GHENT UNIVERSITY

PRETREF WORKSHOP
GHENT, 16th OCTOBER 2019

Tutorial of stirring tank with viscous fluid using OpenFOAM in HPC

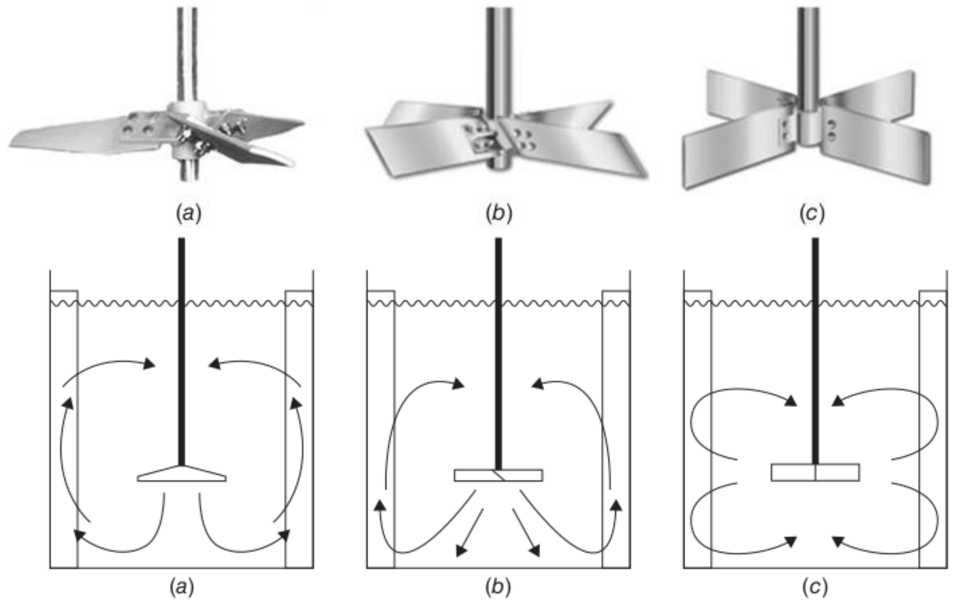
David Fernandes del Pozo (Biomath, UGent)

BIOMATH
Model-based analysis and optimisation of bioprocesses
DEPARTMENT OF MATHEMATICAL MODELLING, STATISTICS AND BIOINFORMATICS

PRETREF

Stirring tanks

- Mixing equipment for the reduction of inhomogeneity(C, phase, T, etc.) in order to achieve a desired process result
- Momentum is transferred to the fluid through the rotating force action of the blades
- Failure to provide the necessary mixing may result in severe manufacturing problems on scale-up
- Optimal mixing highly depends on the process and its desired performance



