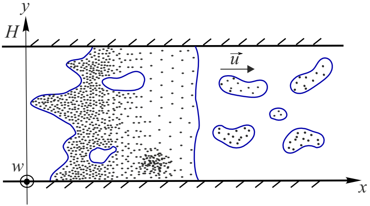
Modelling slurry flow in a hydraulic fracture: from mathematical model to deployable software

Lavrentyev Institute of Hydrodynamics SB RAS
Kovtunenko P., Baykin A., Skopintsev A., Ermishina V.

Novosibirsk State University, Akademgorodok,
2nd of July, 2019

Mathematical model



Governing equations describing the flow

$$\frac{\partial w}{\partial t} + \nabla \cdot (w\mathbf{u}) = -q_l \left(1 - \frac{c_p}{c_{max}}\right); \quad \frac{\partial (c_p w)}{\partial t} + \nabla \cdot (c_p w \mathbf{u}_p) = 0.$$

The effective viscosy approach with single velocity will be used here:

$$\mathbf{u} = \mathbf{u}_f = \mathbf{u}_p = -\Lambda(|\nabla p|, c_p, w)(\nabla p + \rho g e_y); \quad \Lambda = \frac{w^2}{12\mu(c_p)}. \qquad (1)$$

Discretization

For the sake of brevity, we don't consider proppant settling velocity and the gravity terms are also omited. This leads to the following equations:

$$\frac{\partial}{\partial x}\Big(\Lambda \frac{\partial p}{\partial x}\Big) + \frac{\partial}{\partial y}\Big(\Lambda \frac{\partial p}{\partial y}\Big) = \bar{f}, \tag{2}$$

$$\frac{\partial}{\partial t}(wc_p) + \nabla \cdot (wc_p \mathbf{u}) = 0. \tag{3}$$

Introducing the computational grid, (2) is integrated over each cell (FVM):

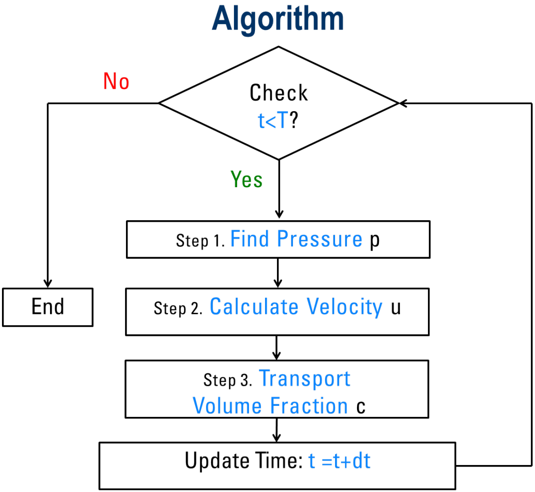
$$p_{i-1,j}\Lambda_{i-\frac{1}{2},j}\frac{h_y}{h_x} + p_{i,j-1}\Lambda_{i,j-\frac{1}{2}}\frac{h_x}{h_y} + p_{i+1,j}\Lambda_{i+\frac{1}{2},j}\frac{h_y}{h_x} + p_{i,j+1}\Lambda_{i,j+\frac{1}{2}}\frac{h_x}{h_y} -$$
$$-p_{i,j}\left[(\Lambda_{i-\frac{1}{2},j} + \Lambda_{i+\frac{1}{2},j})\frac{h_y}{h_x} + (\Lambda_{i,j-\frac{1}{2}} + \Lambda_{i,j+\frac{1}{2}})\frac{h_x}{h_y}\right] = \bar{f}_{i,j} h_x h_y.$$

Proppant transport (3) is modelled using numerical scheme (by R.J. Leveque):

$$(cw)^{n+1}_{i,j} = (cw)^n_{i,j} - \frac{\Delta t}{h_x}(F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}) - \frac{\Delta t}{h_y}(G_{i,j+\frac{1}{2}} - G_{i,j-\frac{1}{2}}).$$

These equations are tied by a certain formula for effective viscosity $\mu = \mu(c)$.

Application and extensions



Application and extensions

Features and bottlenecks

- The implemented algorithm provides flux coherence (stencil of the FVM adjusts on border nodes)
- To find the pressure distribution SLAE must be solved
- The SLAE matrix is a sparse 5-diagonal matrix, but very poorly conditioned
- Big sensitivity to volume balance violation (on poisson equation part)
- No ghost cells are needed, but boundary cells must be dealt with caution
- The problem of implementing fracture geometry

For this very simple model numerous extensions can be applied:

- Non–Newtonian fluids
- Proppant settling
- Transition to a multiphase model (fluids and proppants)
- Proppant bridging criterion
- Thermodynamical effects
- etc.

Implementation

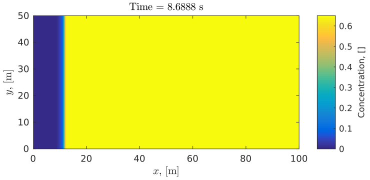
The model has been coded using C++, cmake, python, Gtest, git. By choosing this stack we benefit from power and flexibility in exchange for complexity

- Cross-platform development and deployment
- Ability to use external libraries for acceleration
- Automated testing
- Pack into .dll or use the standalone version

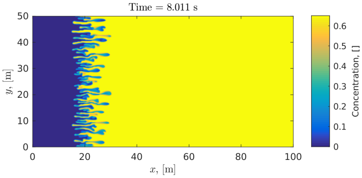
Tips painfully derived from our team's experience

- Always check the math (we are still doing science)
- Keep in mind technical complexity (programming is a craft)
- Create architecture before-head, use best programming practices
- Use modern tools and compilers, don't neglect tests, try python
- Create the most simple and versatile (but ugly) interface to the module
- Avoid writing your own standard or common methods and algorithms
- Use git

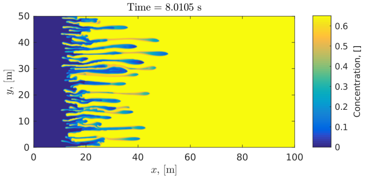
Viscosity relation μ_2/μ_1 effects the instability. $t \approx 8$s



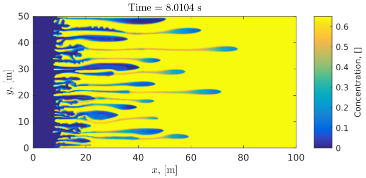
(а) $\mu_2/\mu_1 = 0.1$



(б) $\mu_2/\mu_1 = 10$

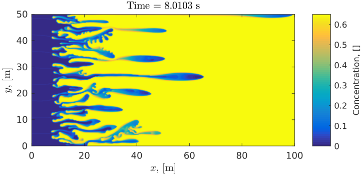


(в) $\mu_2/\mu_1 = 31$

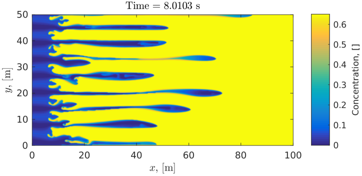


(г) $\mu_2/\mu_1 = 100$

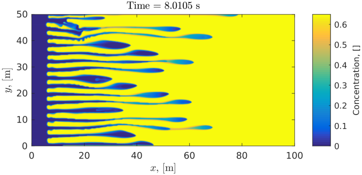
Proppant concentration perturbation $c_p = \varepsilon c_{max} \left(\sin\left(\frac{2\pi \mathbf{m} y}{L_y} \right) + 1 \right)$ affecting the formation of viscous fingering. $\mu_2/\mu_1 = 31$, $t \approx 8$c.



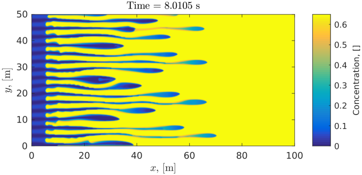
(a) $m = 8; \varepsilon = 0.01$



(b) $m = 8; \varepsilon = 0.05$

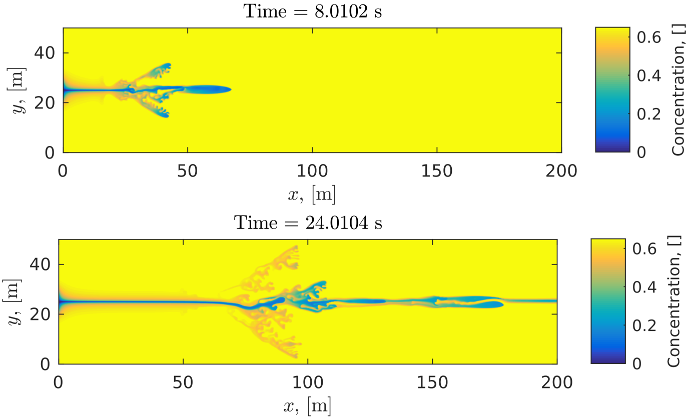


(c) $m = 16; \varepsilon = 0.01$



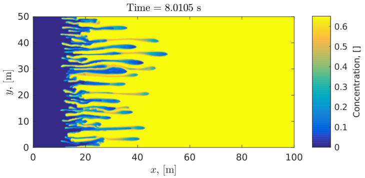
(d) $m = 16; \varepsilon = 0.05$

Single finger propagation

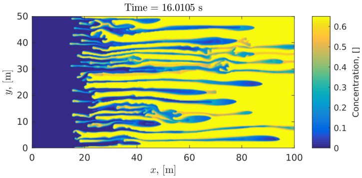


Rough walls simulation

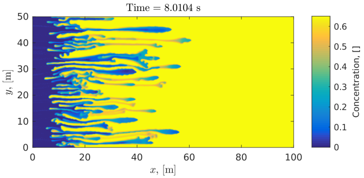
Random walls perturbations with amplitude ε^*.



