TWO-PHASE FLOW SOLVER FOR ANISOTROPIC POROUS MEDIA

-

C. Soulaine[1,*], M. Quintard[1,2]

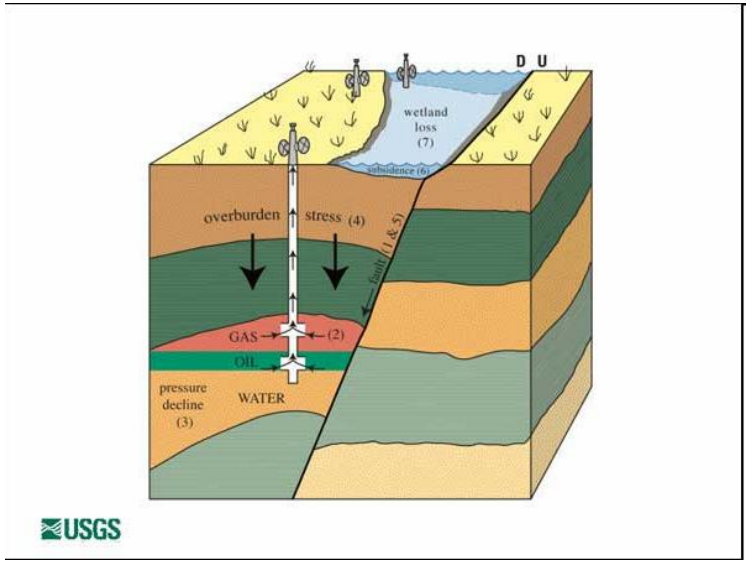
[1] Institut de Mécanique des Fluides de Toulouse, France
[2] C.N.R.S, France

[*] cyprien.soulaine@imft.fr
CFD online nickname: Cyp

7th OpenFOAM® Workshop
26th june, 2012

Multiphase flow in porous media

Chemical engineering (distillation processes, trickle beds…)



Petroleum reservoirs

Pollution of soils

Single-phase flow in porous media

Either tensor (anisotropic medium) or scalar

- **Darcy's law** $\mathbf{V}_\beta = -\frac{1}{\mu_\beta}\mathbf{K}_\beta.(\nabla P_\beta - \rho_\beta \mathbf{g})$ (1)
- **Continuity equation for incompressible fluid:** $\nabla.\mathbf{V}_\beta = 0$ (2)
- (1) **in** (2) **= pressure equation :**

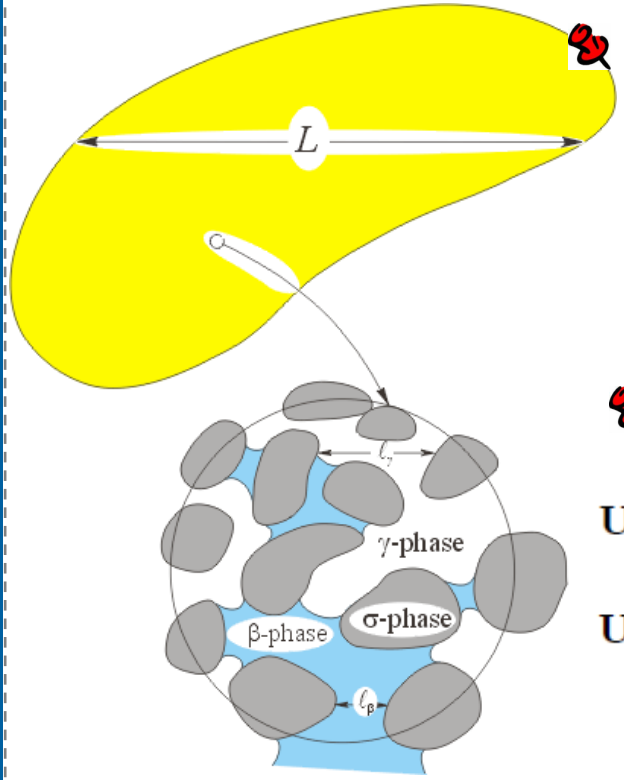
$$\nabla \cdot (-\frac{\mathbf{K}_\beta}{\mu_\beta} \cdot \nabla P_\beta) = 0$$

```
#include "BC.H"

for (int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
{
    fvScalarMatrix pEqn
    (
        fvm::laplacian(-M,p)
    );
    pEqn.setReference(pRefCell, pRefValue);
    pEqn.solve();

    if (nonOrth == nNonOrthCorr)
    {
        phi = pEqn.flux()+phiGG;
        U = fvc::reconstruct(phi);
        U.correctBoundaryConditions();
    }

}
```

fvm::laplacian(M,p) already accounts for tensor

How to deal with two-phase flow ?

Two-phase flow in porous media: theory (1/2)



Continuity equations :

$$\varepsilon\frac{\partial S_\gamma}{\partial t} + \nabla.\mathbf{U}_\gamma = 0$$

$$\varepsilon\frac{\partial S_\beta}{\partial t} + \nabla.\mathbf{U}_\beta = 0$$

$$S_\beta + S_\gamma = 1$$

Generalised Darcy's law (with/without viscous term):

$$\mathbf{U}_\gamma = -\frac{\mathbf{K}_\gamma}{\mu_\gamma}\cdot(\nabla P_\gamma - \rho_\gamma \mathbf{g}) \qquad \mathbf{U}_\gamma = -\frac{\mathbf{K}_\gamma}{\mu_\gamma}\cdot(\nabla P_\gamma - \rho_\gamma \mathbf{g}) + \mathbf{K}_{\gamma\beta}.\mathbf{U}_\beta$$

$$\mathbf{U}_\beta = -\frac{\mathbf{K}_\beta}{\mu_\beta}\cdot(\nabla P_\beta - \rho_\beta \mathbf{g}) \qquad \mathbf{U}_\beta = -\frac{\mathbf{K}_\beta}{\mu_\beta}\cdot(\nabla P_\beta - \rho_\beta \mathbf{g}) + \mathbf{K}_{\beta\gamma}.\mathbf{U}_\gamma$$

The multiphase permeabilities depend on the saturation profiles and Reynold number (in case of inertial flow)