Different impellers create different hydrodynamic patterns

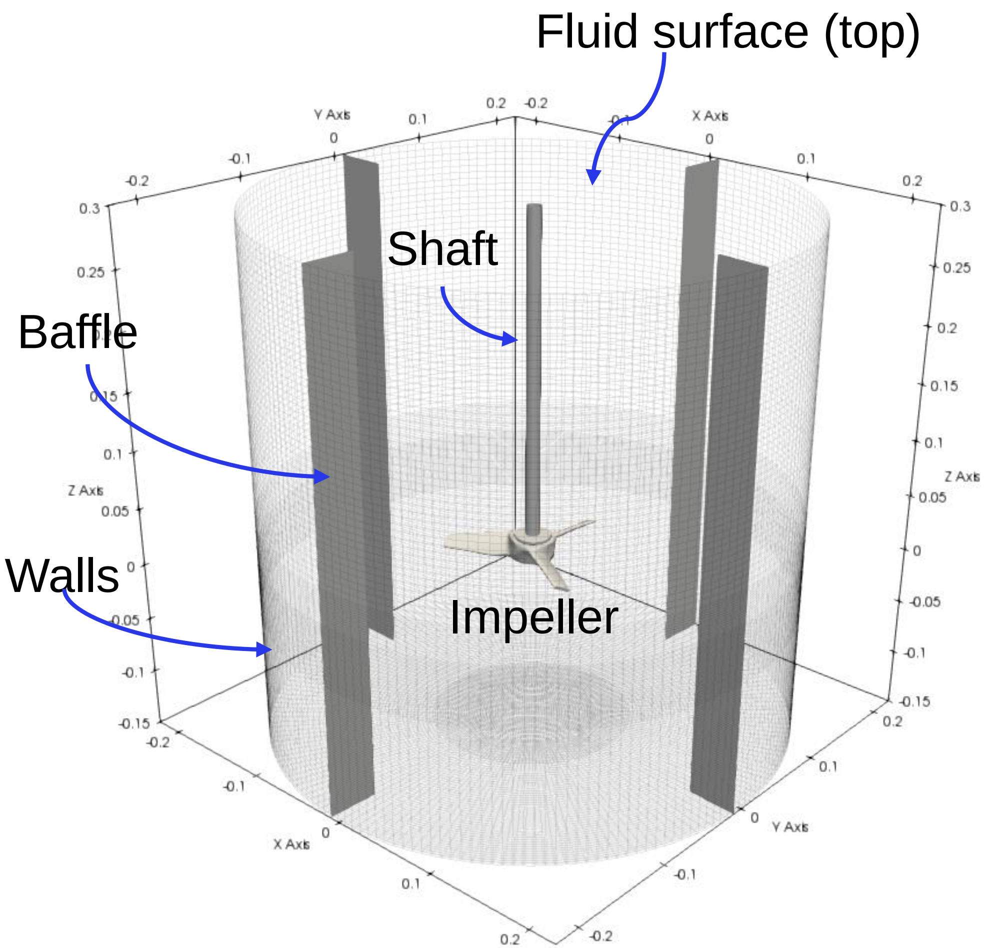
Power consumption

Lab-scale:
0.001 m^3 * 100 W/m^3 = 0.1 W

Breda's WWTP Full-scale digester:
9000m^3*2W/m^3 = 18000 W

Take-home message: Use CFD to explore different scenarios to find optimal mixing at reduced power consumption

Stirring tank case of viscous fluid



CFD model:

1) Dimension: 3D simulation

2) Phases: Single incompressible fluid

3) Turbulence modelling: No (laminar)

4) Fluid: Viscous sludge

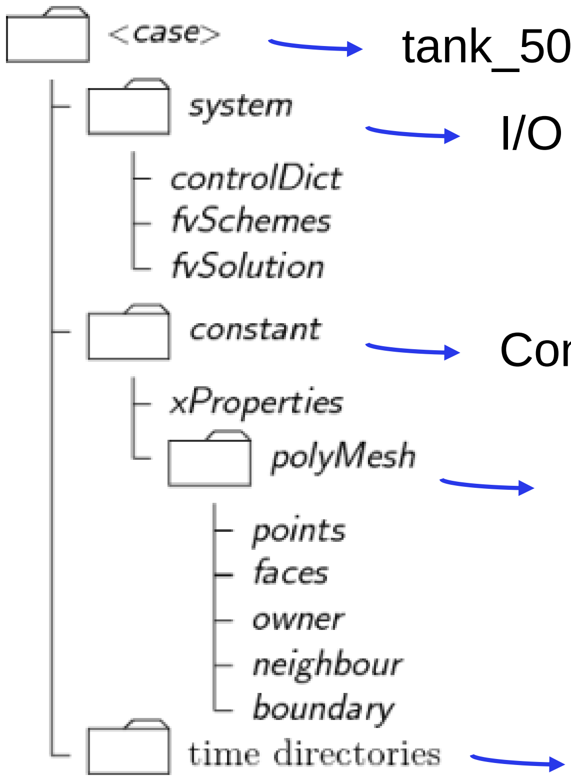
5) Fluid behaviour: Non-Newtonian (Herschel-Bulkley model)

6) Steady-state: MRF (Multiple Reference Frame)

