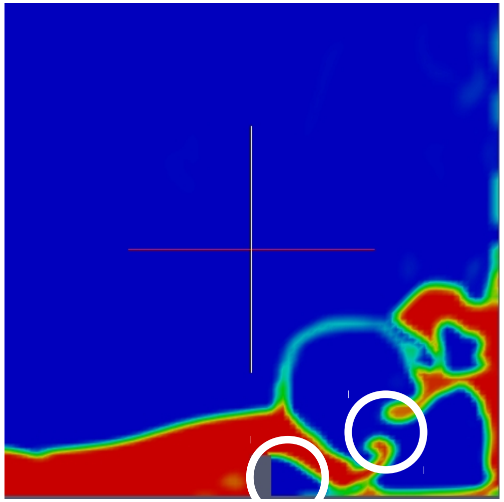


Time=0.2 s.
The water hits the obstacle and collapses. The bounce of the water in the finer resolution has a detailed structure interpreted as splash.



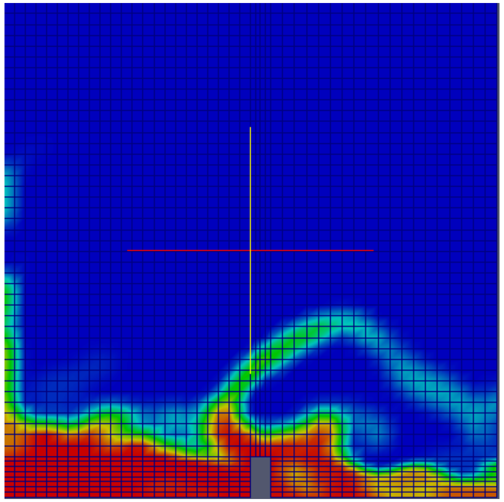


Time=0.6 s.
The behaviour of the water after collapsing is much complicated but sharp for the finer resolution. One can see a clear void in the opposite side of the obstacle in the finer resolution case.

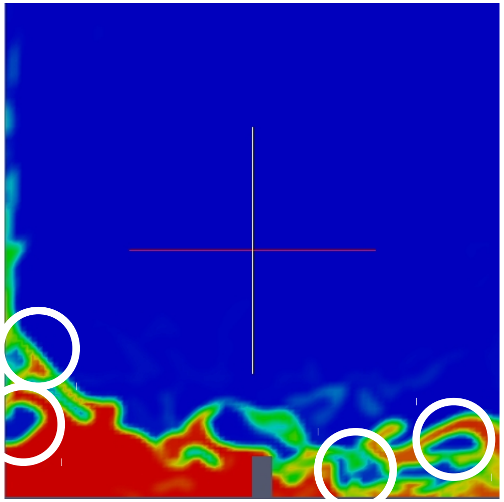


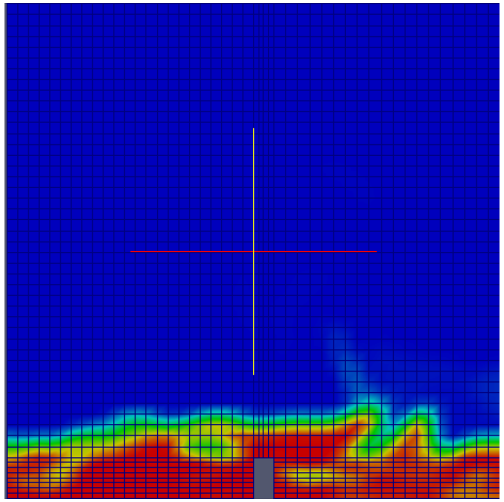
Comparison in terms of **mesh resolutions**, of a transient flow of two fluids (continued)

finer resolution

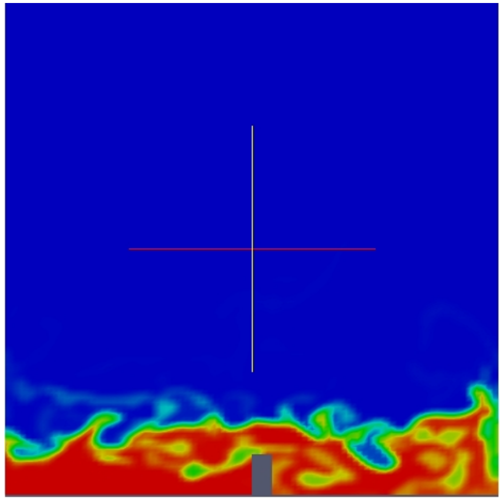


Time=1.0 s.
The bounce-off sharpness of the returning wave is different: the shaper, the finer. There are still several voids seen in the finer resolution case.





Time=1.6 s.
The free surface of the water, i.e. the interface between water and air, is still complicated for the finer resolution case.



Run Time and Parallel Processing

```
Courant Number mean: 0.160146 max: 0.9648
Interface Courant Number mean: 0.0106555 max: 0.417974
deltaT = 0.00287879
Time = 2

PIMPLE: iteration 1
smoothSolver:  Solving for alpha.water, Initial residual = 0.00159868, Final residual = 5.1263e-09, No Iterations
3
Phase-1 volume fraction = 0.127393  Min(alpha.water) = 0  Max(alpha.water) = 1
MULES: Correcting alpha.water
MULES: Correcting alpha.water
Phase-1 volume fraction = 0.127393  Min(alpha.water) = 0  Max(alpha.water) = 1
DICPCG:  Solving for p_rgh, Initial residual = 0.0112584, Final residual = 0.000337618, No Iterations 2
time step continuity errors : sum local = 0.000275361, global = -8.50229e-07, cumulative = 0.00288109
DICPCG:  Solving for p_rgh, Initial residual = 0.0004304, Final residual = 2.06183e-05, No Iterations 8
time step continuity errors : sum local = 1.67787e-05, global = -2.96018e-07, cumulative = 0.0028808
DICPCG:  Solving for p_rgh, Initial residual = 7.01197e-05, Final residual = 8.69268e-08, No Iterations 37
time step continuity errors : sum local = 7.06489e-08, global = 2.17675e-09, cumulative = 0.0028808
ExecutionTime = 13.03 s  ClockTime = 13 s

End
```

The Courant number is variable because that the meshes are non-uniform and the cells are fractionated.

The number of iterations (`No Iterations`) has no meaning for simulating a transient flow.

