Axisymmetric Clarifier Simulation

María Elena Valle Medina,

CNRS, ICUBE, ENGEES,

FRANCE

Index

- Overview
- Main files in OpenFOAM
- Meshing
- Setting-up a case(pre-processing)
  - 0 folder
  - Constant
  - System
- Processing
- Post-Processing
  - Paraview and other tools

Overview

Watermatex 2019

- Flow characteristics

  $Q = 540\ m^3.h^{-1}$

  $Q_{in} = Q + Q_{rec}$

  $Q_{rec} = 1.15 * Q_{in}$

- Sludge Characteristics

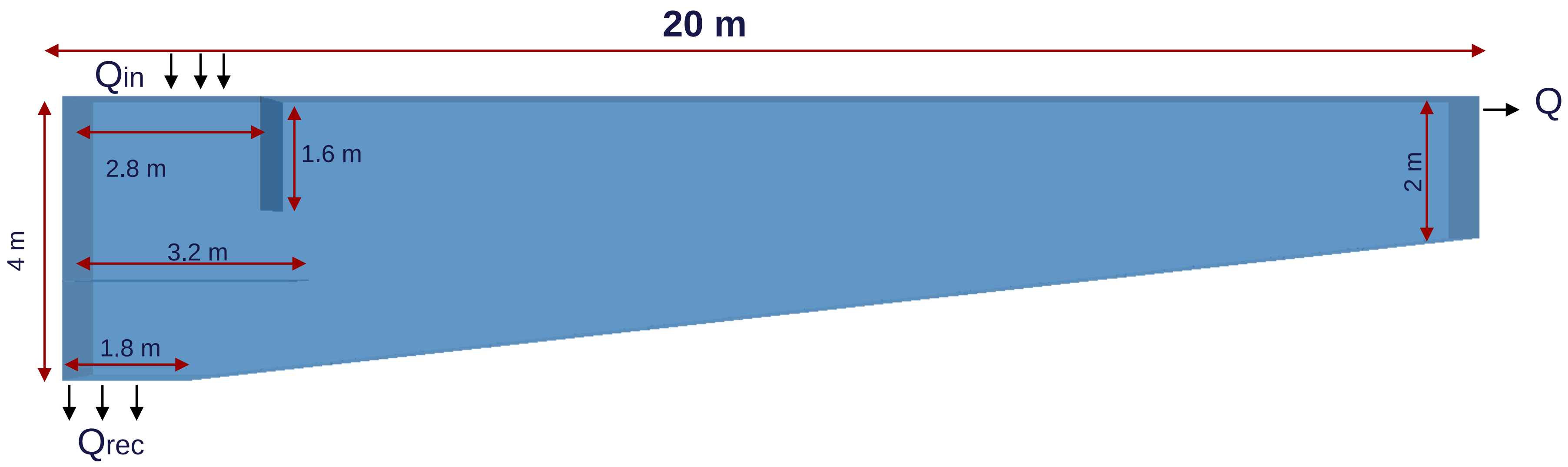
  $X_{in} = 3.9\ Kg.m^{-3}$

  $\rho_s = 1050\ Kg.m^{-3}$

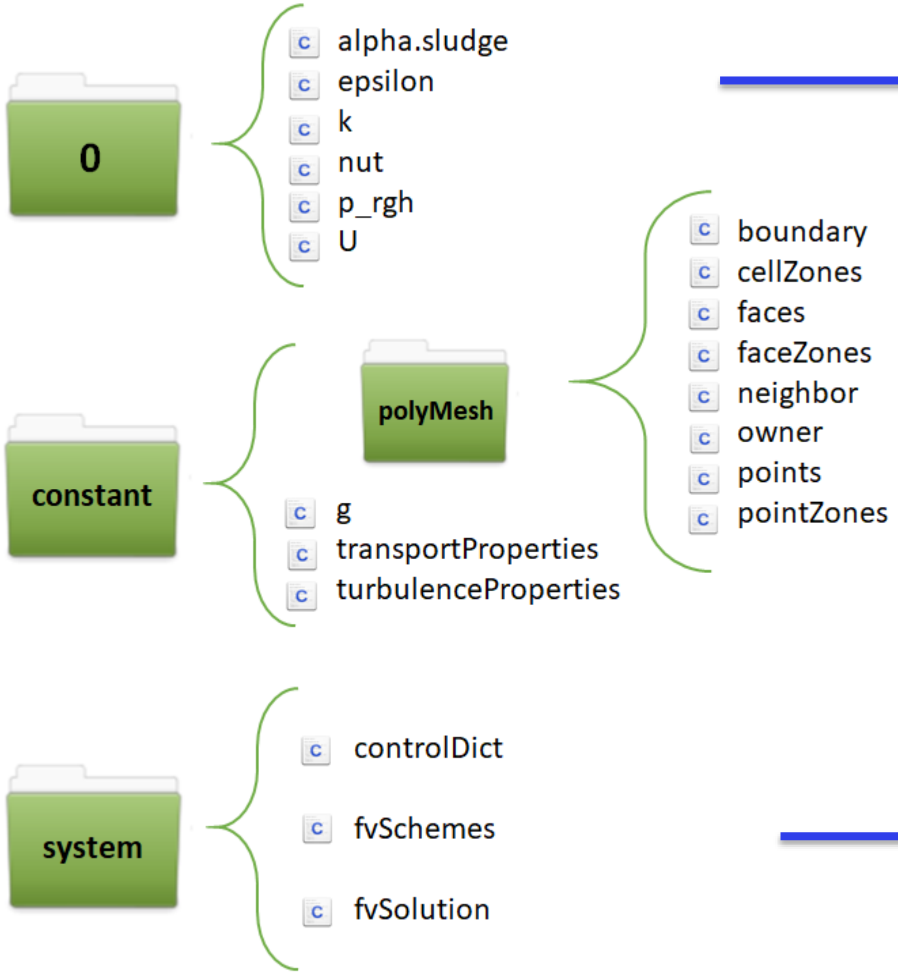
- Water Characteristics

  $\rho_w = 998\ Kg.m^{-3}$

  $v = 1.7871*10^{-6}\ m^2.s$



Main files



Variables during the simulation

Constants, in this case for:

- Settling velocity model
- Rheology model
- Turbulence model

Simulation controls:

- Time step
- Numerical schemes

BE ALWAYS AWARE TO NAME THE FILES WITHOUT BLANK SPACES.

Meshing

1. Create a folder called "triSurace" and place the stl files within it. The *triSurface* folder is then placed on the constant directory. <u>It is not necessary to have other files within this folder while meshing.</u>

- back.stl
- baffled.stl
- baffleu.stl
- front.stl
- inlet.stl
- outlet.stl
- removal.stl
- surface.stl
- symmetry.stl