7) Geometry: Axial impeller

Cylindrical tank with 4 baffles (V=70 L; D=0.45 m; d=0.15m)

Overview of case directory



tank_50rpm_blockMesh, tank_50rpm_snappy, tank_50rpm_fine

I/O data, Solver and FVM settings (Run time modifiable)

Constant options during run time (fluid properties)

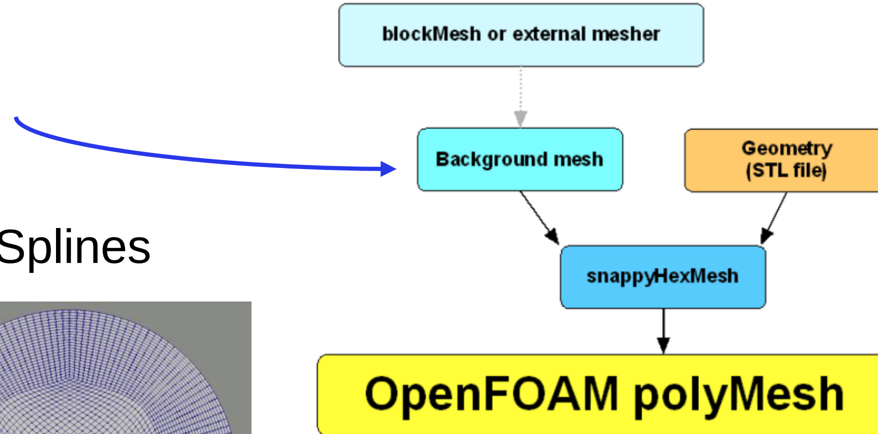
Location of mesh

Starting folder is usually 0 (contains BC)

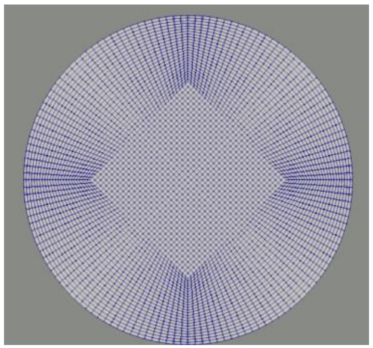
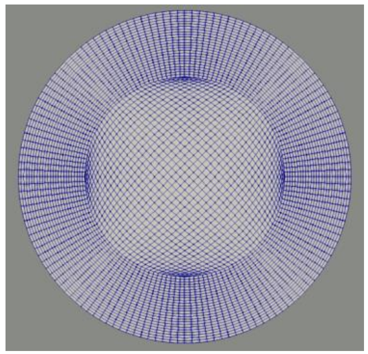
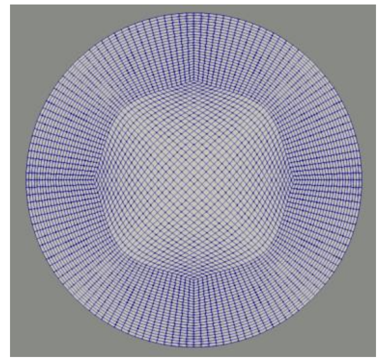
We can now submit the job to create the mesh (check path case!):