Curvature of the interface beta/gamma is represented by a macroscale capillary pressure:

$$P_\beta - P_\gamma = P_c(S)$$

Two-phase flow in porous media: theory (2/2)

- General formulation of the two-phase pressure/velocity laws in porous media:

$$\mathbf{U}_\gamma = -\mathbf{M}_\gamma \cdot \nabla P + \mathbf{L}_\gamma \cdot \mathbf{g}$$

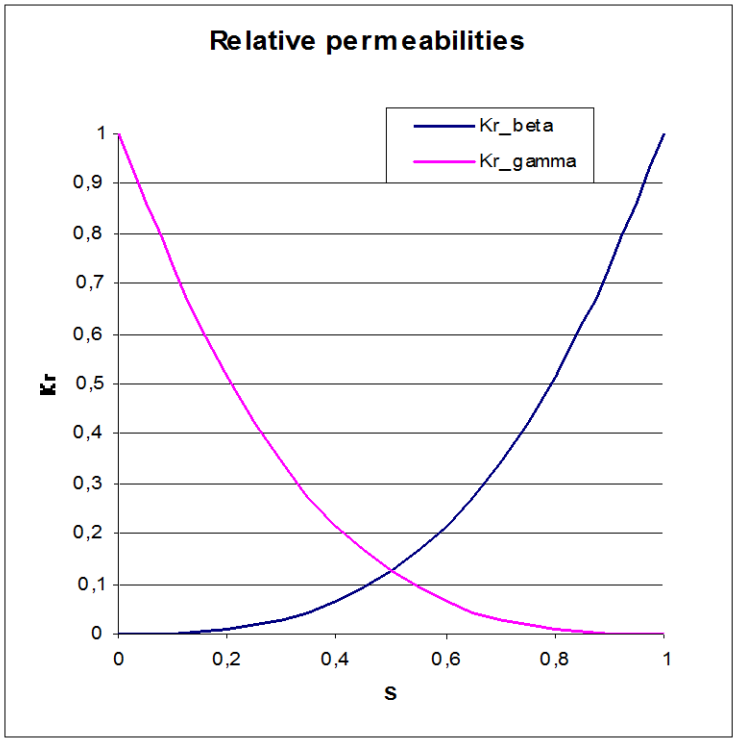
$$\mathbf{U}_\beta = -\mathbf{M}_\beta \cdot \nabla P + \mathbf{M}_\beta \cdot \nabla P_c + \mathbf{L}_\beta \cdot \mathbf{g}$$

- The multiphase permeabilities depend on the saturation profiles.

  For example, the Brooks-Corey correlation relates S and K

$$\mathbf{K}_\gamma = \mathbf{K}\, k_{r\gamma}(S_\gamma) \longrightarrow \mathbf{K}_\gamma = \mathbf{K}(1-S)^n$$

$$\mathbf{K}_\beta = \mathbf{K}\, k_{r\beta}(S_\beta) \longrightarrow \mathbf{K}_\beta = \mathbf{K}S^n$$



Solution procedure for the two-phase flow system

IMPES = **IM**plicit **P**ressure **E**xplicit **S**aturation

- Well-known of the oil industry (see *Aziz and Settari, 1979*)

- The two-phase flow problem is reformulated in term of
  - a gas pressure *(P)* equation
  - a liquid saturation *(S)* equation

- Boundary conditions depending on inlet velocity values but the IMPES method solves a pressure equations…

Implicit pressure equation

Gas pressure equation

$$\mathbf{U} = \mathbf{U}_\gamma + \mathbf{U}_\beta$$

$$\mathbf{U} = -\mathbf{M}\cdot\nabla P + \mathbf{M}_\beta \cdot \nabla P_c(S) + \mathbf{L}\cdot\mathbf{g} = 0$$

$$\begin{cases} \varepsilon\dfrac{\partial S_\beta}{\partial t} + \nabla.\mathbf{U}_\beta = 0 \\ \oplus \qquad\qquad = \qquad \nabla\cdot\mathbf{U} = 0 \\ \varepsilon\dfrac{\partial S_\gamma}{\partial t} + \nabla.\mathbf{U}_\gamma = 0 \end{cases}$$

$$\nabla\cdot(-\mathbf{M}\cdot\nabla P) + \nabla\cdot(\mathbf{M}_\beta\cdot\nabla P_c(S) + \mathbf{L}\cdot\mathbf{g}) = 0$$

- M and L depend on the saturation field and are updated every time-steps
- The velocity fields Ua and Ub are reconstructed using generalized Darcy's laws

```
{

    surfaceScalarField phiGG = linearInterpolate(L & g) & mesh.Sf();

    surfaceScalarField phiPcS =
            linearInterpolate(Mb*dPcdS & fvc::grad(S)) & mesh.Sf();

    phiGG += phiPcS;

    #include "BC.H"

    for(int nonOrth=0; nonOrth<=nNonOrthCorr; nonOrth++)
    {
       fvScalarMatrix pEqn
        (
            fvm::laplacian(-M, p) + fvc::div(phiGG)
        );

        pEqn.setReference(pRefCell, pRefValue);
        pEqn.solve();

        if (nonOrth == nNonOrthCorr)
        {
            phi = pEqn.flux()+phiGG;
            U = fvc::reconstruct(phi);
            U.correctBoundaryConditions();
        }
    }

}
```

Explicit saturation equation

Liquid saturation equation

$$\varepsilon \frac{\partial S_\beta}{\partial t} + \nabla . \mathbf{U}_\beta = 0$$

```
word sScheme("div(phi,S)");

for (int scorr=0; scorr<nSCorr; scorr++)
{

    surfaceScalarField phiS = fvc::flux((phib/linearInterpolate(S)),S,sScheme);

    fvScalarMatrix SEqn
    (
          eps*fvm::ddt(S) + fvc::div(phiS)
    );
    SEqn.relax();
    SEqn.solve();

}
```

