SedFoam

An open-source multi-dimensional two-phase flow model for sediment transport applications

THESIS symposium
University of Delaware, DE (US)

Julien CHAUCHAT, **Cyrille BONAMY and Antoine MATHIEU**
LEGI, Grenoble Institute of Technology, CNRS, Grenoble, France

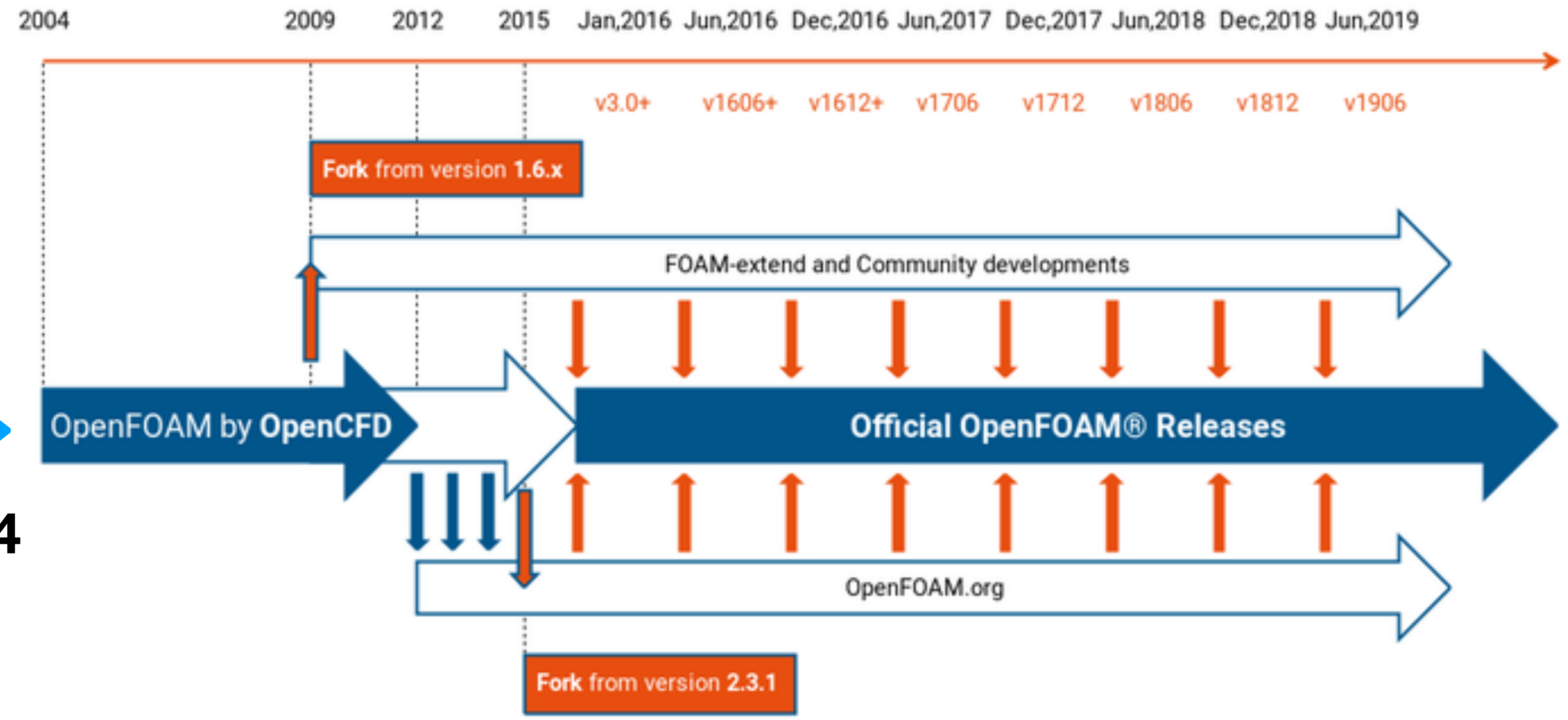
A brief history of openFOAM: Field Operation And Manipulation

FOAM

Henry Weller
JASAK Hrvoje

Nabal Ltd
Imperial College (UK)

1998 → **2004**



- **A set of top level classes for finite volume on unstructured grids**

- **User defined solvers based on top level classes: algorithms are written in a math-like syntax**

Example: PISO algorithm for incompressible Navier-Stokes equations

Equations

The solver uses the **PISO** algorithm to solve the continuity equation:

$$\nabla \cdot \mathbf{u} = 0$$

and momentum equation:

$$\frac{\partial}{\partial t}(\mathbf{u}) + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \nabla \cdot (\nu \nabla \mathbf{u}) = -\nabla p$$

Where:

$\mathbf{u}$ = Velocity

p = Kinematic pressure

```
//set up the linear algebra for the momentum
equation.  The flux of U, phi, is treated explicity
//using the last known value of U.

        fvVectorMatrix UEqn
        (
            fvm::ddt(U)
          + fvm::div(phi, U)
          - fvm::laplacian(nu, U)
        );

// solve using the last known value of p on the
RHS.  This gives us a velocity field that is
// not divergence free, but approximately satisfies
momentum.  See Eqn. 7.31 of Ferziger & Peric
        solve(UEqn == -fvc::grad(p));
```

The PISO algorithm consists in building an elliptic equation for the pressure to ensure the velocity field is divergence free *(not shown here)*

A brief history

collaboration UD/LEGI

twoPhaseEulerFoam — Henrik Rusche PhD at IC — 2002

sedFOAM-1.0 — Zhen Cheng PhD at UD — 2011

2014

sedFOAM-2.0 — 2015

Tim Nagel PhD at LEGI

Yeulwoo Kim PhD at UD

sedFOAM-3.x — 2019

Antoine Mathieu PhD at LEGI

Ali Salimi PhD at UD

Benjamin Tsai PhD at UD

LEGI: J. Chauchat, C. Bonamy, T. Nagel, A. Mathieu, H. Rousseau

University of Delaware: T.-J. Hsu, Z. Cheng, Y. Kim, A. Salimi Tarazouj, B. Tsai

PennState: X. Liu

ONR: S. Bateman

P. Higuera (U. Singapore)

G. Keetels (U. Delft)

Motivations

Sediment transport during floods

Salmon River during the 1964 flood

Scour around hydraulic structures

O, Link (UdeC, Chili)

Storms at coast

Sand Engine - The Netherland

Luijendijk et al. CE (2017)

Modeling approaches

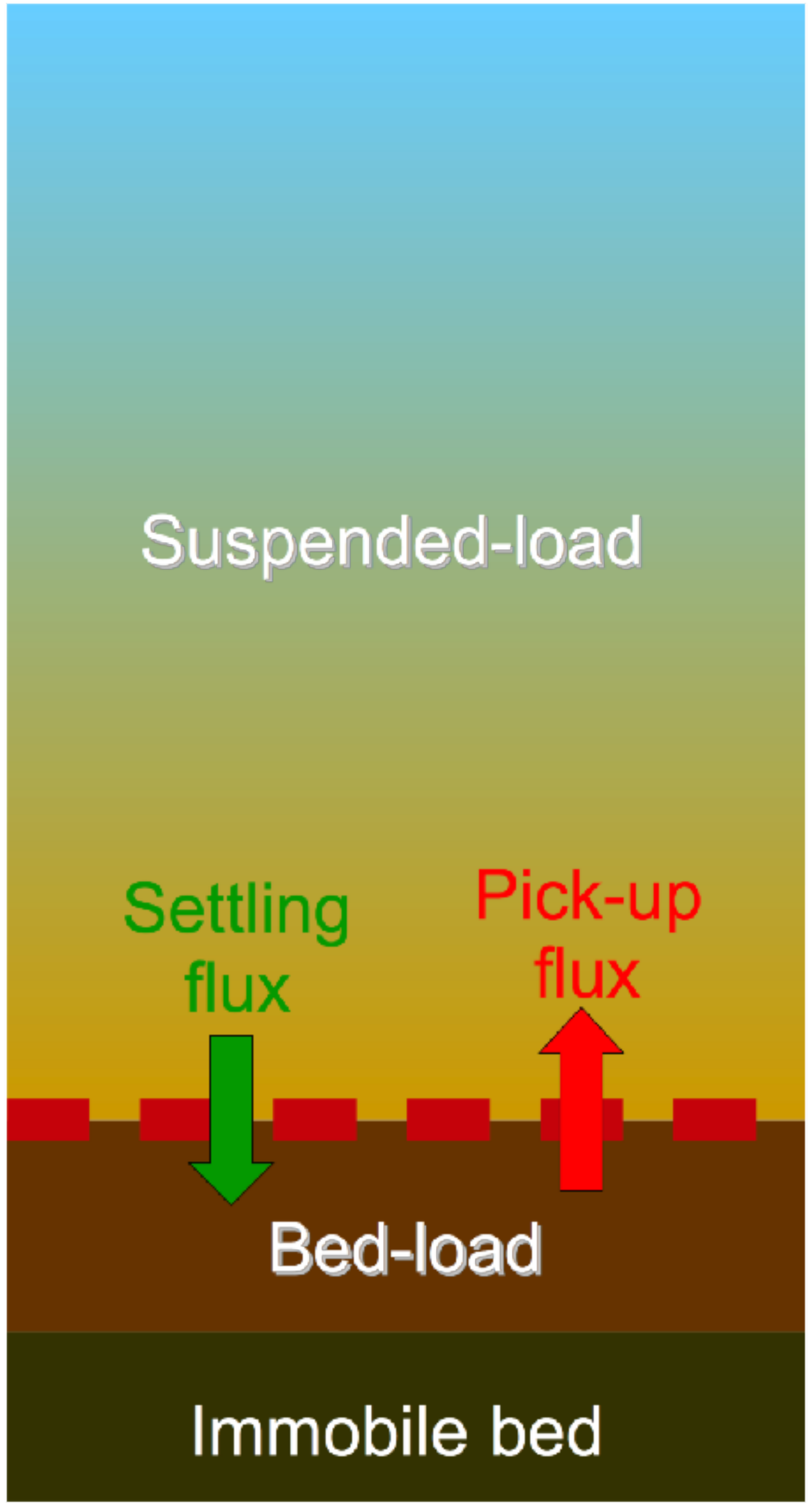
Pros

- Simple
- Applicable at large-scale

Cons

- Empirical formulas
  - large scatter
  - Missing physics
- Arbitrary separation between bed-load and suspended-load

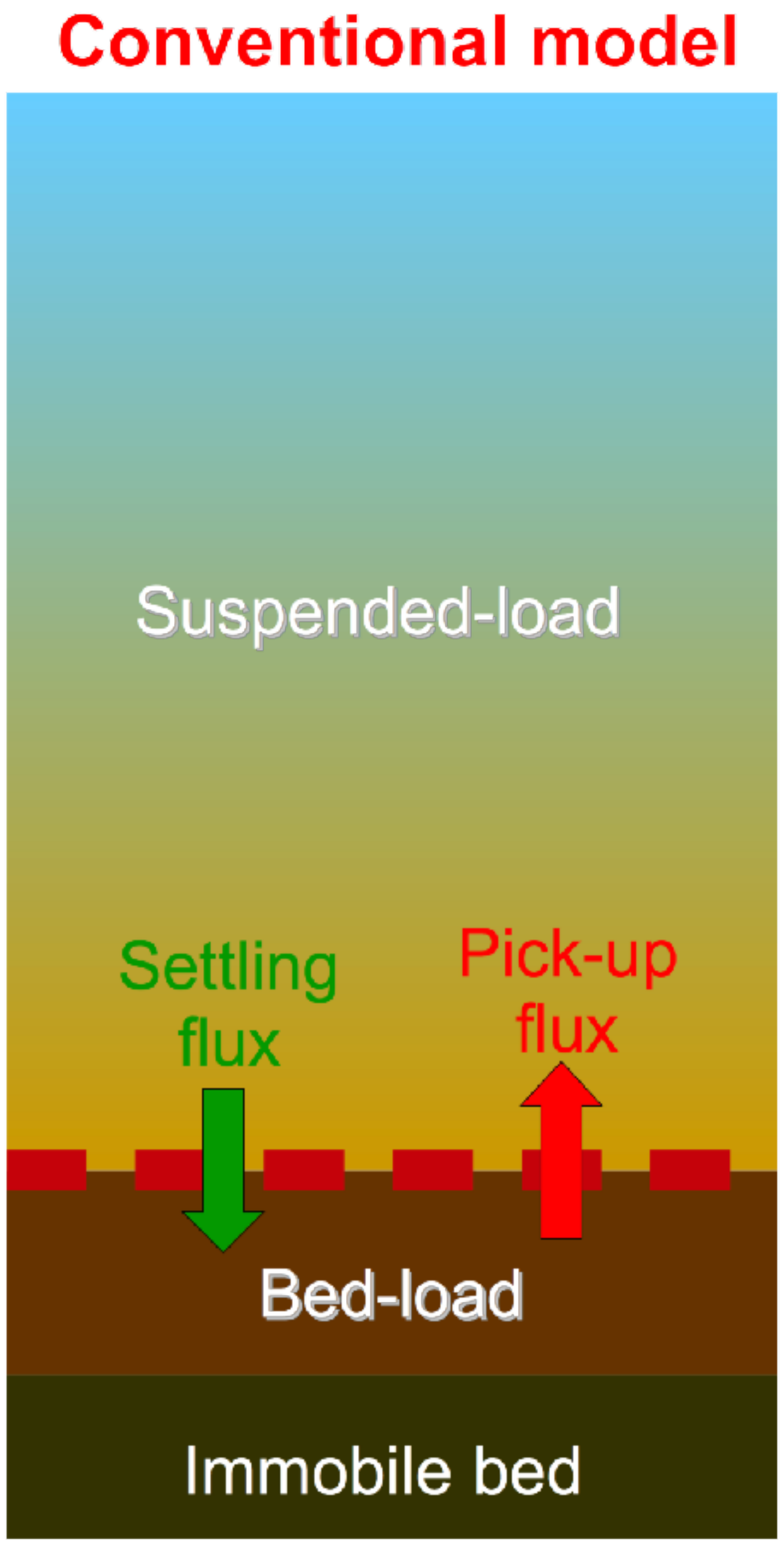
Conventional model

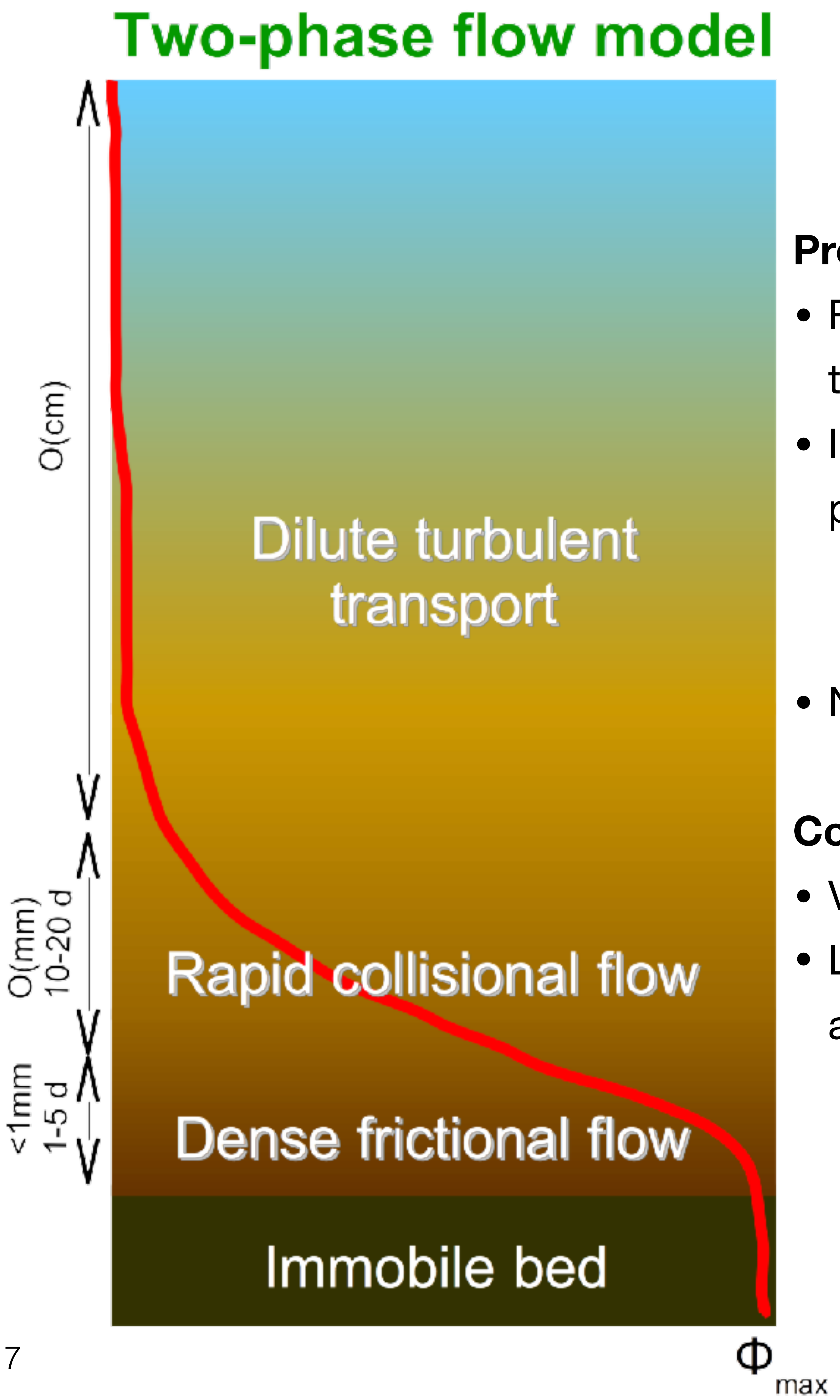


Modeling approaches

Pros
- Simple
- Applicable at large-scale

Cons
- Empirical formulas
  - large scatter
  - Missing physics
- Arbitrary separation between bed-load and suspended-load





Pros
- Resolve continuously sediment transport profile
- Incorporate fine-scale processes:
  - Turbulence
  - Particle-particle interactions
- No arbitrary separation

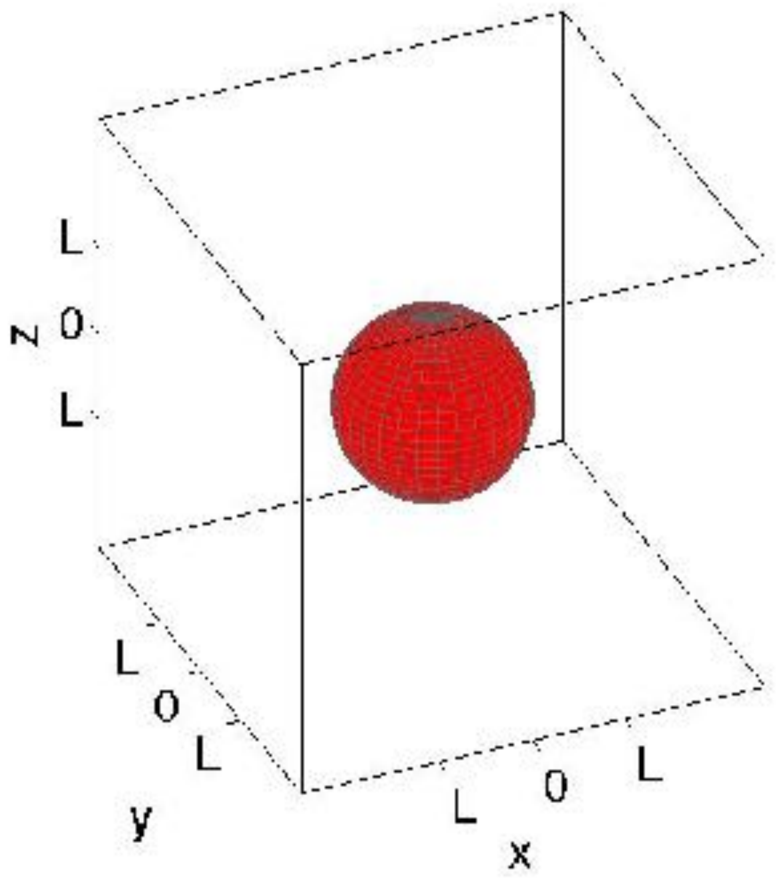
Cons
- Very expensive
- Limited to ‘small scale’ applications

Jenkins and Hanes JFM (1998)