The above is a run log for the dam break simulation. The number of cells for this simulation is **2268** and it took **13.03 s** for the 2-second simulation. Very simply, the execution time per mesh took **5.75 ms**.

Run Time and Parallel Processing (continued)

```
Courant Number mean: 0.206533 max: 0.904299
Interface Courant Number mean: 0.0246906 max: 0.904299
deltaT = 0.00227741
Time = 2

PIMPLE: iteration 1
smoothSolver:  Solving for alpha.water, Initial residual = 0.00348366, Final residual = 1.90826e-09, No Iterations 4
Phase-1 volume fraction = 0.121584  Min(alpha.water) = 0  Max(alpha.water) = 1
MULES: Correcting alpha.water
MULES: Correcting alpha.water
Phase-1 volume fraction = 0.121584  Min(alpha.water) = 0  Max(alpha.water) = 1
DICPCG:  Solving for p_rgh, Initial residual = 0.0400558, Final residual = 0.000788878, No Iterations 2
time step continuity errors : sum local = 0.000596737, global = 6.05325e-07, cumulative = 0.00047935
DICPCG:  Solving for p_rgh, Initial residual = 0.000906242, Final residual = 4.4466e-05, No Iterations 10
time step continuity errors : sum local = 3.32898e-05, global = 4.11157e-08, cumulative = 0.000479391
DICPCG:  Solving for p_rgh, Initial residual = 0.000140724, Final residual = 9.82252e-08, No Iterations 76
time step continuity errors : sum local = 7.35061e-08, global = 1.7932e-09, cumulative = 0.000479393
ExecutionTime = 92.95 s  ClockTime = 93 s

End
```

The above is a run log for the dam break simulation **with the higher mesh resolution**. The number of cells for this simulation is **7700** and it took **92.95 s** for the 2-second simulation. Very simply, the execution time per mesh took **12.07 ms** and **it took almost double time** compared to the previous case.

Run Time and Parallel Processing (continued)

The CPU specification of the machine is

Intel® Core™ i7-2620M(@2.70GHz) x 2 cores (x 2 threads)

Therefore it is possible to execute 4-CPU parallel processing.
The below is the beginning of a run log for multi (e.g. 4 for this case) parallel processing.

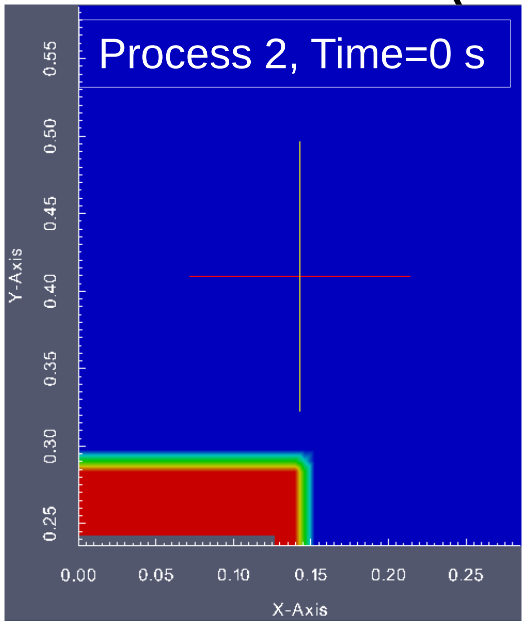
```
/*---------------------------------------------------------------------------*\
| =========                 |                                                 |
| \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox           |
|  \\    /   O peration     | Version:  2.4.0                                 |
|   \\  /    A nd           | Web:      www.OpenFOAM.org                      |
|    \\/     M anipulation  |                                                 |
\*---------------------------------------------------------------------------*/
Build  : 2.4.0-f0842aea0e77
Exec   : interFoam -parallel
Date   : [redacted]
Time   : [redacted]
Host   : [redacted]
PID    : 1162
Case   : [redacted]/interFoam/laminar/damBreakFine
nProcs : 4
Slaves :
3
(
"[redacted]1163"
"[redacted]1164"
"[redacted]1165"
)

Pstream initialized with:
    floatTransfer      : 0
    nProcsSimpleSum    : 0
    commsType          : nonBlocking
    polling iterations : 0
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster
allowSystemOperations : Allowing user-supplied system call operations
```

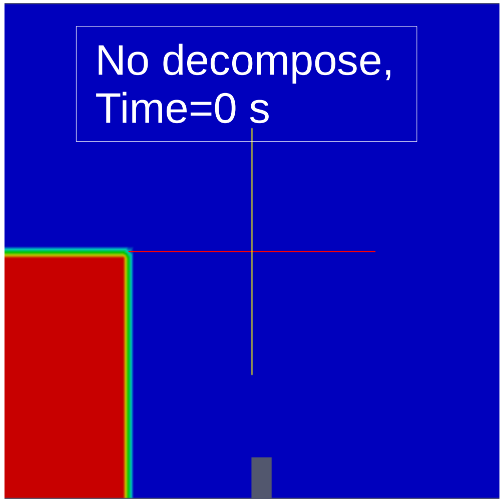
Run Time and Parallel Processing (continued)

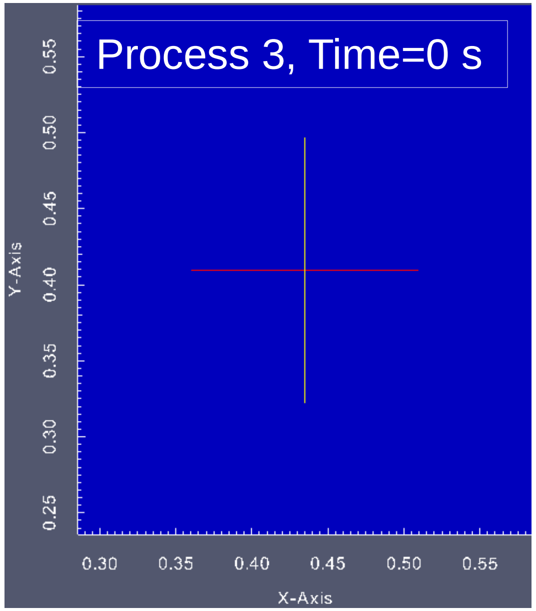
```
Courant Number mean: 0.196853 max: 0.885829
Interface Courant Number mean: 0.01863 max: 0.663802
deltaT = 0.0018679
Time = 2

PIMPLE: iteration 1
smoothSolver:  Solving for alpha.water, Initial residual = 0.00322972, Final residual = 9.11648e-10, No Iterations 4
Phase-1 volume fraction = 0.121654  Min(alpha.water) = -3.14759e-10  Max(alpha.water) = 1
MULES: Correcting alpha.water
MULES: Correcting alpha.water
Phase-1 volume fraction = 0.121654  Min(alpha.water) = -3.6111e-08  Max(alpha.water) = 1
DICPCG:  Solving for p_rgh, Initial residual = 0.0480139, Final residual = 0.00135963, No Iterations 2
time step continuity errors : sum local = 0.00070948, global = 2.05648e-07, cumulative = 0.000397546
DICPCG:  Solving for p_rgh, Initial residual = 0.00153384, Final residual = 7.52928e-05, No Iterations 10
time step continuity errors : sum local = 3.8974e-05, global = 6.54736e-07, cumulative = 0.000398201
DICPCG:  Solving for p_rgh, Initial residual = 0.000230692, Final residual = 9.06872e-08, No Iterations 79
time step continuity errors : sum local = 4.69235e-08, global = -1.51287e-09, cumulative = 0.0003982
ExecutionTime = 63.33 s  ClockTime = 65 s

End

Finalising parallel run
```