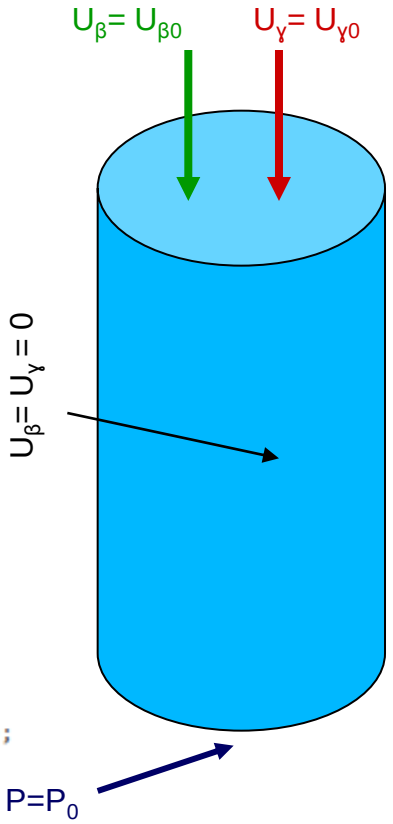
- This equation is hyperbolic and required a flux limiter for a better accuracy of the solution. We choose a van Leer scheme for the simulations.

- Stability of the convergence is ensured by a CFL condition.

Boundary conditions

- If the inlet and wall boundary conditions are velocity fixed values, they should be transformed into pressure gradient conditions using:

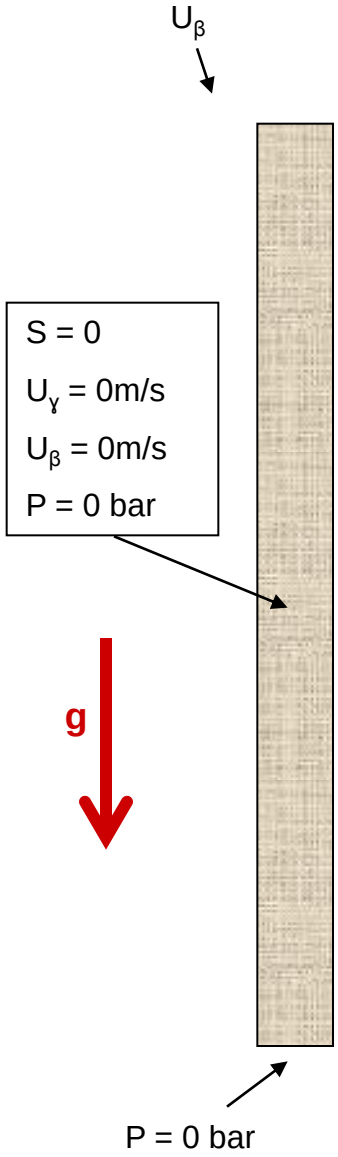
$$\nabla P \cdot \mathbf{n} = \left[\mathbf{M}^{-1}(-\mathbf{U} + \mathbf{L} \cdot \mathbf{g})\right] \cdot \mathbf{n}$$



- For the moment, it is hard-coded in OpenFOAM:

```
#include "fixedGradientFvPatchFields.H"


surfaceScalarField phiUM = linearInterpolate(invM &  U) & mesh.Sf();
surfaceScalarField phiGGM = linearInterpolate (invM & (L & g)) & mesh.Sf();

scalarField fluxP = -(phiUM.boundaryField()[patchID]-phiGGM.boundaryField()[patchID])
            /(mesh.magSf().boundaryField()[patchID]);


fixedGradientFvPatchScalarField& gradUPatch=refCast<fixedGradientFvPatchScalarField>(p.boundaryField()[patchID]);
scalarField& gradUField =  gradUPatch.gradient();
gradUField = fluxP;


scalarField fluxPwall = -(-phiGGM.boundaryField()[patchWall])
            /(mesh.magSf().boundaryField()[patchWall]);
```

Validation: Buckley-Leverett analysis (1/4)



1D generalized Darcy's law

$$U_\beta = -k\frac{S^3}{\mu_\beta}\left(\frac{\partial P}{\partial z} - \rho_\beta g_z\right)$$

$$U_\gamma = -k\frac{(1-S)^3}{\mu_\gamma}\left(\frac{\partial P}{\partial z} - \rho_\gamma g_z\right)$$

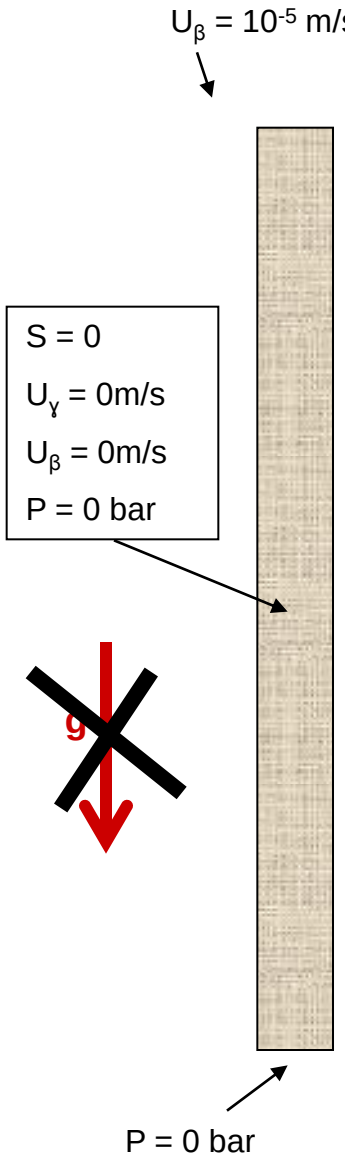
Input data

$\rho_\beta = 1000 kg/m^3$	$\rho_\gamma = 1 kg/m^3$
$\mu_\beta = 1\ \ 10^{-3} Pa.s$	$\mu_\gamma = 17.6\ \ 10^{-6} Pa.s$
$K = 10^{-11} m^2$	$\epsilon = 0.5$

Buckley-Leverett theory -> there are analytical solutions that depend on the inlet velocity value

- if $U_0 < U_{lim}$ then the flow is governed by gravity effects
- else if $U_0 > U_{lim}$ then the flow is described by Buckley-Leverett solutions

Validation: Buckley-Leverett analysis (2/4)