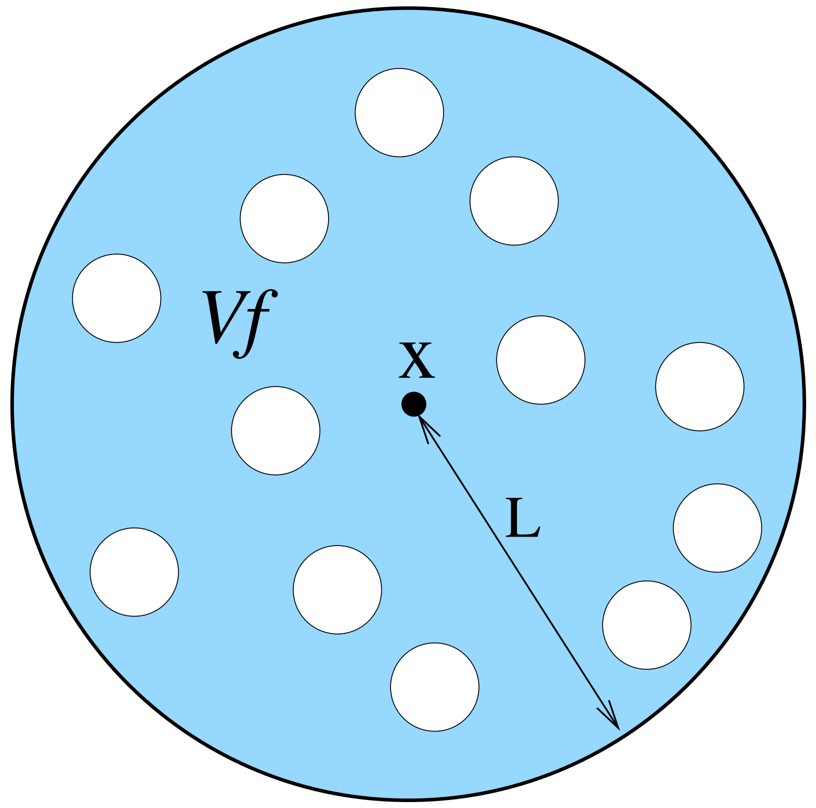
Eulerian-Eulerian two-phase flow equations

Local mass & momentum conservation for a fluid-particle mixture

$$\nabla.\vec{u} = 0 \quad \text{and} \quad \frac{d\rho\vec{u}}{dt} + \nabla.(\rho\vec{u} \otimes \vec{u}) = \nabla.\overline{\overline{\sigma}} + \rho\vec{g}$$

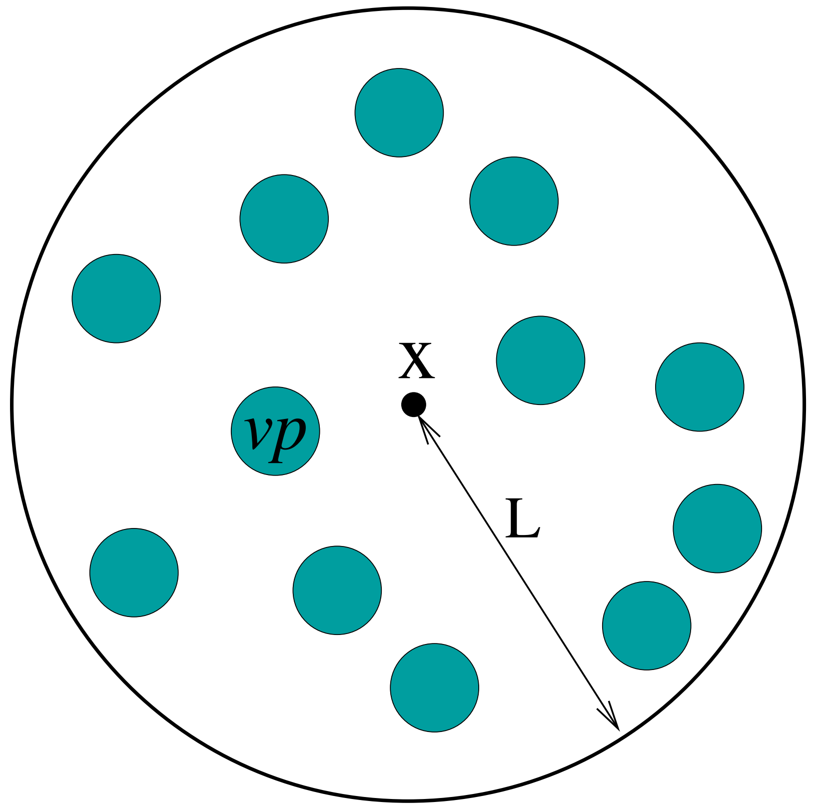


G_L is a 3D door function



Local spatial averaging

Jackson (2000)



$$\langle f \rangle^f (\vec{x}, t) = \frac{1}{\epsilon} \int_{V_f(t)} f(\vec{y}, t) G_L(|\vec{x} - \vec{y}|) dV_y$$

$$\langle f \rangle^p (\vec{x}, t) = \frac{1}{\phi} \sum_p \int_{v_p} f(\vec{y}, t) G_L(|\vec{x} - \vec{y}|) dv_y$$

Fluid phase mass and momentum equations

$$\frac{\partial \epsilon}{\partial t} + \nabla.\left(\epsilon \vec{u^f}\right) = 0$$

$$\rho_f \left[\frac{\partial \epsilon \vec{u^f}}{\partial t} + \nabla.\left(\epsilon \vec{u^f} \otimes \vec{u^f}\right)\right] = \nabla.\overline{\overline{\sigma^f}} - n\vec{f} + \epsilon\rho_f\vec{g}$$

Solid phase mass and momentum equations

$$\frac{\partial \phi}{\partial t} + \nabla.\left(\phi \vec{u^p}\right) = 0$$

$$\rho_p \left[\frac{\partial \phi \vec{u^p}}{\partial t} + \nabla.\left(\phi \vec{u^p} \otimes \vec{u^p}\right)\right] = \nabla.\overline{\overline{\sigma^p}} + n\vec{f} + \phi\rho_p\vec{g}$$

Governing equations

Effective fluid stress
= include particle perturbations

Fluid phase equations

$$\frac{\partial \epsilon}{\partial t} + \nabla.\left(\epsilon \vec{u^f}\right) = 0$$

$$\rho_f \left[\frac{\partial \epsilon \vec{u^f}}{\partial t} + \nabla.\left(\epsilon \vec{u^f} \otimes \vec{u^f}\right)\right] = -\nabla p^f + \nabla.\overline{\overline{\tau^f}} - n\vec{f} + \epsilon \rho_f \vec{g}$$

Fluid-particle interactions
= fluid flow at the particle scale

Solid phase equations

$$\frac{\partial \phi}{\partial t} + \nabla.\left(\phi \vec{u^p}\right) = 0$$

$$\rho_p \left[\frac{\partial \phi \vec{u^p}}{\partial t} + \nabla.\left(\phi \vec{u^p} \otimes \vec{u^p}\right)\right] = -\nabla p^p + \nabla.\overline{\overline{\tau^p}} + n\vec{f} + \phi \rho_p \vec{g}$$

Granular stresses
= particle-particle interactions

Details of the flow at the particle scale are missing due to averaging

➡ Need to model grain-scale physics

Fluid-particle interactions

$$n\vec{f} = n\vec{f_B} + n\vec{f_D} + \ldots$$

Generalized buoyancy

$$nf_B = \phi \nabla . \left(-p^f \overline{\overline{I}} + \overline{\overline{\tau^f}} \right)$$

Jackson (2000)

Archimede Local fluid acceleration

Drag

$$nf_D = \frac{\phi \ \rho^p}{t_p} \left(\vec{u^f} - \vec{u^p} \right)$$

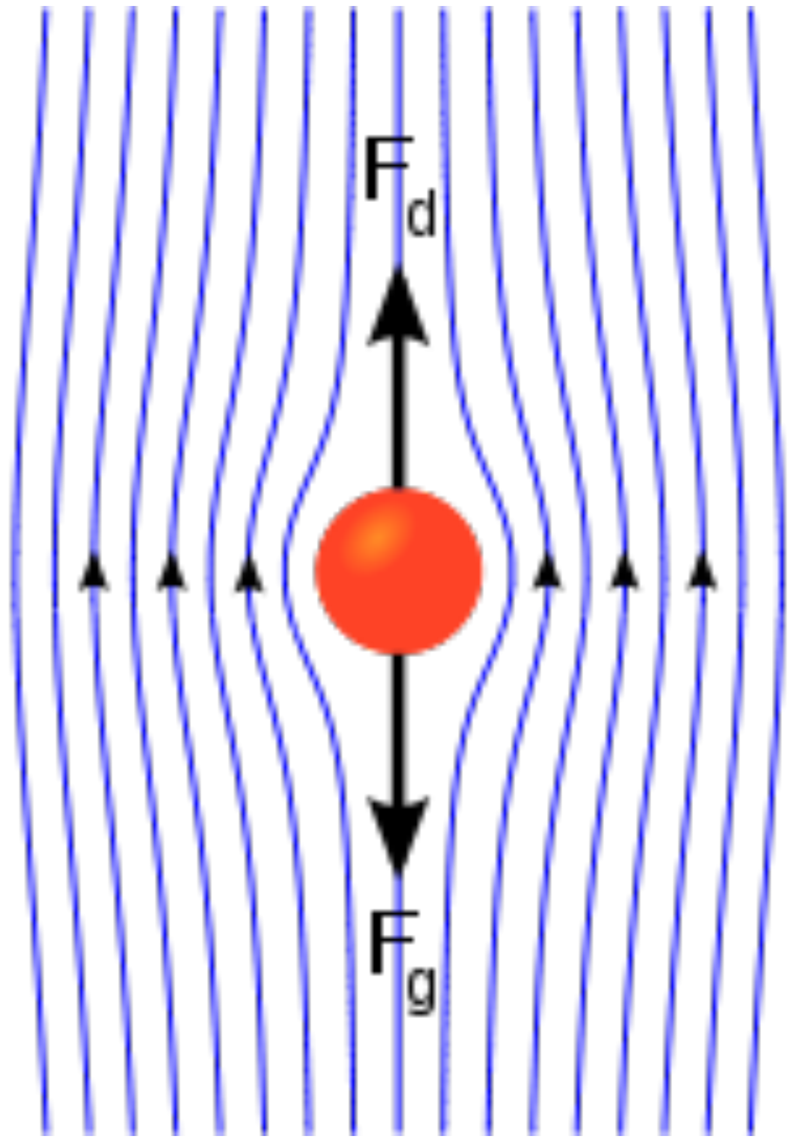
where t_p *is the particle response time*

Stokes drag around a single particle $f_D = 3 \ \pi \ d_p \ \eta^f \ \left(\vec{u^f} - \vec{u^p} \right)$

x particle number density: n

x hindrance function

$$nf_D = \phi\rho^p \underbrace{\frac{18 \ \eta^f}{\rho^p \ d_p^2} (1-\phi)^{-2.65}}_{=1/t_p} \left(\vec{u^f} - \vec{u^p} \right)$$



sedFoam: a 3D two-phase numerical model for sediment transport

- Finite Volume Method
- PISO algorithm for pressure-velocity coupling
- Based on twoPhaseEulerFoam from H. Rusche (2002) implemented in OF-2.4
- Publically available on github: *https://github.com/SedFoam/sedfoam*
- Fluidfoam: a python pre/post-processing package for OpenFOAM *https://bitbucket.org/sedfoam/fluidfoam*
- sedFOAM-3.1 is available and is compatible with OF5.x, OF6, OF7, OF1712+ to OF1906+

Chauchat *et al. (2017) - Geoscientific Model Development*


Geoscientific Model Development
An interactive open-access journal of the European Geosciences Union
EGU.eu | EGU Journals | EGU Highlight Articles | Contact | Imprint |

https://doi.org/10.5194/gmd-2017-101
© Author(s) 2017. This work is distributed under the Creative Commons Attribution 3.0 License.

Discussion papers

Abstract | Discussion | Metrics

Model description paper 07 Jun 2017

SedFoam-2.0: a 3D two-phase flow numerical model for sediment transport

Julien Chauchat[1], Zhen Cheng[2,a], Tim Nagel[1], Cyrille Bonamy[1], and Tian-Jian Hsu[2]
[1]University of Grenoble Alpes, LEGI, G-INP, CNRS, F-38000 Grenoble, France
[2]Civil and Environmental Engineering, Center for Applied Coastal Research, University of Delaware, Newark, DE 19711, USA
[a]now at: Applied Ocean Physics & Engineering, Woods Hole Oceanographic Institution, Woods Hole, MA 02543, USA

Received: 22 Apr 2017 – Accepted for review: 05 Jun 2017 – Discussion started: 07 Jun 2017

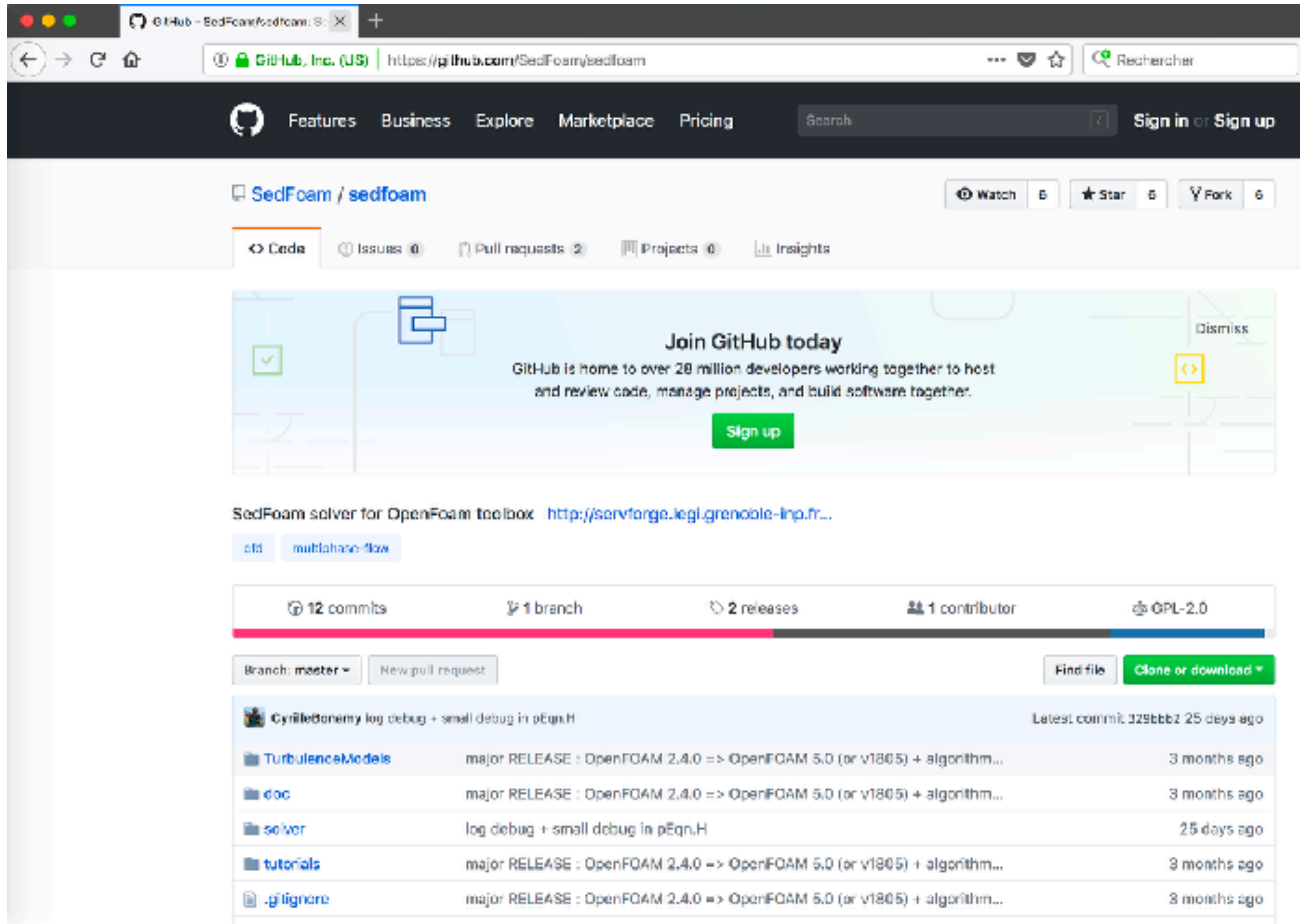
Abstract. In this paper, a three-dimensional two-phase flow solver, SedFoam-2.0, is presented for sediment transport applications. The solver is extended upon twoPhaseEulerFoam available in the 2.1.0 release of the open-source CFD toolbox OpenFOAM. In this approach the sediment phase is modeled as a continuum, and constitutive laws have to be prescribed for the sediment stresses. In the proposed solver, two different inter-granular stress models are implemented: the kinetic theory of granular flows and the dense granular flow rheology $\mu(I)$. For the fluid stress, laminar or turbulent flow regimes can be simulated and three different turbulence models are available for sediment transport: a simple mixing length model (one-dimensional configuration only), a k-ε and a k-ω model. The numerical implementation is first demonstrated by two validation test cases, sedimentation of suspended particles and laminar bed-load. Two applications are then investigated to illustrate the capabilities of SedFoam-2.0 to deal with complex turbulent sediment transport problems with different combinations of inter-granular stress and turbulence models.

Citation: Chauchat, J., Cheng, Z., Nagel, T., Bonamy, C., and Hsu, T.-J.: SedFoam-2.0: a 3D two-phase flow numerical model for sediment transport, Geosci. Model Dev. Discuss., https://doi.org/10.5194/gmd-2017-101, in review, 2017.

Review status
This discussion paper is under review for the journal Geoscientific Model Development (GMD).

Search articles | Download | Short summary: This manuscript presents the development and validation of a community two-phase flow... ▸ Read more | Citation: BibTeX, EndNote | Share

About | Editorial board | Articles | Special issues | Highlight articles | Subscribe to alerts | Peer review | For authors | For editors and referees | User ID | Password | Follow @EGU_GMD | Journal metrics | IF 3.458 | IF 5-year



Installation and technical aspects

BOOT DIRECTLY FROM USB-STICK

- AT LAPTOP STARTUP, BOOT FROM THE USB-STICK :
  F2 OR *DEL* OR …
  (DEPENDING OF LAPTOP)
- ADVANTAGES :
  - NO DOWNLOAD
  - NO VIRTUALISATION -> FASTEST
  - NO VIRTUALBOX SOFTWARE
- DISADVANTAGES :
  - IO ACCESS NOT VERY FAST
  - NO FULL CONTROL
- NOT RECOMMENDED FOR MAC

BOOT USB-STICK VIA VIRTUALBOX

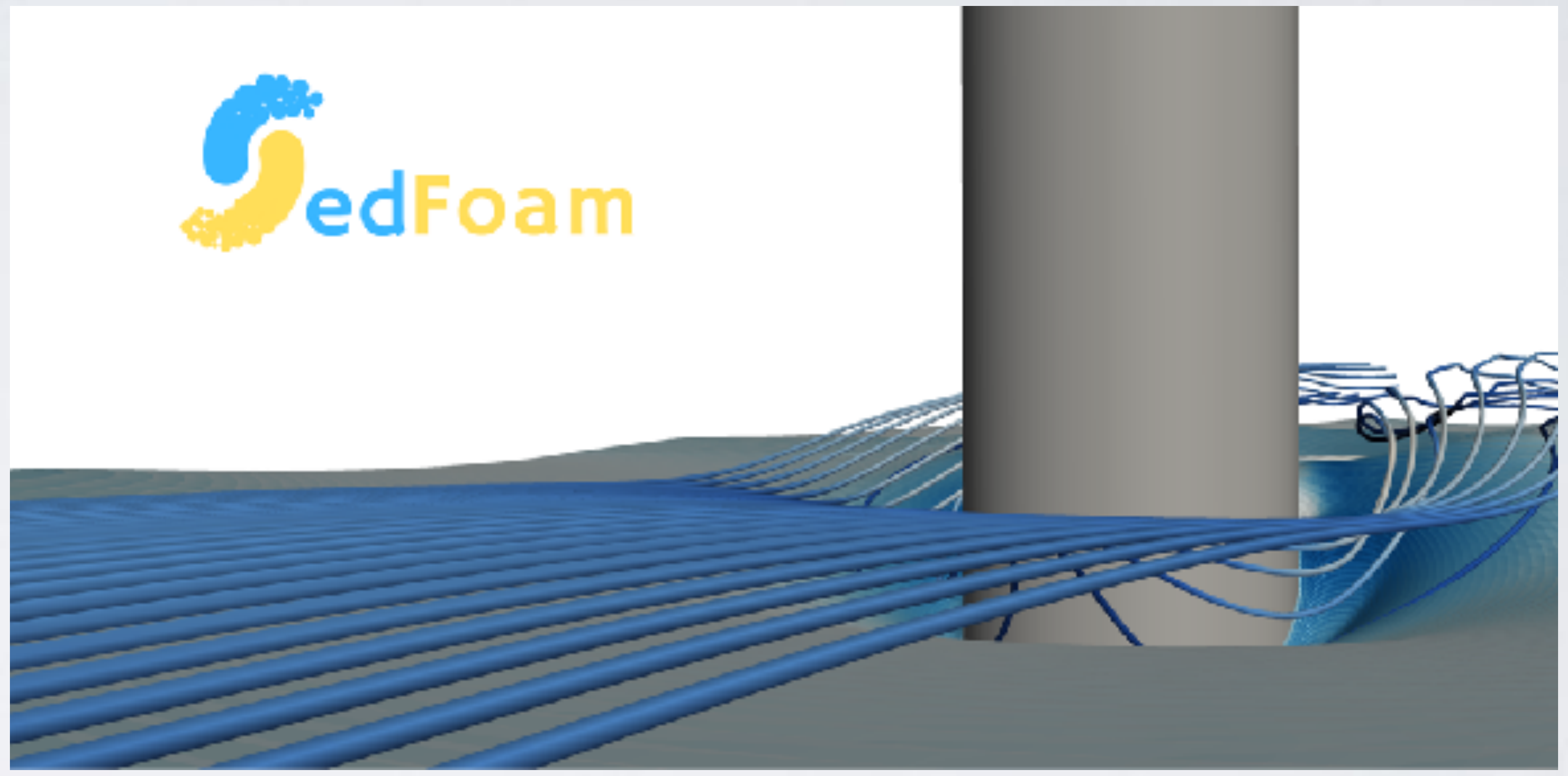
- VIRTUALBOX 6.0 NEEDED (CF. README_USB.TXT)
- ADVANTAGES :
  - NO DOWNLOAD
  - NO STARTUP PROBLEMS
- DISADVANTAGES :
  - IO ACCESS NOT VERY FAST
  - NO FULL CONTROL
- RECOMMENDED FOR MAC

LAUNCH PRE-DOWNLOADED VIRTUAL MACHINE

- VIRTUALBOX 6.0 NEEDED (CF. README_FULL.TXT)
- ADVANTAGES :
  - FULL CONTROL
  - VERY FAST
- DISADVANTAGES :
  - RISK OF BREAKING THE VM
  - DOWNLOAD NEEDED
- RECOMMENDED FOR SPEED

SPECIFICATIONS OF THE ENVIRONMENT

- Linux OS : Ubuntu
- Username : lubuntu
- Password : lubuntu
- OpenFoam v1812 (ESI version)
- Python 3.7
- Latest official sedfoam
- Latest official fluidfoam
- Important tools :
  terminal, python notebook



- Directory of openfoam sources :
  /opt/openfoam/1812plus/
- Directory of sedfoam (sources, tutorials, turbulent models, post processing functions…) :
  /home/lubuntu/Documents/sedfoam

Linux Survival Guide

- To launch terminal, just click on icon of the desktop or icon of launch bar
- List of useful classical commands/tools in terminal :
  - **cd** : change directory;

    example : cd /home/lubuntu/Documents/sedfoam
  - **ls** : list directory contents of files and directories;

    example : ls /home/lubuntu/Documents
  - **touch** : create empty file;

    example : touch /home/lubuntu/Documents/empty.file
  - **rm** : remove file or directory (**-r** option needed for directory);

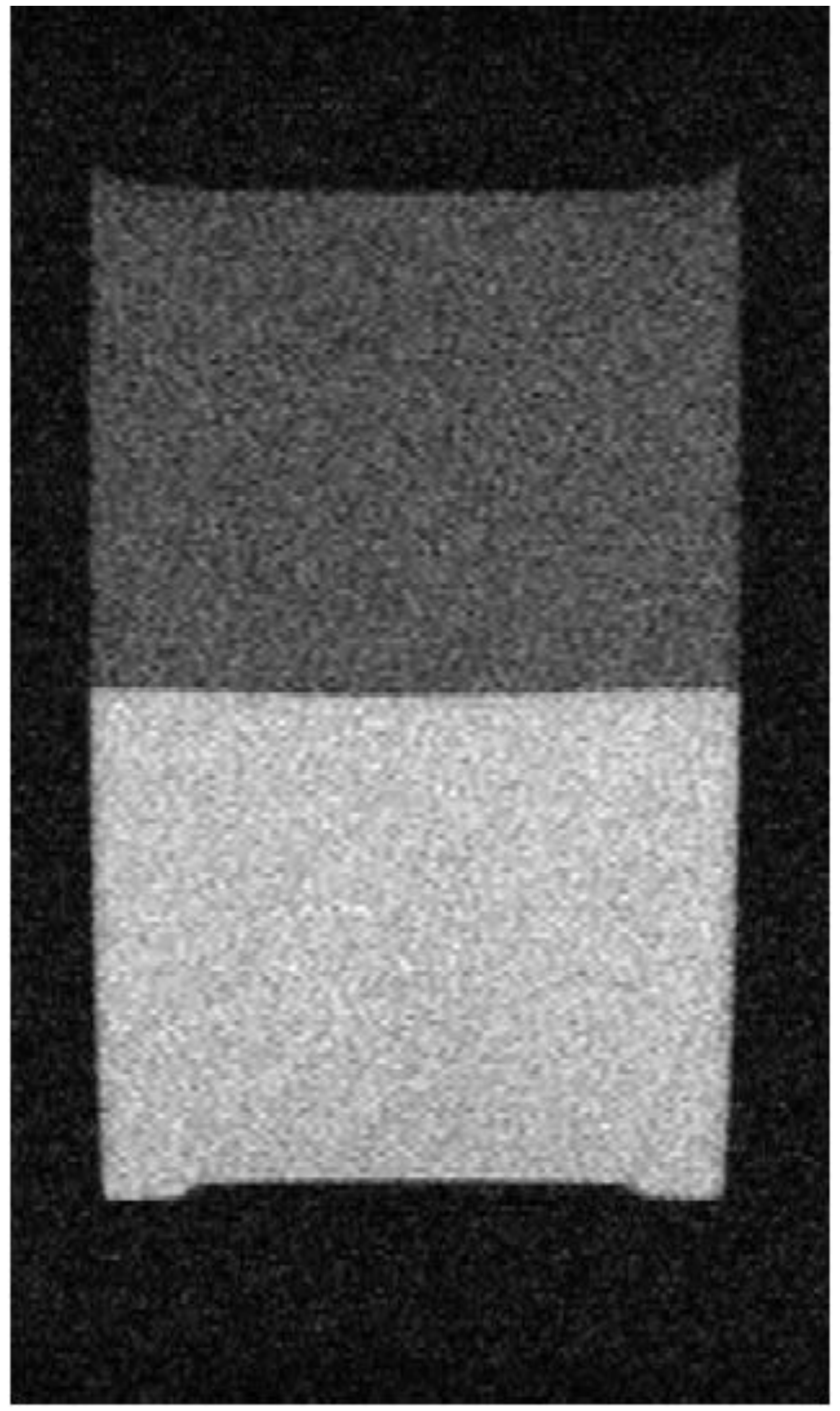
    example : rm /home/lubuntu/Documents/empty.file
  - **gedit** : classical editor to modify files

    other editors : **vi, emacs, nano, atom, vscode**…

- **Paraview** : visualisation tools (very useful for 3D output)
- To launch python notebook, just type : **jupyter-lab** in terminal
- List of shortcuts for notebook :
  - shift+return : execute the notebook box
  - return : go to the line

Test case 1: Sedimentation of particles at low particulate-Reynolds number

Polystyrene beads in silicon oil



Physical parameters: LMSGC experiment - MRI measurements

Pham Van Bang et al. (2006)

Fluid phase:

- $\eta_f = 20.10^{-3}$ Pa s (200 x water)
- $\rho_f = 0.95$ kg m^{-3}

Solid phase:

- $d = 0.29 \pm 0.03$ mm
- $\rho_p = 1.05$ kg m^{-3}
- $\phi^0 = 0.48$

Model ingredients:

- Stokes drag + hindrance function
- Particle pressure due to enduring contacts: $p^p = P_0 \dfrac{(\phi - \phi_{rlp})^5}{(\phi_m - \phi)^3}$ *Johnson & Jackson (1987)*

where P_0 is a modulus (in Pa) and ϕ_{rlp} is the random loose packing fraction

Numerical parameters:

- Ny= 120 ; Δt=0.2 s; first order schemes

Test case 1: run the case

- Open a terminal

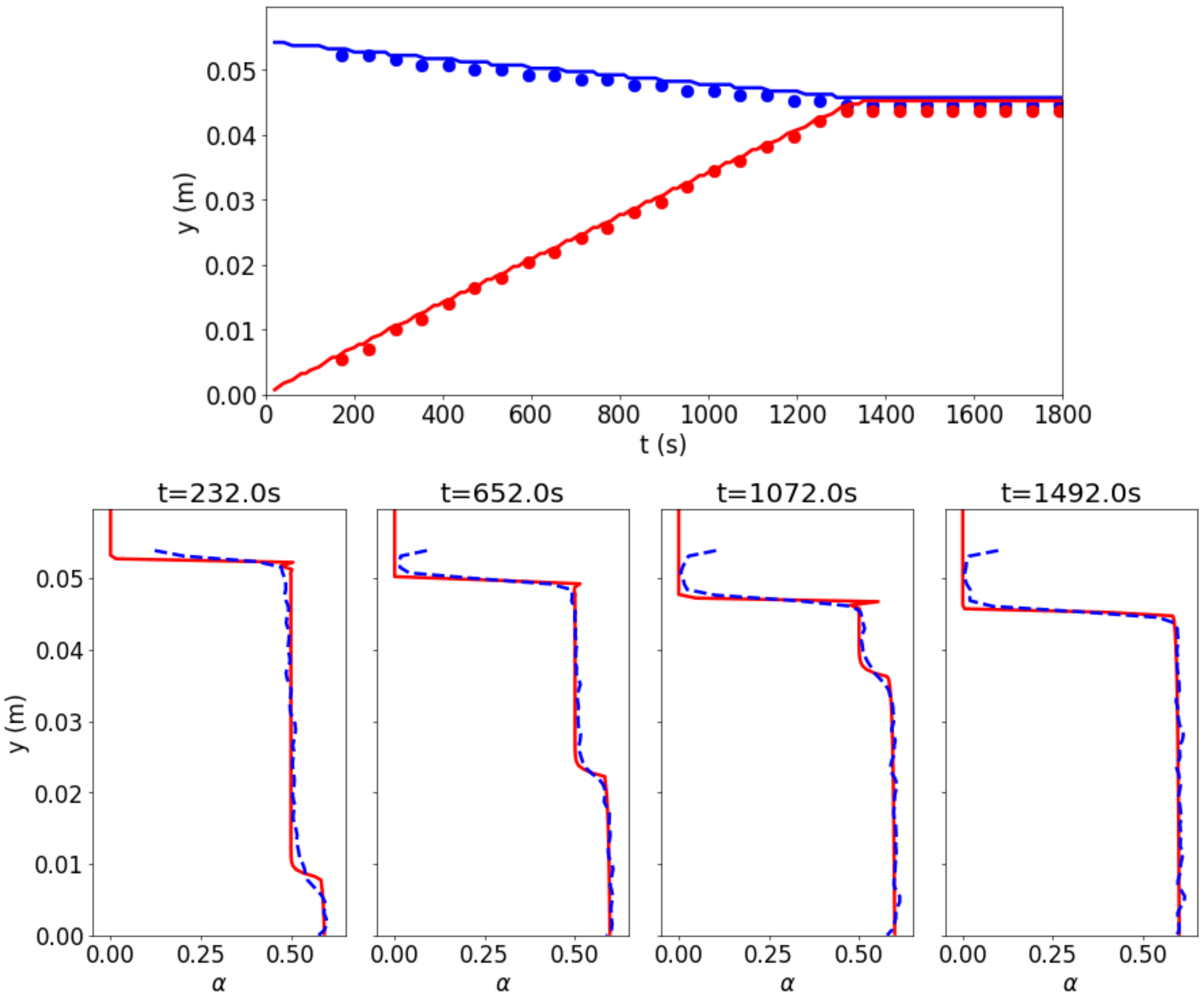
  cd /home/lubuntu/Documents/sedfoam/tutorials/

- Open the jupyter-notebook:

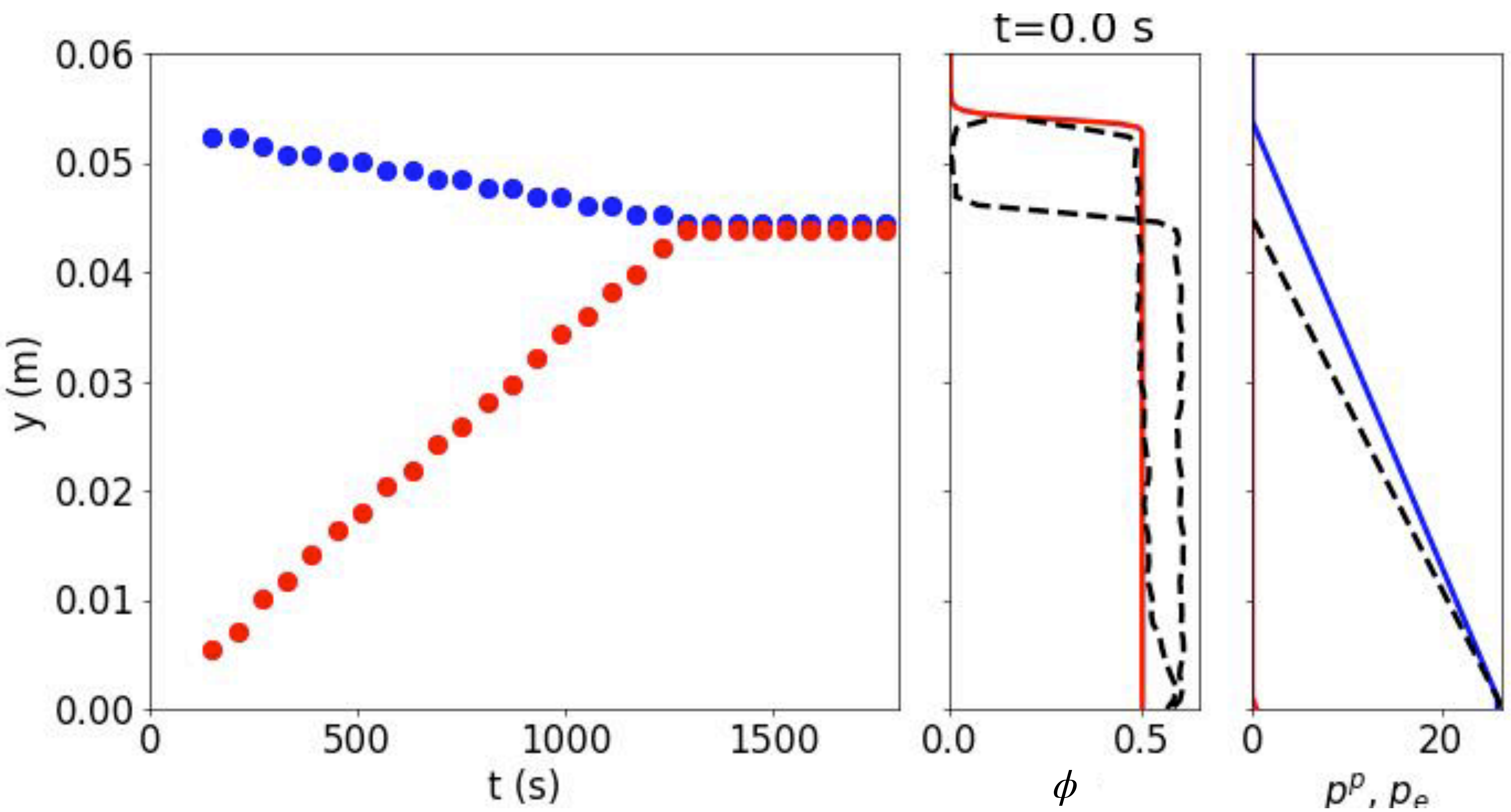
  jupyter-lab THESIS.ipynb &

- Follow the steps!

Test case 1: run the case



Test case 1: Sedimentation of particles at low particulate-Reynolds number



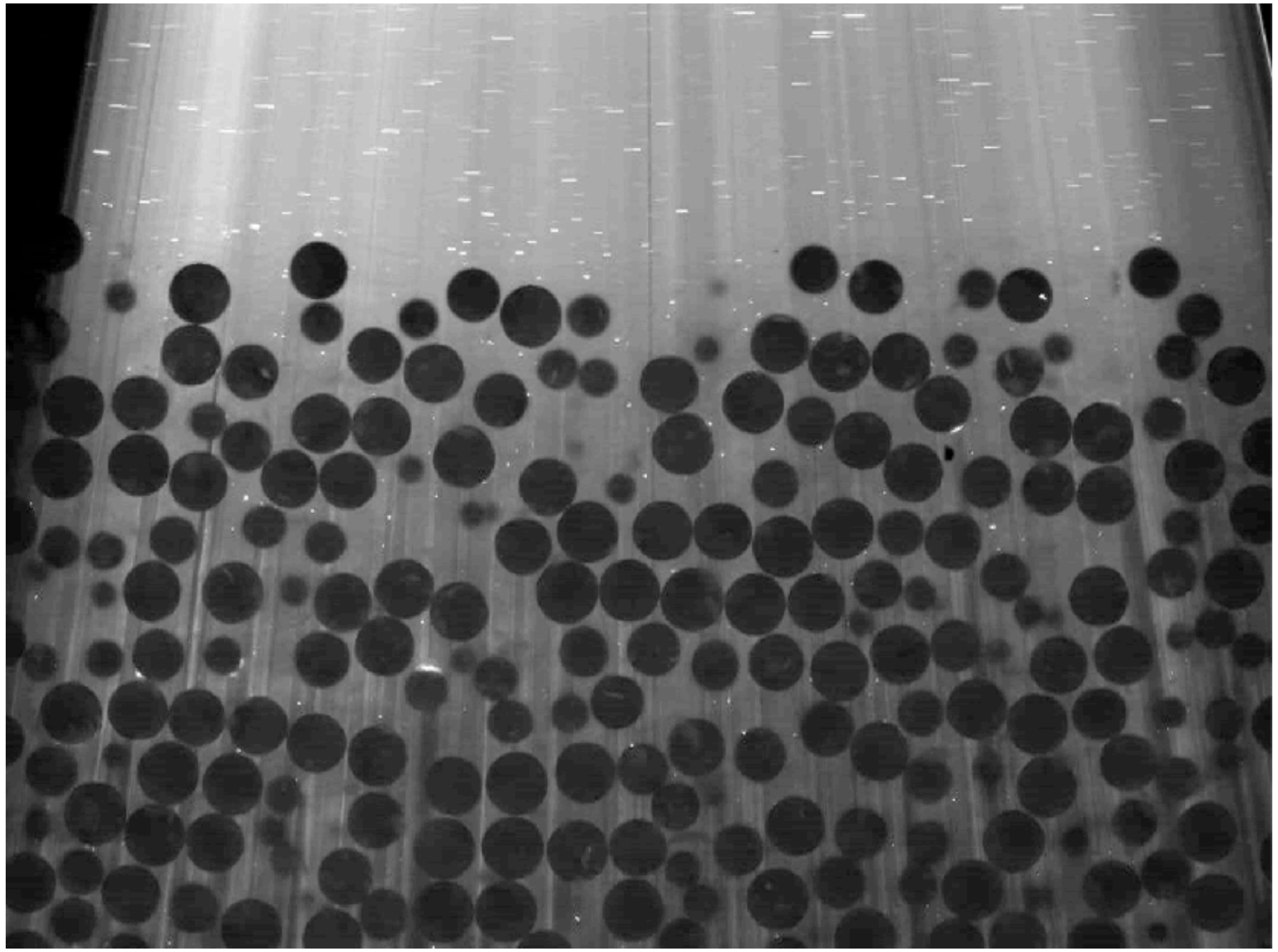
Excess pore pressure: $p_e = p^f - \rho_f \; g \; y$

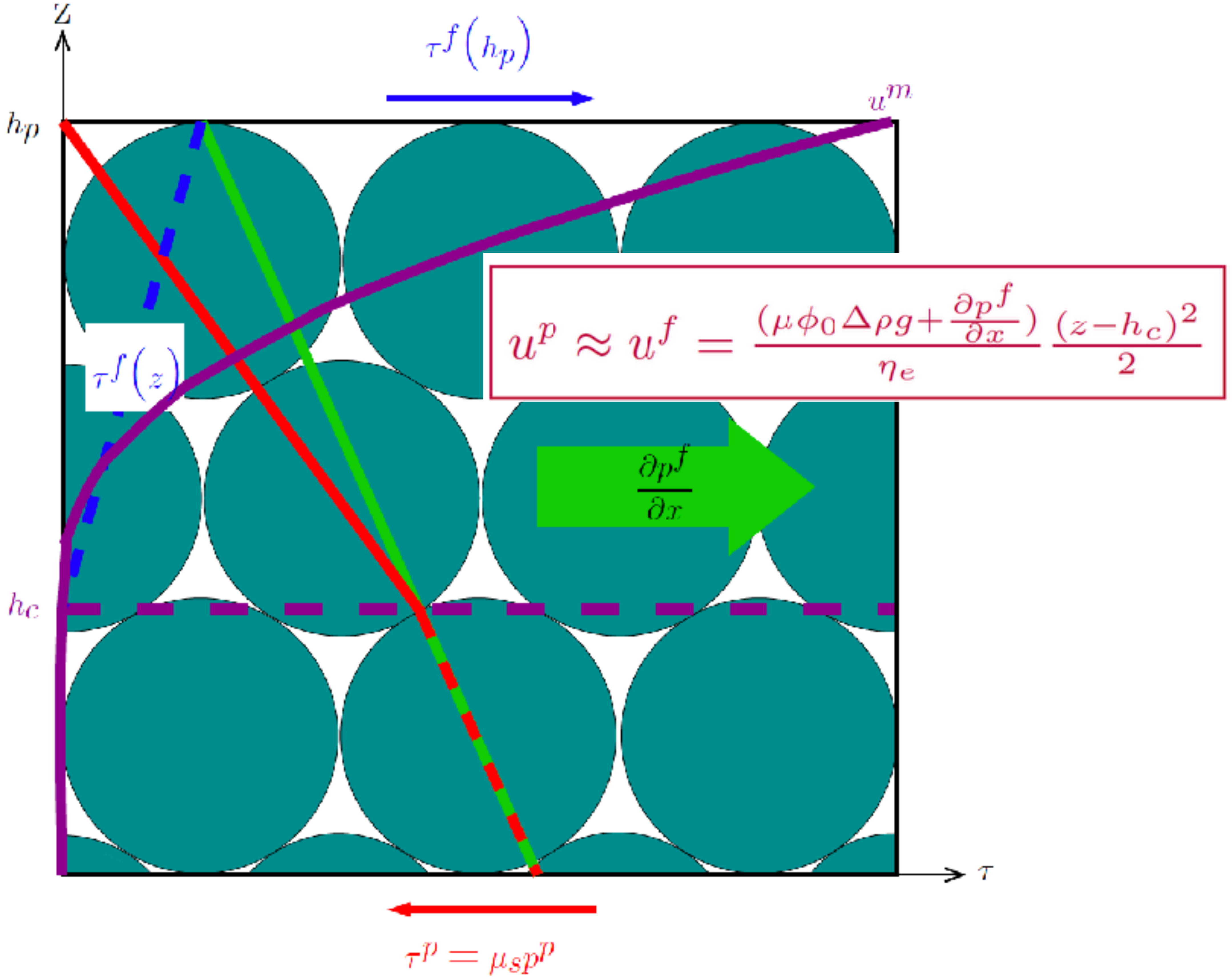
Mixture momentum balance: $\dfrac{D\rho_m w_m}{Dt} = -\dfrac{dp^f}{dy} - \rho_m g - \dfrac{dp^p}{dy}$

The dense granular flow rheology depends on p^p => essential to predict it accurately

Chauchat et al. GMD (2017)

Test case 2: Laminar bed-load





Index-matching experiments

- Particles: d_p=2mm PMMA ; $\rho_p/\rho_f = 1.2$
- Fluid: Triton X-100
- Re ~ 1

(Aussillous et al., JFM 2013)

Analytical solution

- Einstein viscosity
- Coulomb friction: μ = constant
- Parabolic velocity profile

(Ouriemi et al., JFM 2009)

Governing equations

Fluid phase equations

$$\frac{\partial \epsilon}{\partial t} + \nabla . \left(\epsilon \vec{u^f} \right) = 0$$

$$\rho_f \left[\frac{\partial \epsilon \vec{u^f}}{\partial t} + \nabla . \left(\epsilon \vec{u^f} \otimes \vec{u^f} \right) \right] = -\nabla p^f + \nabla . \overline{\overline{\tau^f}} - n\vec{f} + \epsilon \rho_f \vec{g}$$

Effective fluid stress

= include particle perturbations

Fluid-particle interactions

= fluid flow at the particle scale

Solid phase equations

$$\frac{\partial \phi}{\partial t} + \nabla . \left(\phi \vec{u^p} \right) = 0$$

$$\rho_p \left[\frac{\partial \phi \vec{u^p}}{\partial t} + \nabla . \left(\phi \vec{u^p} \otimes \vec{u^p} \right) \right] = -\nabla p^p + \nabla . \overline{\overline{\tau^p}} + n\vec{f} + \phi \rho_p \vec{g}$$

Granular stresses

= particle-particle interactions

Details of the flow at the particle scale are missing due to averaging

➡ Need to model grain-scale physics

Granular stresses: particle-particle interactions

Dense granular flow rheology: μ(I) *(GDR Midi, 2004)*

Represent frictional-collisional interactions in dense granular flows

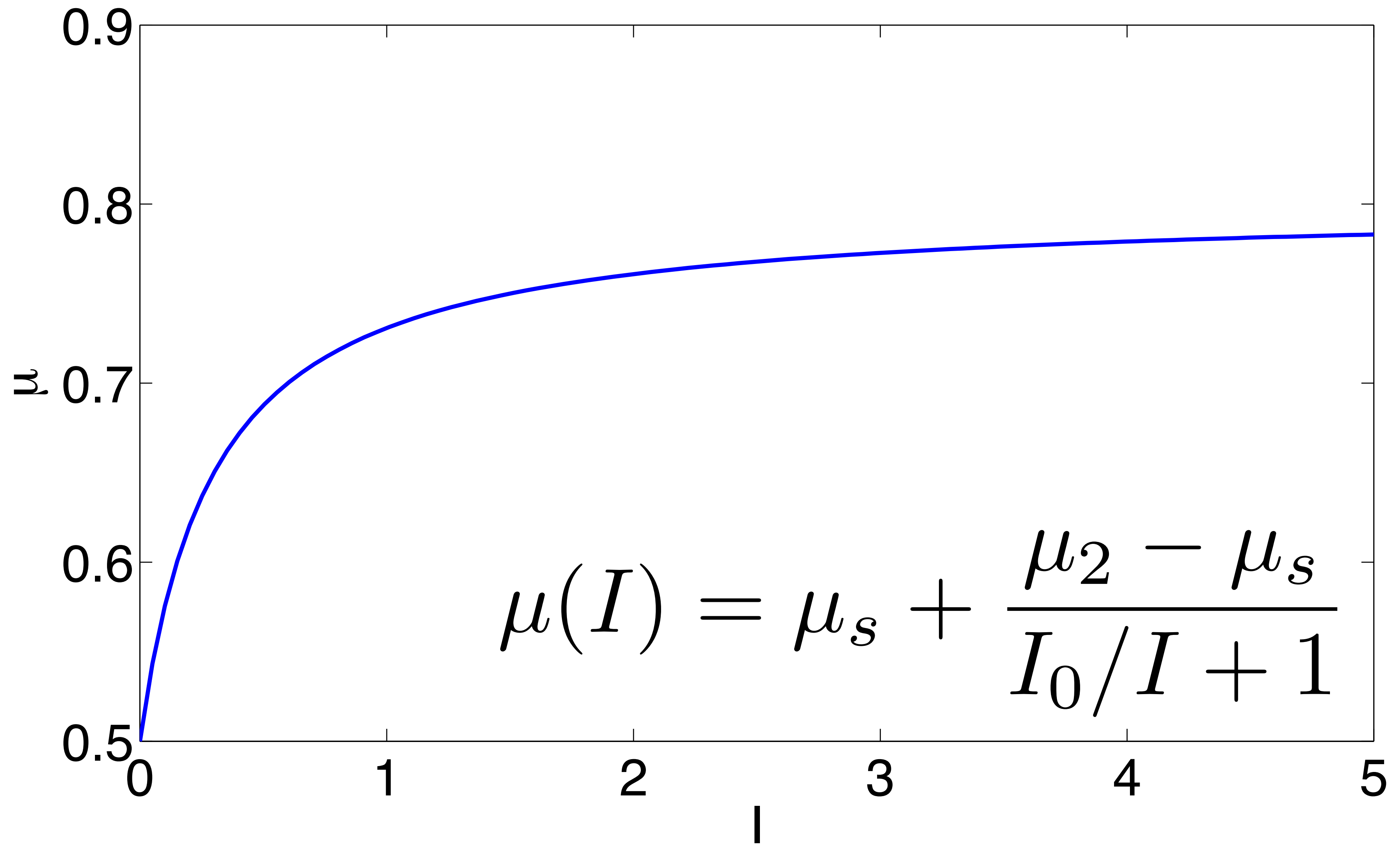
- **Shear stress**

$$\overline{\overline{\tau^p}} = \mu(I) p^p \frac{\overline{\overline{S^p}}}{||\overline{\overline{S^p}}||}$$

(Jop et al., 2006)

with $\overline{\overline{S^p}} = \nabla \vec{u^p} + (\nabla \vec{u^p})^T - \frac{2}{3} tr(\nabla . \vec{u^p})$

Visco-plastic rheology: contain a yield stress (need regularization) and a non-linear viscous term

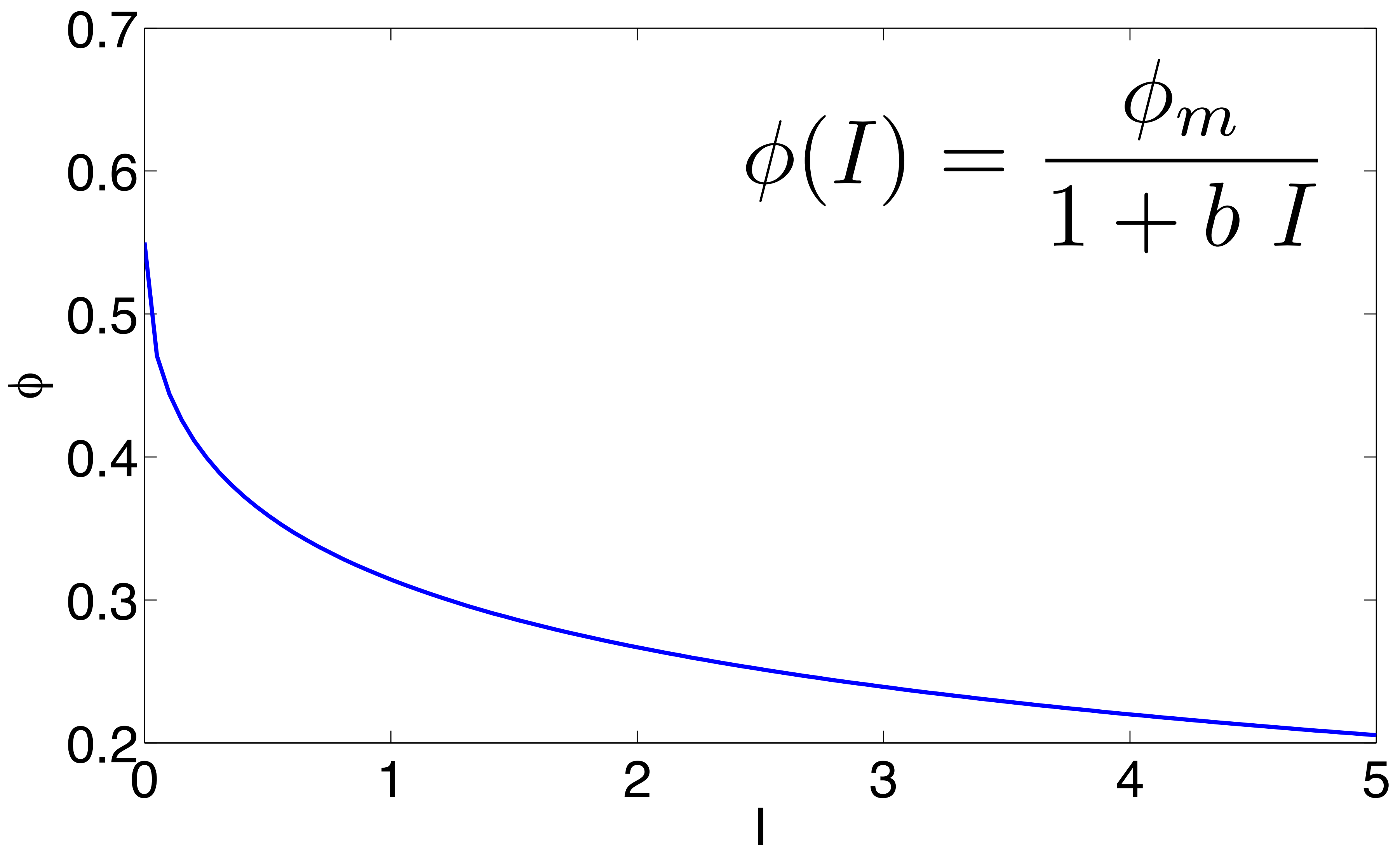


- **Particle pressure**

$$p^p = \left(\frac{b\ \phi}{\phi_m - \phi} \right)^2 \rho^p d_p^2\ ||\nabla \vec{u^p}||^2$$

Shear-induced pressure: lead to bed decompaction *(Maurin et al., 2016)*

+ pressure due to enduring contact *(Johnson & Jackson, 1987)*



Control parameter = Inertial number: $I = \frac{||\nabla \vec{u^p}||\ d_p}{\sqrt{p^p / \rho^p}}$

Effective fluid stress

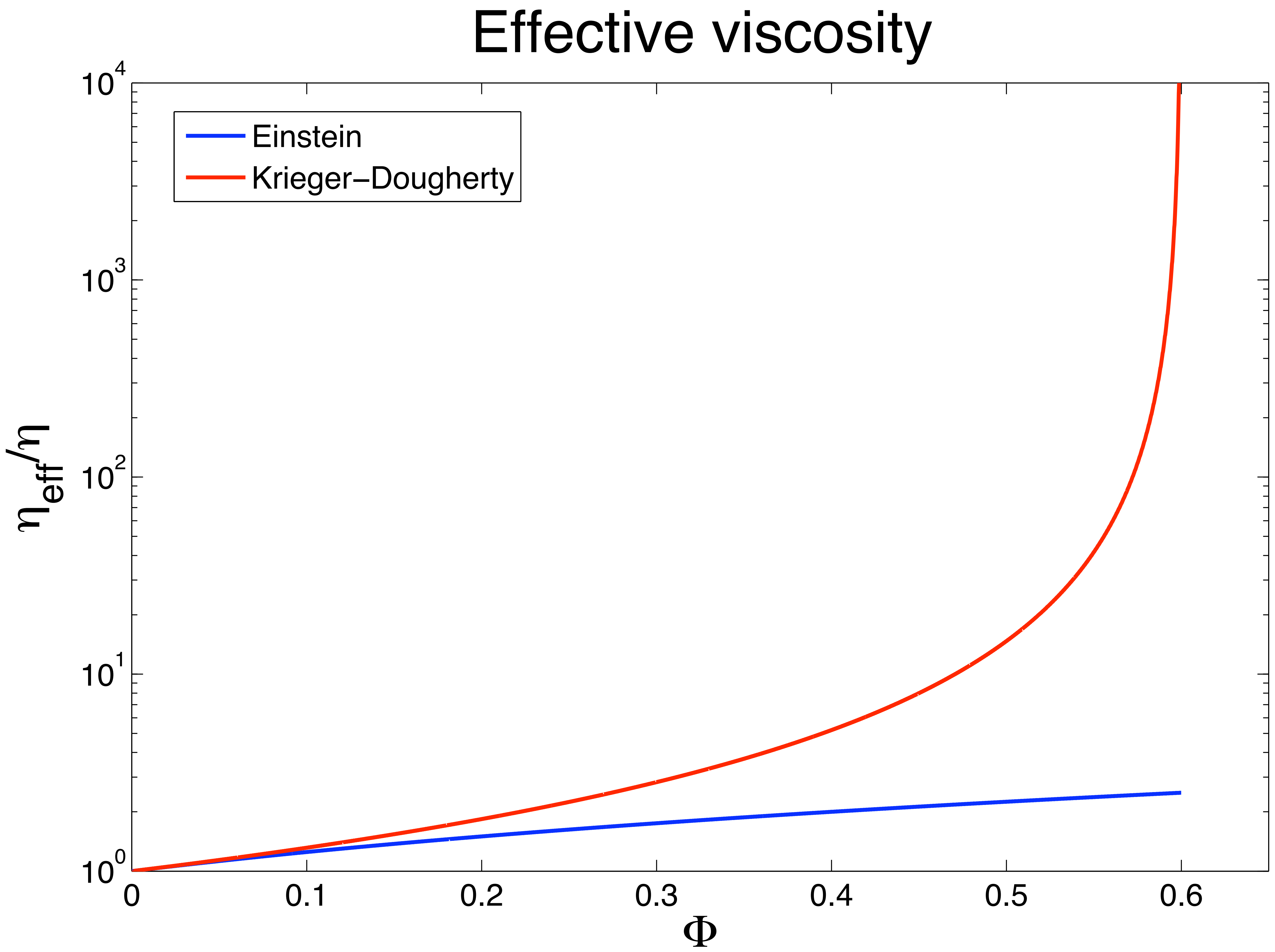
Shear stress: $$\overline{\overline{\tau^f}} = \eta_e \, \overline{\overline{S^f}}$$ with $\overline{\overline{S^f}} = \nabla\vec{u^f} + (\nabla\vec{u^f})^T - \frac{2}{3}tr(\nabla.\vec{u^f})$ the velocity shear rate

Effective viscosity models depends on volume fraction

- Einstein (1906) model: $$\eta_e = \eta^f\left(1 + \frac{5}{2}\phi\right)$$
- Krieger-Dougherty (1957) model:

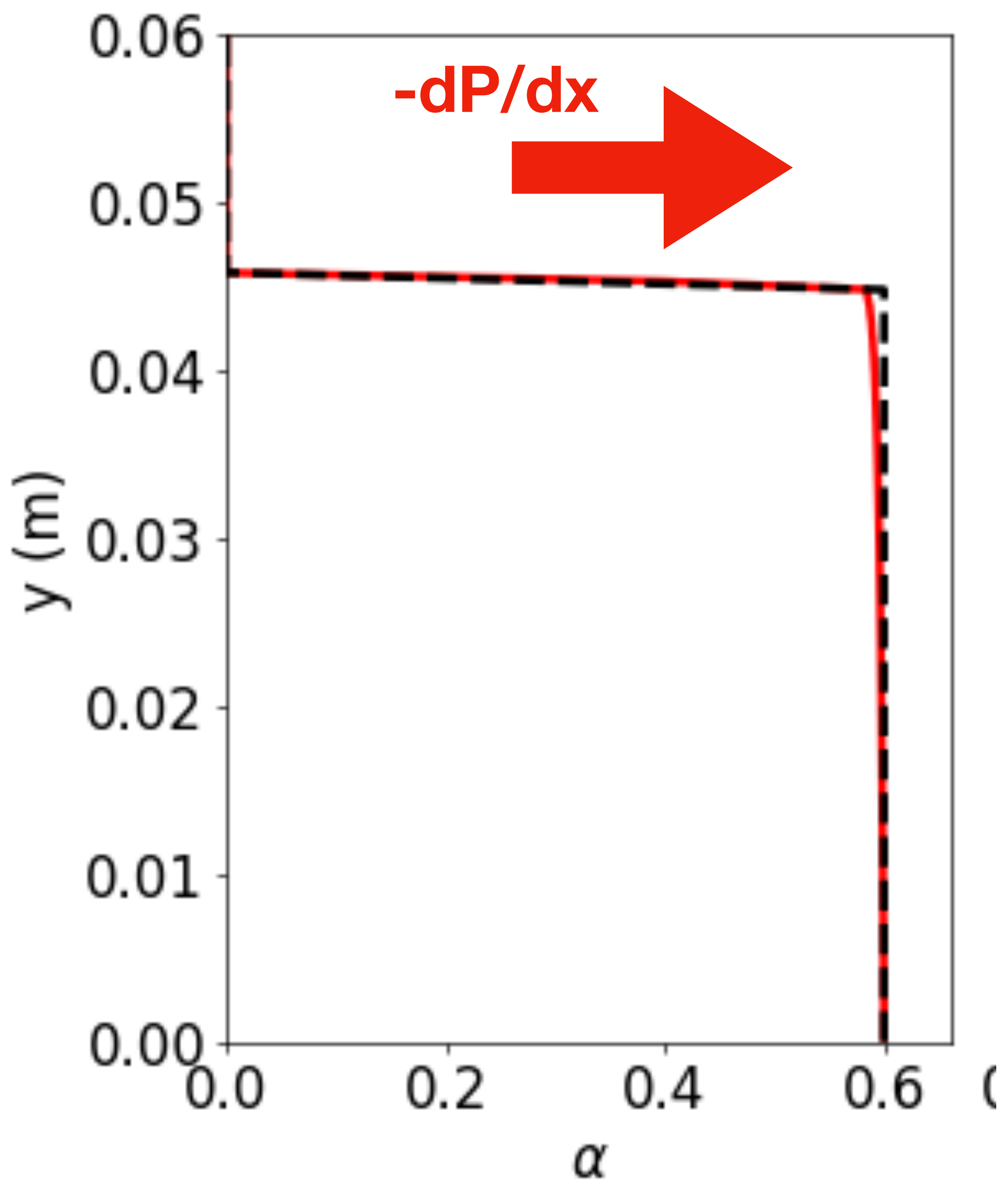
$$\eta_e = \eta^f\left(1 - \frac{\phi}{\phi_{max}}\right)^{-\frac{5}{2}\phi_{max}}$$

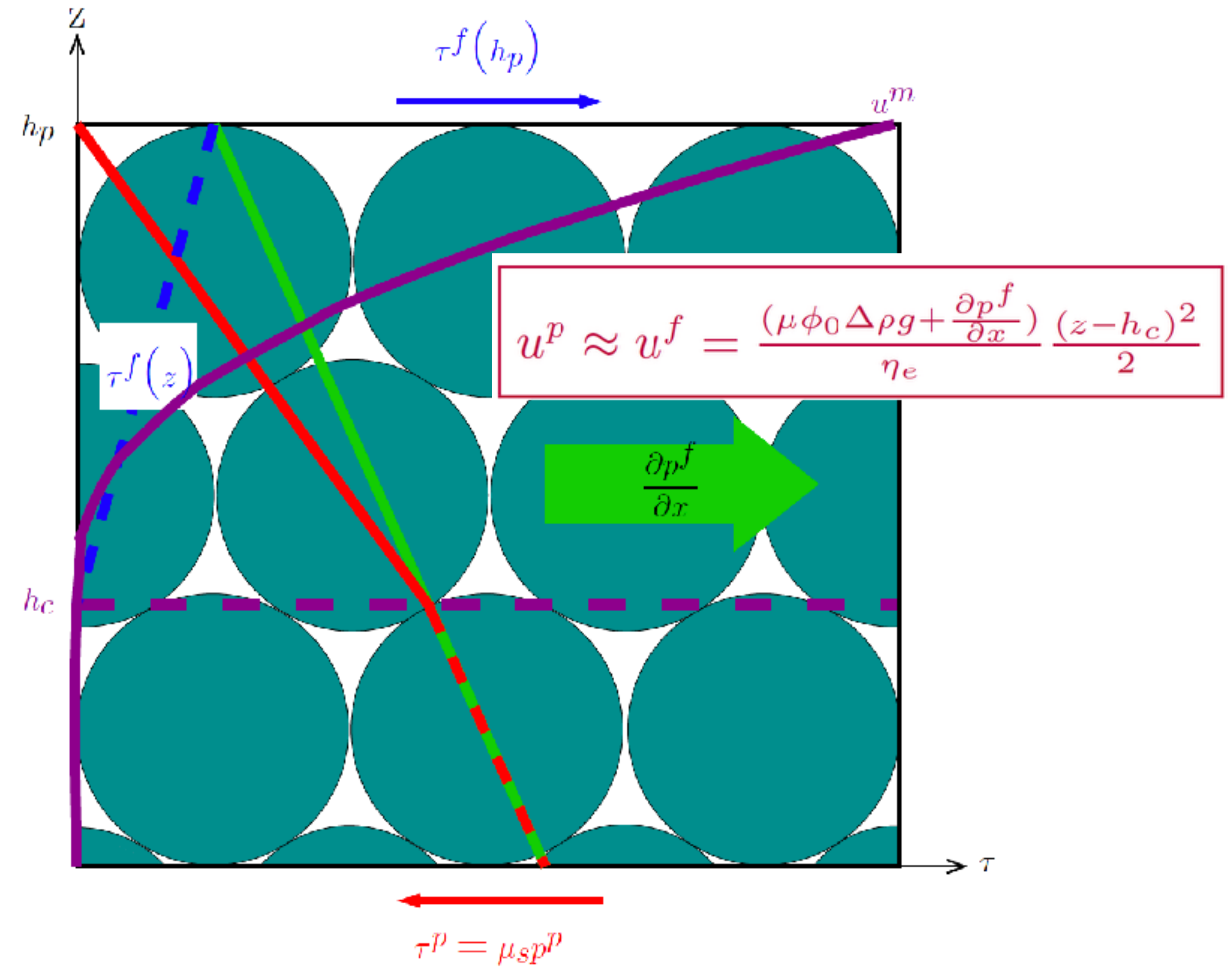
- Viscosity increases with volume fraction



Test case 2: Laminar bed-load

- Once the particles are deposited we set a streamwise pressure gradient to drive the fluid flow above the granular bed.

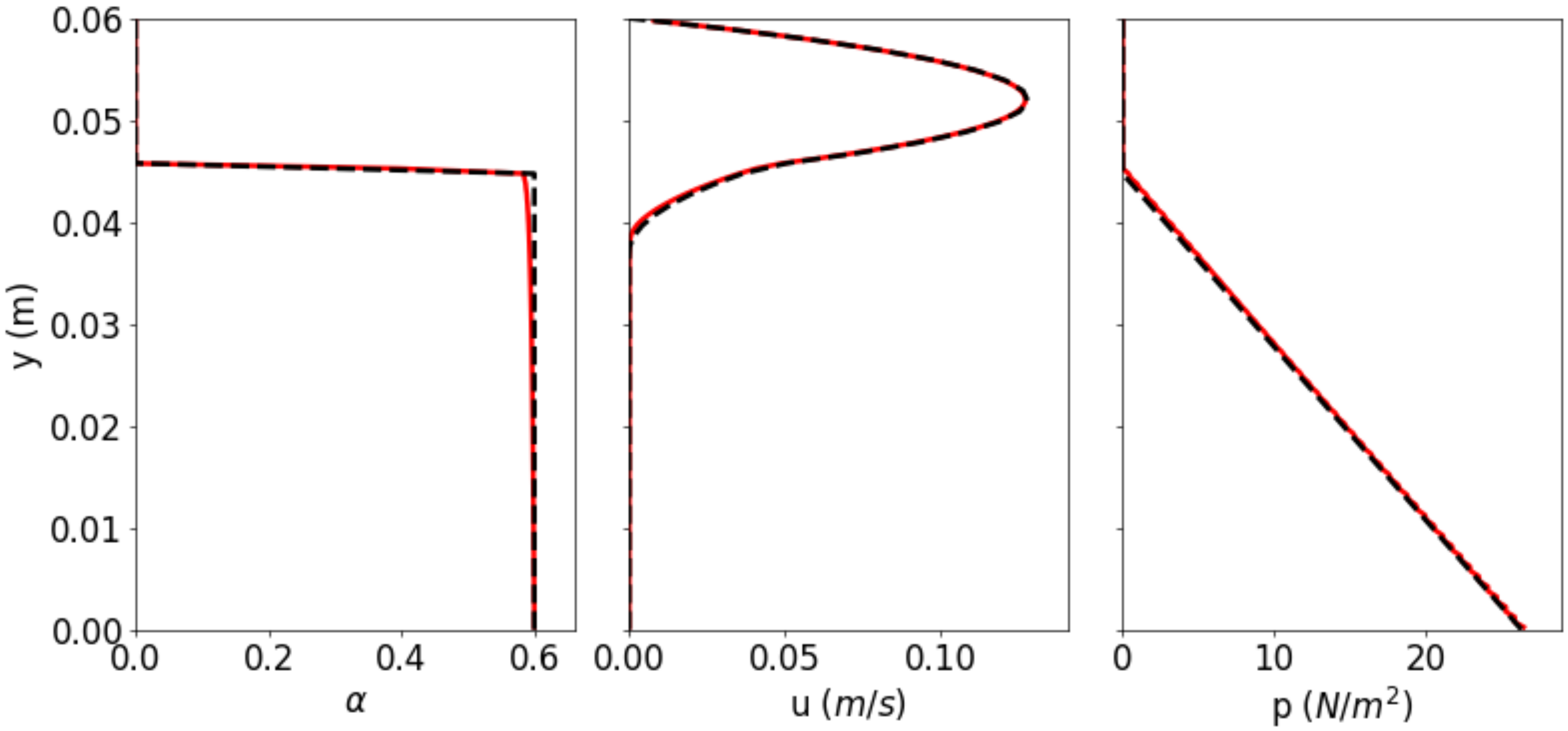




- Modify the input files (see NoteBook) and run the model

- Numerical parameters: Ny=120, Δt=0.2 s, first order schemes

Test case 2: Laminar bed-load with Coulomb rheology

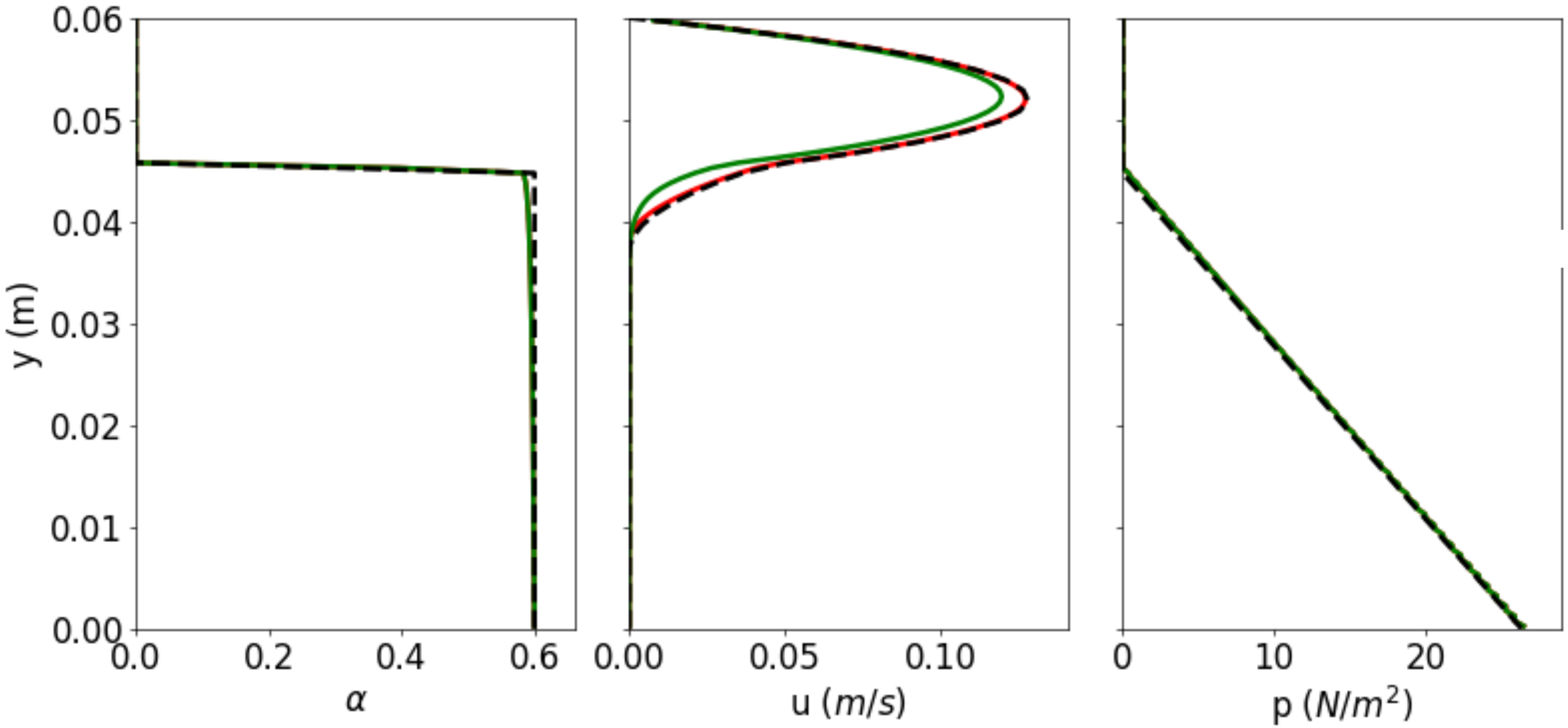


- Comparison with analytical solution: Coulomb rheology + Einstein viscosity model
  - Numerical implementation of granular flow rheology is validated

- Numerical parameters: Ny=200; first order schemes

Chauchat et al. GMD (2017)

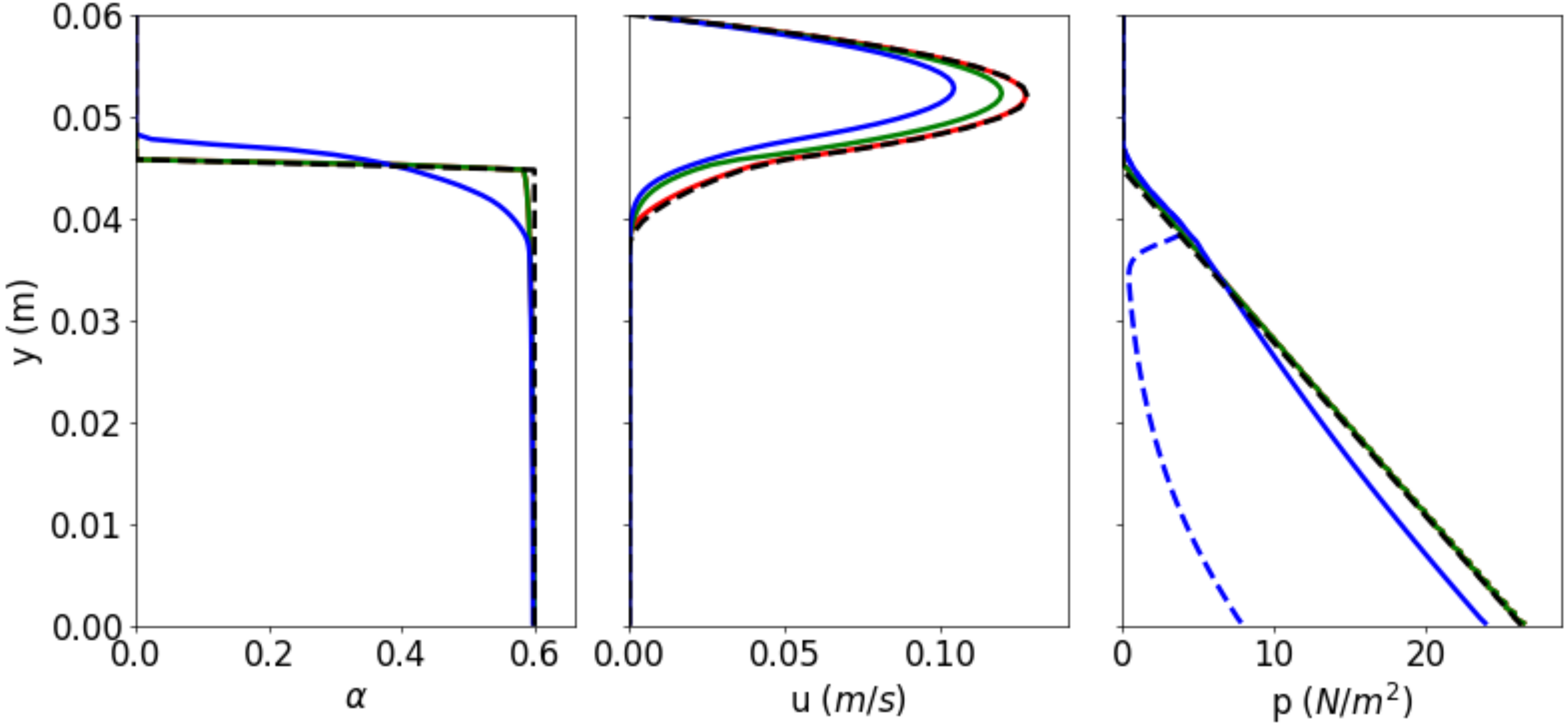
Test case 2: Laminar bed-load with μ(I) rheology



- Comparison with numerical solution: μ(I) rheology + Einstein viscosity model
  - Numerical implementation of granular flow rheology is validated
- Numerical parameters: Ny=120; first order schemes

Chauchat et al. GMD (2017)

Test case 2: Laminar bed-load with μ(I) rheology and dilatancy law



- Numerical solution: μ(I) rheology + phi(I) + Einstein viscosity model
- Numerical parameters: Ny=120; first order schemes

Chauchat et al. GMD (2017)

Favre-averaged two-phase flow equations

Favre-averaging: Ensemble averaging $\langle \phi \rangle = \lim_{N \to \infty} \sum_{k=1}^{N} \phi_k$ Favre-average velocities $\widetilde{\vec{u^f}} = \frac{\langle (1-\phi)\vec{u^f} \rangle}{1 - \langle \phi \rangle}$

Concentration fluctuations $\phi'_k = \phi_k - \langle \phi \rangle$ Velocity fluctuations $\Delta \vec{u^f} = \vec{u^f} - \widetilde{\vec{u^f}}$

Turbulent shear stress

$$\overline{\overline{R^f}} = -\rho^f \langle (1-\phi) \Delta \vec{u^f} \otimes \Delta \vec{u^f} \rangle$$

Fluid phase equations

$$\frac{\partial \langle \epsilon \rangle}{\partial t} + \nabla . \left(\langle \epsilon \rangle \widetilde{\vec{u^f}} \right) = 0$$

$$\rho_f \left[\frac{\partial \langle \epsilon \rangle \widetilde{\vec{u^f}}}{\partial t} + \nabla . \left(\langle \epsilon \rangle \widetilde{\vec{u^f}} \otimes \widetilde{\vec{u^f}} \right) \right] = -\langle \epsilon \rangle \nabla \langle p^f \rangle + \nabla . \left(\overline{\overline{\tau^f}} + \overline{\overline{R^f}} \right) - \frac{\langle \phi \rangle \rho^p}{t_p} \left(\widetilde{\vec{u^f}} - \widetilde{\vec{u^p}} + \vec{u_d} \right) + \langle \epsilon \rangle \rho_f \vec{g}$$

Kinetic shear stress

$$\overline{\overline{R^p}} = -\rho^p \langle \phi \Delta \vec{u^p} \otimes \Delta \vec{u^p} \rangle$$

Drift velocity

$$\vec{u_d} = \frac{\langle \phi' \Delta \vec{u^f} \rangle}{\langle \phi \rangle}$$

Solid phase equations

$$\frac{\partial \langle \phi \rangle}{\partial t} + \nabla . \left(\langle \phi \rangle \widetilde{\vec{u^p}} \right) = 0$$

$$\rho_p \left[\frac{\partial \langle \phi \rangle \vec{u^p}}{\partial t} + \nabla . \left(\langle \phi \rangle \widetilde{\vec{u^p}} \otimes \widetilde{\vec{u^p}} \right) \right] = -\nabla \langle p^p \rangle + \nabla . \left(\overline{\overline{\tau^p}} + \overline{\overline{R^p}} \right) - \langle \phi \rangle \nabla \langle p^f \rangle + \frac{\langle \phi \rangle \rho^p}{t_p} \left(\widetilde{\vec{u^f}} - \widetilde{\vec{u^p}} + \vec{u_d} \right) + \langle \phi \rangle \rho_p \vec{g}$$

Fluid turbulence modeling

Reynolds shear stress:

$$\overline{\overline{R^f}} = \rho^f (1 - \langle\phi\rangle)\ \nu_t^f\ \overline{\overline{S^f}}$$

Eddy viscosity models:

- Two-equation models: k-ε or k-ω models

Drag damping term *Density stratification term*

➡ Modified TKE equation: $\rho^f \frac{Dk}{Dt} = P + D - \rho^f \varepsilon - \frac{\rho^p \langle\phi\rangle}{(1-\phi)t_p}\left(2k - \frac{\langle\phi \Delta\vec{u^f} \Delta\vec{u^p}\rangle}{\langle\phi\rangle}\right) - (\rho^p - \rho^f)\frac{\nu_t^f}{\sigma_s}\frac{\nabla\langle\phi\rangle}{\langle 1-\phi\rangle}.\vec{g}$

correlations between fluid and sediment velocity fluctuations: $\frac{\langle\phi \Delta\vec{u^f} \Delta\vec{u^p}\rangle}{\langle\phi\rangle} = 2e^{-BSt}k$ (*Cheng et al., 2017*)

- Large Eddy Simulation: Dynamic Smagorinsky

Drift velocity model:

$$\vec{u_d} = \frac{\langle\phi' \Delta\vec{u^f}\rangle}{\langle\phi\rangle}$$

- Gradient diffusion model: $\vec{u_d} = -\frac{\nu_t^f}{\sigma_c}\frac{\nabla\langle\phi\rangle}{\langle\phi\rangle}$ where σ_c is a turbulent Schmidt number

Drift velocity is equivalent to Reynolds flux in Rouse profile *(Chauchat, 2018)*

Granular stress modeling

Kinetic Theory of Granular Flows = analogy with molecular gases

Collisional and kinetic stresses:

$$\overline{\overline{R^p}} = 2\ \eta^p\ \overline{\overline{S^p}} + \lambda\ tr(\nabla.\vec{u^p})$$

- **Shear viscosity:** $\eta^p = f_\eta(\phi, e)\ \rho^p\ d\ \Theta^{1/2}$
- **bulk viscosity:** $\lambda = f_\lambda(\phi, e)\ \rho^p\ d\ \Theta^{1/2}$
- **Collisional pressure:** $p^p = f_p(\phi, e)\ \rho^p\ \Theta$

➡ Depend on the granular temperature: $\Theta = \langle \Delta\vec{u^p} \Delta\vec{u^p} \rangle$ counterpart of the TKE for a fluid

- **Transport equation for the granular temperature:**

$$\frac{3}{2}\rho^p \left[\frac{\partial \phi \Theta}{\partial t} + \nabla.\left(\phi \vec{u^p} \Theta\right) \right] = \left(-p^p \overline{\overline{I}} + R^p \right) \nabla.\vec{u^p} - \nabla.\vec{q} - \gamma + \frac{\phi \rho^p}{(1-\phi) t^p} \left(2 e^{-BSt} k - 3\Theta \right)$$

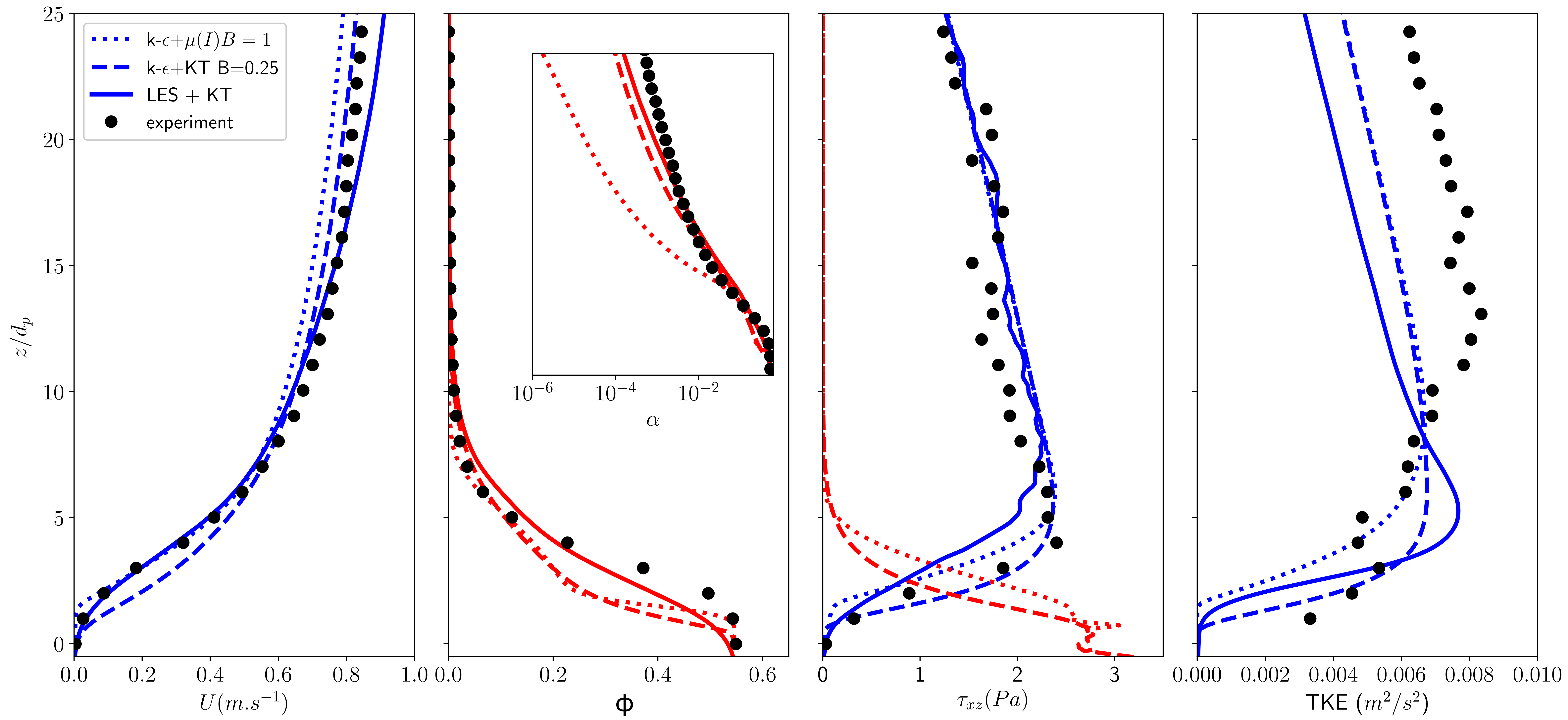
Jenkins and Savage (1983); Ding and Gidaspow (1990)

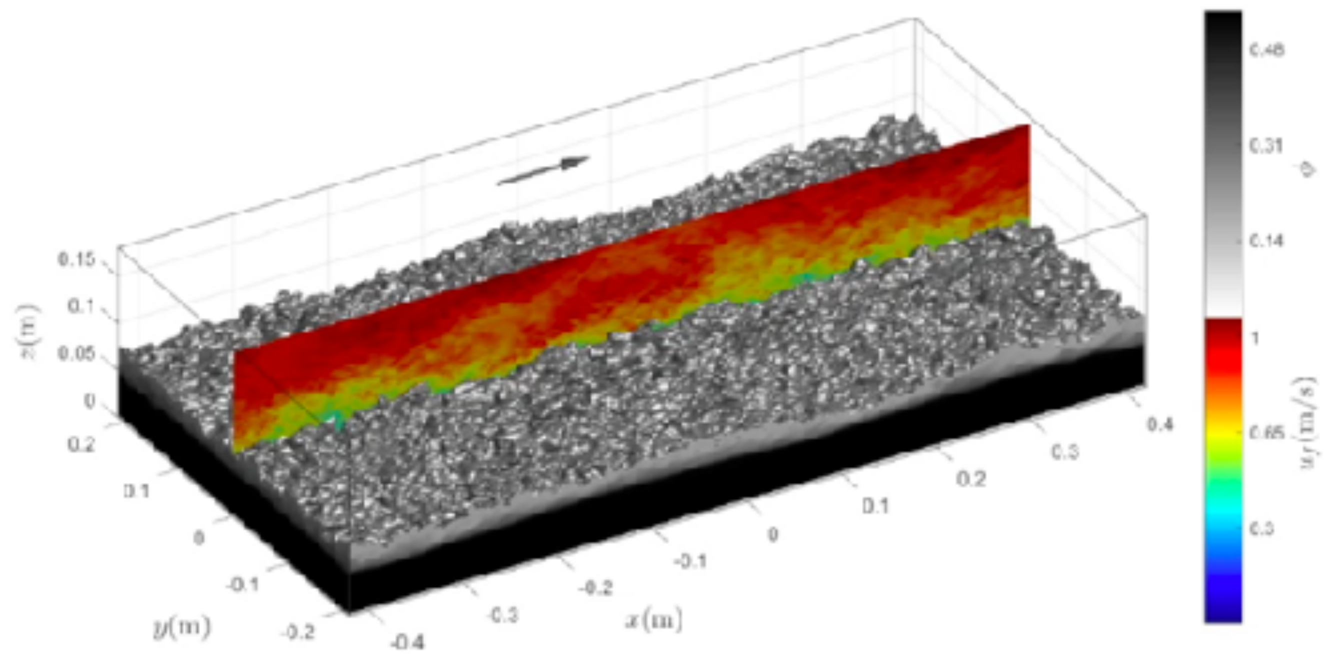
Application to unidirectional sheet-flow

Sheet flow experiment of Revil-Baudard et al. JFM (2015, 2016)

- d=3mm, s~1.2 / h=0.17m ; Slope=0.005 ; θ=0.5
- Acoustic Concentration and Velocity Profiler (Hurther et al., CE 2011)
- Collocated velocity and concentration measurements at 100Hz and 3mm resolution

Application to sheet-flow: Eulerian-Eulerian simulations

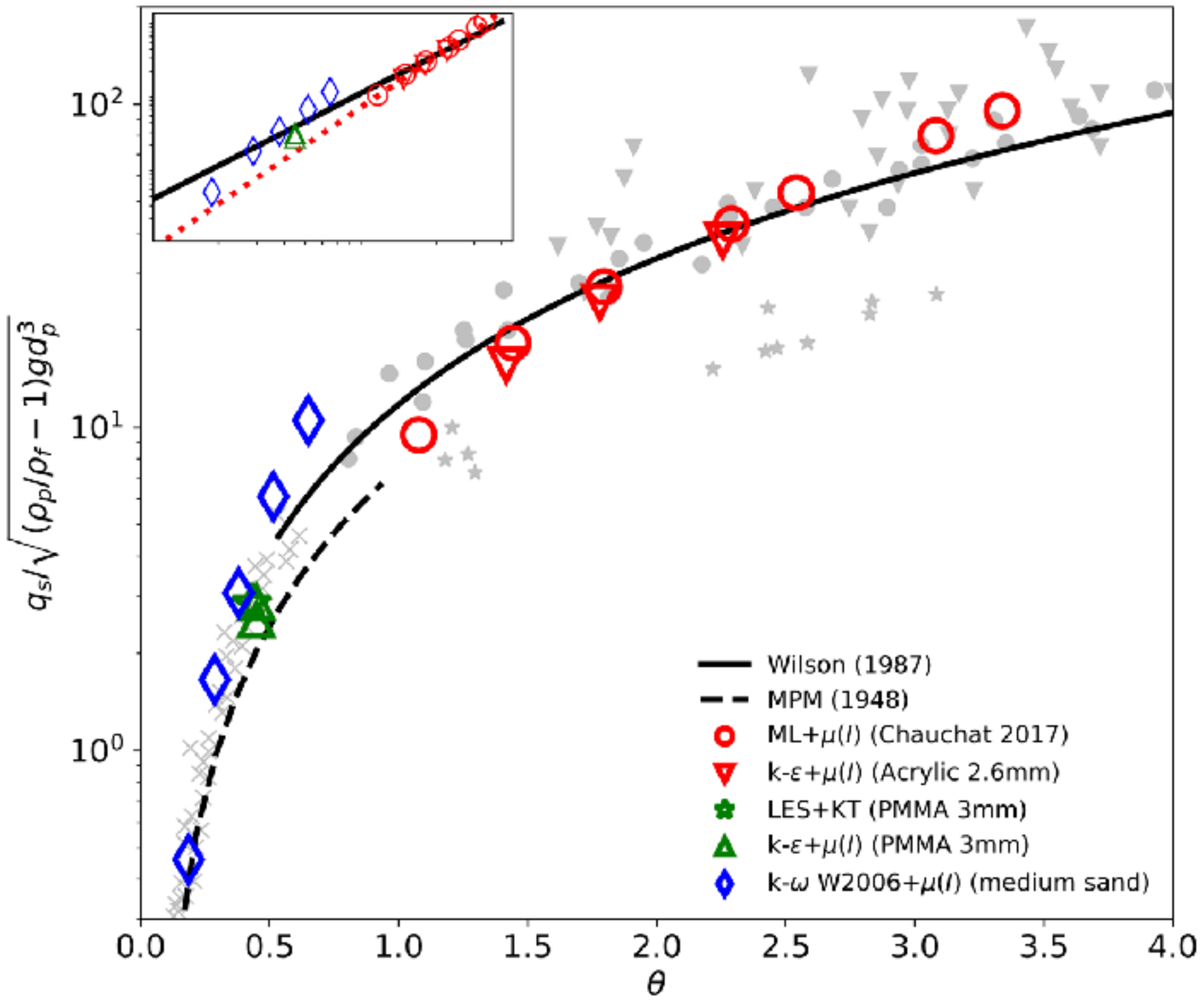


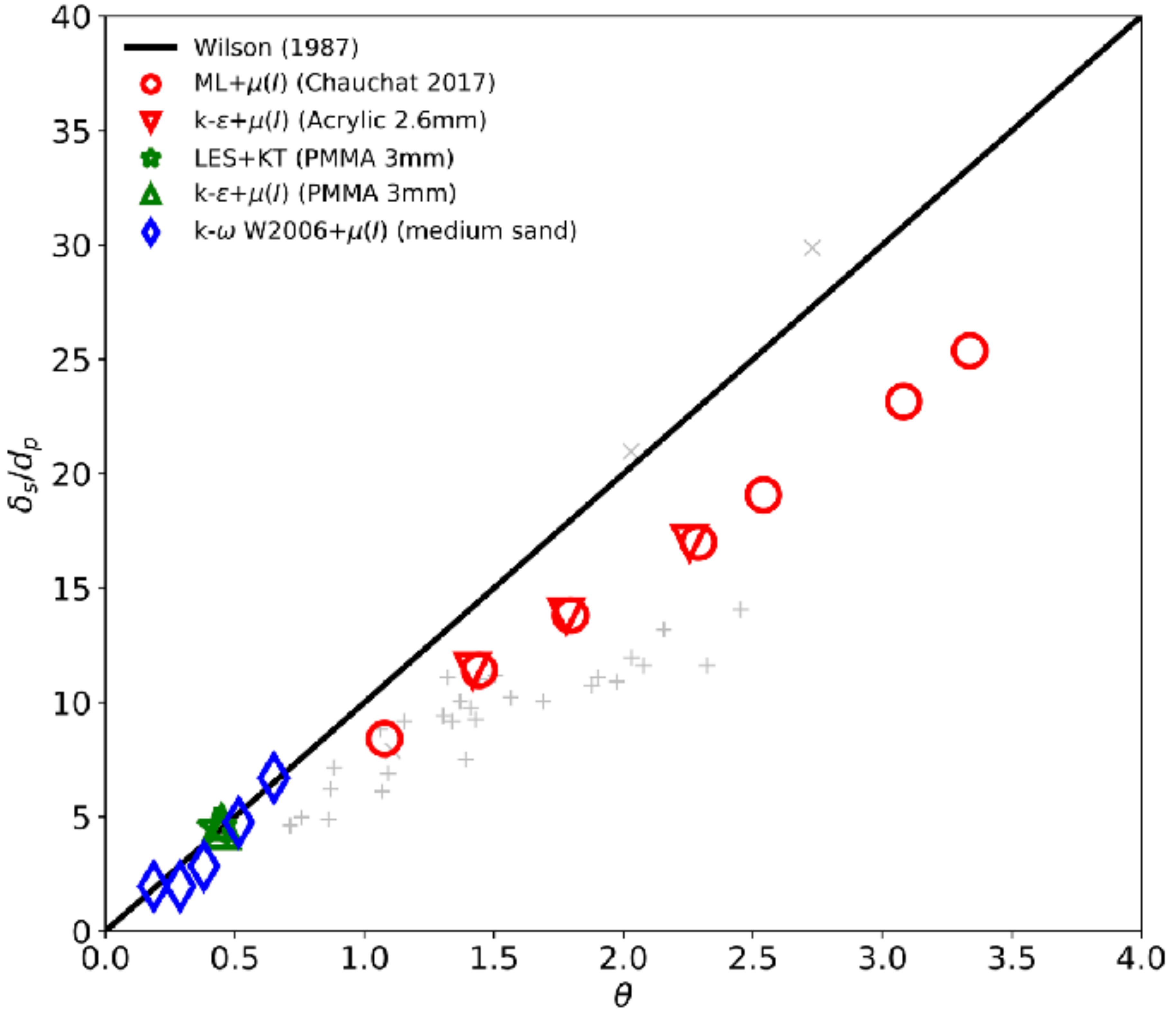


Simulations		Turbulence model	Granular stress model
Run 1	1D	k-ε	μ(I) rheology
Run 2	1D	k-ε	Kinetic Theory
Run 3	3D	LES	Kinetic Theory

Chauchat et al. GMD (2017) ; Cheng et al. AWR (2018)

Sediment flux and transport layer thickness





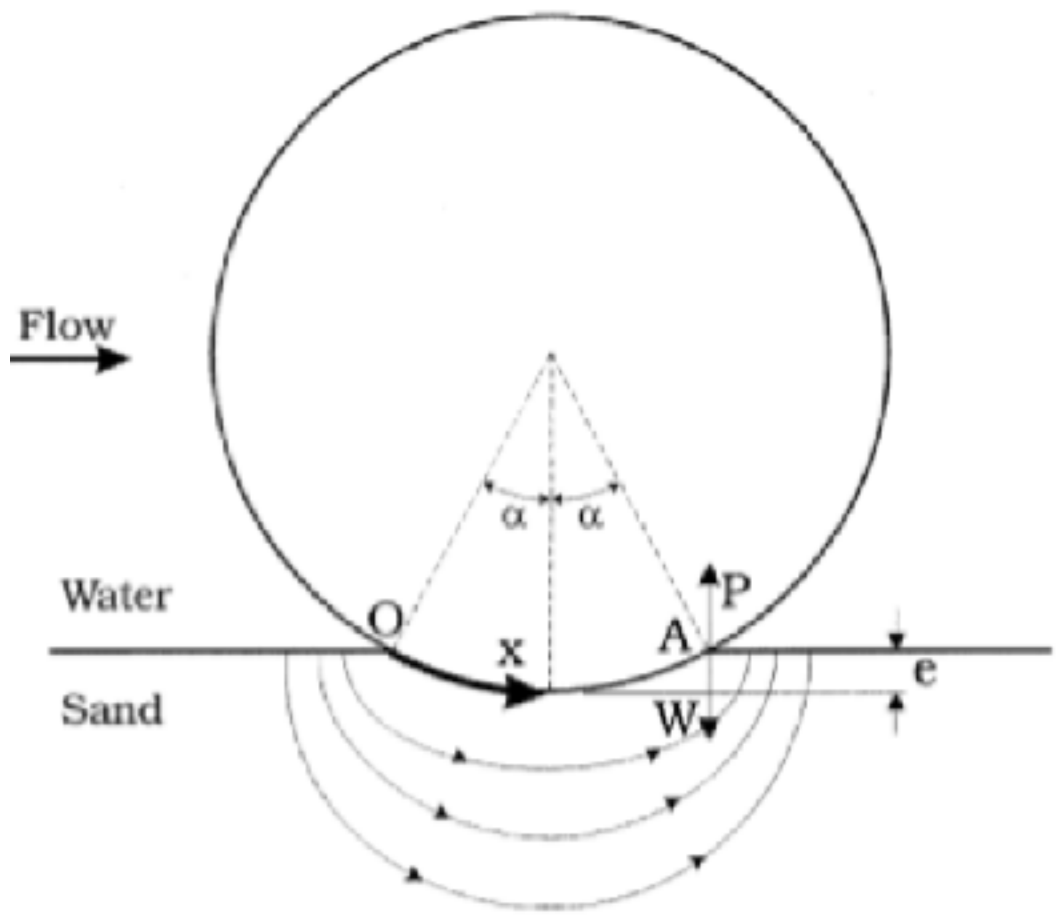
- Meyer-Peter and Müller (1948): $\dfrac{q_s}{\sqrt{(s-1)gd_p^3}} = 8\,(\theta-\theta_c)^{3/2}$