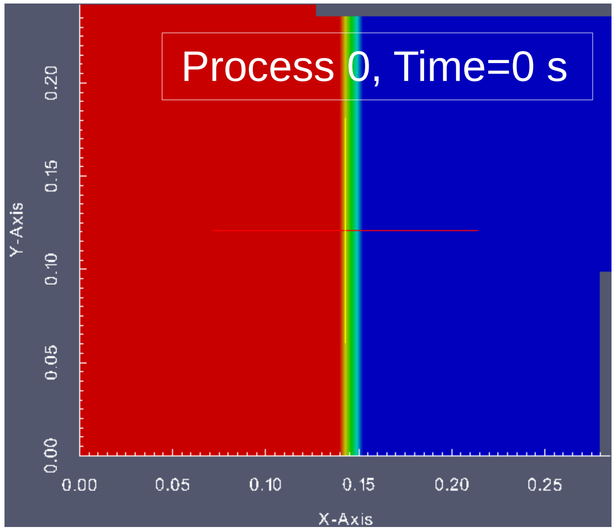
The above is a run log for the dam break simulation **with the higher mesh resolution**. The number of meshes for this simulation is **7700** and it took only **63.33 s** for the 2-second simulation, and **reducing about 32%** of the execution time thanks to the 4-CPU parallel processing.

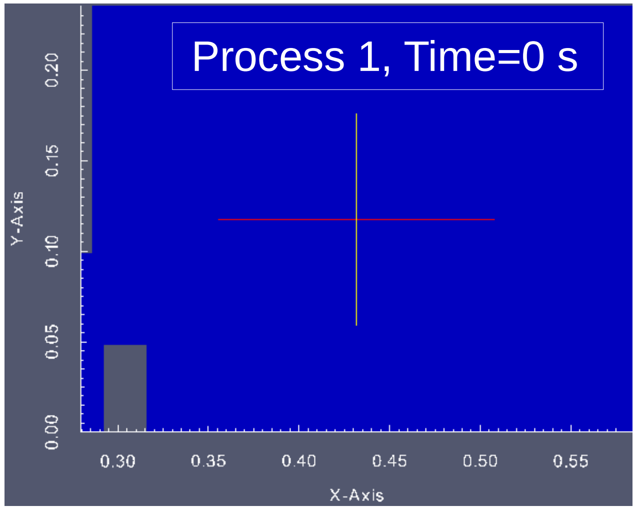
Comparison between single-processed and multi-processed (**domain decomposition**)



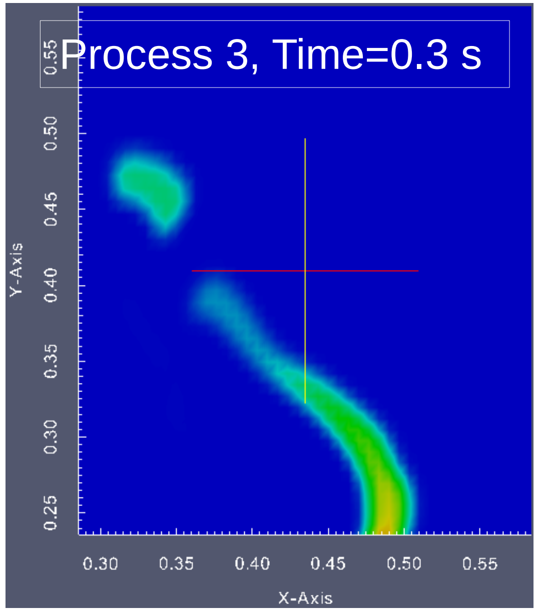


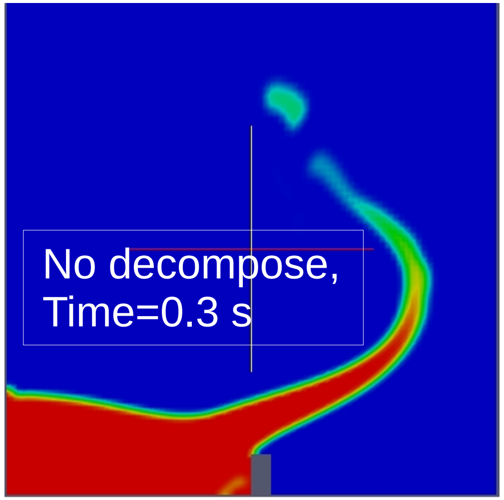


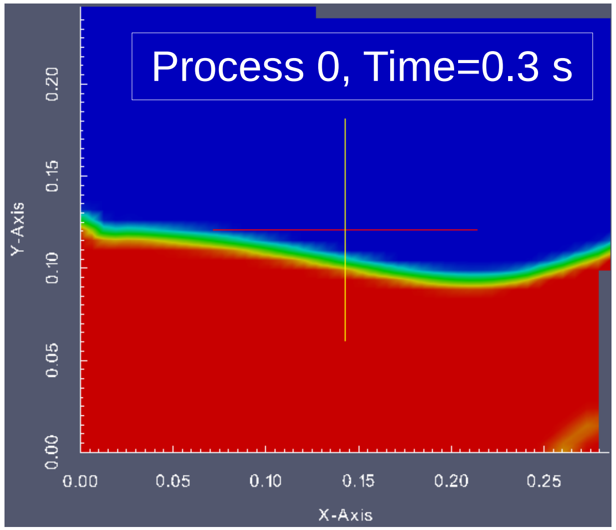


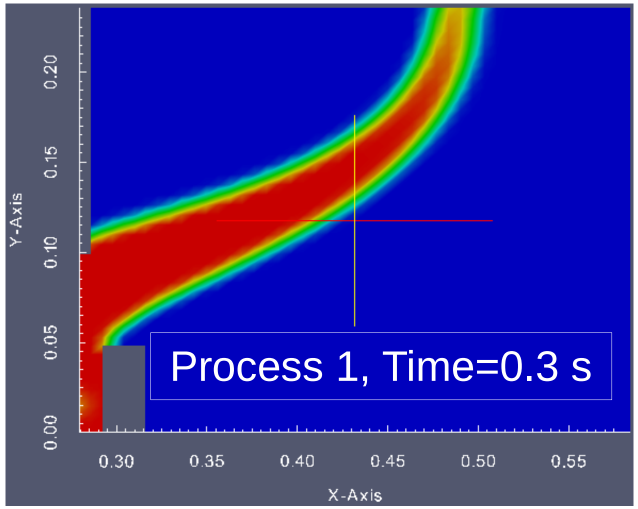


Comparison between single-processed and multi-processed (**domain decomposition**) (continued)

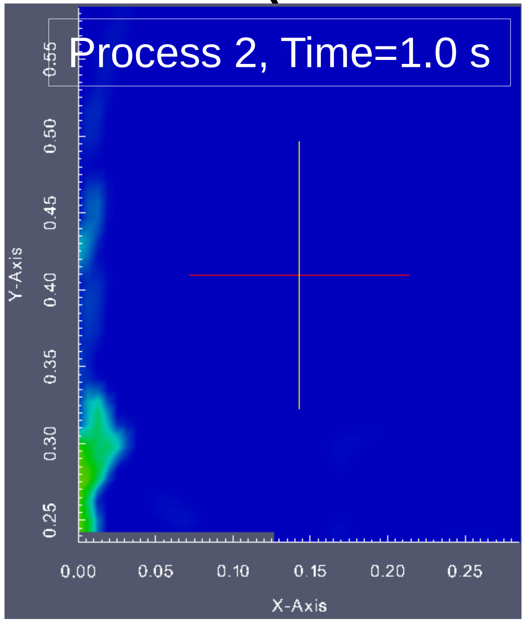


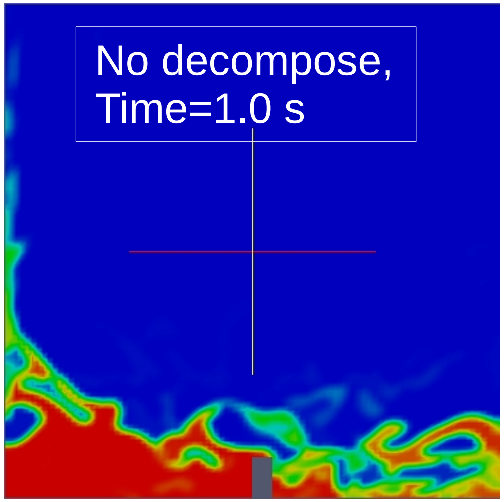




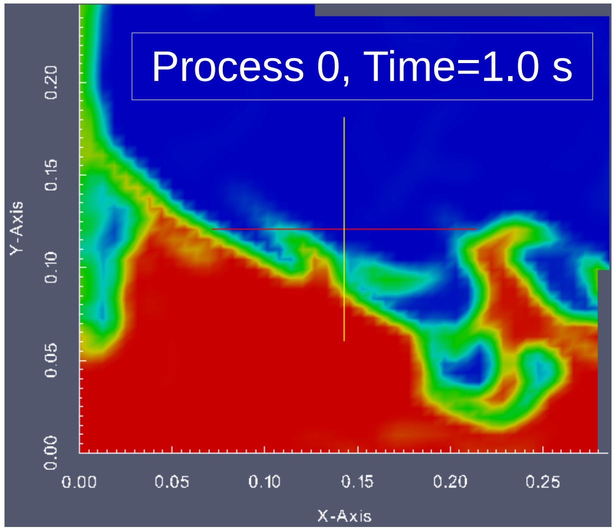


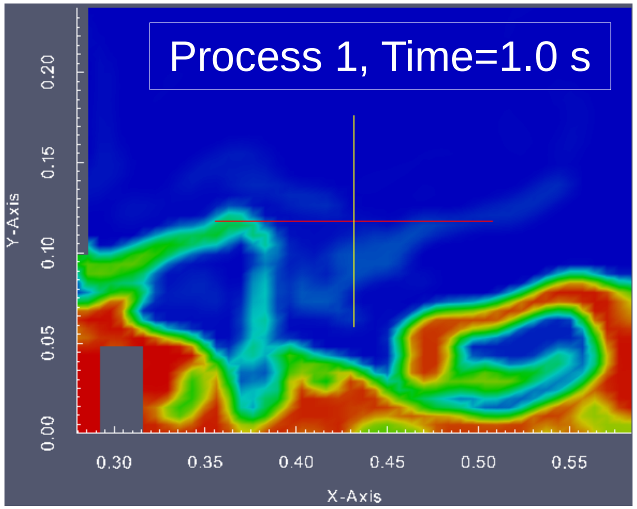
Comparison between single-processed and multi-processed (**domain decomposition**) (continued)





In detail, there are many differences between single-processed result and multi-processed result at Time=1.0 s. Such a discrepancy takes place as the calculation time passes. So far, no investigation to explain such discrepancies has been done.

