7th OpenFOAM® Workshop, Darmstadt, 26/06/12

Application of OpenFOAM to transfers in soils : First developments and perspectives

Laurent Orgogozo[1], Nicolas Renon[2], Florent Hénon[3], Cyprien Soulaine[3]

[1] GET (Géosciences Environnement Toulouse), 14, avenue Édouard Belin, 31400 Toulouse, France
[2] CALMIP/DTSI, Université Paul Sabatier, University of Toulouse, 118, route de Narbonne, 31400 Toulouse, France
[3] IMFT (Institut de Mécanique des Fluides de Toulouse), Allée Camille Soula, F-31400 Toulouse, France

Transfers of water in soils

agriculture (nitrates transfer, …)

environnemental engineering (waste storage, …)

earth sciences (weathering processes, …)

Flow in non-saturated porous media : Richards equation

Flow in water saturated soils : diffusivity equation (e.g.: de Marsilly, 1986)

$$S\frac{\partial h}{\partial t} = div(\mathbf{K}.\nabla(h+z))$$

S the storage coefficient $\left[L^{-1}\right]$

$\mathbf{K}$ the permeability tensor $\left[L.T^{-1}\right]$

z the vertical coordinate $\left[L\right]$

h the matrix potential $\left[L\right]$ $\left(= p/\left(\rho_{water} g\right),\ h \geq 0\right)$

Flow in unsaturated soils : basically two-phase (air-water) flow, and thus two equations of conservation, but …

Flow in non-saturated porous media : Richards equation

usual conditions of pressure and temperature, fully connected air phase: a single equation for the unsaturated flow (Richards, 1931)

$$c(h)\frac{\partial h}{\partial t} = div(k(h)\mathbf{K}.\nabla(h+z))$$

c, the capillary capacity $\left[L^{-1}\right]$, $\begin{cases} c = S \text{ if } h \geq 0 \\ c = c(h) \text{ if } h < 0 \end{cases}$

k, the relative permeability $\left[-\right]$, $\begin{cases} k = 1 \text{ if } h \geq 0 \\ k = k(h) \text{ if } h < 0 \end{cases}$

$c(h)$ and $k(h)$ functions : experimentally fitted following various models (e.g. van Genuchten 1980, Brooks and Corey 1964)

Flow in non-saturated porous media : Richards equation

usual conditions of pressure and temperature, fully connected air phase: a single equation for the unsaturated flow (Richards, 1931)

$$c(h)\frac{\partial h}{\partial t} = div(k(h)\mathbf{K}.\nabla(h+z))$$

c, the capillary capacity $\left[L^{-1}\right]$, $\begin{cases} c = S \text{ if } h \geq 0 \\ c = c(h) \text{ if } h < 0 \end{cases}$

k, the relative permeability $\left[-\right]$, $\begin{cases} k = 1 \text{ if } h \geq 0 \\ k = k(h) \text{ if } h < 0 \end{cases}$

Non linearity

Picard method

In order to deal with non linearity, one can use a Picard algorithm

(e.g.: Ababou et al., 1992, Weill et al., 2009)

For each time step,

do iteratively

$$c(h_{i-1})\frac{\partial h_i}{\partial t} = div(k(h_{i-1})\mathbf{K}.\nabla(h_i + z))$$

until

$$\left\| h_i - h_{i-1} \right\| \leq \varepsilon$$

Implementation in OpenFOAM 1 – Picard algorithm

```
while (runTime.loop())
  {

      #include "readTimeControls.H"
      #include "setDeltaT.H"

      for (int pic=0; pic<nIterPicard; pic++)
         {
            psim1=psi;
            psi = psi_tmp;

-------------------------------------------------------------------------------------------------
            RESOLUTION (new psi)
            UPDATING OF c(h) AND k(h) (by taking into account the new psi)
-------------------------------------------------------------------------------------------------

            err=psi-psim1
            if (sqrt(sum(magSqr(err))).value()/nbMesh < precPicard)
               {
                  currentPicard=pic;
                  break;
               }
         }

      if (sqrt(sum(magSqr(err))).value()/nbMesh >= precPicard)
         {
            currentPicard=nIterPicard;
         }

      psi_tmp = psi;

      runTime.write();
  }
```

Implementation in OpenFOAM 2 – van Genuchten soil model

$$c, \text{the capillary capacity } \left[L^{-1}\right], \begin{cases} c = S \text{ if } h \geq 0 \\ c = \left(\theta_s - \theta_r\right) \alpha n \left(1 - \left(1/n\right)\right) \left(-\alpha h\right)^{n-1} \left(1 + \left(-\alpha h\right)^n\right)^{-\left(2-\left(1/n\right)\right)} \text{ if } h < 0 \end{cases}$$

$$k, \text{the relative permeability } \left[-\right], \begin{cases} k = 1 \text{ if } h \geq 0 \\ k = \left(\left(1 + \left(-\alpha h\right)^n\right)^{-\left(1-\left(1/n\right)\right)}\right)^{\left(1/2\right)} \left(1 - \left(1 - \left(\left(1 + \left(-\alpha h\right)^n\right)^{-\left(1-\left(1/n\right)\right)}\right)^{\left(n/n-1\right)}\right)^{1-\left(1/n\right)}\right)^2 \text{ if } h < 0 \end{cases}$$

```
Thtil = 0.5*(
          (1+sign(psi))
        + (1-sign(psi))*pow((1+pow(mag(alpha*psi),n)),-(1-(1/n)))
       ) ;

Krel = 0.5*(
         (1+sign(psi))*K
       + (1-sign(psi))*K*pow(thtil,0.5)*pow((1-pow((1-pow(thtil,(n/(n-1)))),(1-(1/n)))),2)
      );

Crel = 0.5*(
         (1+sign(psi))*S
       + (1-sign(psi))*((thetas-thetar)*n*alpha*(1-(1/n))*pow(mag(alpha*psi),(n-1))*pow((1+pow(mag(alpha*psi),n)),-(2-(1/n))))
      );
```