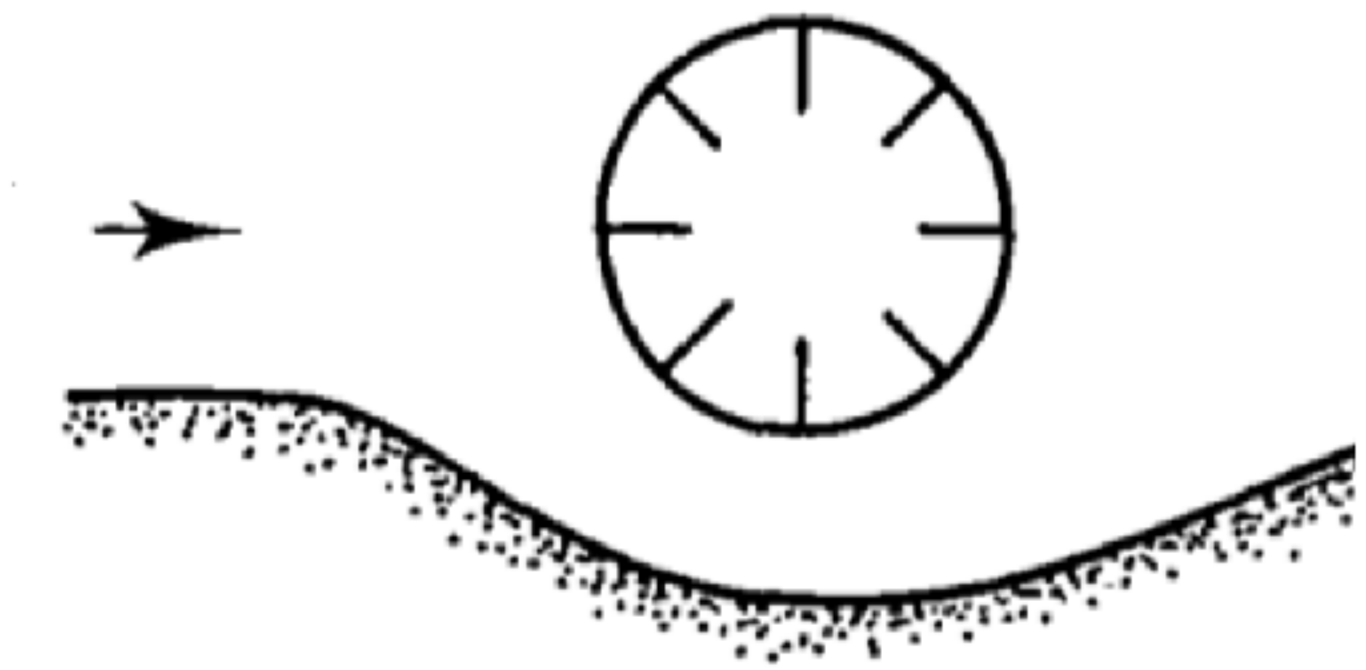
- Comparison of **Eulerian-Eulerian model predictions** with experimental data
- **Wide range of** particle **properties**: medium sand - 2.6mm acrylic - 3mm PMMA particles
- **Wide range of Shields** number: θ = 0.1-3.5

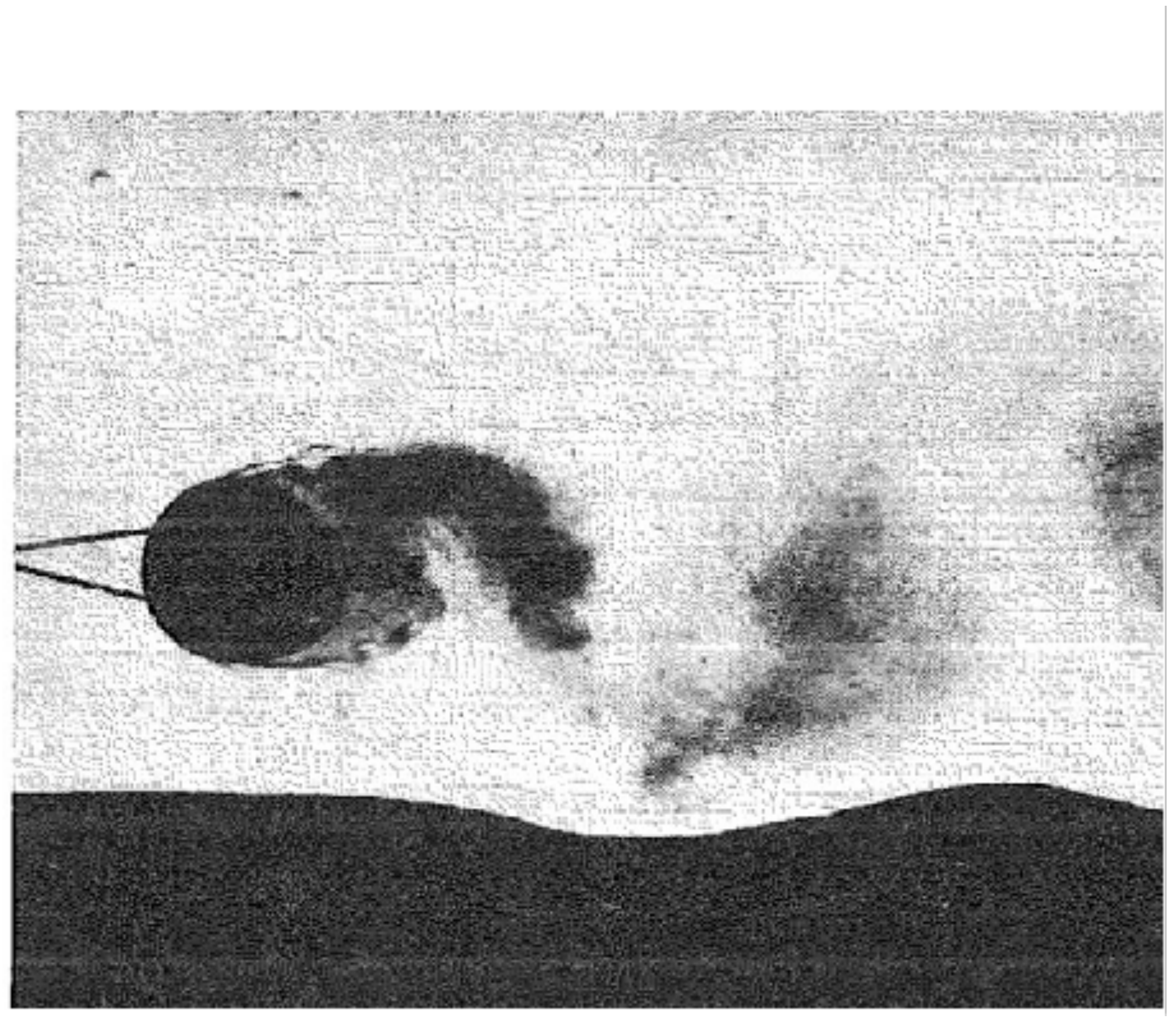
Chauchat (2018); Nagel (PhD Thesis)

Test case 3: Application to scour around a pipeline

3 stages of scour below a pipeline: onset, tunneling, lee-wake erosion

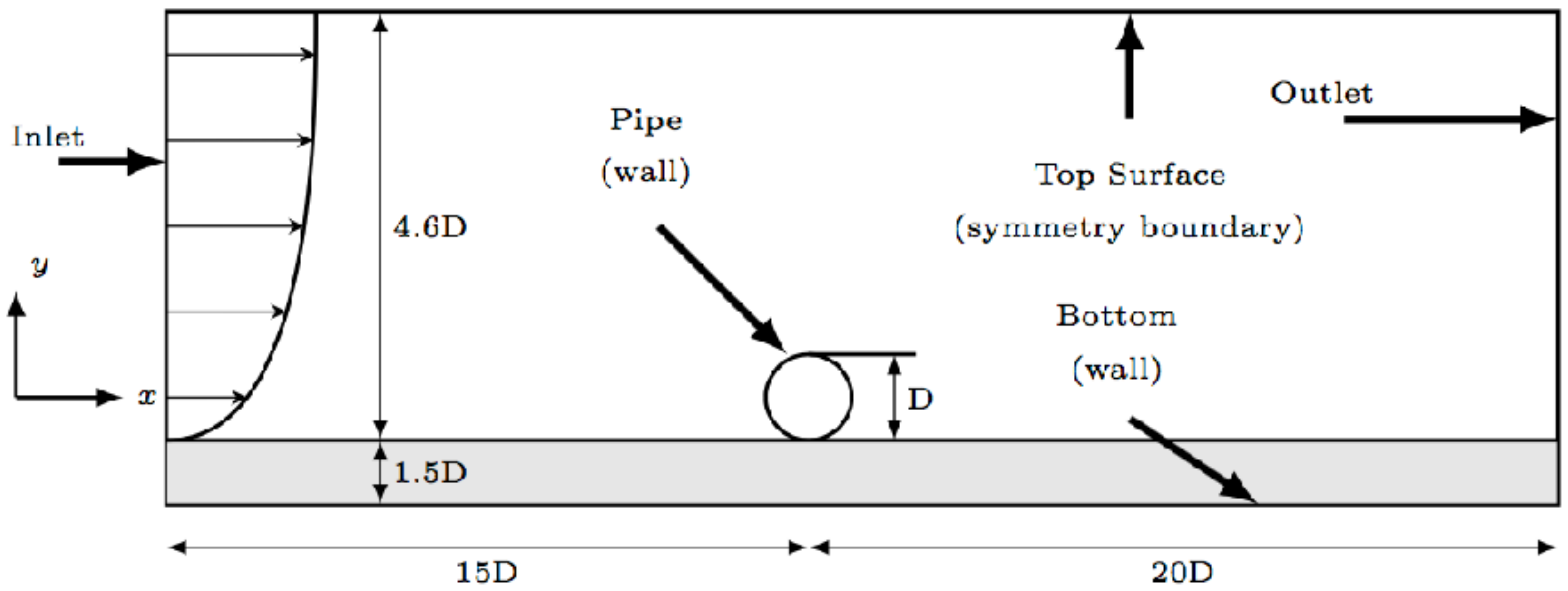






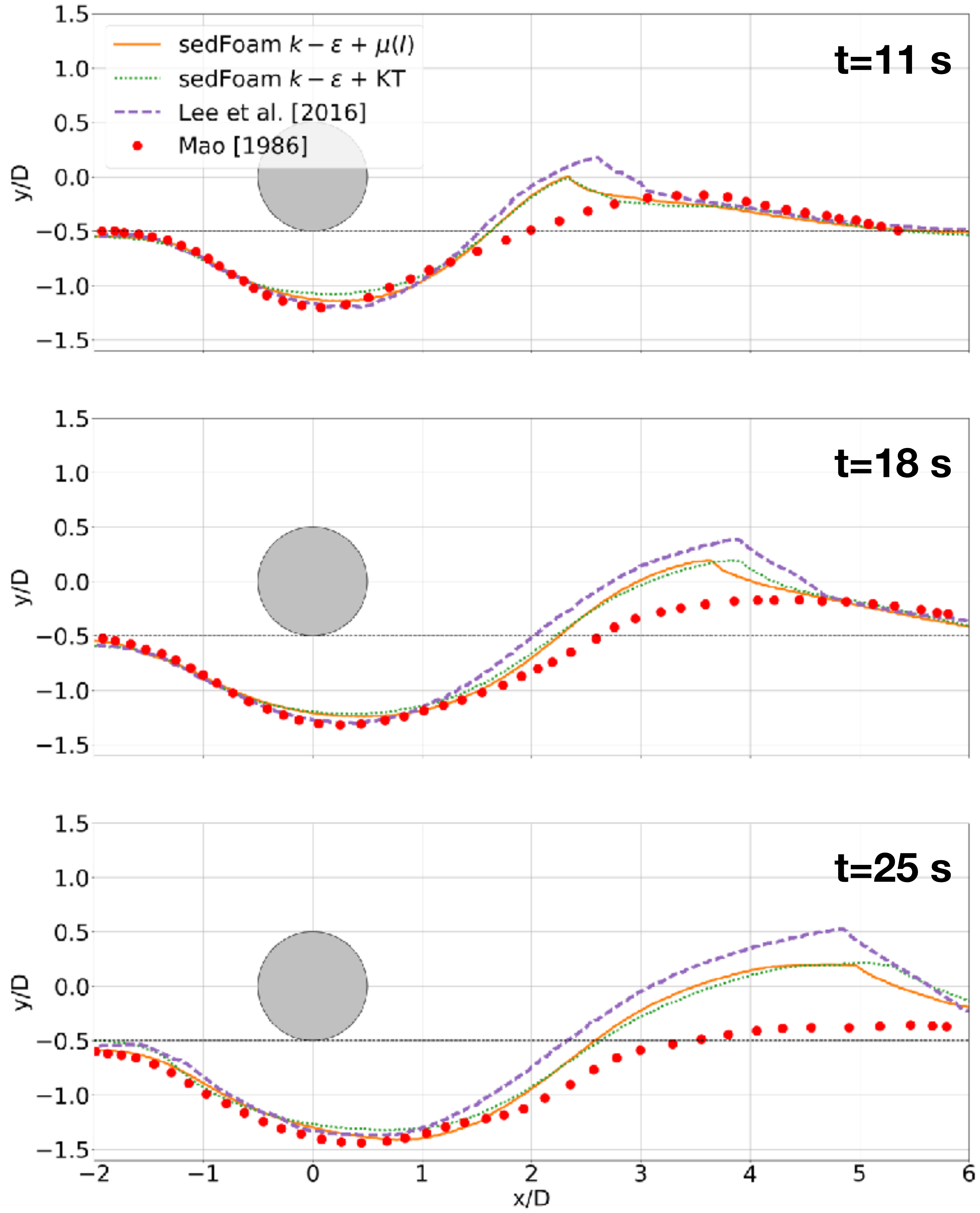
- Non-structured grid: N~200 000 cells

  (Δx~Δy ~ 0.75 - 3 mm)
- D=0.05m, Re_D=4.3 10^4
- Medium sand: d=360µm, ρ_s=2650 kg/m³, θ_0=0.33
- μ(I) rheology + two-equation turbulence models



Mathieu et al. Water (2019)

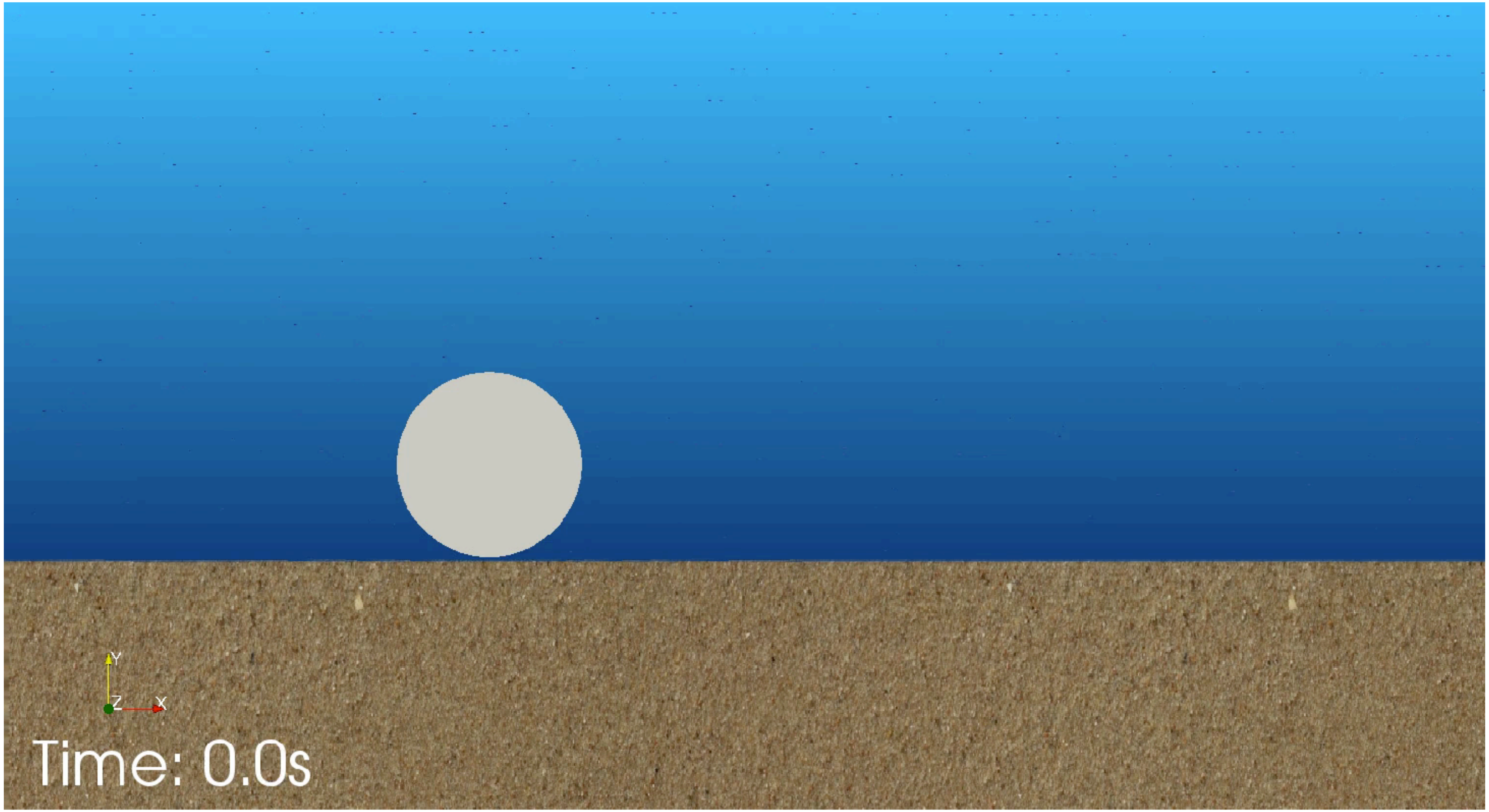
Test case 3: Applications to scour around a pipeline



- k-ε model is not able to reproduce vortex shedding
- We developed a hybrid k-ε/k-ω model to simulate both the tunneling and the lee-wake erosion stages.
  - k-ε behavior in the near bed region
  - k-ω behavior near solid walls
- More work has to be done on turbulence modeling…