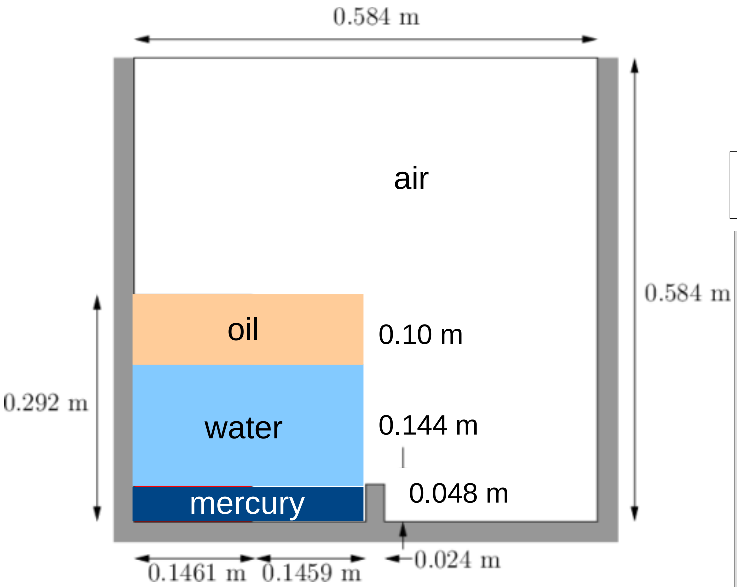




CFD Applications

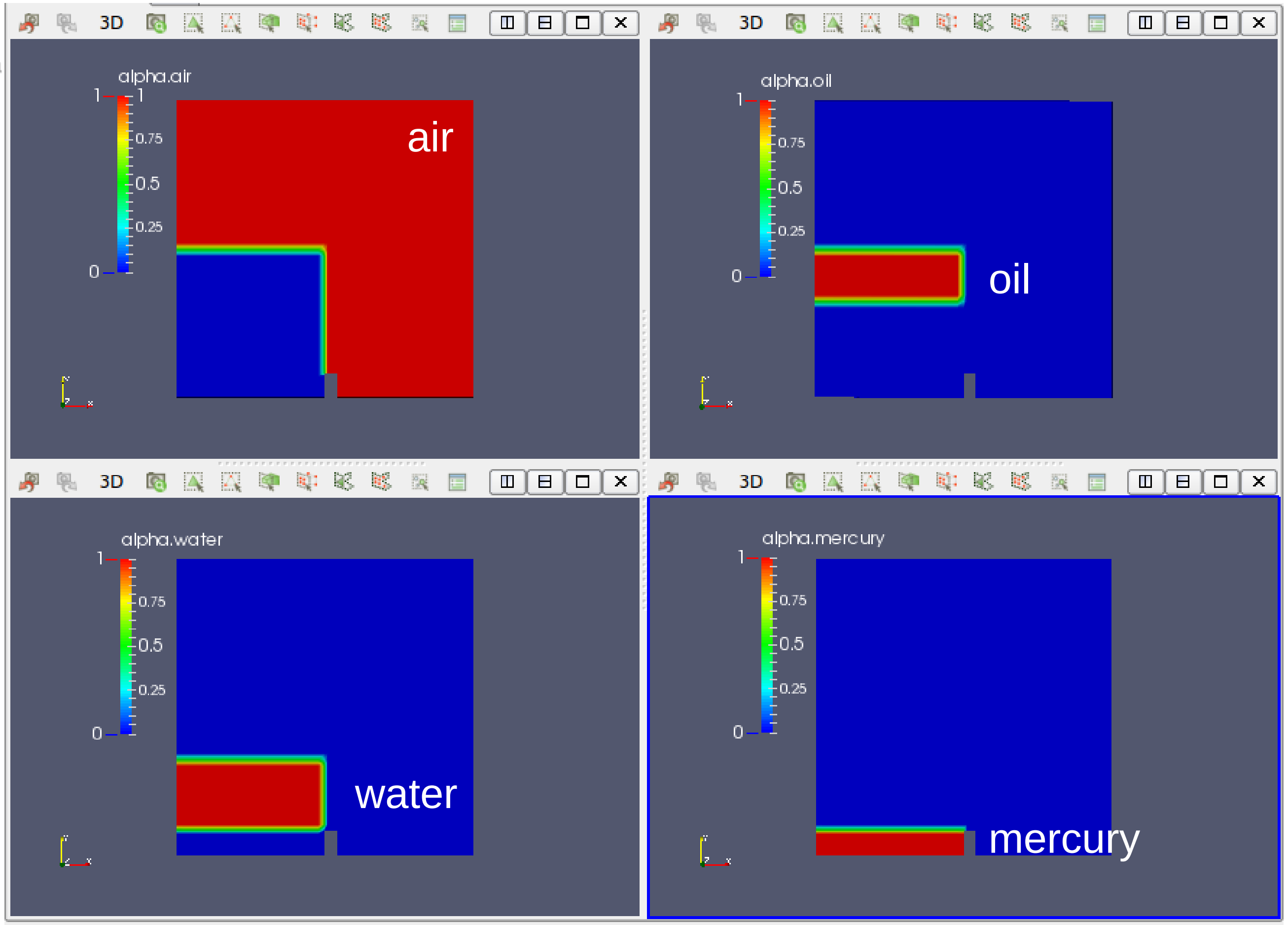
- Hydrological applications of CFD
  - **Example 1**: Dam of high concentration of heavy metal (mercury) in the bottom and being covered by oil layer is going to break.
  - This case is handling multi phases, surface tensions and contact angle to the walls (in terms of wettability).
  - **Extension**: Flow of multi-phase, immersible fluids through a porous media, e.g. penetration of water containing heavy metals into the soil (multi-phase infiltration**)**