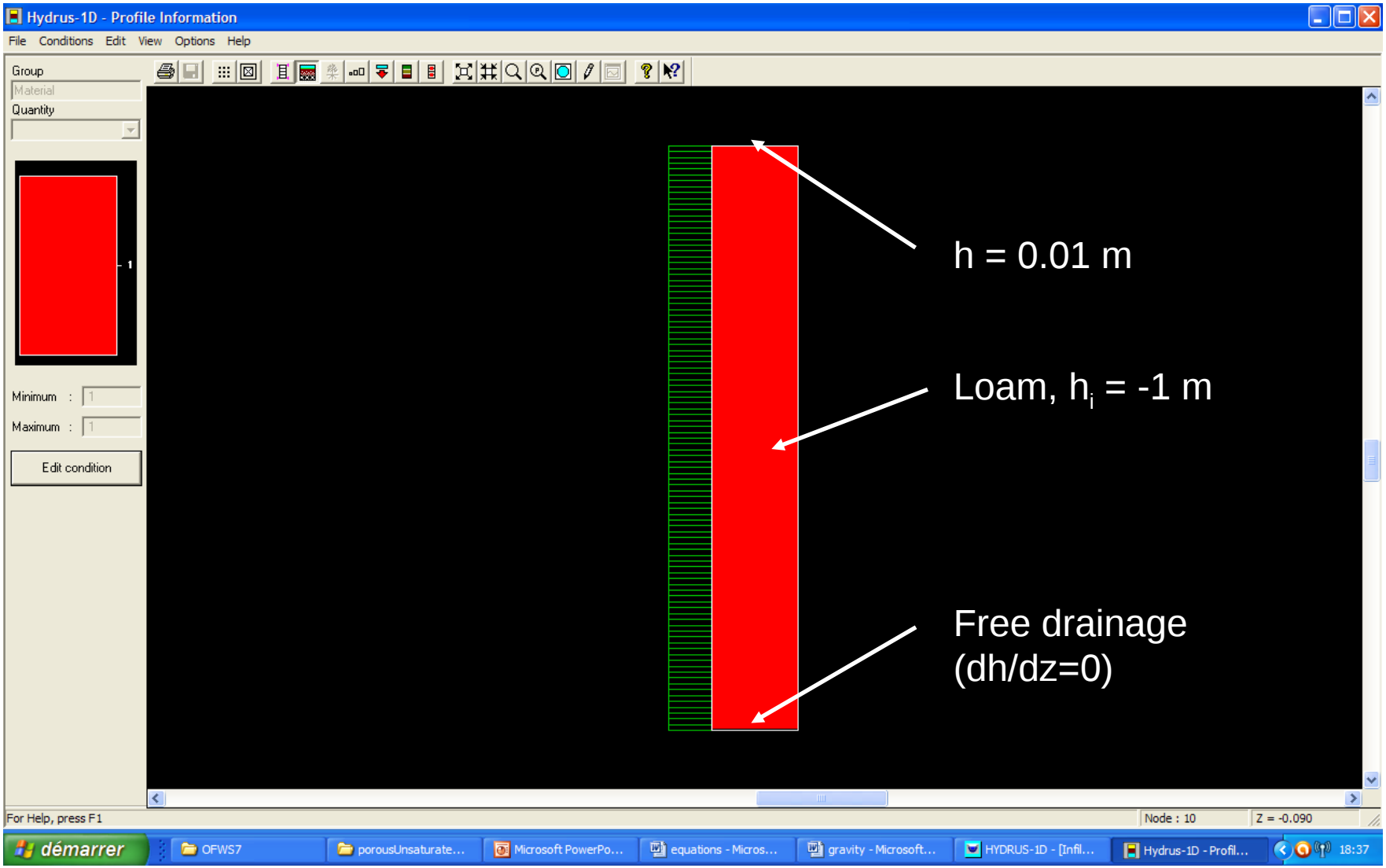
Implementation in OpenFOAM 3 – gravity term

$$c(h)\frac{\partial h}{\partial t} = div(k(h)\mathbf{K}.\nabla(h+z)) \Leftrightarrow c(h)\frac{\partial h}{\partial t} = div(k(h)\mathbf{K}.\nabla(h)) + \frac{\partial\left(k(h)K_{zz}\right)}{\partial z}$$

$$\mathbf{U} = -k(h)\mathbf{K}.\nabla(h+z) \Leftrightarrow \mathbf{U} = -k(h)\mathbf{K}.\left(\nabla(h) + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}\right)$$

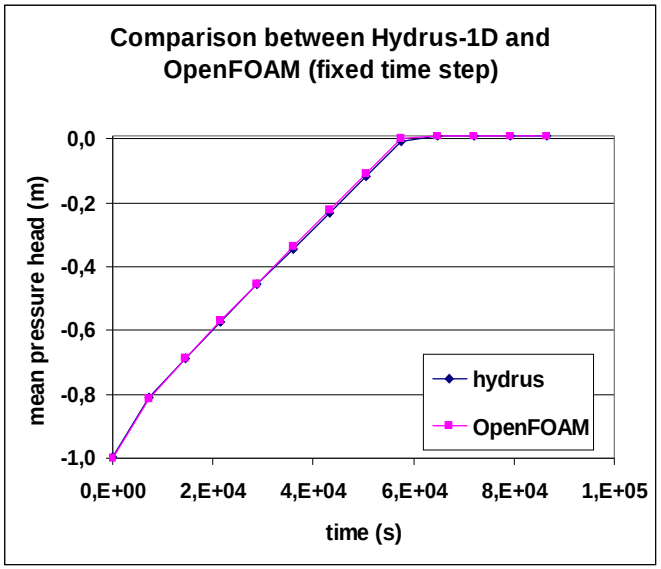
```
gradk=fvc::grad(Krel);

gradkz=gradk.component(vector::Z);

fvScalarMatrix psiEqn
   (
      Crel*fvm::ddt(psi)
      ==
      fvm::laplacian(Krel,psi,"laplacian(Krel,psi)")
      +gradkz
   );

U=-Krel*((fvc::grad(psi))+vuz);
```