~$: *qsub* --pass=reservation=PRETREF *Allrun.pre.blockMesh*

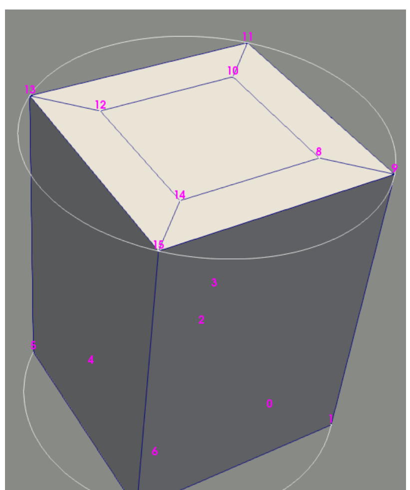
BlockMeshDict



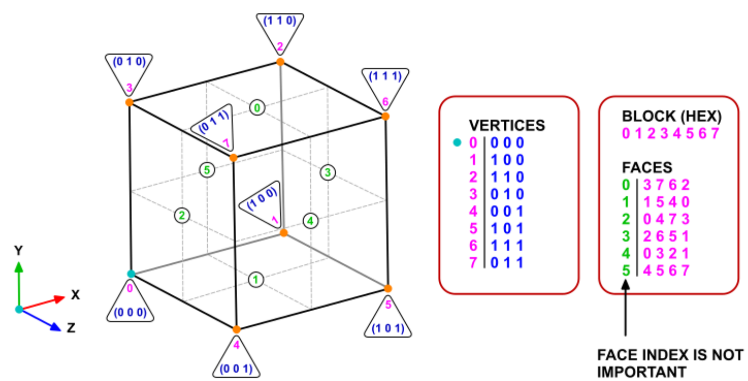
Topologies:

Square	O-grid	Splines
		

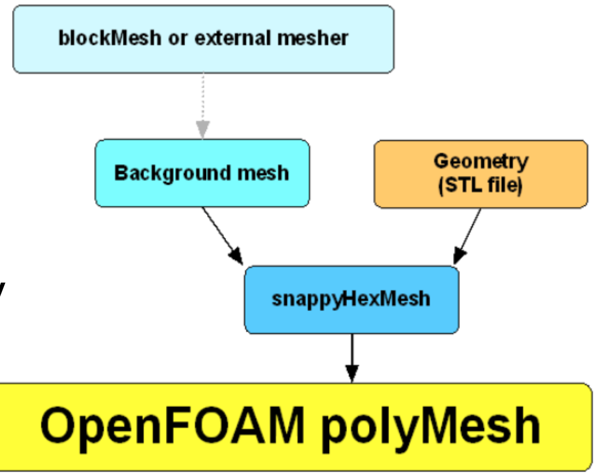
Vertices



blockMesh workflow



SnappyHexMeshDict(1)



Automatic Hex-dominant meshing OF tool

We will run *Allrun.pre.snappy* on tank_50rpm_snappy

```
// which of the steps to run
castellatedMesh true;
snap            true;
addLayers       false;

geometry
{
    A310_scaled.stl
    {
        type triSurfaceMesh;
        name Impeller;
    }
    MRF_cylinder
    {
        type            searchableCylinder;
        point1          (0 0 -0.002);
        point2          (0 0  0.0231);
        radius          0.079;
    }
  refinementcylinder
  {
        type            searchableCylinder;
        point1          (0 0 -0.15);
        point2          (0 0  0.18);
        radius          0.2;
        /*type searchableBox;
        min ( 0.0  0.0  0.0);
        max ( 0.16 0.16 0.05);*/
  }
};
```

Load of STL file

Additional simple geometries

Refinement level

```
refinementSurfaces
{
    Impeller
    {
        level (1 1);
        patchInfo
        {
            type wall;
        }
    }
    MRF_cylinder
    {
        level (1 1);
        //regions {}
        faceType internal;
        faceZone rotating; //name of faceZone
        cellZone rotating; // name of cellZone
        cellZoneInside inside;
    }
```

