Flow Analysis of Porous Structure Using OpenFOAM

Jae-Ryul Shin, Byoung-Yun Kim
Research & Development Center
NEXT*f*oam CO., LTD.

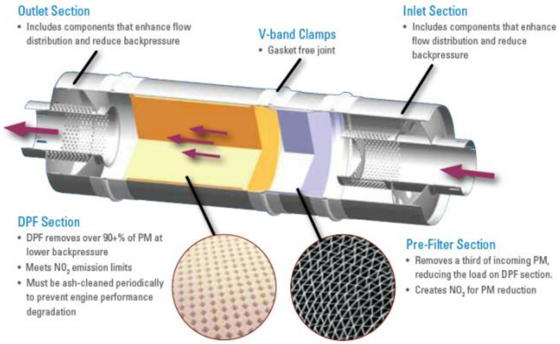
Contents

- Introduction
  - Objective
  - OpenFOAM
- Flow Dynamic Modeling
  - Governing Equations
  - Solution Method
- Computational Result
  - Porous Dam Break 2D / 3D
- Summary and Remark

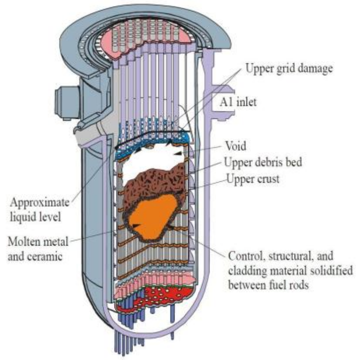
Introduction

- Motivation
  - Multiphase or multi-fluid flow with porous media in engineering fields

Breakwaters



DPF

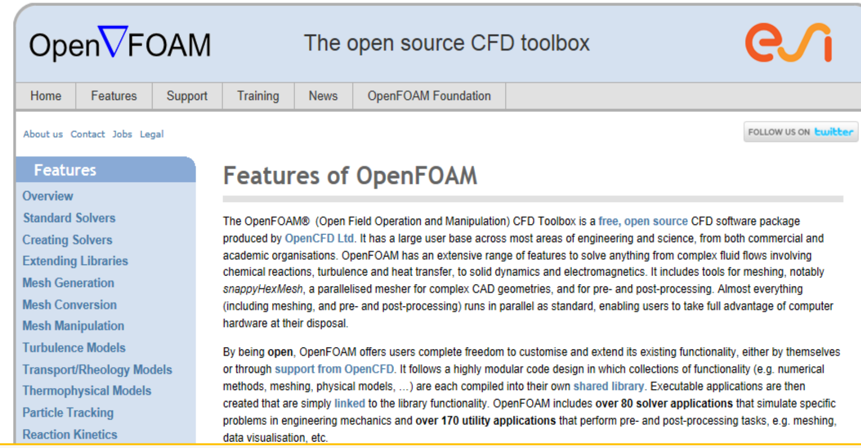


Reactors

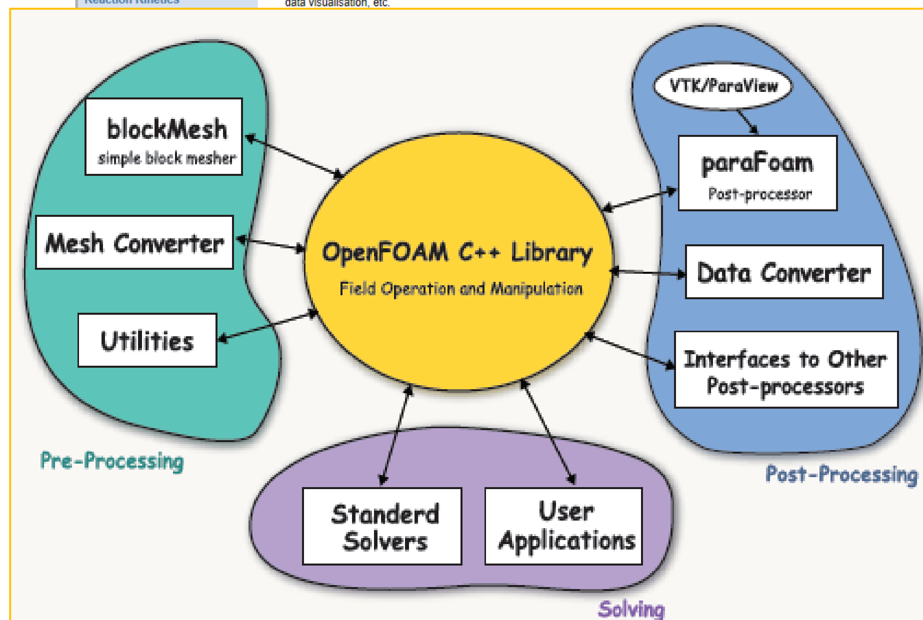
- Objectives
  - Validation of precompiled OpenFOAM solver
    - Minimum functions solve a flow field
    - Find out problems and modifying functions to appropriate solution