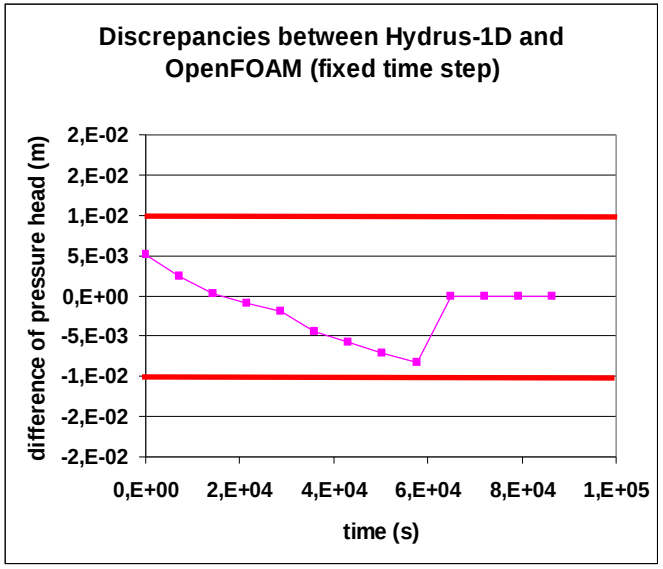
Validation : comparison with Hydrus 1D (PC-Progress)

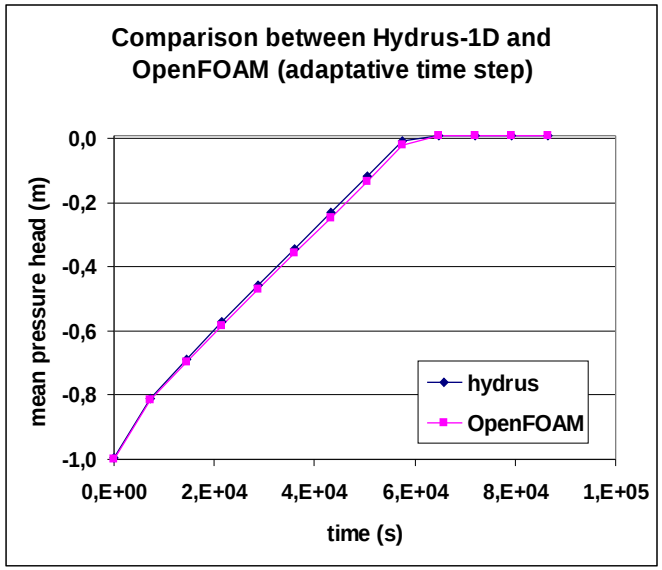


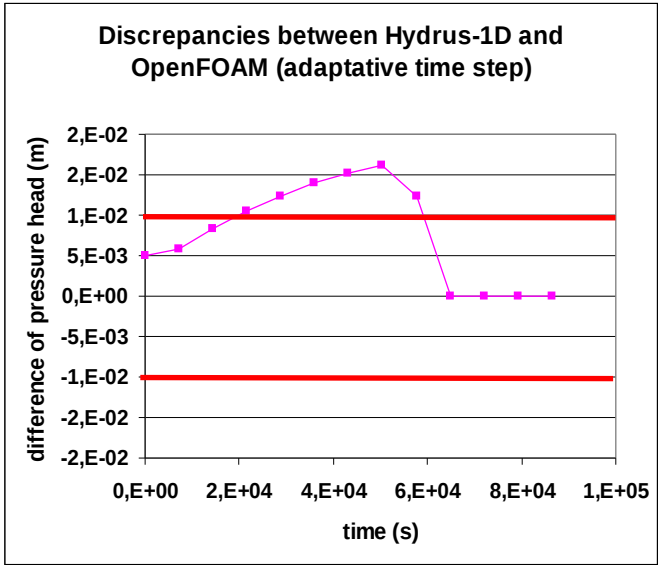
(e.g.: Šimůnek et al., 2008)

Validation : comparison with Hydrus 1D (PC-Progress)