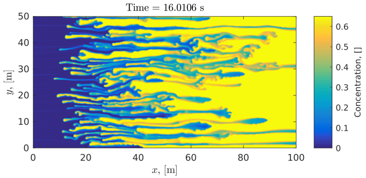
(a) $\varepsilon^* = 0$; $t \approx 8$ c



(b) $\varepsilon^* = 0$; $t \approx 16$ c

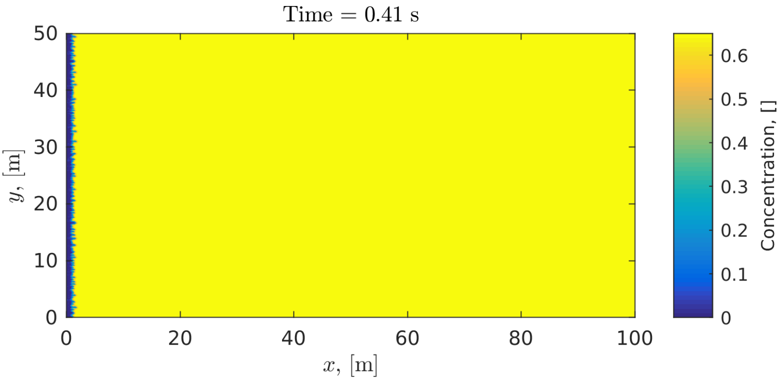


(c) $\varepsilon^* = 0.05$; $t \approx 8$ c



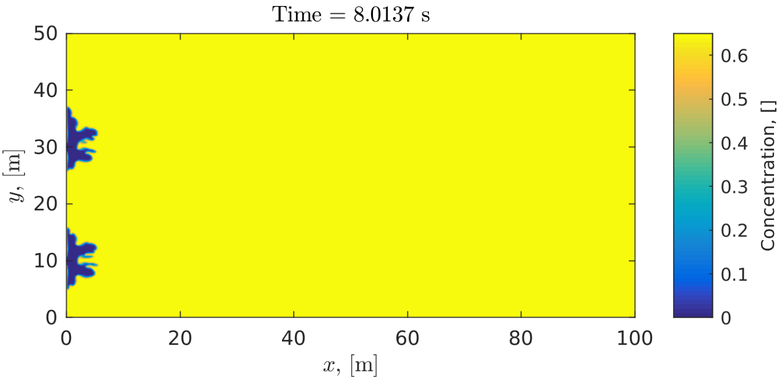
(d) $\varepsilon^* = 0.05$; $t \approx 16$ c

Rough walls simulation



perturbation amplitude $\varepsilon^* = 0.05$, viscosity relation$\mu_2/\mu_1 = 31$

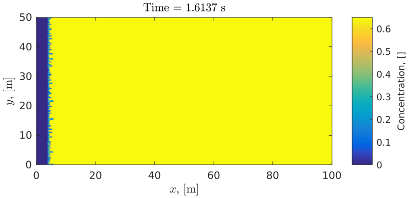
Fluid inflow through a perforation



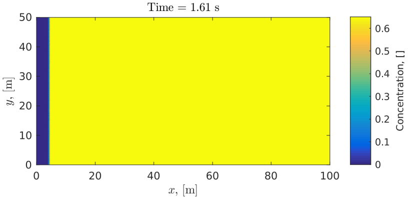
Viscosity relation is $\mu_2/\mu_1 = 31$