Example 1: Heavy metal outflow

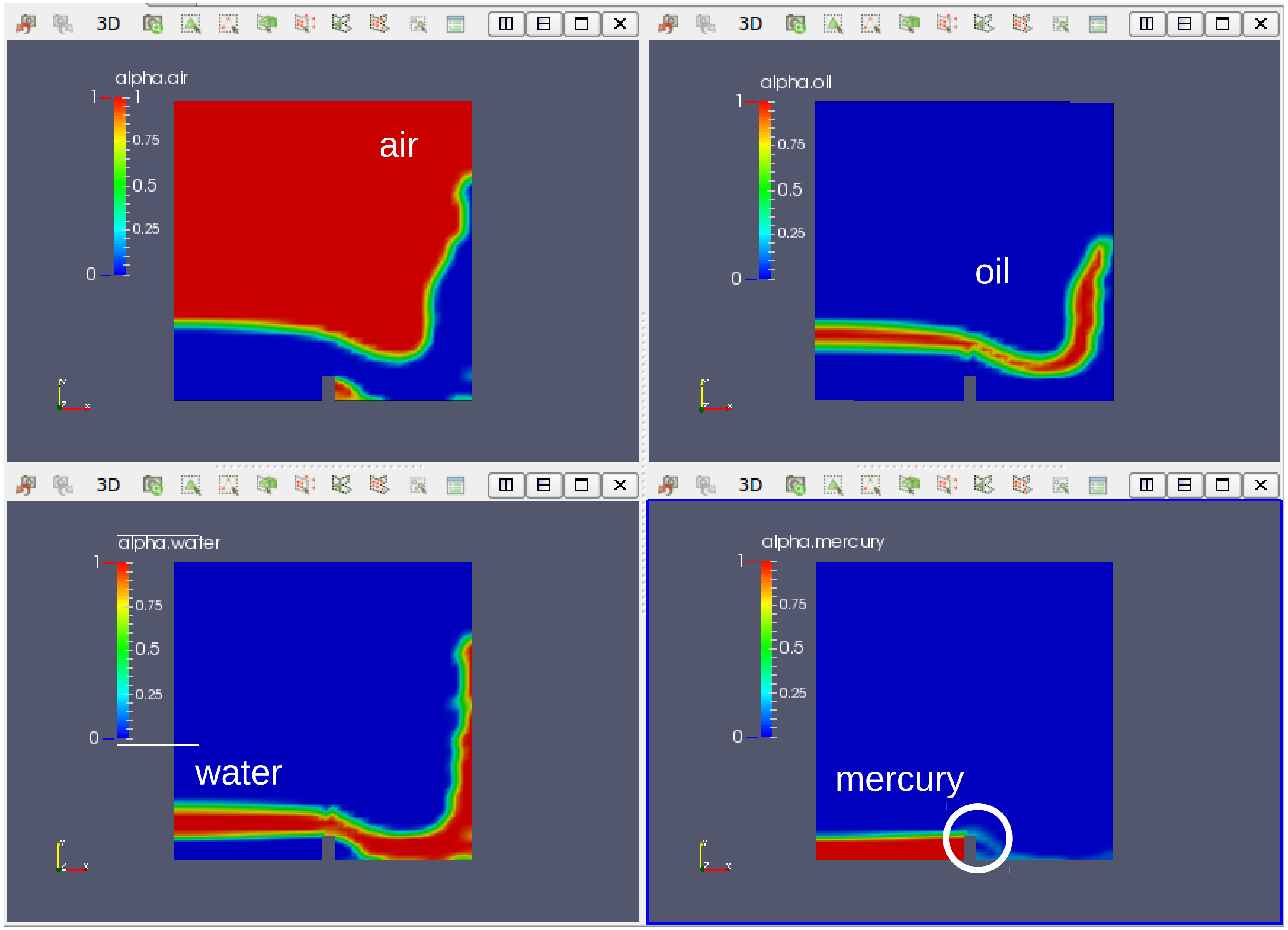


Time=0 s

If the fraction ratio is 1 (100%), it is shown by red colour.



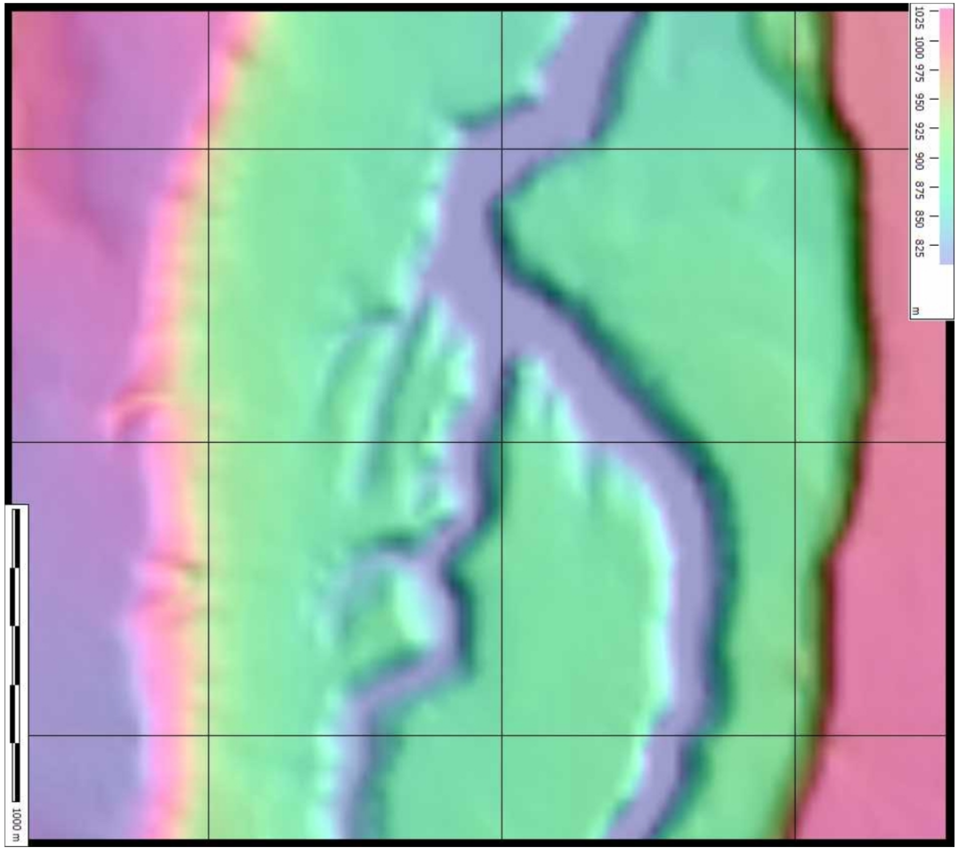
Example 1: Heavy metal outflow (continued)