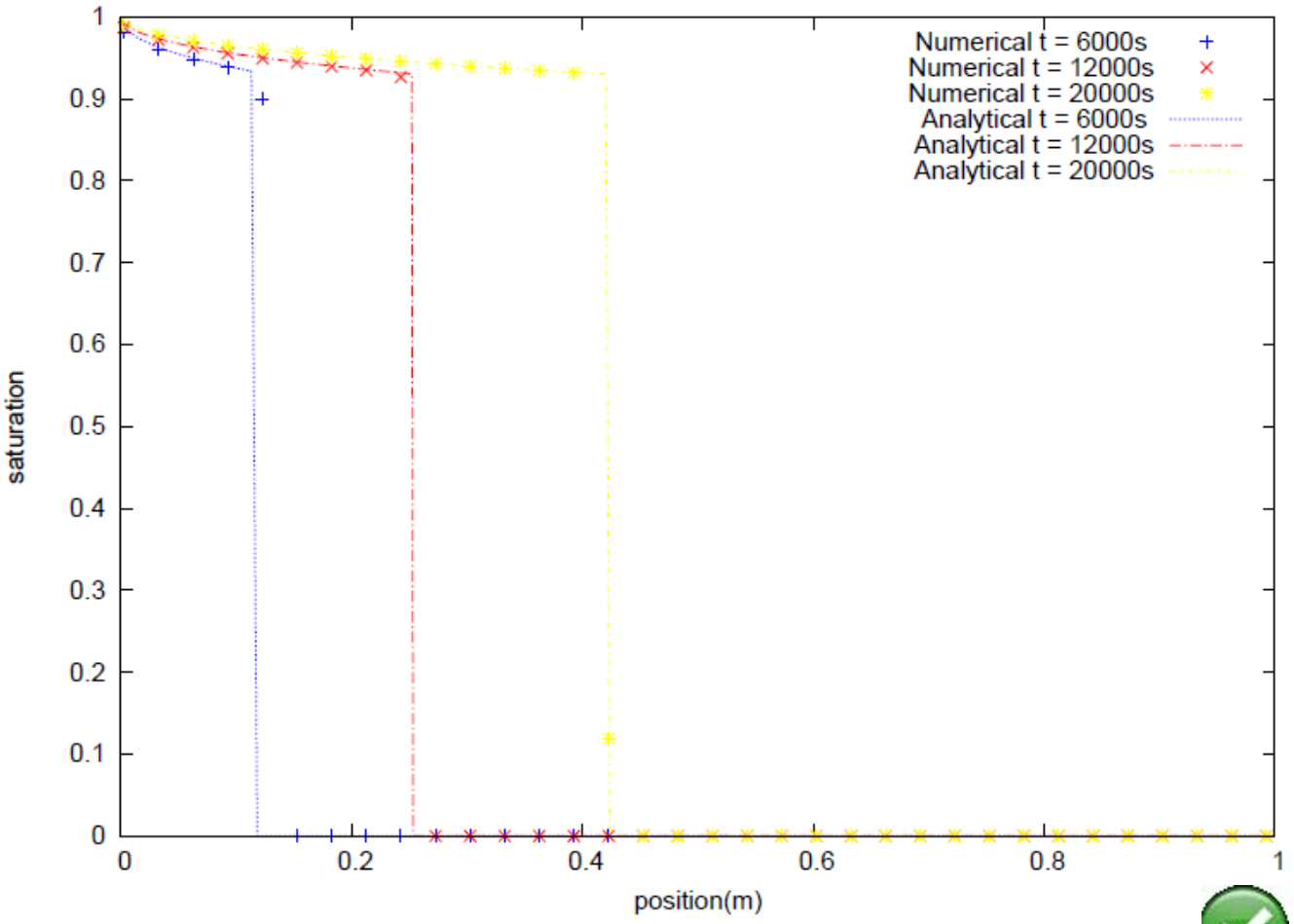
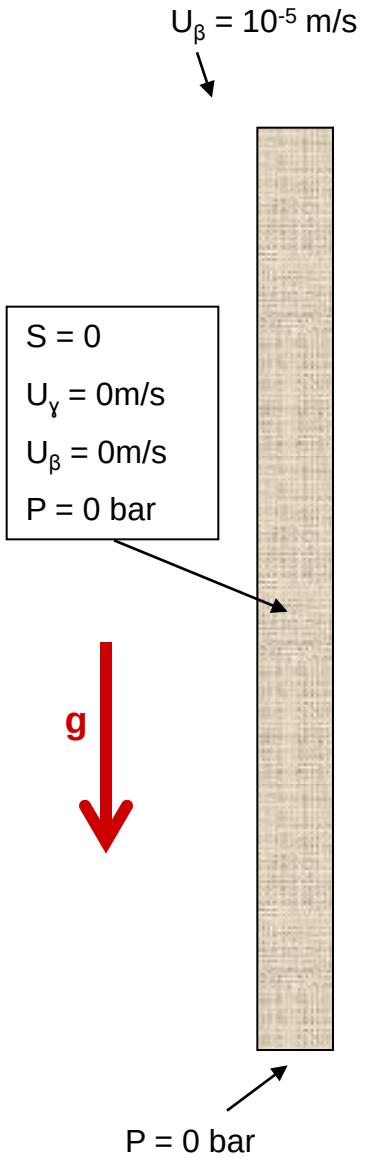


Figure 2: Buckley-Leverett test case : $U_0 = 10^{-5} m/s$ and $g_z = 0 m/s^2$

Validation: Buckley-Leverett analysis (3/4)