- walls.stl

2. In the *system* Folder, open the *SurfaceFeatureExtractDict* file and introduce the name of all the stl.files. The order of the boundary names is not important.

```
baffled.stl
{
    // How to obtain raw features (extractFromFile || extractFromSurface)
    extractionMethod    extractFromSurface;

    extractFromSurfaceCoeffs
    {
        // Mark edges whose adjacent surface normals are at an angle less
        // than includedAngle as features
        // - 0  : selects no edges
        // - 180: selects all edges
        includedAngle   150;
    }

    // Write options

        // Write features to obj format for postprocessing
        writeObj                yes;
}
```

3. Then, open the blockMeshDict file to create the main mesh bounding the geometry.

```
convertToMeters 1;
vertices
(
   ( -1.0 -1.0 -1.0)
   ( 22.0 -1.0 -1.0)
   ( 22.0 6.0 -1.0)
   ( -1.0 6.0 -1.0)
   ( -1.0 -1.0 2.0)
   ( 22.0 -1.0 2.0)
   ( 22.0 6.0 2.0)
   ( -1.0 6.0 2.0)
);
```

```
blocks
( hex
   ( 0 1 2 3 4 5 6 7)
   ( 23 7 3) simpleGrading
   ( 1 1 1)
);
```

```
patches
( wall ffminx
  (
  ( 0 4 7 3)) wall ffmaxx
  (
  ( 1 2 6 5)) wall ffminy
  (
  ( 0 1 5 4)) wall ffmaxy
  (
  ( 3 7 6 2)) wall ffminz
  (
  ( 0 3 2 1)) wall ffmaxz
  (
  ( 4 5 6 7))
);
```