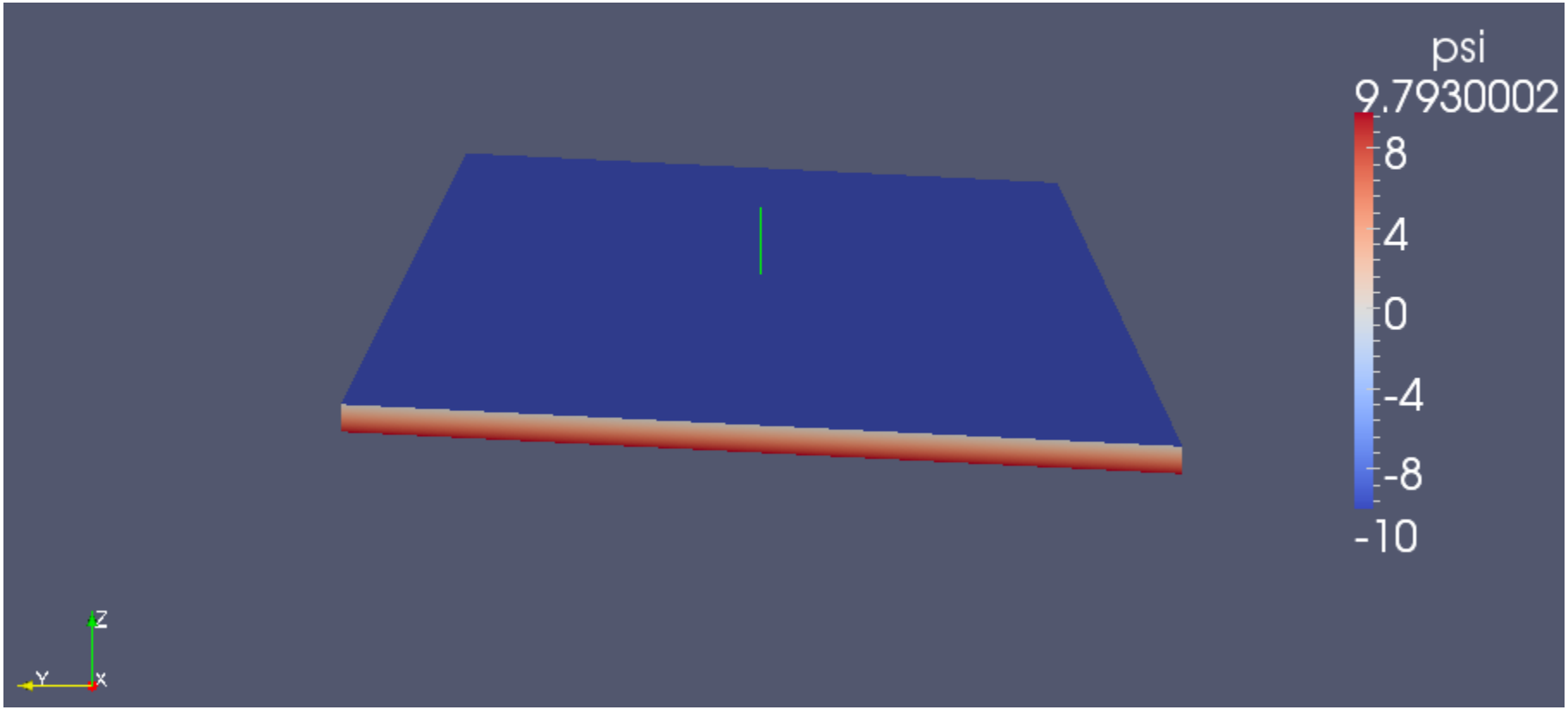




3D case : geometry, boundary and initials conditions

A slope of 300m*300m with a 10m thick layer of loam, and a river at the bottom.

The slope is inclined of 20° and the river of 3°. Mesh 300*300*200 max ts 30 mn.



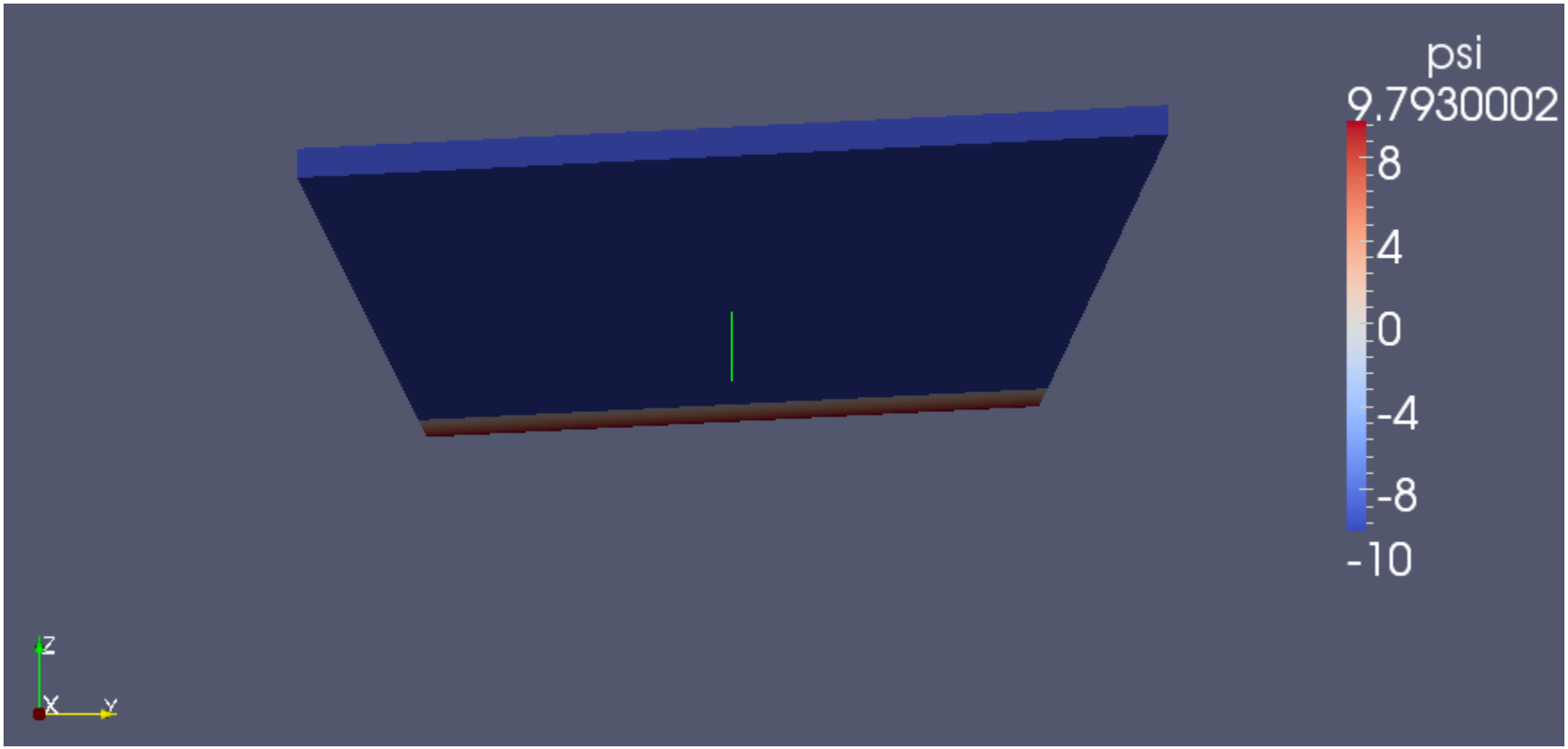
BC : hydrostatic pressure at the river, h=-0.1 m at the top and No flux elsewhere

IC : hydrostatic pressure below the river level, h=-10m above.

3D case : geometry, boundary and initials conditions

A slope of 300m*300m with a 10m thick layer of loam, and a river at the bottom.

The slope is inclined of 20° and the river of 3°. Mesh 300*300*200 max ts 30 mn.