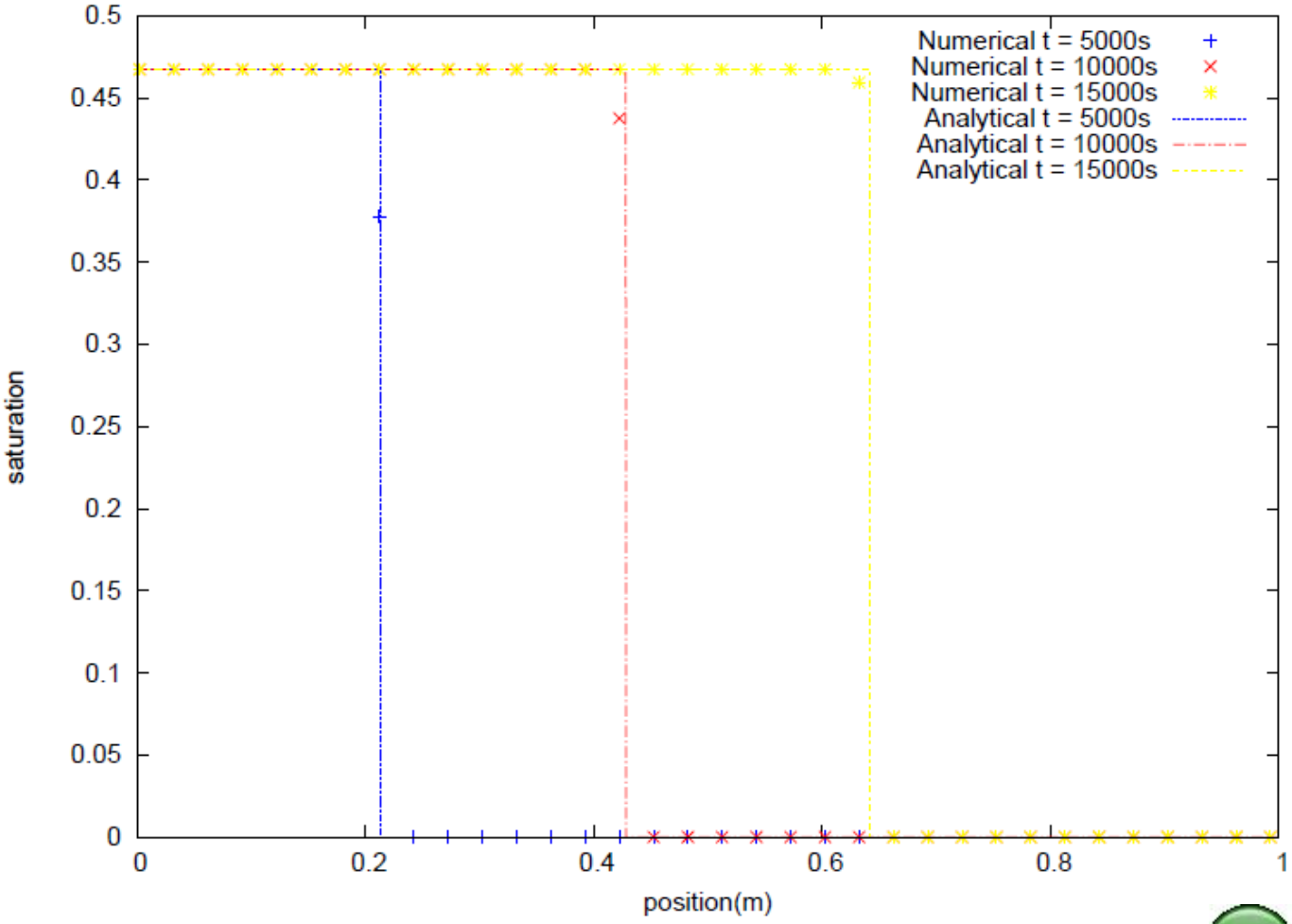
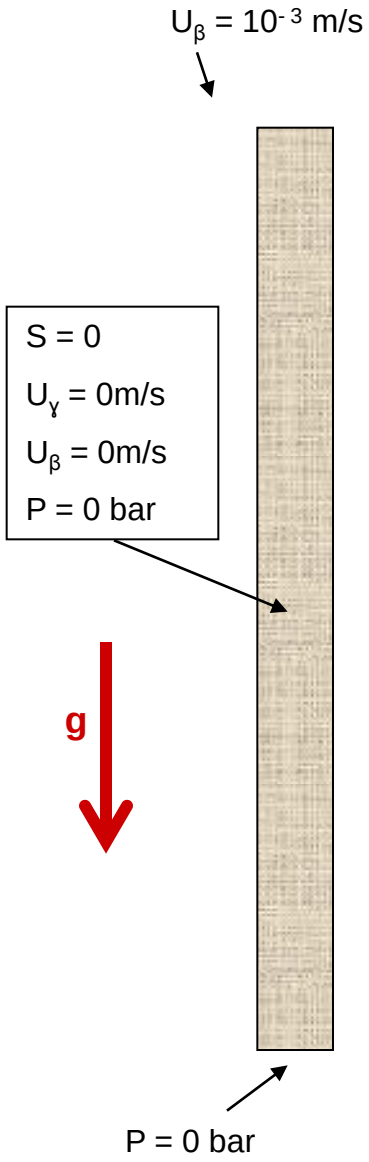


Figure 3: Buckley-Leverett test case : $U_0 = 10^{-5} m/s$ and $g_z = 9.81 m/s^2$

Validation: Buckley-Leverett analysis (4/4)



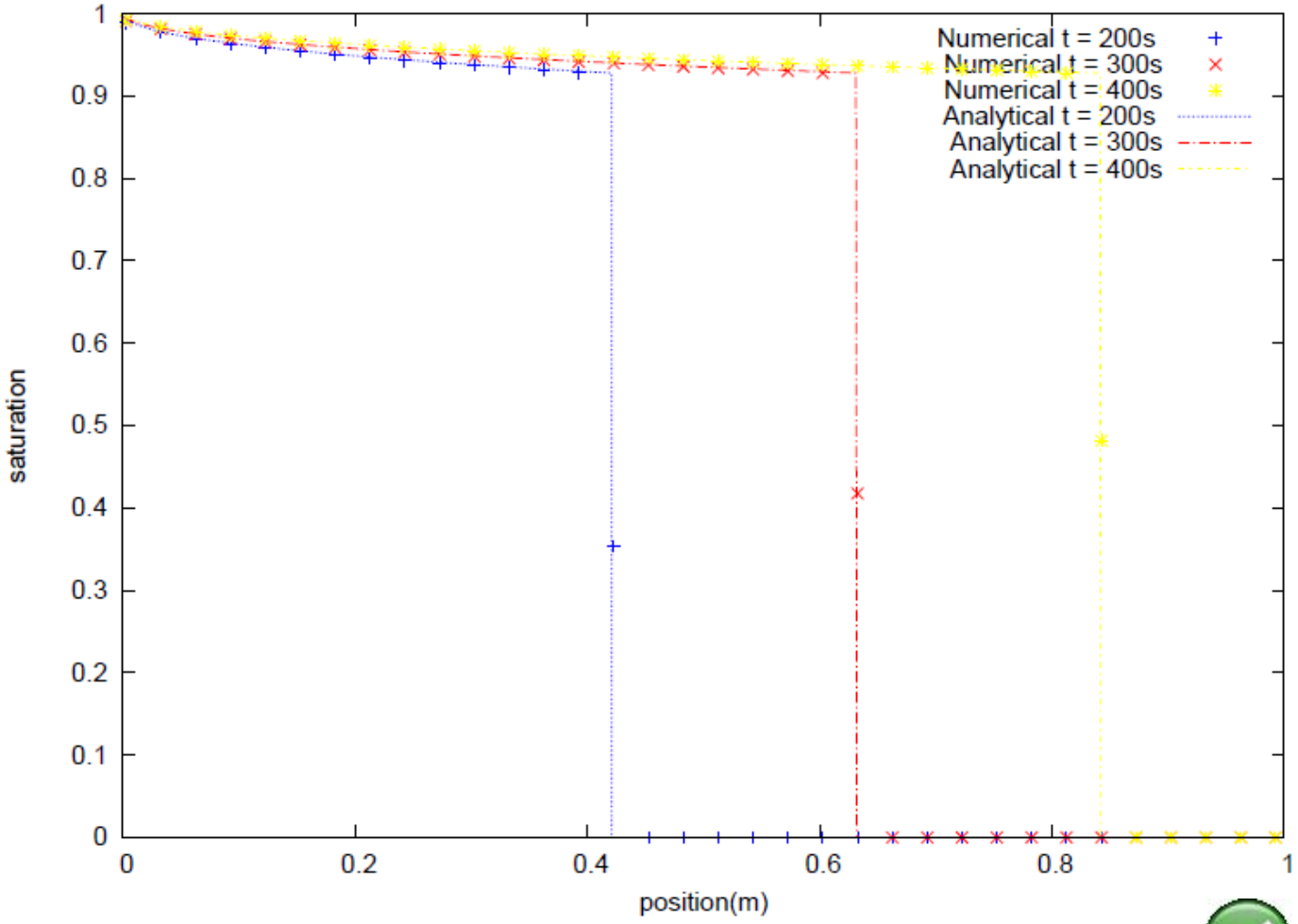
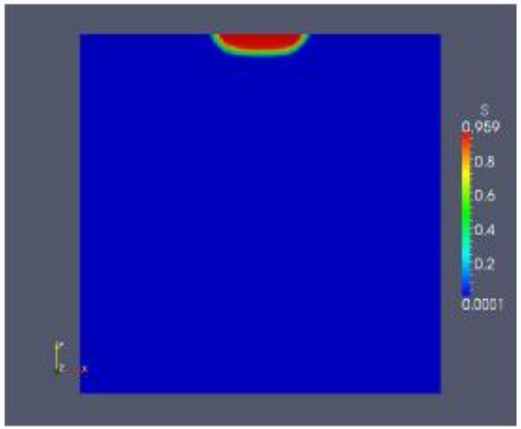


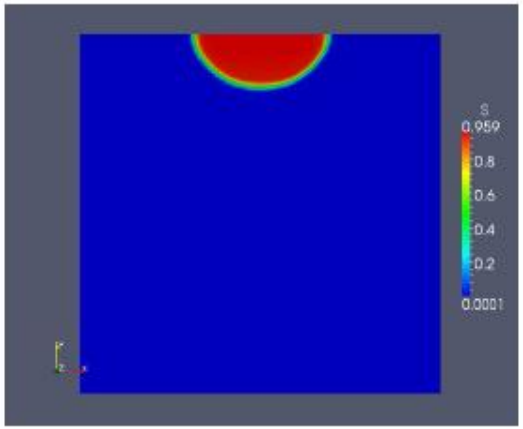
Figure 4: Buckley-Leverett test case : $U_0 = 10^{-3} m/s$ and $g_z = 9.81 m/s^2$

Simulation examples: 2D isotropic medium (1/2)

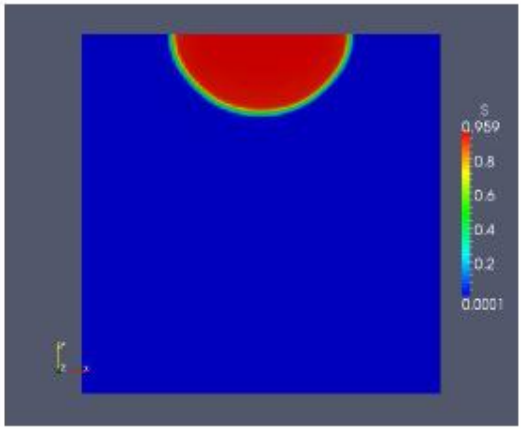
Simulation with no gravity ($U_0=10^{-5}$ m/s)