Non-Newtonian fluid

Fluid with power-law rheology, consistency indexes contrast is $K_2/K_1 = 100$

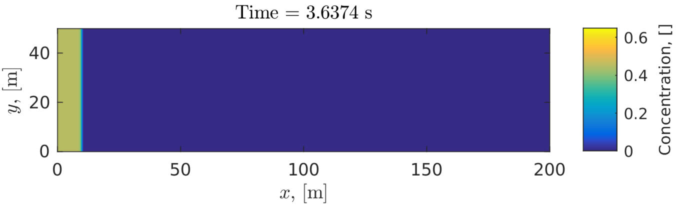


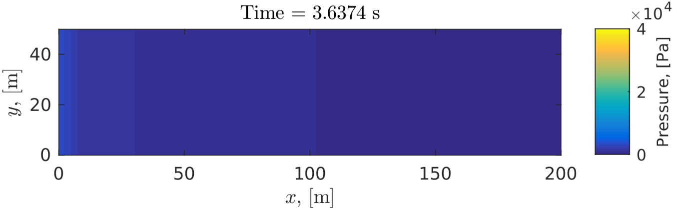
(a) n = 0.5



(b) n = 0.7

Non zero Leak-off

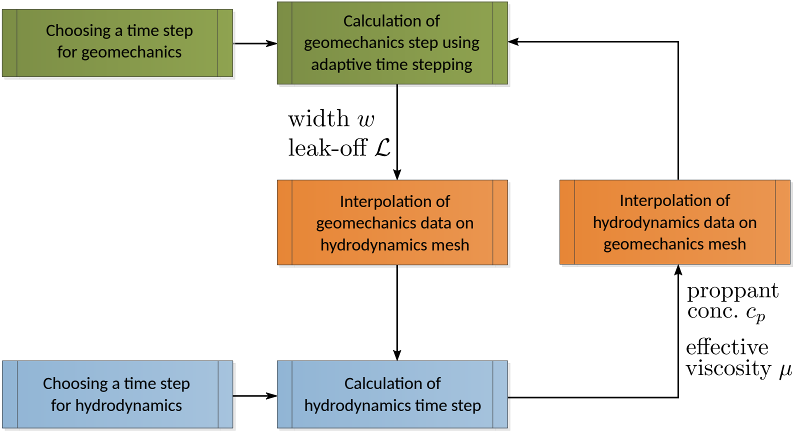




Coupling with the geomechanical model

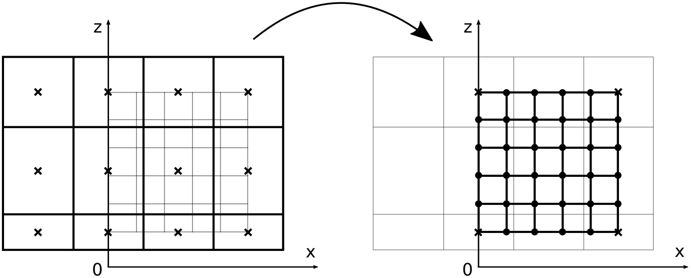
We try to keep the hydrodynamics module architecturally and logically decoupled from other modules, the coupling has been performed with:

- Planar 3D ILSA model (report by Valov A.)
- EP3D model (report by Skopintsev A.)



Coupling with the geomechanics: problematics

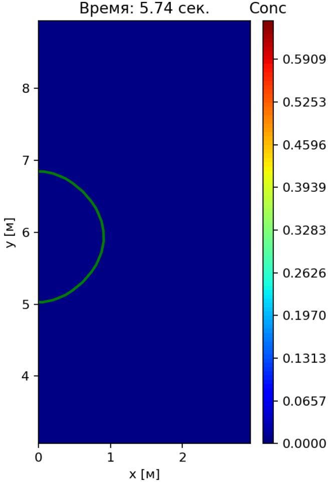
Interpolating data is essential. Bilinear algorithm can be considered, but a more advanced conservative method may become crucial.



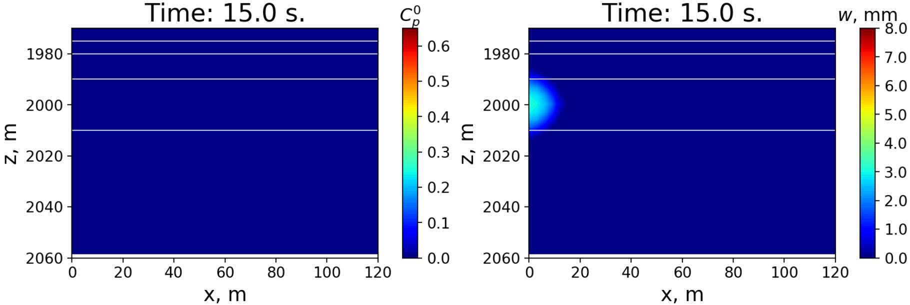
- Volume balance might be violated during interpolation
- Bringing fracture geometry onto hydrodynamics mesh
- Calculating on various time scales
- Duplicated calculations, starting with Poisson equation
- General lack of data in each module

Radial fracture with leak-off

Volumetric proppant concentration on the inlet is $c_p = 0.55$.



Pulsative injection



$E = 22$ GPa, $\nu = 0.23$, $\mu = 0.05$ Pa·s, $K_{I_C} = 1$ MPa·m$^{1/2}$,
$C_l = 2.0 \cdot 10^{-5}$ m·s$^{-1/2}$, $Q = 0.06$ m^3/s

Thank you for your attention!