Meshing

4. Open the *snappyHexMeshDict* file. Fill the all names of boundaries in the geometry subsection.

```
geometry
{
    back.stl
    {
        type triSurfaceMesh;
        name back;
        appendRegionName false;
    }
```

5. *snappyHexMesh* can refine specifically the mesh at certain edges called "feature edges" usually defined by a specific angle. These feature edges are extracted from the geometry using the *surfaceFeatureExtract* command. Then, the call is made on the *castellatedMeshControls* to the created lines in the *triSurface* folder. One can adjust the level of refinement.
6. Add the refinement level for each boundary.
7. Locate a point inside the main body of the geometry

```
castellatedMeshControls
{
    features
    (

    {
        file "back.eMesh";
        levels ((0.0 4));
        refineFeatureEdgesOnly false;
    }
```

```
refinementSurfaces
{
    back
    {
        level ( 3 3 );
    }
```

```
locationInMesh ( 10.005000114440918 2.005000114440918 0.5 );
maxLocalCells 100000;
maxGlobalCells 2000000;
minRefinementCells 0;
nCellsBetweenLevels 1;
resolveFeatureAngle 30.0;
allowFreeStandingZoneFaces false;
planarAngle 30.0;
maxLoadUnbalance 0.1;
```

Meshing

Watermatex 2019

8. Now, let's build the mesh, in the command line type
   - *blockMesh*
   - *surfaceFeatureExtract*
   - *snappyHexMesh*
   - *checkMesh*

9. Open the *boundary* file within the created *polyMesh* folder inside the constant directory and change the name of the patch from wall to *wedge* for the front and back boundaries.

10. Open the *system/ExtrudeMeshDict* file to build the axisymmetric clarifier, select the *wedge* mode to extrude. And then type in the terminal:
    - *ExtrudeMesh*
    - *CheckMesh*

```
// If construct from patch/mesh:
sourceCase ".";
sourcePatches (back);

// If construct from patch: patch to use for back (can be same as sourcePatch)
exposedPatchName front;

//- Wedge extrusion of a single layer
//  with wedge patches on front and back
extrudeModel        wedge;
```

```
linearNormalCoeffs
{
    thickness       1;
}
```

Watermatex 2019

Processing

Solver section

driftfluxfoam

Multiphase flow in which the relative motion between the phases is governed by a particular subset of the flow parameters.

A single set of continuity and momentum equations is solved for the mixture.

Can be used for:

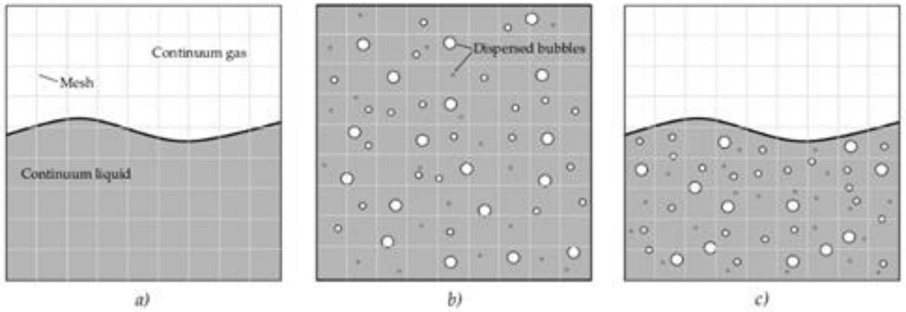
- Water-sludge modelling.
- Fluidized bed (granular solids – liquid/gas).
- Pool boiling crisis.
- Cyclone separators,
- Bubbles in heat exchangers,
- Anular flow in refineries