(a) t = 2500s

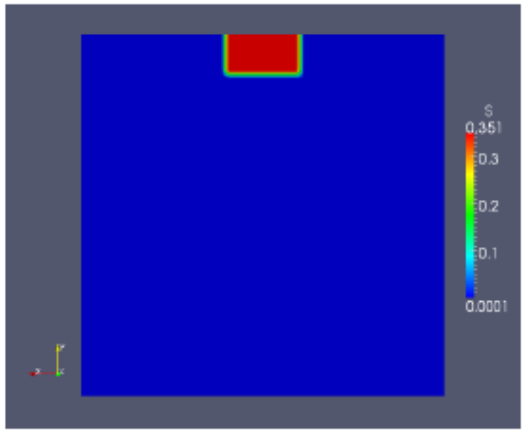


(b) t = 10000s

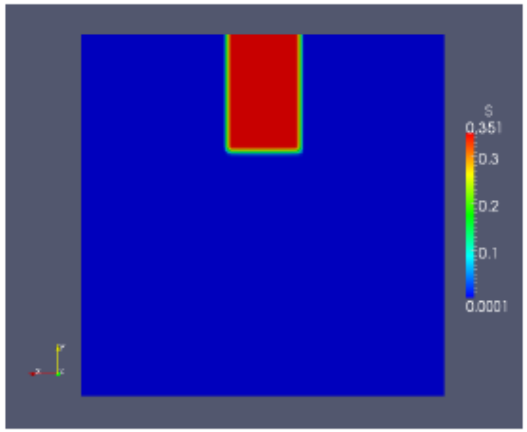


(c) t = 20000s

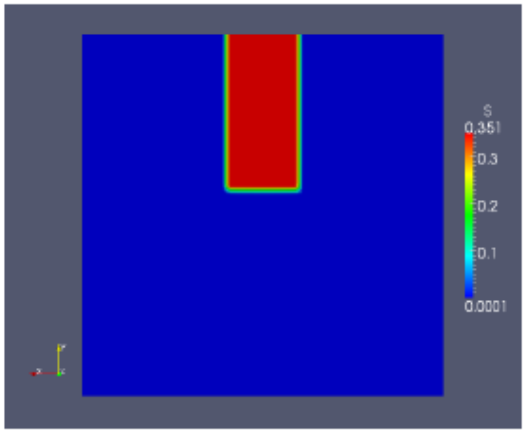
Simulation with gravity ($U_0=10^{-5}$ m/s -> gravity flow)



(a) t = 2500s



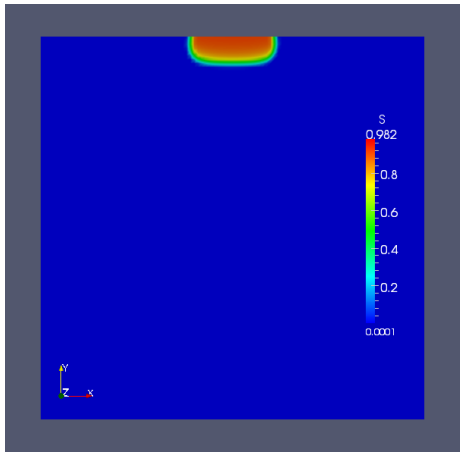
(b) t = 7500s



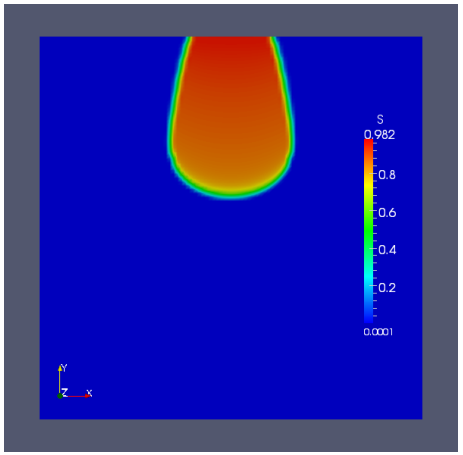
(c) t = 20000s

Simulation examples: 2D isotropic medium (2/2)

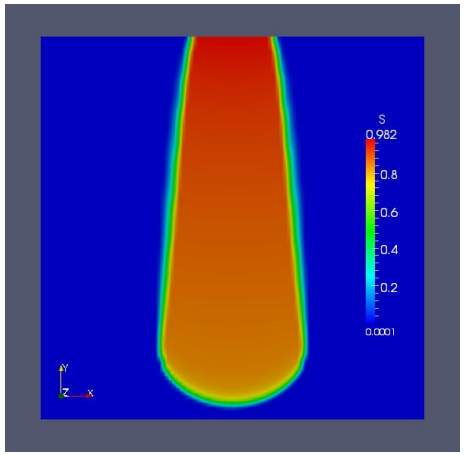
Simulation with gravity ($U_0=10^{-4}$ m/s)



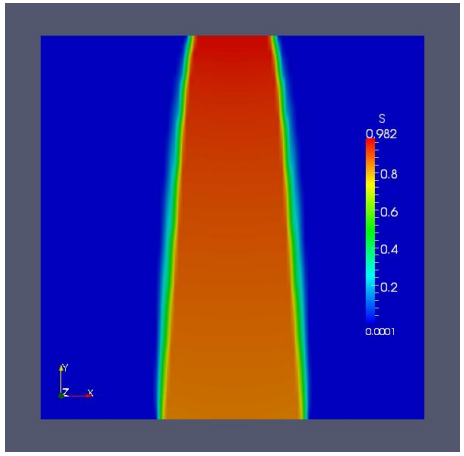
t=300s



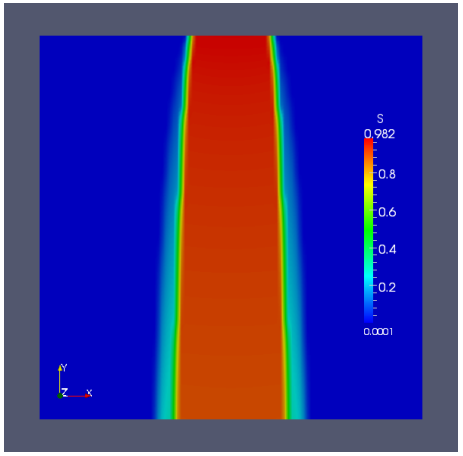
t=2500s



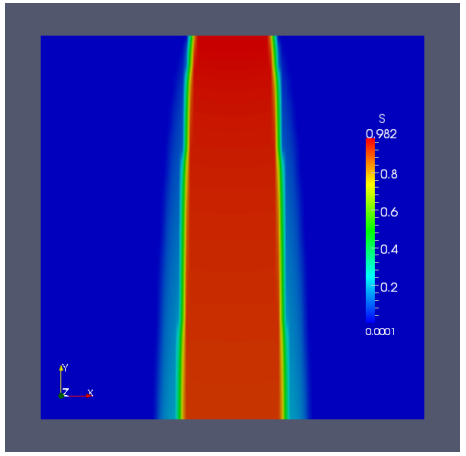
t=6500s



t=8000s



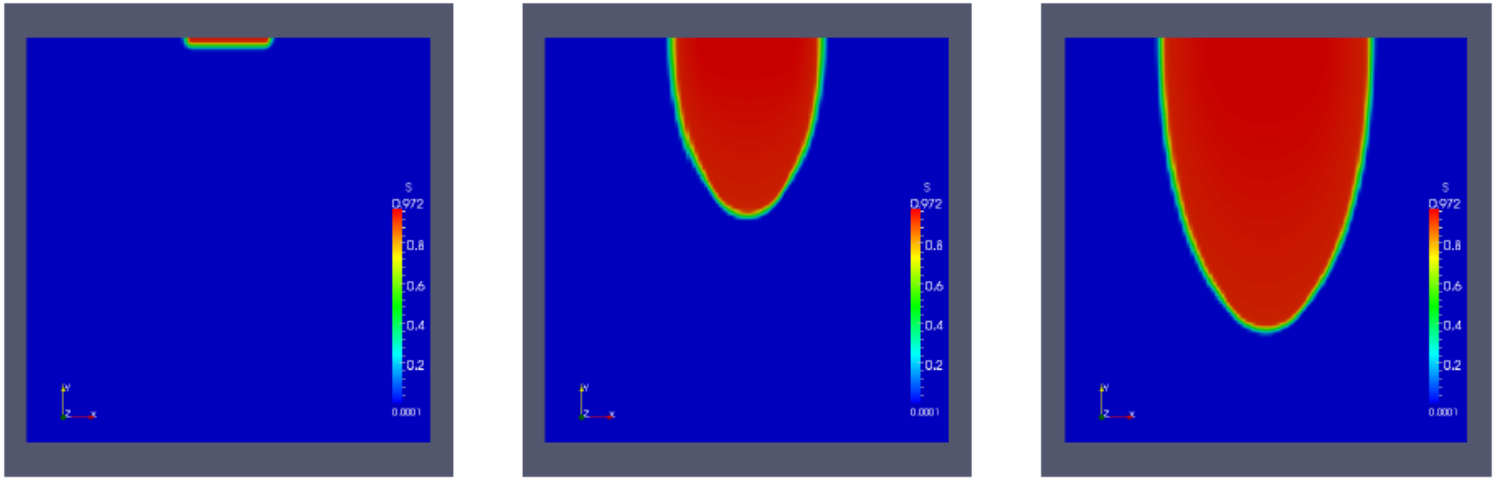
t=20000s



t=40000s

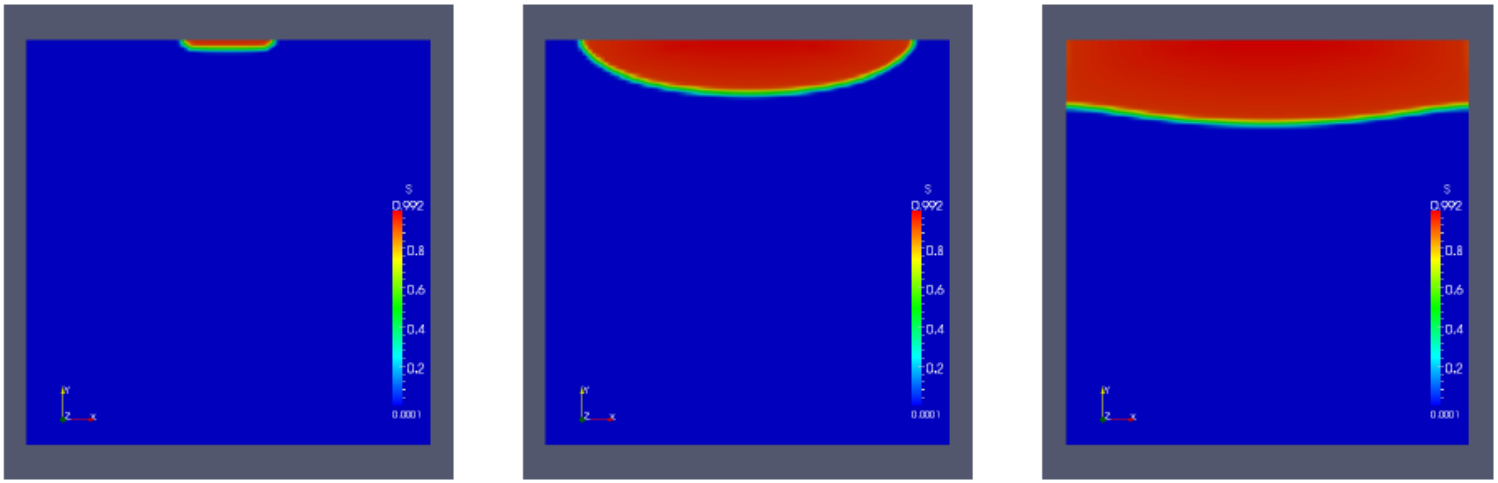
Simulation examples: 2D anisotropic medium

Set of values: $K_{xx}=10^{-11}m^2$; $K_{yy}=10^{-10}m^2$; $K_{xy}=K_{yx}=0m^2$ ($U_0=10^{-4}$ m/s; g=0)



(a) t = 100s (b) t = 3000s (c) t = 7000s

Set of values: $K_{xx}=10^{-10}m^2$; $K_{yy}=10^{-11}m^2$; $K_{xy}=K_{yx}=0m^2$ ($U_0=10^{-4}$ m/s)



(a) t = 100s (b) t = 2000s (c) t = 4500s

Conclusions and perspectives

Summary:

- Two-phase flow solver for anisotropic porous media
- Validations through a Buckley-Leverett analysis
- Simulation examples

Next steps:

- Improvement of the CPU time using a sub time-step for the saturation equation
- Program pressure boundary conditions in a more elegant way
- Program libraries to account for different models of the capillary pressure and different models of relative permeabilities
- The extension to 3-phase (or more) flow in porous media is easy

Thank you for your attention...

* cyprien.soulaine@imft.fr
CFD online nickname: Cyp