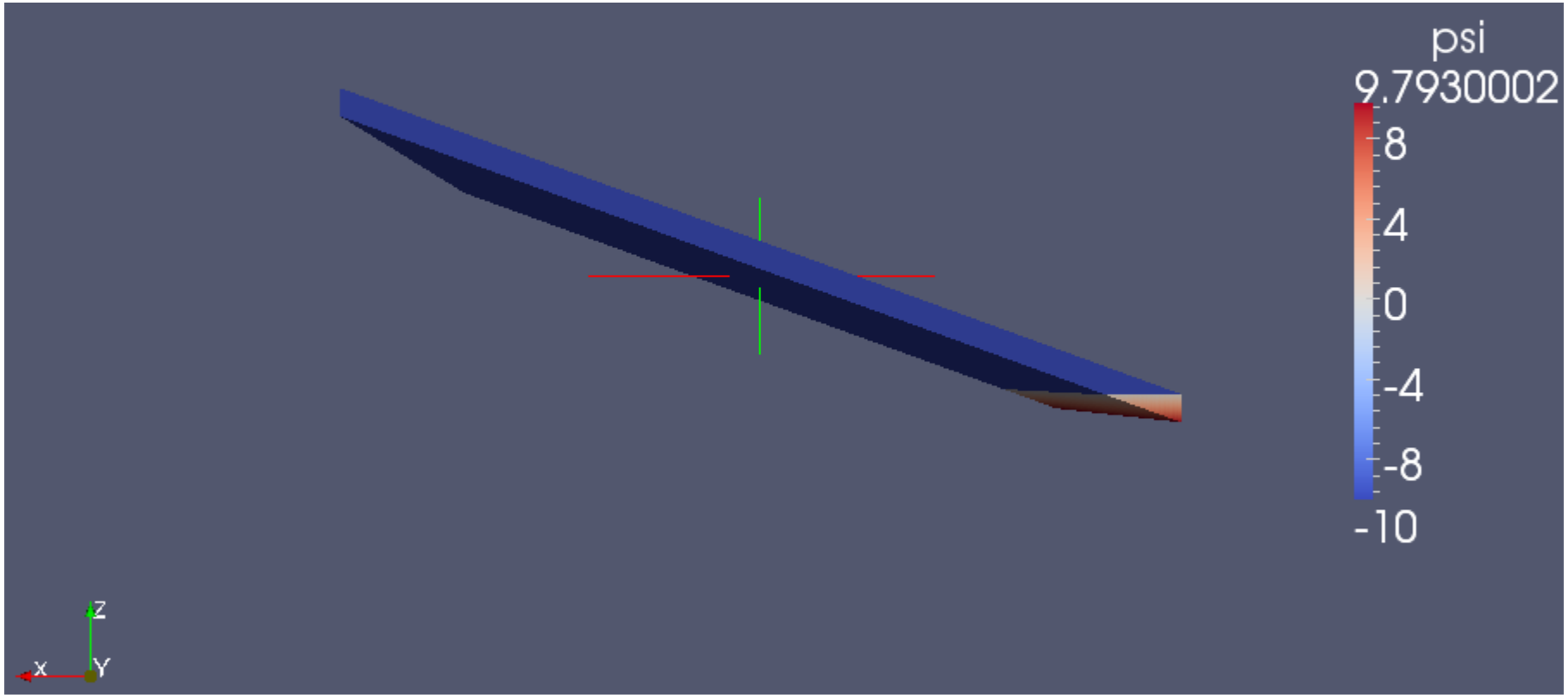
BC : hydrostatic pressure at the river, h=-0.1 m at the top and No flux elsewhere

IC : hydrostatic pressure below the river level, h=-10m above.

3D case : geometry, boundary and initials conditions

A slope of 300m*300m with a 10m thick layer of loam, and a river at the bottom.

The slope is inclined of 20° and the river of 3°. Mesh 300*300*200 max ts 30 mn.



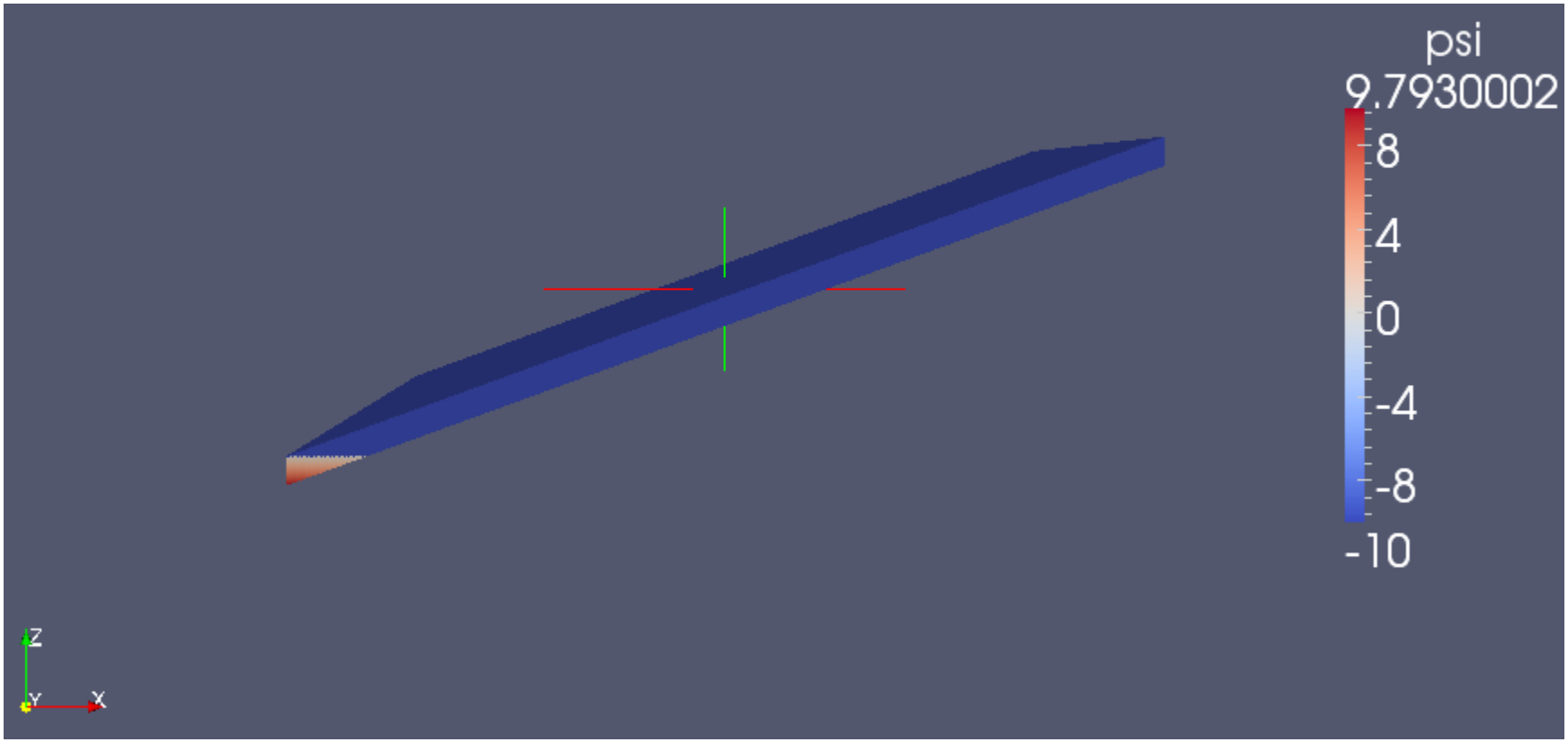
BC : hydrostatic pressure at the river, h=-0.1 m at the top and No flux elsewhere