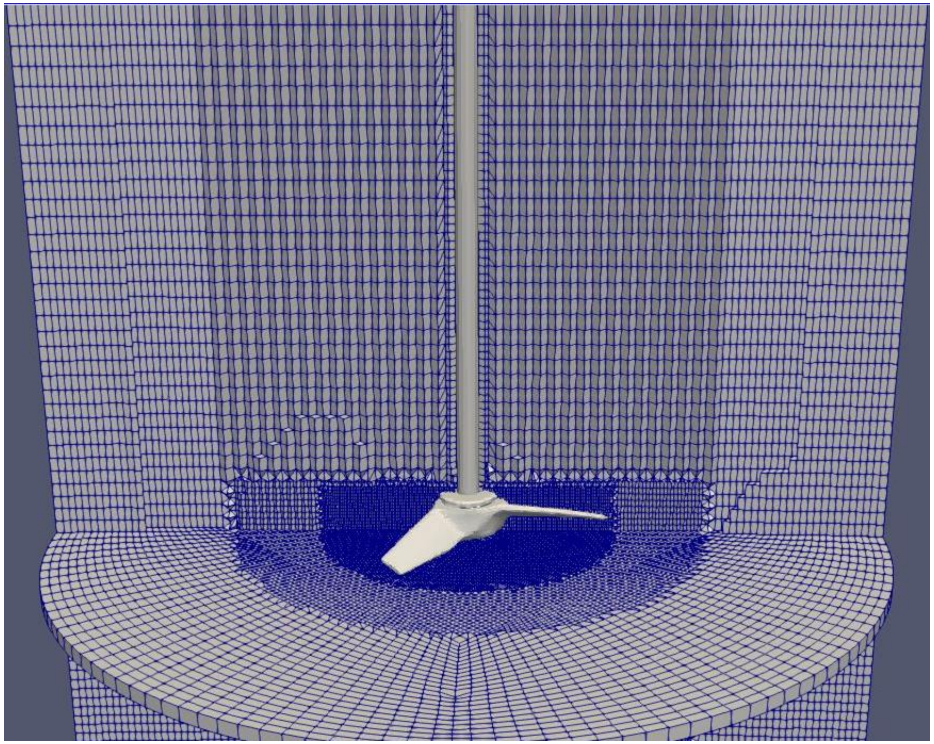
```
refinementRegions
{

    MRF_cylinder
    {
        mode inside;
        levels ((1e15 2));
    }
    refinementcylinder
    {
        mode inside;
        levels ((1e15 1));
    }
}
```

snappyHexMeshDict(2)

1) We can now visualise slices of the mesh with Paraview

~$: Module load ParaView/5.4.1-intel-2018a-mpi
~$: paraview

2) Load state in paraview to visualise the mesh
File>load state> mesh_slices.pvsm



CheckMesh (Mesh Quality)

```
Mesh stats
    points:           803805
    faces:            2330773
    internal faces:   2295313
    cells:            763907
    faces per cell:   6.0558235
    boundary patches: 10
    point zones:      0
    face zones:       1
    cell zones:       2
```

Total number of cells

Max. Non-orthogonality is especially important in OF

```
Checking geometry...
    Overall domain bounding box (-0.224998 -0.224998 -0.15) (0.224998 0.224998 0.3)
    Mesh has 3 geometric (non-empty/wedge) directions (1 1 1)
    Mesh has 3 solution (non-empty) directions (1 1 1)
    Boundary openness (6.7223673e-16 8.8341327e-17 -8.5007276e-16) OK.
    Max cell openness = 3.3217426e-16 OK.
    Max aspect ratio = 8.6711266 OK.
    Minimum face area = 8.2539221e-08. Maximum face area = 0.00010871995.  Face area magnitudes OK.
    Min volume = 4.0914001e-10. Max volume = 5.6590945e-07.  Total volume = 0.071481415.  Cell volumes OK.
    Mesh non-orthogonality Max: 65.215566 average: 7.6008523
    Non-orthogonality check OK.
    Face pyramids OK.
    Max skewness = 3.219436 OK.
    Coupled point location match (average 0) OK.
 ***Error in face tets: 144 faces with low quality or negative volume decomposition tets.
  <<Writing 130 faces with low quality or negative volume decomposition tets to set lowQualityTetFaces
    Min/max edge length = 2.0338852e-05 0.010572216 OK.
  <<Writing 2136 near (closer than 7.7941825e-07 apart) points to set nearPoints
   *There are 730 faces with concave angles between consecutive edges. Max concave angle = 79.995541 degrees.
  <<Writing 730 faces with concave angles to set concaveFaces
    Face flatness (1 = flat, 0 = butterfly) : min = 0.29997056  average = 0.99988878
   *There are 49 faces with ratio between projected and actual area < 0.8
    Minimum ratio (minimum flatness, maximum warpage) = 0.29997056
  <<Writing 49 warped faces to set warpedFaces
    Cell determinant (wellposedness) : minimum: 0.039789017 average: 8.2448685
    Cell determinant check OK.
 ***Concave cells (using face planes) found, number of cells: 9302
  <<Writing 9302 concave cells to set concaveCells
    Face interpolation weight : minimum: 0.053099091 average: 0.4925026
    Face interpolation weight check OK.
    Face volume ratio : minimum: 0.013606612 average: 0.96019292
    Face volume ratio check OK.
```