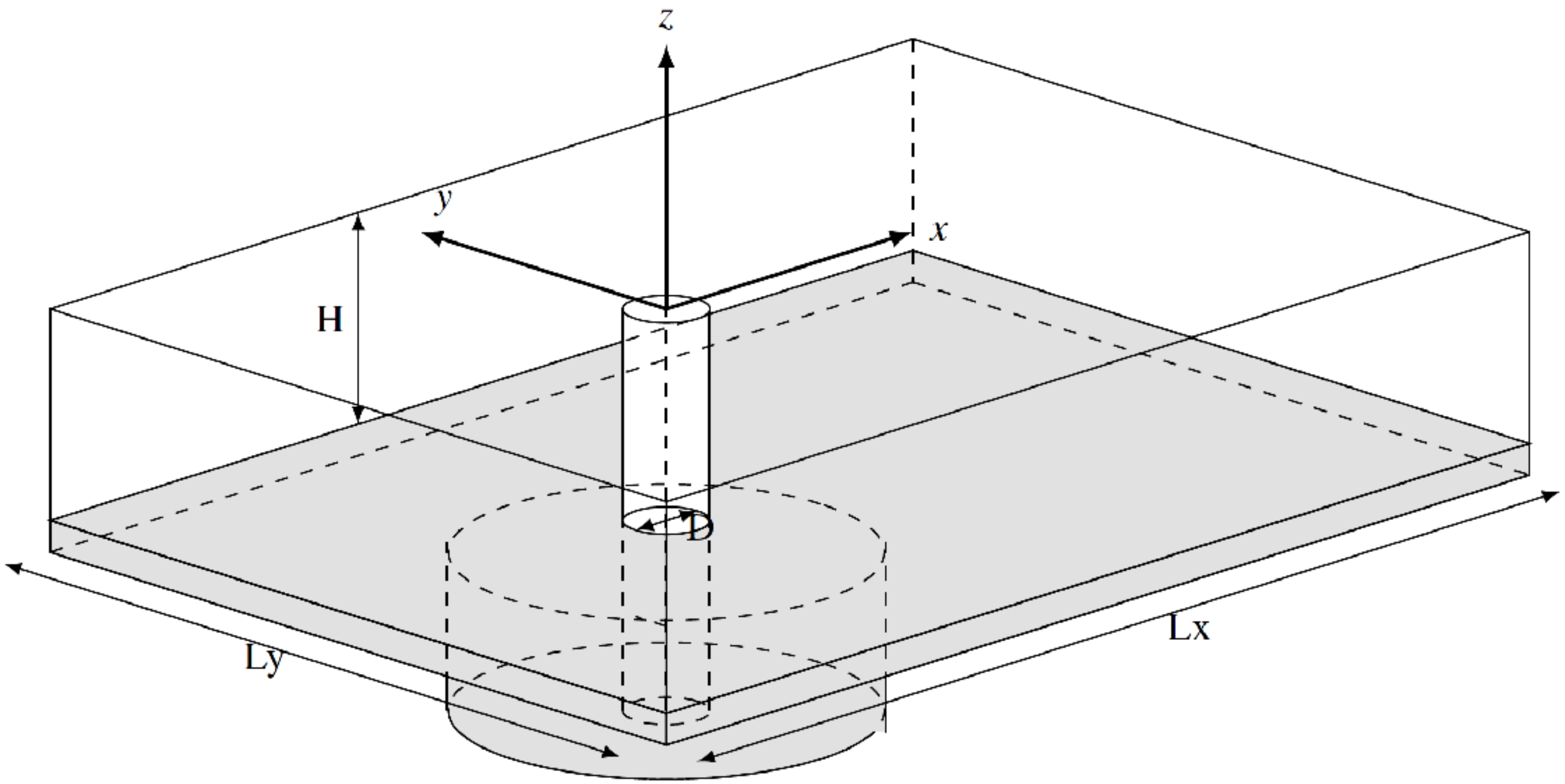
Mathieu et al. Water (2019)

Test case 3: Applications to scour around a pipeline



Mathieu et al. Water (2019)

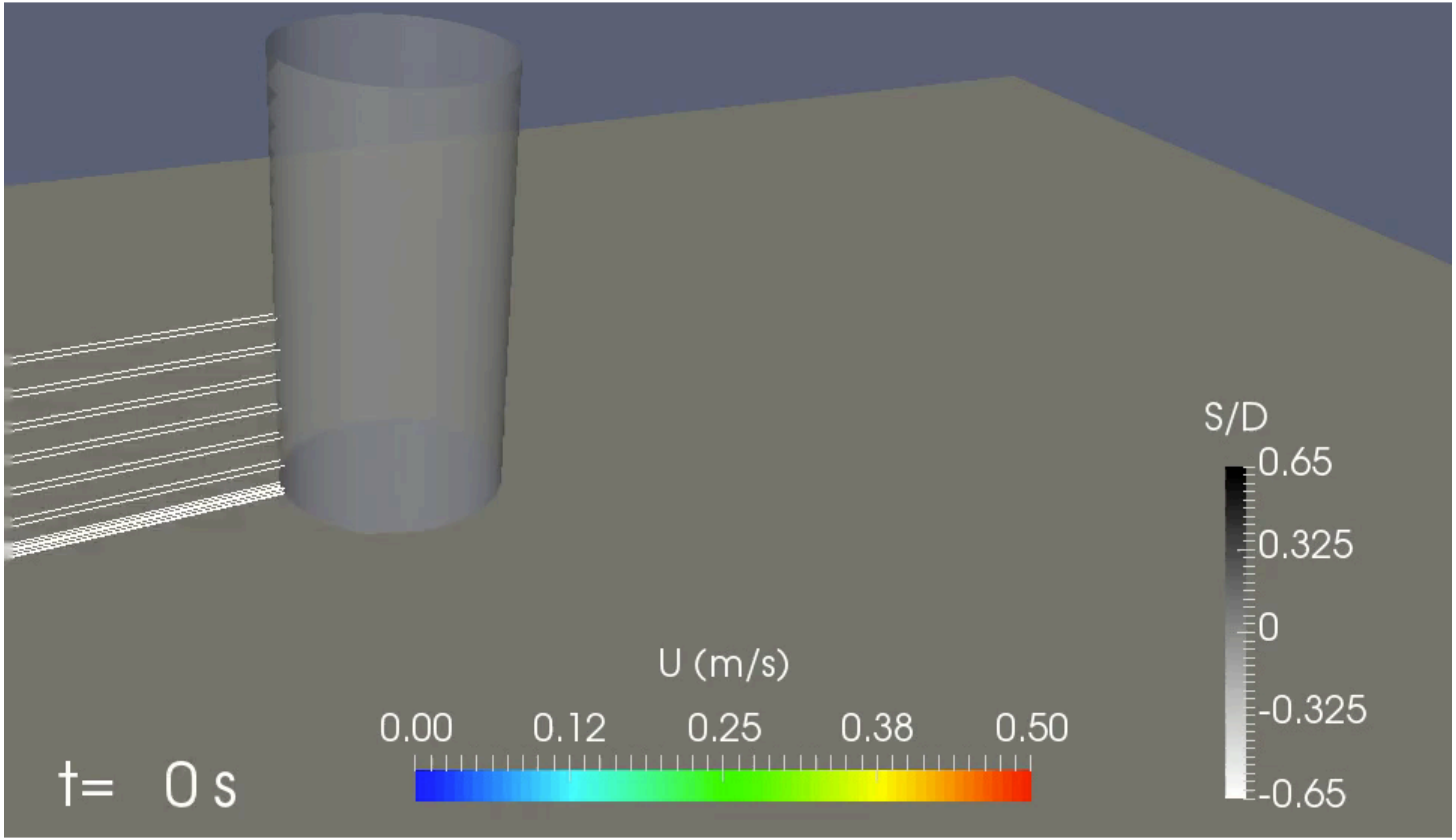
Test case 4: Applications to scour around a bridge pile



- **3D scour:** 5 million cells
  - 600s of dynamics = 110 000 CPU hours
  - ~ 20 days on 224 CPUs ~ 12 yrs on 1CPU
- D=0.1m, Re_D=4.6 10^4
- Medium sand: d=260µm, ϱ_s=2650 kg/m^3
- Live-bed configuration: θ_0=0.2
- μ(I) rheology + k-omega Wilcox 2006

Nagel et al. ADWR (in prep.)

Test case 4: Applications to scour around a bridge pile



Nagel et al. ADWR (in prep.)

Conclusions

- **Open-source framework for two-phase flow modeling** of sediment transport
- Basic **validation on fundamental problems**: sedimentation & laminar bed-load
- **Turbulence modeling** using « classical » 2 equations models: **k-ε & k-ω models**
- **Granular stress models:** μ**(I)** and **Kinetic Theory**
- **Validation on sheet-flows**: vertical structure + sediment flux and transport layer thickness Vs θ
- **Application** to multi-dimensional problems: **scour** around a pipeline and « bridge pier »

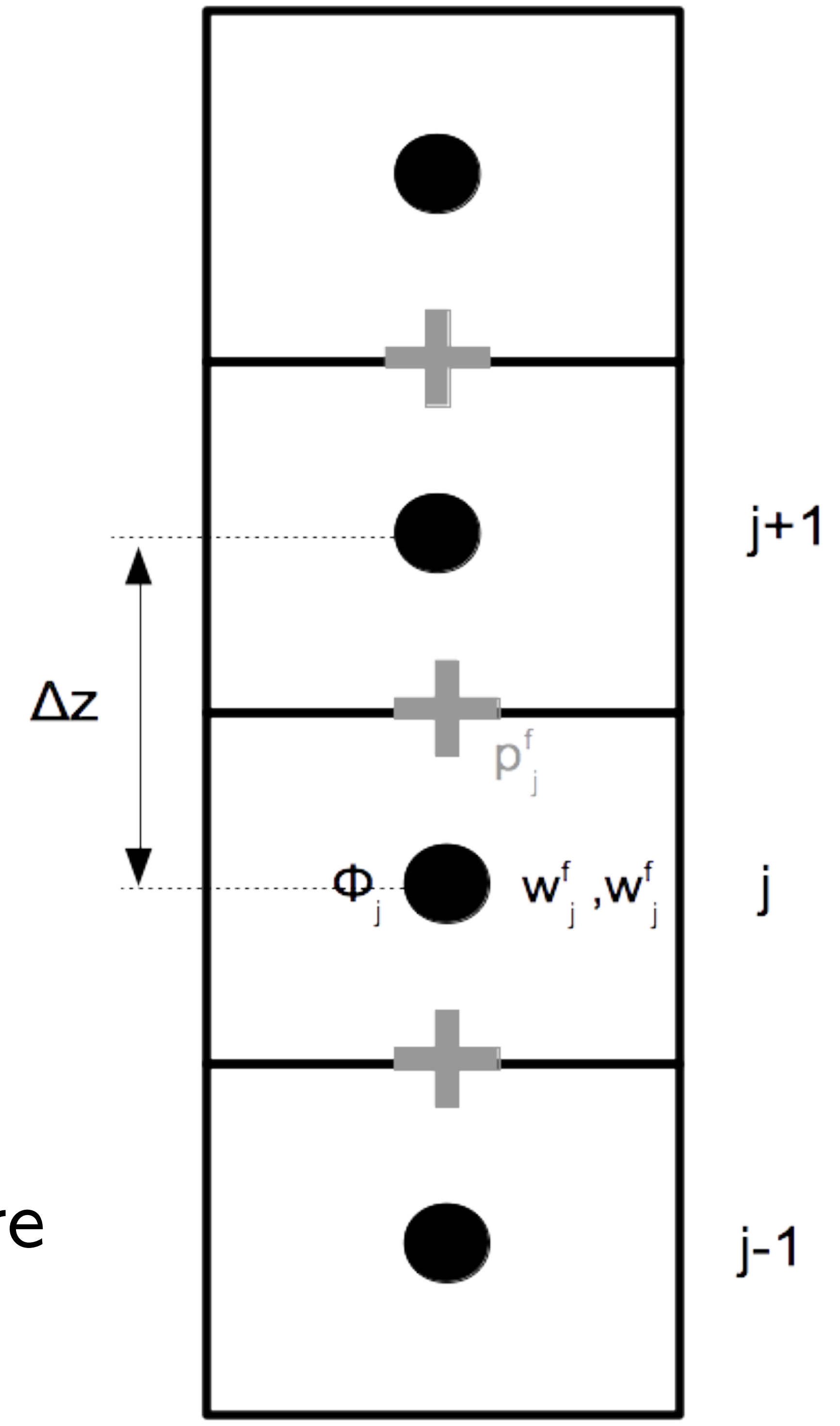
Perspectives

- Develop **more reliable turbulence models** to account for the presence of sediment particles
- Develop **accurate sub grid scale models for LES** - A. Mathieu PhD 2018-2021
- Implement **extended kinetic theory** and better elastic stress models
- Develop a **multi-class model** to reproduce **grain size sorting** mechanisms - H. Rousseau PhD 2018-2021
- Implement **dilatancy** and **pore-pressure coupling** - B. Tsai PhD (UD)
- Perform **ripple migration** simulations to disentangle suspended/bedload/near bed suspended load - A. Salimi PhD (UD)
- Develop a **free surface resolving two-phase flow model** (Kim *et al.*, 2018)

Numerical algorithm for the pressure-velocity coupling

The algorithm is based on the following steps:

1. Solve for $\{\phi\}_j^{n+1}$ using $\dfrac{\partial \phi}{\partial t} + \dfrac{\partial \phi w_p}{\partial z} = 0$
2. Solve for $\{\epsilon\}_j^{n+1}$ using $\epsilon = 1 - \phi$
3. Solve for intermediate velocities $\{w_f\}_j^*$ and $\{w_s\}_j^*$
4. Solve for the pressure $\{p_f\}_j^*$ using the poisson equation
5. Correct the velocities $\{w_f\}_j^{n+1}$ and $\{w_s\}_j^{n+1}$ using the new pressure



There are different methods for solving the pressure-velocity coupling, they are almost all based on predictor-corrector algorithm. In the following the PISO (Pressure Implicit with Splitting of Operators) algorithm is detailed.

Pressure Implicit with Splitting of Operators (PISO) algorithm

The PISO algorithm requires the momentum equations to be written in a semi-discretized form. We start by writing the fluid phase momentum equation in the phase intensive form:

$$\frac{\partial w_f}{\partial t} + w_f \frac{\partial w_f}{\partial z} = -\frac{}{\rho_f}\frac{\partial p_f}{\partial z} - g - \frac{\phi \ \rho_p}{\epsilon \rho_f t_p}\ (w_f - w_s)$$

The semi-discrete form of the equation can be written in matrix form as:

$$\left[A^f\right]_{ij} \{w_f\}_j^* = \left\{H^f\right\}_j - \frac{1}{\rho_f}\frac{\partial \{p_f\}_j^*}{\partial z}$$

where $\left[A^f\right]_{ij}$ contains implicit advection and drag terms, $\left\{H^f\right\}_j$ contains explicit source terms including temporal derivative, gravity, explicit drag term (solid phase contribution) and the index j represents the j^{st} grid node in the mesh.

For example, using a first order Euler scheme for the time derivative, the vector $\left\{H^f\right\}_j$ can be written as: $\left\{H^f\right\}_j = \frac{1}{\Delta t}\{w_f\}_j^n - g + \frac{\{\phi\}_j^{n+1} \ \rho_p}{\{\epsilon\}_j^{n+1} \ \rho_f t_p}\{w_s\}_j^n$

where Δt the time step.

Pressure Implicit with Splitting of Operators (PISO) algorithm

Using a first order Upwind scheme, the matrix coefficients $\left[A^f\right]_{ij}$ are given by:

$$\left[A^f\right]_{ij} = -\frac{\left\{w_f\right\}_j^n}{\Delta z}\delta_{i\ j-1} + \left(\frac{\left\{w_f\right\}_j^n}{\Delta z} + \frac{\{\phi\}_j^{n+1}\ \rho_p}{\{\epsilon\}_j^{n+1}\ \rho_f t_p}\right)\delta_{i\ j}$$

where $\delta_{i\ j}$ is the Kronecker symbol and Δz the grid size assumed uniform for simplicity. In matrix form, it reads:

$$\left[A^f\right] \times \left\{w_f\right\}^{n+1} = \begin{bmatrix} \ddots & & \ddots & \\ -\frac{\left\{w_f\right\}_j^n}{\Delta z} & & \frac{\left\{w_f\right\}_j^n}{\Delta z} + \frac{\{\phi\}_j^{n+1}\ \rho_p}{\{\epsilon\}_j^{n+1}\ \rho_f t_p} & \\ \ddots & & \ddots & \end{bmatrix} \begin{pmatrix} \vdots \\ \left\{w_f\right\}_{j-1} \\ \left\{w_f\right\}_j \\ \vdots \end{pmatrix}^{n+1}$$

1. Velocity predictor step

Using the discretized momentum equation, the predictor step for the fluid phase can be written formally as:

$$\left\{w_f\right\}_j^* = \left[A^f\right]_{ij}^{-1} \left\{H^f\right\}_j$$

This step requires the inversion of the matrix $\left[A^f\right]_{ij}$.

Pressure Implicit with Splitting of Operators (PISO) algorithm

Similarly, the solid phase momentum equation in matrix form can be written as:

$$[A^s]_{ij} \{w_s\}_j^* = \{H^s\}_j - \frac{1}{\rho_p} \frac{\partial \{p_f\}_j^*}{\partial z}$$

where the term $\{H^s\}$ also contains the particle pressure contribution:

$$\{H^s\}_j = \frac{1}{\Delta t} \{w_s\}_j^n - g + \frac{1}{t_p} \{w_f\}_j^n$$

Using the discretized momentum equation, the predictor step for the solid phase can be written formally as:
$\{w_s\}_j^* = [A^s]_{ij}^{-1} \{H^s\}_j$

The velocity correction equations integrate the fluid pressure gradient correction and provide the corrected velocity fields $\{w_f\}_j^{**}$ and $\{w_s\}_j^{**}$:

$$\{w_f\}_j^{**} = \{w_f\}_j^* - \frac{[A^f]_{ij}^{-1}}{\rho_f} \frac{\partial \{p_f\}_j^*}{\partial z}$$

$$\{w_s\}_j^{**} = \{w_s\}_j^* - \frac{[A^s]_{ij}^{-1}}{\rho_p} \frac{\partial \{p_f\}_j^*}{\partial z}$$

Pressure Implicit with Splitting of Operators (PISO) algorithm

2. Pressure solution

The corrected velocity fields should be divergence-free for the volume-averaged mixture velocity: $\{w_m\}_j^{**} = \{\epsilon\}_j^{n+1}\{w_f\}_j^{**} + \{\phi\}_j^{n+1}\{w_s\}_j^{**}$

$$\frac{\partial \{w_m\}_j^{**}}{\partial z} = 0$$

$$\Leftrightarrow \frac{\partial}{\partial z}\left(\{\epsilon\}_j^{n+1}\{w_f\}_j^{**} + \{\phi\}_j^{n+1}\{w_s\}_j^{**}\right) = 0$$

$$\Leftrightarrow \frac{\partial}{\partial z}\left[\left(\frac{\{\epsilon\}_j^{n+1}}{\rho_f\left[A^f\right]_{ij}} + \frac{\{\phi\}_j^{n+1}}{\rho_p\left[A^s\right]_{ij}}\right)\frac{\partial\{p_f\}_j^*}{\partial z}\right] = \frac{\partial}{\partial z}\left(\{\epsilon\}_j^{n+1}\{w_f\}_j^* + \{\phi\}_j^{n+1}\{w_s\}_j^*\right)$$

Using a staggered grid for between the pressure and the velocity to avoid Rhie and Chow oscillations, the Poisson equation can be discretized as:

$$\Leftrightarrow \frac{1}{2\Delta z}\left[\left(\frac{\{\epsilon\}_{j+1}^{n+1}}{\rho^f\left[A^f\right]_{ij}} + \frac{\{\phi\}_{j+1}^{n+1}}{\rho^p\left[A^s\right]_{ij}}\right)\frac{\{p_f\}_{j+1}^* - \{p_f\}_j^*}{\Delta z} - \left(\frac{\{\epsilon\}_j^{n+1}}{\rho^f\left[A^f\right]_{ij}} + \frac{\{\phi\}_j^{n+1}}{\rho^p\left[A^s\right]_{ij}}\right)\frac{\{p_f\}_j^* - \{p_f\}_{j-1}^*}{\Delta z}\right]$$

$$= \frac{\{\epsilon\}_{j+1}^{n+1}\{w^f\}_{j+1}^* + \{\phi\}_{j+1}^{n+1}\{w^p\}_{j+1}^* - \left(\{\epsilon\}_j^{n+1}\{w^f\}_j^* + \{\phi\}_j^{n+1}\{w^p\}_j^*\right)}{\Delta z}$$

Pressure Implicit with Splitting of Operators (PISO) algorithm

3. Velocity corrector step

Using the newly computed pressure field $\{p_f\}_j^*$, the velocity correction equations can be used to correct the velocity fields:

$$\{w_f\}_j^{**} = \{w_f\}_j^* - \frac{[A^f]_{ij}^{-1}}{\rho_f} \frac{\partial \{p_f\}_j^*}{\partial z}$$

$$\{w_s\}_j^{**} = \{w_s\}_j^* - \frac{[A^s]_{ij}^{-1}}{\rho_p} \frac{\partial \{p_f\}_j^*}{\partial z}$$

Remarks:

1. In one-dimensional problems the solution of this equation is cheap and a simple double sweep algorithm can be used Thomas (1995) however for three-dimensional problems this can become very expensive as Bi-Conjugate Gradient algorithms might become necessary to resolve the algebraic system associated with the previous equation.
2. The algorithm presented above is implemented in sedFOAM an open-source Eulerian-Eulerian two- phase flow model developed under the open-source CFD toolbox openFOAM. The 3D version of this Gravity driven settling: sedimentation of non-cohesive particles in the viscous regime algorithm is presented in Chauchat et al. (2017).