The mixture approach solids transport equation is solved for the volume fraction (α_d)

$$\frac{\partial \alpha_{\mathrm{d}}}{\partial t} = -\nabla\cdot\left(\alpha_{\mathrm{d}}\,\vec{v}_{\mathrm{m}}\right) - \nabla\cdot\left(\frac{\alpha_{\mathrm{d}}\rho_{\mathrm{c}}}{\rho_{\mathrm{m}}}\,\vec{v}_{\mathrm{dj}}\right) + \nabla\cdot\left(d_{\mathrm{comp}}\nabla\alpha_{\mathrm{d}}\right) + \nabla\cdot\left(\Gamma\nabla\alpha_{\mathrm{d}}\right)$$

Where:

$$\alpha_d = \frac{X}{\rho_s}$$

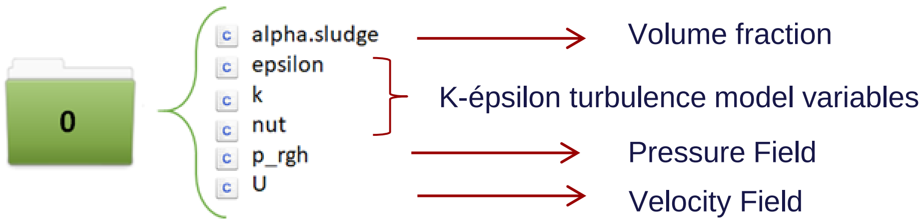


Representation of short and long geometrical scales in a bubbly flow. a) Long scale interfaces, b) short scale interfaces, c) presence of both scales simultaneously.

Pre-processing

1. Set the boundary conditions (*patch, wall, symmetry, wedge*) in the boundary file.

2. In the 0 folder set the initial conditions for the variables used during the simulations, only the boundaries set as *patch* are filled with numerical values.



Example for alpha.sludge file

```
boundaryField
{
    back
    {
        type            wedge;
    }
    wbaffled
    {
        type            zeroGradient;
    }
    wbaffleu
    {
        type            zeroGradient;
    }
    front
    {
        type            wedge;
    }
    inlet
    {
        type            fixedValue;
        value           uniform 0.00362;
    }
```

3. Set the rheology and settling velocity parameters in the *transportProperties file* located inside the constant directory

```
phases (sludge water);

sludge
{
    transportModel  BinghamPlastic;

    "(plastic|BinghamPlastic)Coeffs"
    {
        coeff       0.0016;
        exponent    79.96;

        BinghamCoeff    0.00503;
        BinghamExponent 141.3629;
        BinghamOffset   0;

        muMax       10;
    }

    rho         1050;
}

water
{
    transportModel  Newtonian;

    nu              1.7871e-06;
    rho             998;
}
```

Chose the rheology model.

Made the proper conversion, the equations are in base

Set the sludge density

Set the water density and kinematic viscosity

```
relativeVelocityModel simple;

"(simple|general)Coeffs"
{
    V0              (0 -0.006103 0);
    a               276.685866;
    a1              5217.369;
    residualAlpha   0;

}
```

Chose the settling velocity model.

Made the proper conversion, the equations are in base .

Simple stands for a Vesilind equation.

General stands for Takacs equation.

Pre-processing and Processing

Watermatex 2019

4. Finally set the time controls. In the system directory, open the *controlDict* file and introduce the conditions:

```
application     compressionDriftFluxFoam;
startFrom       latestTime;
startTime       0;
stopAt          endTime;
endTime         262800;
deltaT          0.01;
writeControl    adjustableRunTime;
writeInterval   1800;
purgeWrite      0;
writeFormat     ascii;
writePrecision  8;
writeCompression uncompressed;
timeFormat      general;
timePrecision   8;
runTimeModifiable yes;
adjustTimeStep  on;
maxCo           1;
maxDeltaT       1;
```