Constant folder

PolyMesh: Contains the mesh (boundary, faces, cellZones, etc.)

TriSurface: Contains geometry files (e.g. STL)

TurbulenceProperties: simulationType: laminar

TransportProperties: transportModel: Herschel-Bulkley rheological model

MRFProperties: Specify stirring MRF cellzone and stirring speed

System folder

ControlDict: Specify solver, times, monitoring functions, etc.

fvSolution: Solver settings for the different variables (tolerances, relaxation)

fvSchemes: Discretisation schemes for ALL terms (2nd order solution)

DecomposeParDict: Specify number of processors and how to distribute cells

MapFieldsDict: Specify mapping options from other meshes

Other handy dicts (monitoring funcs, mesh manipulation, etc.)

0 folder

0.org: First copy of 0 folder for safety reasons (for a clean start)

0 folder:

Boundary conditions and initialization fields for ALL variables (e.g. U)

```
rev  50; // [rev/min] but conversion to rad/s is done in 'omega'

dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (0 0 0);

boundaryField
{
    //- Set patchGroups for constraint patches
    //#includeEtc "caseDicts/setConstraintTypes"

  wall_reactor
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
  top
    {
        type            slip; //slip; "Symmetry plane" for a perfect flat surface;
                              // Use "slip" for curved surfaces too but works with flat too
    }
  bottom
    {
        type            fixedValue;
        value           uniform (0 0 0);
    }
  "Impeller" // Running MRF implies fix value, use movingWallVelocity in AMI
    {
        type            fixedValue;
        value           uniform (0 0 0);
        /*type            movingWallVelocity;
        value           uniform (0 0 0);*/
    }
```

```
"Shaft"
{
        type      rotatingWallVelocity;
  origin      (0 0 0);
  axis        (0 0 1);
  // Negative is clock-wise rotation
  omega       #calc "-$rev*2*$pi/60";// rad/s
}
// If a part of the shaft is in MRF, use this BC
"Shaft_low"
{
  type            fixedValue;
  value           uniform (0 0 0);
  /*type            movingWallVelocity;
  value           uniform (0 0 0);*/
  /*type      rotatingWallVelocity;
  origin      (0 0 0);
  axis      (0 0 1);
  omega       -20.9439510239;*/
}
//.* is the Unix wildcard as * in Windows
"baffle_.*"
{
  type            fixedValue;
  value           uniform (0 0 0);
}
  "AMI.*"
  {
      type        cyclicAMI;
  }
```

Run script & monitoring in HPC

~$: *qsub Allrun (don't do it, its already run for you ☺)*

We can monitor on-the-fly the solution using the monitoring functions defined in *controlDict* using gnuplot in HPC

Files are written in PostProcessing folder:

~$: *module load gnuplot/5.2.2-intel-2018a*

- **Pressure and velocity residuals:**

~$: *gnuplot plot_residuals*

- **Integrating torque on impeller and baffles + walls surfaces**
~$: *gnuplot plot_sum_torque*

- **Probe near the impeller**
~$: *gnuplot plot_probes*

Post-process with ParaView

1) We load the necessary module:

Module load ParaView/5.4.1-intel-2018a-mpi

(module avail paraview to check for the HPC available modules*)*

2) If you want to open the case:

paraview <case_name>.foam

(type ‘*touch <case_name>.foam*’ if it does not appear)

3) We can now generate azimuthal averages to obtain radial profiles

File>load state> contours.pvsm