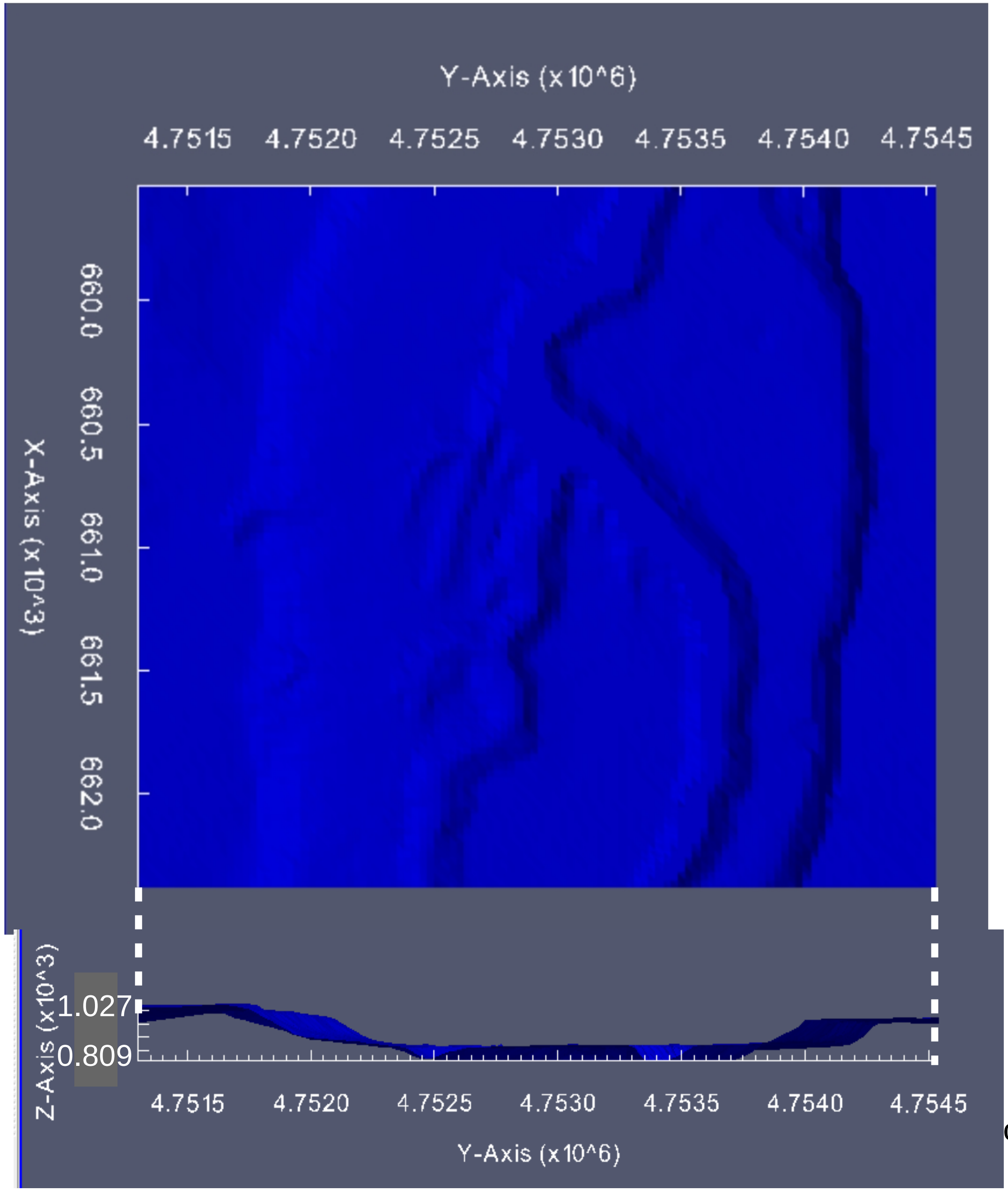
Despite the bottom layer of heavy metal is dammed by the remaining barrier, **a small amount of the heavy metal is leaked** being accompanied by the outflow of the upper layer of water. (Time=0.5 s)

CFD Applications

- Hydrological applications of CFD
  - **Example 2**: Mesh generation and conversion for complicated geometries, e.g. natural terrains.
  - Dynamic mesh (displacing river across the canyon), **Monitoring the changing wind path** (the compressible flow (air) interacting with both the fixed wall (terrain) and incompressible flow (water in the river)) due to the displacing river.