IC : hydrostatic pressure below the river level, h=-10m above.

3D case : geometry, boundary and initials conditions

A slope of 300m*300m with a 10m thick layer of loam, and a river at the bottom.

The slope is inclined of 20° and the river of 3°. Mesh 300*300*200 max ts 30 mn.



BC : hydrostatic pressure at the river, h=-0.1m at the top and No flux elsewhere

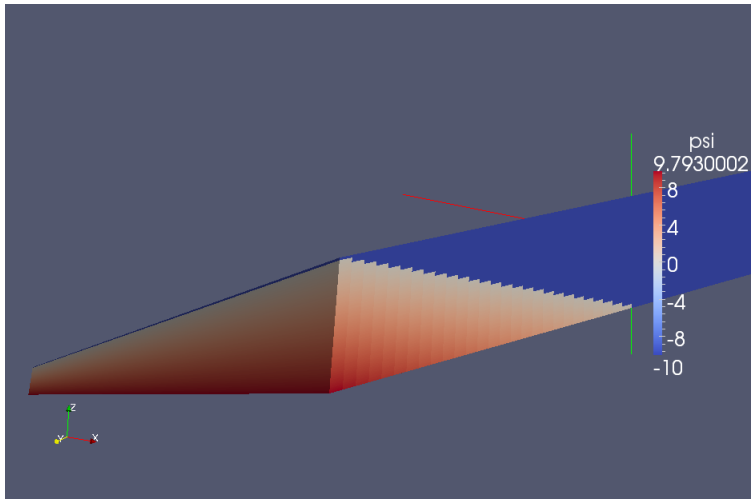
IC : hydrostatic pressure below the river level, h=-10m above

3D case : geometry, boundary and initials conditions

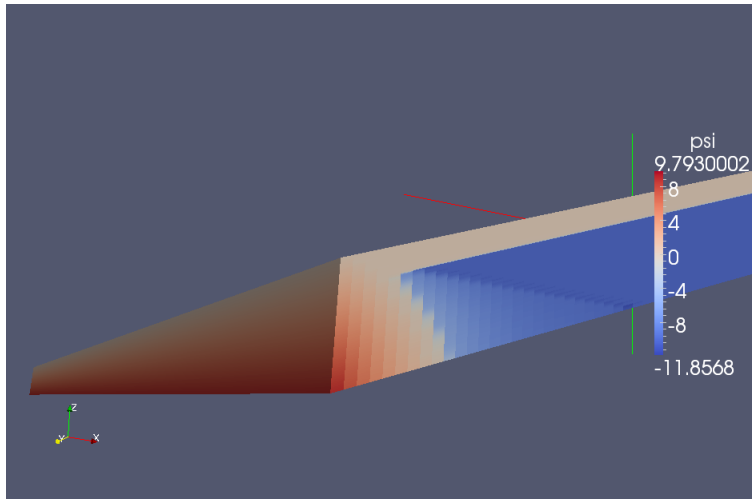
Set up of the case :

1) **A « solver » to build up a case with h = z + y*5/100 everywhere (i.e. hydrostatic pressure in the river)**
2) **mapFields from this case to the case of resolution**
3) **swak4Foam (funkySetFields) to set h = -10m above the water table.**
4) **decomposePar**
5) **Resolution**
6) **reconstructPar**

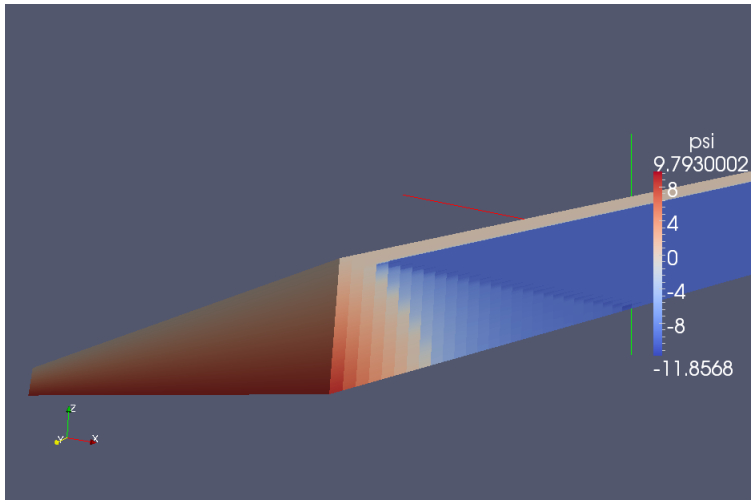
3D case : geometry, boundary and initials conditions