Introduction

- OpenFOAM
  - **Open** F**ield** **O**peration **a**nd **M**anipulation
  - Open source CFD toolbox written in C++
  - Freely available and open source
  - Licensed under the GNU General Public License
  - Homepage: http://www.openfoam.org



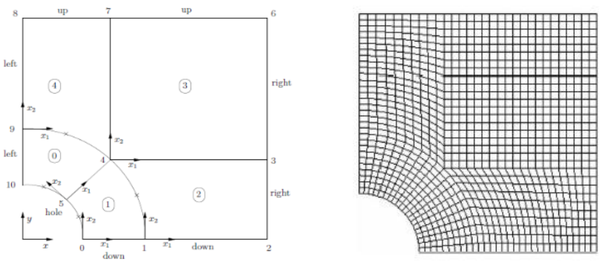
- Solver
  - Over 80 solvers
  - Over 170 applications
- Library
  - Turbulence, Thermophysical, Chemistry
- Mesh
  - Mesh Generator, Converting
- Core Tech.
  - Numerical Method
  - Dynamic Mesh, Parallel Computing



Introduction

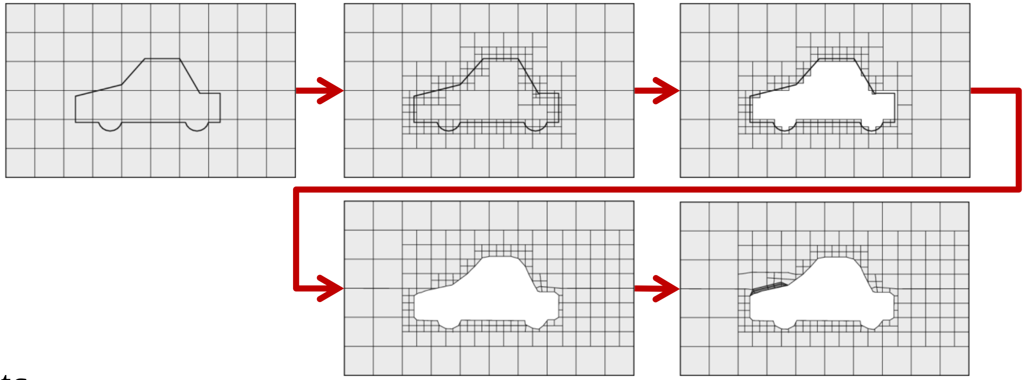
- Pre-processing (Mesh generating: blockMesh, snappyHexMesh, mesh converting)
- Setup Boundary Condition, choose PDE solver, Run
- Post-processing (Visualization: paraview)

- Pre-processing
  - Block mesh
    - Vertex / Edge / Face
    - Block



Block mesh

  - Snappy hex mesh (Automatic Mesh generating)
    - Create base mesh
    - Refine base mesh
    - Remove unused cells
    - Snap mesh to surface
    - Add layers



Automatic mesh procedure

  - Mesh converting to Foam
    - Ansys, CFX, Fluent, gambit, plot3D, etc

Flow Dynamic Modeling

- Governing Equations
  - Navier-Stokes Equation
    - Continuity

$$\frac{\partial u_i}{\partial x_i} = 0$$

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

    - Momentum

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_j u_i) = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j}(2\mu S_{ij}) + B_i$$

    - Body Force = porous drag (Darcy-Forchheimer) + gravity

$$B_i = -\left[\mu D_{ij} + \frac{1}{2}\rho \,|u|\, F_{ij}\right]u_i - \rho g_i$$

    - Volume of Fraction

$$\frac{\partial \alpha}{\partial t} + \frac{\partial}{\partial x_j}(\alpha u_j) = 0$$

    - Ergun

$$D = \frac{150(1-\Phi)^2}{d_p^2 \Phi^3} \quad F = \frac{1.75(1-\Phi)}{d_p \Phi^3}$$

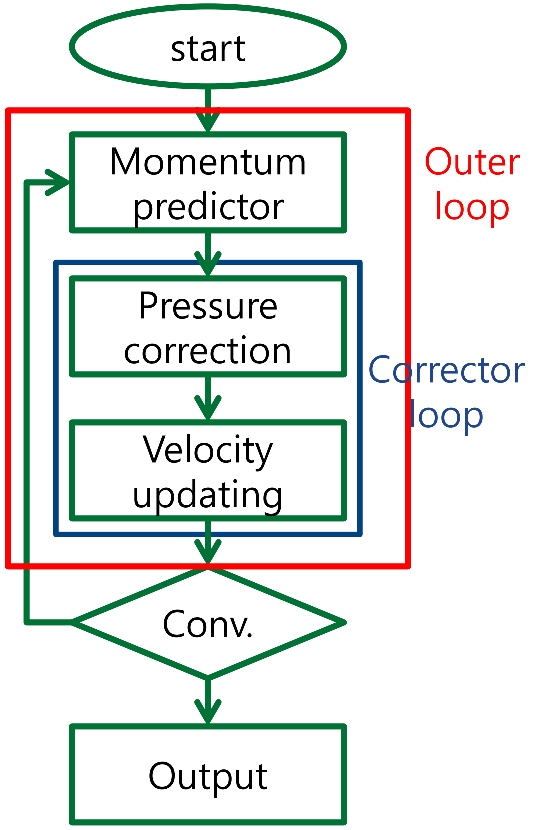
- Turbulence Model
  - Two equation model: k-ω SST model

$$\frac{\partial}{\partial t}(\rho k) + \frac{\partial}{\partial x_j}(\rho u_j k) = \frac{\partial}{\partial x_j}\left[(\mu + \sigma_{k1}\mu_t)\frac{\partial k}{\partial x_j}\right] + P_k - \beta^* \rho k \omega$$

$$\frac{\partial}{\partial t}(\rho \omega) + \frac{\partial}{\partial x_j}(\rho u_j \omega) = \frac{\partial}{\partial x_j}\left[(\mu + \sigma_{\omega 1}\mu_t)\frac{\partial \omega}{\partial x_j}\right] + \alpha_1 \frac{\omega}{k} P_k - \beta_1 \rho \omega^2$$

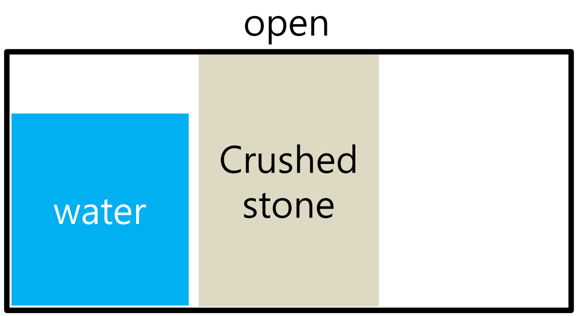
Flow Dynamic Modeling

- Time Integration
  - Backward Euler
  - Crank-Nicholson
- Spatial Discretization
  - Second / Fourth order central difference: linear / cubic
  - First / Second order upwind: upwind / linearUpwind
- Matrix Solver
  - Iterative Method (e.g. Gauss-Seidel): smoothSolver
  - Preconditioned (bi-)conjugate gradient methods: PCG, PBiCG
  - Geometric and algebraic multigrid: GAMG
- Solution Algorithm
  - SIMPLE
  - PISO
  - PIMPLE



Computational Result

- Experiment of porous dam break (Lin, p., 1998)
  – Porous dam in fish tank
  – Two type of granular materials: crushed stone, glass beads



Experiment schematic view

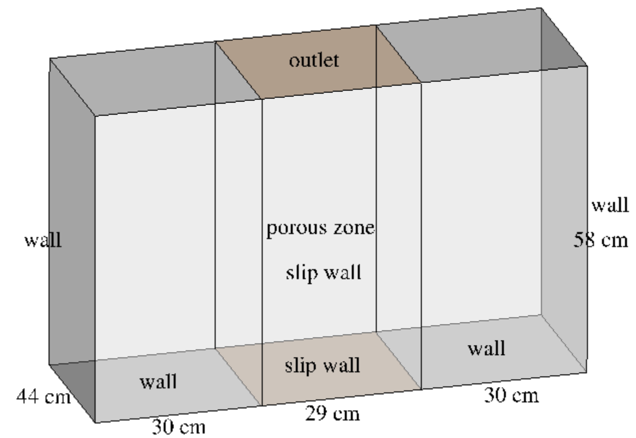
Material	Φ	d_p [cm]
Crushed stone	0.49	1.59
Glass beads	0.39	0.3

Porous media property

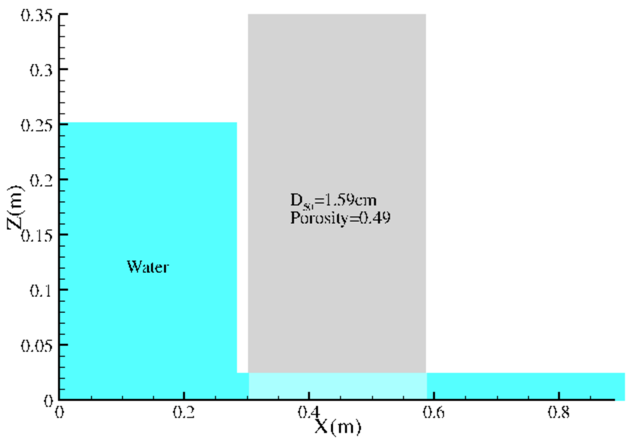
- Benchmark test
  – Darcy-Forchheimer Coefficients
    - 2D simulation
  – k-ω SST turbulence model
    - 2D simulation
  – Grid dependency
    - 3D simulation

Computational Result

- 2D computational Set up
  - Domain Size: 89cm×58cm
  - Cell size: 1cm×0.5cm
  - Total cell No.: 10,324
- 3D computational Set up
  - Domain Size: 89cm×58cm×44cm
  - Cell size
    - Coarse: 1cm×0.5cm×1cm
    - Fine: 0.5cm×0.25cm×0.5cm
  - Total cell No.
    - Coarse: 454,256
    - Fine: 3,634,048
- Boundary and initial condition
  - Wall: non-slip, slip
  - Outlet: extrapolation
  - Initial water height: 25cm



Domain and BC

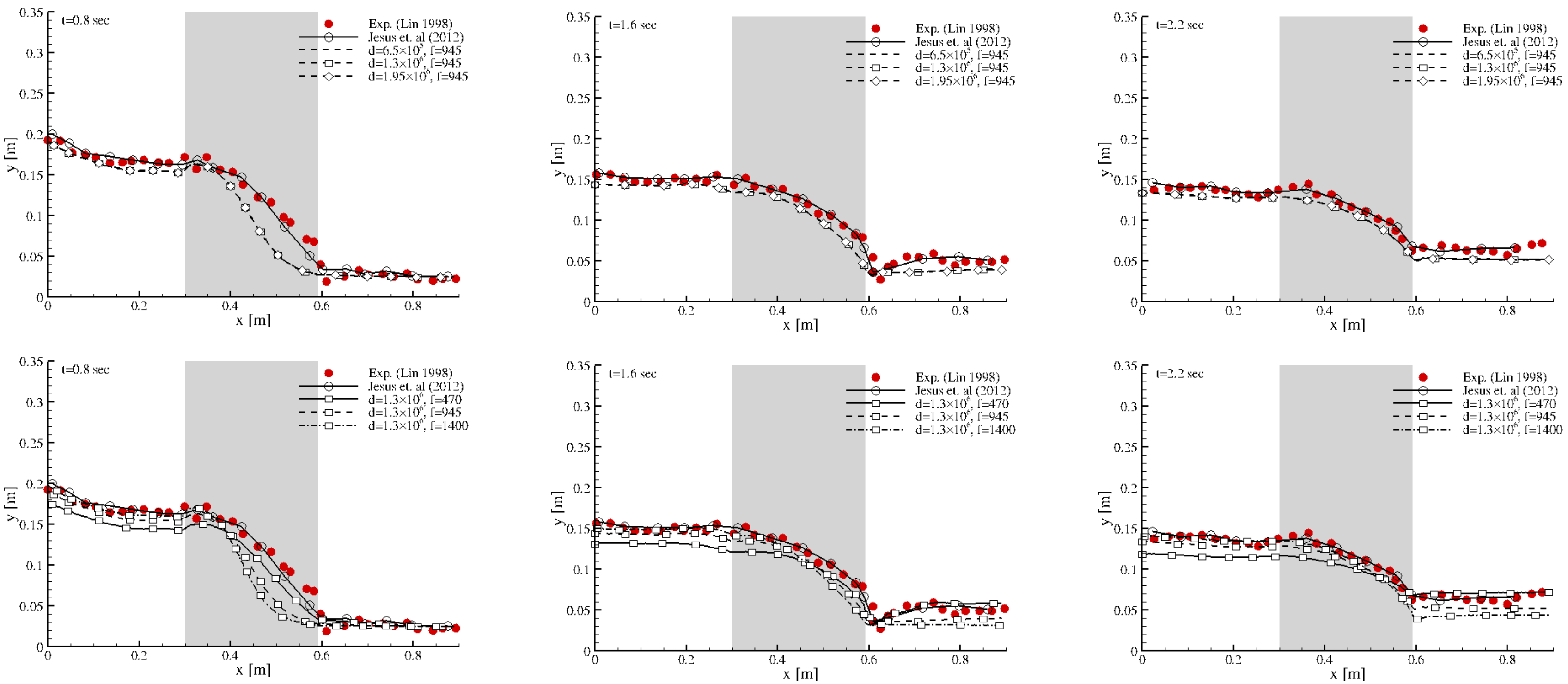


Initial Condition

Computational Result

- Dependency of Darcy-Forchheimer Coefficients
  - Comparison coefficients

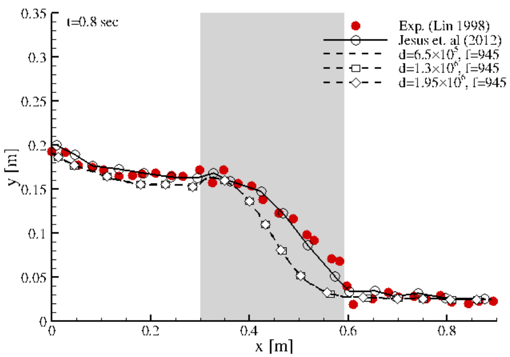
Coeff.	Value		
D	6.5×10^5	1.3×10^6	1.95×10^6
F	470	945	1400

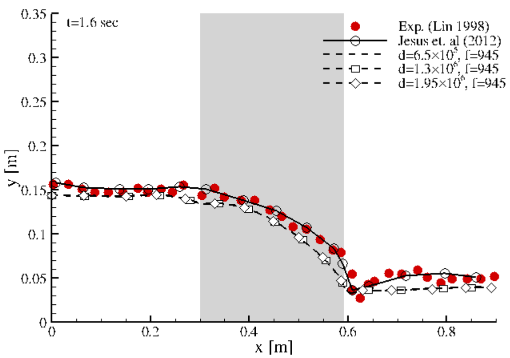


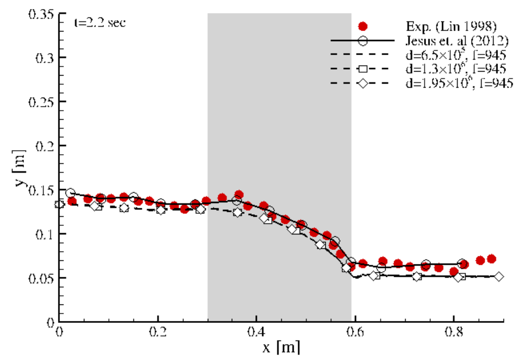
Computational Result

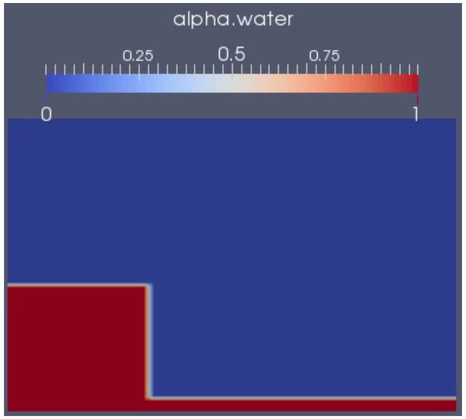
- Dependency of Darcy-Forchheimer Coefficients
  - Comparison coefficients

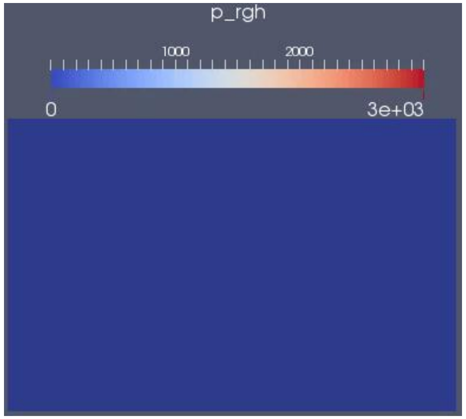
Coeff.		Value	
D	-	1.3×10^6	-
F	-	945	-

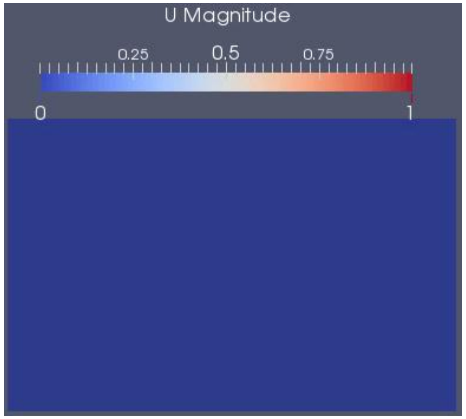








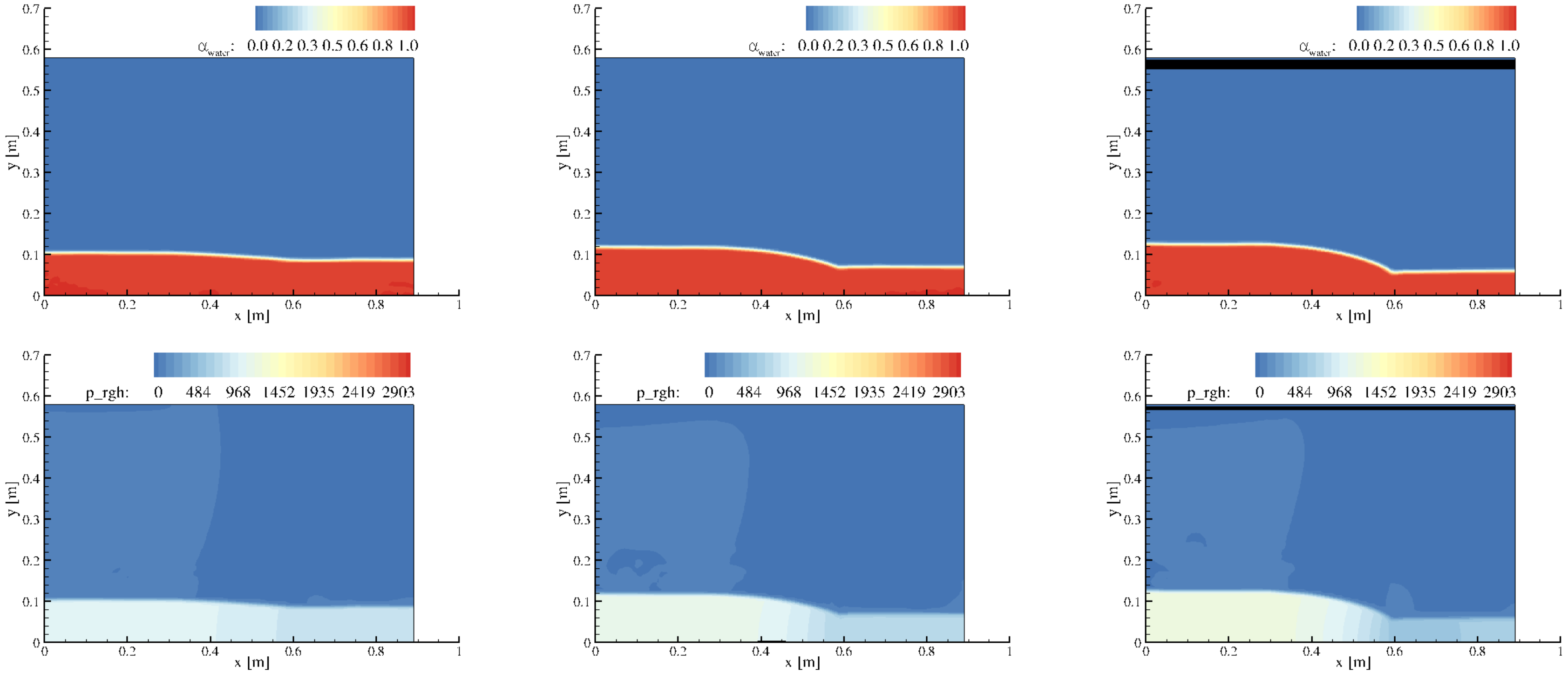




Computational Result

- Dependency of Darcy-Forchheimer Coefficients
  - Comparison coefficients

Coeff.	Value		
D	-	1.3×10^6	-
F	470	945	1400

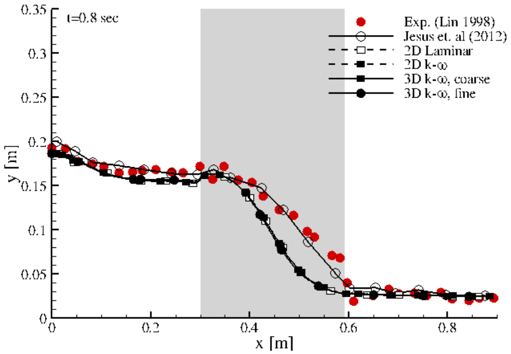


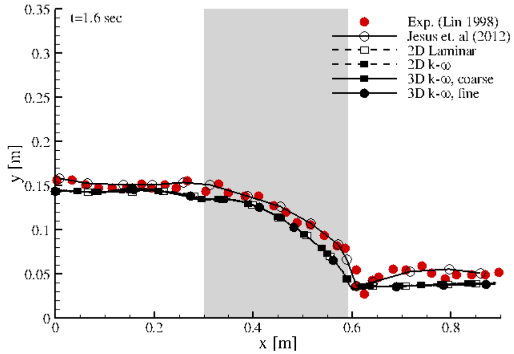
Computational Result

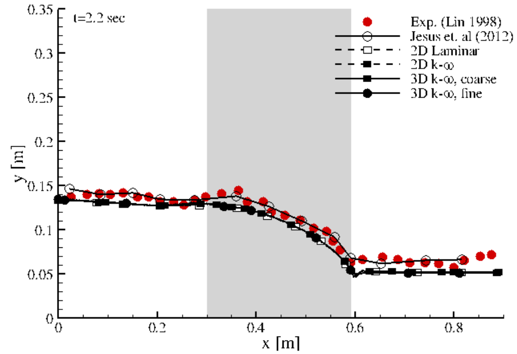
- Grid dependency and turbulence effect
  - Coarse: 1cm×0.5cm×1cm, Fine: 0.5cm×0.25cm×0.5cm, k- ω SST model

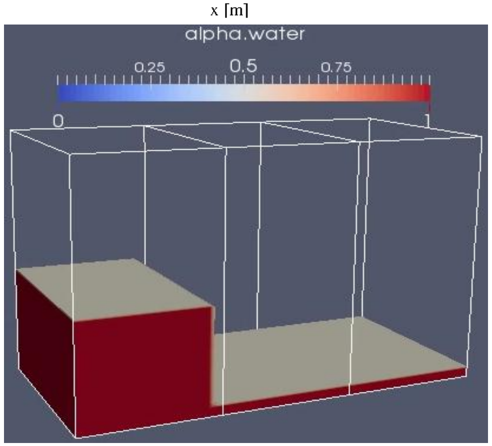
Coeff.		Value	
D	-	1.3×10^{6}	-
F	-	945	-

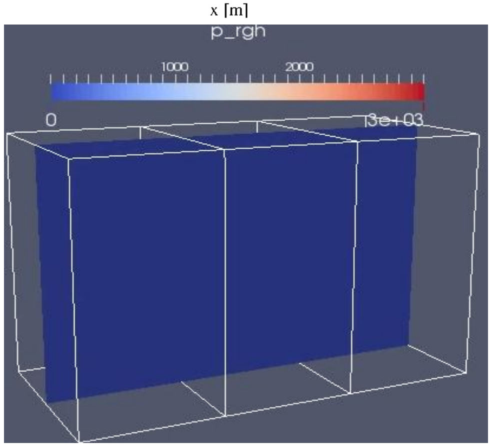
- Turbulence dose not affect the flow field
- Need to consider porosity in turbulence model
- Same result of grid dependency (may BC)

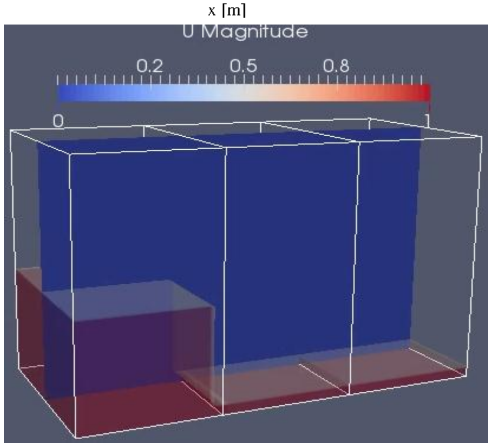












Summary and Remark

- OpenFOAM
  - Used one of precompiled solvers, called porousInterFoam
    - Pressure drop is considered only, as body force without other details
    - But it shows good performance

- Darcy-Forchheimer Coefficients
  - Guess D and F coefficient from fixed porosity and nominal diameter with Ergun equation
  - The result shows very similar behavior of motion
  - But real is not homogenous porous media
  - Need new approach, guessing D and F coefficients. e.g. Gaussian distribution.

- Grid and turbulence
  - Grid dependency
    - Both 2D and 3D are applied same BC (slip wall on sides)
  - Turbulence model dose not affect the flow field
    - Initial distribution of flow field
    - Need test other turbulence models. e.g. k-ε.

Thank you for your attention