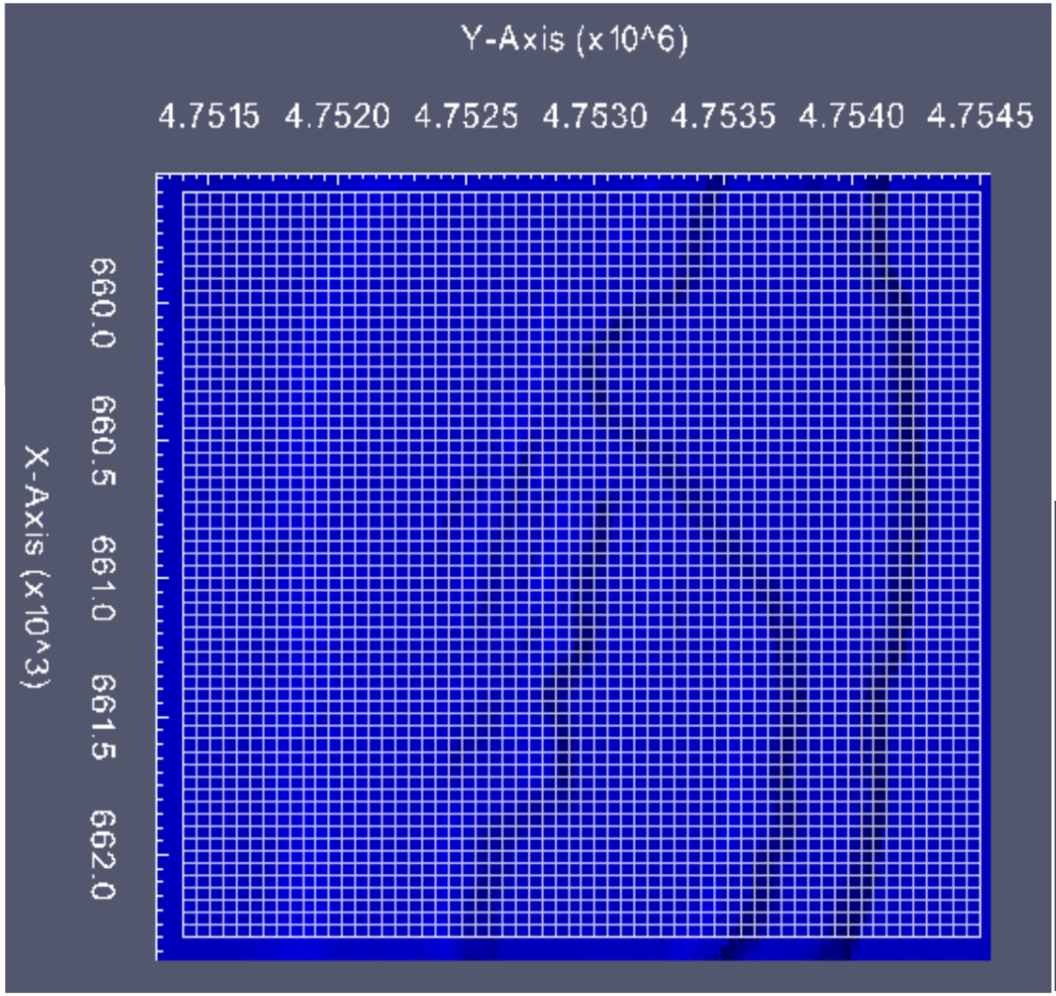


Example 2: Mesh generation and conversion

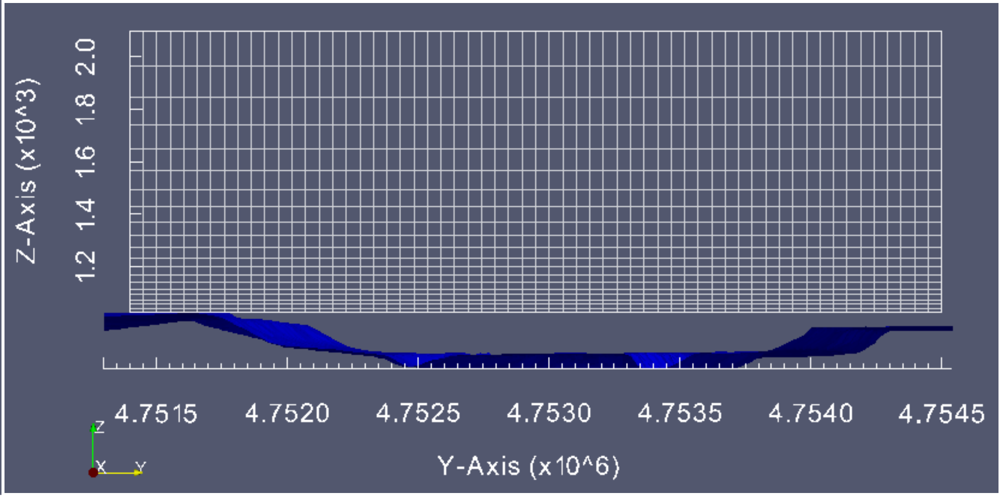


The geometry of the terrain is given as **STL (stereolithography)** file format and the size of this case is **3.0km** x **3.0km** x **218m**. This will be bottom wall (boundary) having partially dynamic (displacing) cells of the computational domain.

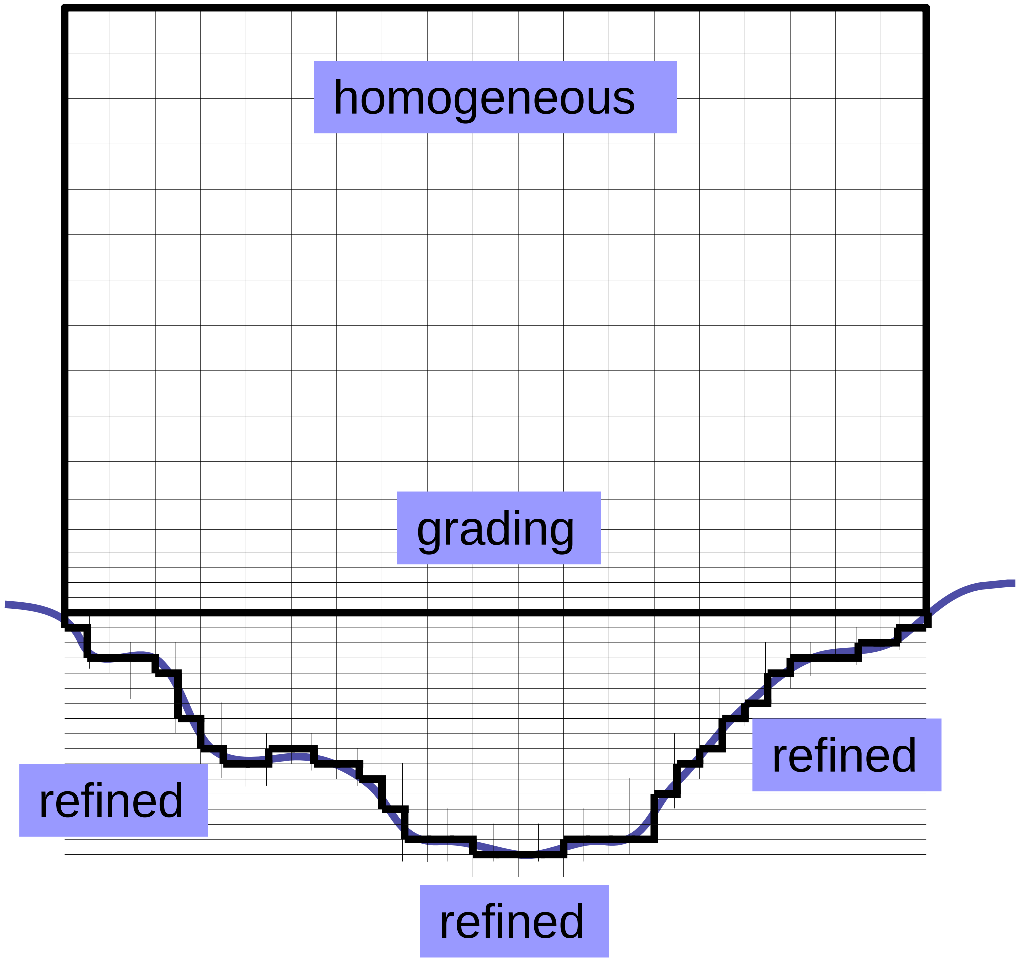
Example 2: Mesh generation and conversion (continued)



The upper half computational domain has a form of box with homogeneous meshes in the upper region and grading meshes (for higher resolution) in the lower region. This is due to that **complicated flow behaviours take place close to the terrain**.



Example 2: Mesh generation and conversion (continued)



The figure on the left side shows a schematic cross-section in an arbitrary YZ-plane, concerning internal meshes (of computational domain).

The domain consists of two parts: the **upper half domain** (a box form) which has **homogeneous meshes** (in 3D, regular hexahedron) in the upper part and **grading meshes** (in 3D, cuboid) in the lower part; the **lower half domain** which has the meshes consistent with the upper half domain in the centre and the **refined meshes** along the terrain.