initial

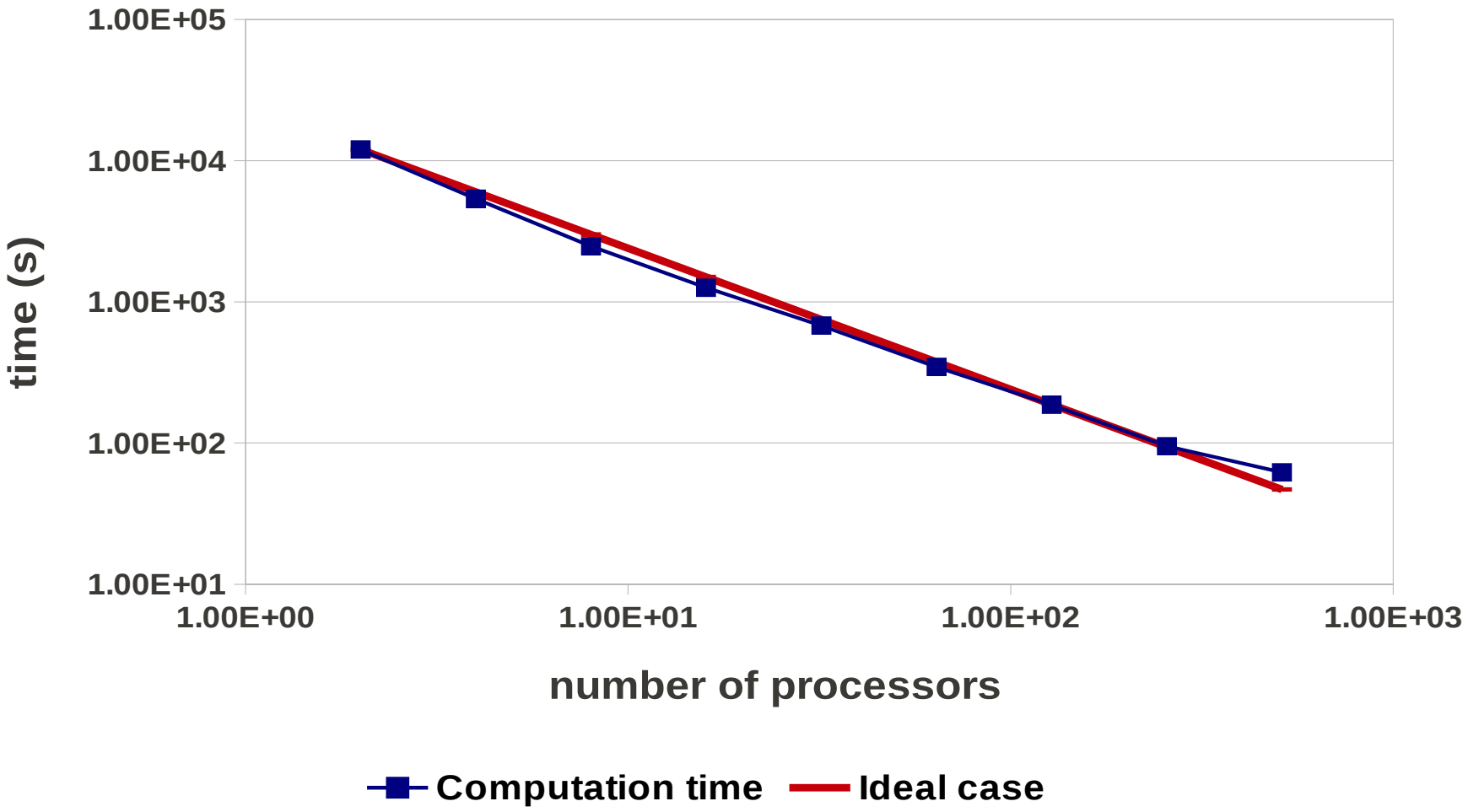


t = 10 j



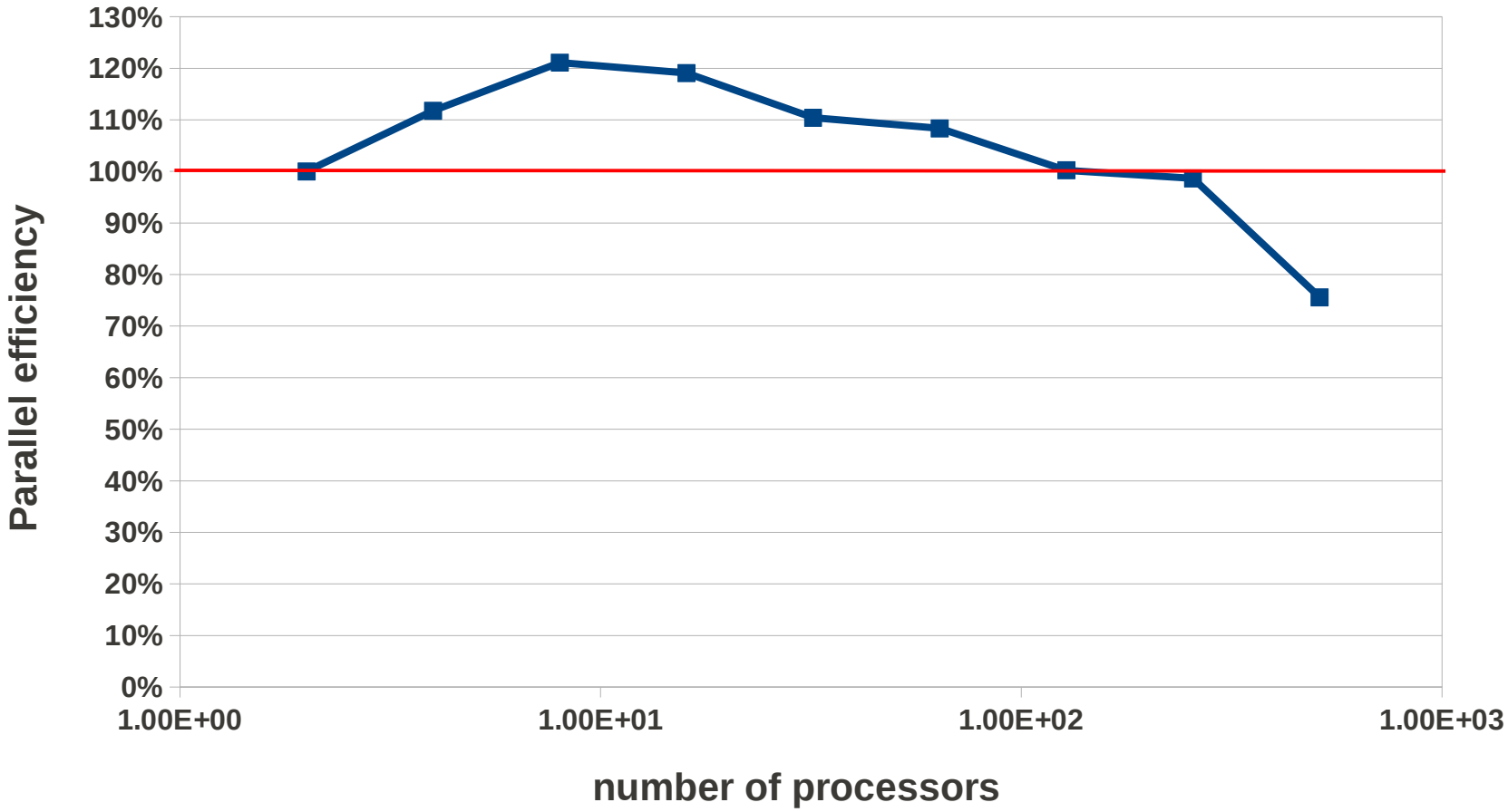
t = 5 j

Parallel computation



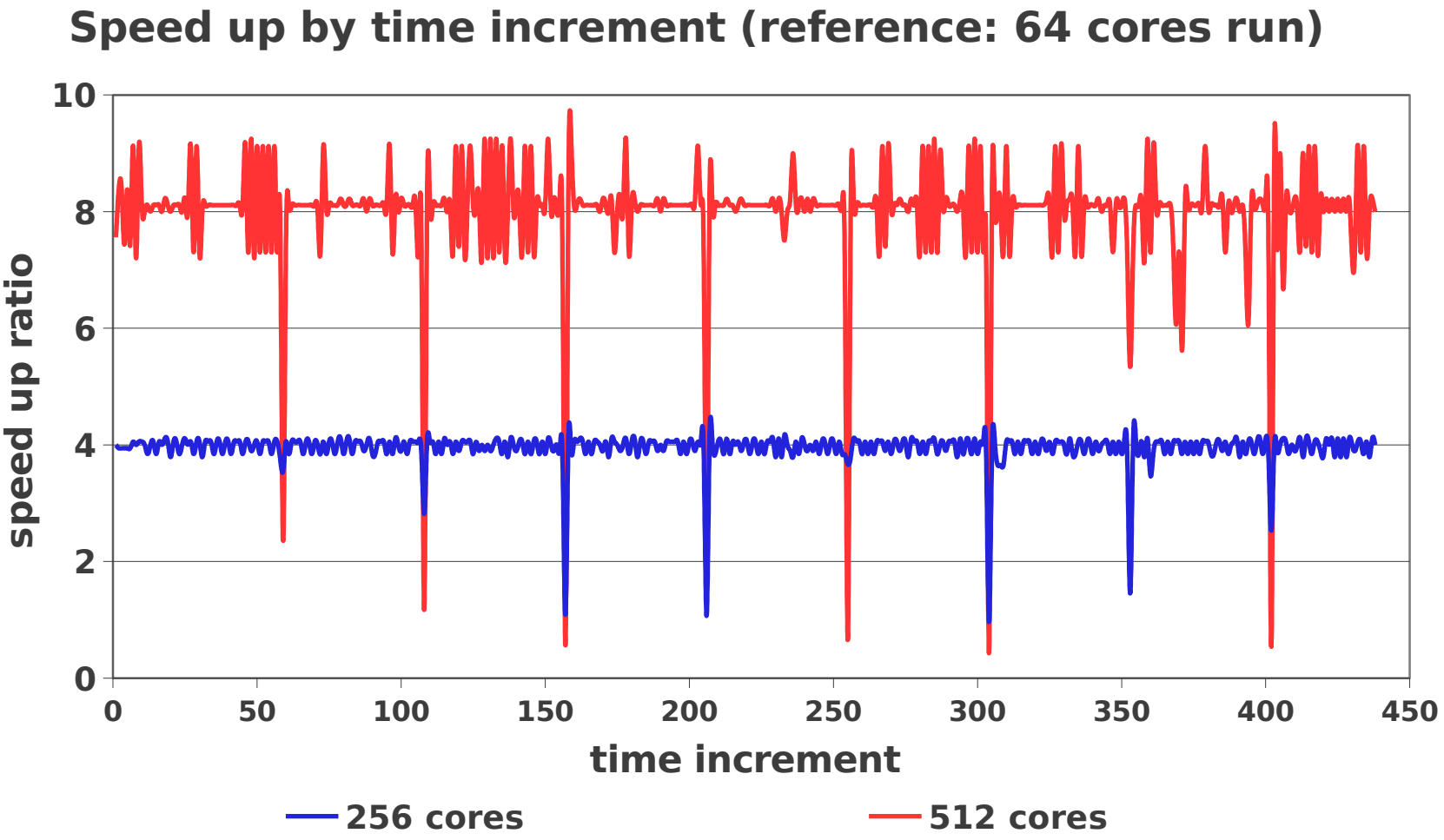
(performed on the CALMIP cluster, http://www.calmip.cict.fr/spip/)

Parallel computation



(performed on the CALMIP cluster, http://www.calmip.cict.fr/spip/)

Parallel computation



(performed on the CALMIP cluster, http://www.calmip.cict.fr/spip/)

Perspectives

Benchmark code to code and with analytical solutions

Coupling with energy and solute transfer

Parallel meshing

Create meshes from DEMs

Build boundary conditions on the basis of observation chronicles

Application to experimental watersheds (e.g.: GDRI CAR WET SIB, ORE-BVET)

Thank you for your attention

Contact : laurent.orgogozo@get.obs-mip.fr

GET
GÉOSCIENCES
ENVIRONNEMENT
TOULOUSE

IMFT
INSTITUT DE MÉCANIQUE
DES FLUIDES DE TOULOUSE