5. To run the case just type in the terminal
   - *driftfluxfoam*

   OpenFOAM will stop the simulation until the set time in the controlDict file

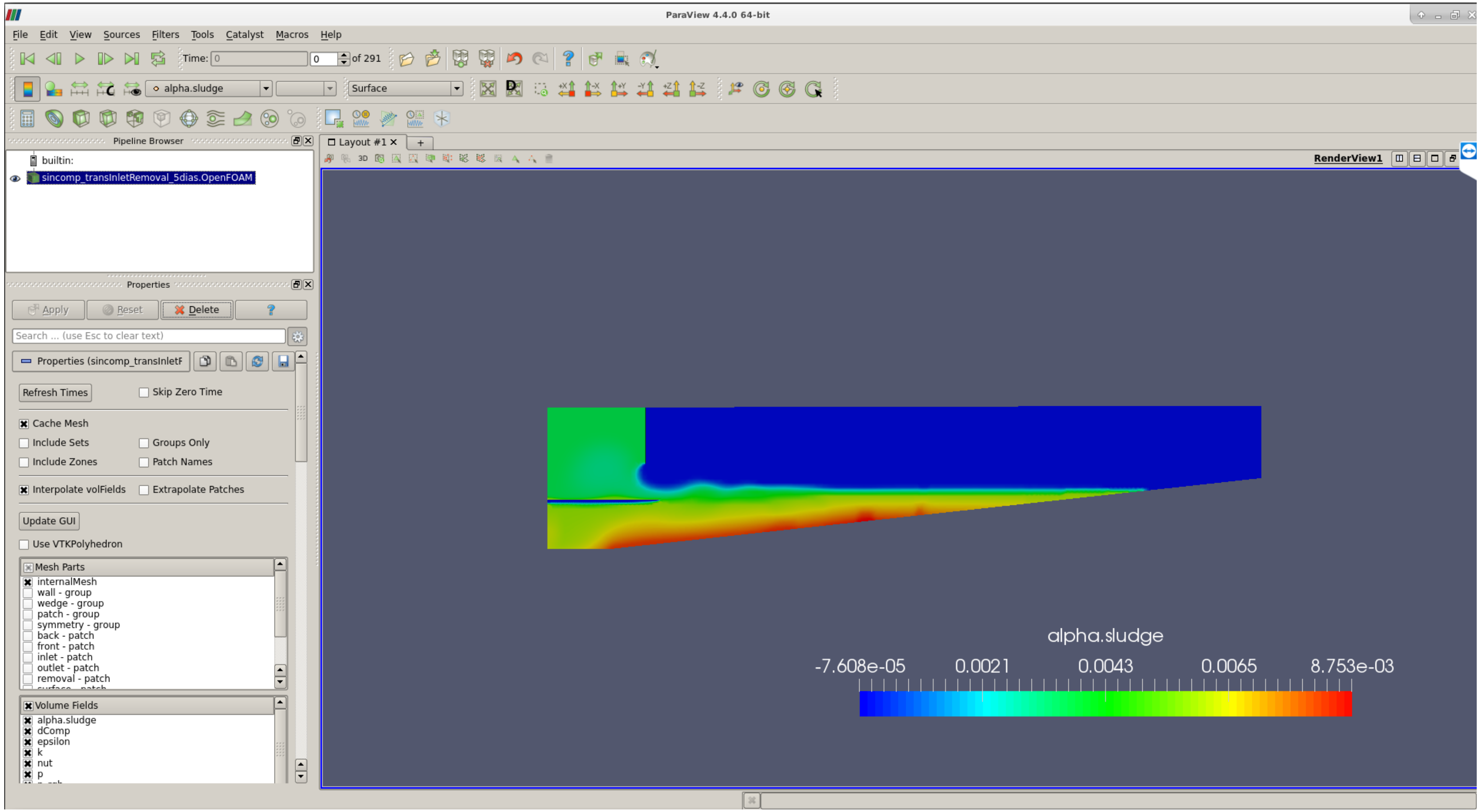
6. If working in parallel:
   - Set the number of processors in the decomposeParDict file
   - In the terminal type:
     - *decomposePar*
     - *mpirun -np n driftFluxFoam -parallel*
     - *recontructPar*

**n is the number of processors*

Post-processing

7. To visualize the data, type in the terminal:

   - *paraFoam*



Official Event of:

Organized and hosted by:

Watermatex 2019

Platinum sponsor:

Gold sponsor:

Silver sponsors:

Supported by: