DRAFT ICAM62
Close-Out Report
19th March 2021

Asiri Obeysekara

Project:	**IMPROVING MULTIPHASE CFD CAPABILITY OF BP**
Researcher:	Dr. Asiri I.B. Obeysekara[1]
Principal Investigator:	Prof. Chris Pain[1]
BP-Mentor:	Dr. Andre Nicolle[2]

[1]AMCG, Imperial College London
[2]BP

Imperial College London

AMCG

TABLE OF CONTENTS

1 INTRODUCTION

This document will summarise and highlight computational fluid-dynamics (CFD) research and engineering that spans the period between October 2018 to the current date in the context of the industrial partnership between AMCG (Imperial College London) and AMT (BP Plc) in ICAM62. This period also includes 1 year and 3 months on the MUFFINS/PREMIERE research project (UKRI-funded). The main part of this document will focus on Phase 2 work of the ICAM62 project, focussing on development and optimisation work of the research software, IC-FERST, and application to a 'Tank Mixing' model. The first part of the document will highlight work completed during the 2 years and 3 months period, alongside issues and challenges we have resolved during this time to further develop and extend the code capability for industrial and high-performance computing (HPC).

1.1 COMPLETED DELIVERABLES

I. Paper submitted to **Advances in Engineering Software (ASE):**
   Obeysekara *et al.* (2021) **Prediction of multi-phase flows using an integrated modelling approach with anisotropic mesh optimisation.**
II. Completed Phase 1 with evaluation cases 1 and 2.
III. Completed full installation and integration of code (with version control) on BP HPC
IV. Tested and optimised with up-to-date libraries (HPC driven development)
V. Knowledge transfer to other members of team (Morgan and Andre)
VI. Supporting indirect deliverables (FSI and slug-flow modelling) to BP
VII. Block matrix solve for momentum in serial (2D and 3D)

1.2 CURRENT TASKS

I. Multi-component tank simulation at different CFL
II. Multi-fluids tank simulation
III. New block matrix development in parallel (20-100% increase in speed)
IV. **(Pablo/Chris)** – P0DG for velocity (1000x increase in speed)

1.3 ISSUES RESOLVED

1.3.1 ICAM

i. HPC optimisation (fabric optimisation of InfiniBand and Omni-path libraries)
ii. Open MPI optimisation (version issues, i.e. leak in version 3.x)
iii. Intel MPI capability added
iv. Adaptivity for massively parallel (Zoltan optimisation)
v. Previous development memory leaks (surface tension)
vi. More friendly GUI – *default optimised parameters*
vii. *Pipemesh – complex pipe and pipe-network generator in Python, open-source and portable*
viii. PETSc debug capability added and tested

1.3.2 MUFFINS/PREMIERE

i. Optimisation of geometry (GMSH)
ii. Surface tension model
iii. Validation and verification
iv. Adaptivity optimisation to model complex flow
v. New external library optimisation (PETSc version upgrades)
vi. PETSc and code profiling methods added
vii. COVID-19 modelling and testing

1.4 CURRENT CHALLENGES

The validation tests of Phase 1 and the Tank Mixing simulations are non-trivial and test the capability of the high-order numerical methods and adaptive meshing technology within a massively parallel simulation.

- Modelling priorities for the framework are underpinned by methods optimising for high-accuracy and high-resolution. (addressed by P0DG methods for velocity solve)
- Accuracy demands more from PETSc and Zoltan, which has required some optimisation and development to handle this. (addressed by upgrading and optimising with external libraries)
- High-accuracy and resolution requires persistent dynamic load-balancing and adaptivity (such as in the Rayleigh Taylor 3D simulations).
- Highly anisotropic and distorted, unstructured meshes occasionally arise when modelling complex multiphase flow (this issue can be addressed by constraining mesh adaptivity parameters, [sometimes] at increased computational cost for sharp interface modelling).
- Load-balancing in a high-performance cluster using some external libraries (i.e. Zoltan ZHG partitioner) has proved to be difficult within an adaptive, dynamic partitioning context (this issue was solved by using Partmetis partitioner).

Table 1 – Timetable of main additions and deliverables met

ICAM62		MUFFINS/PREMIERE
Phase 1	Phase 2 (on-going)	ICAM relevant work
October 2018		
HPC installation	**1st Year ICAM62** 1st Jan 2021	**1st Year ICAM62** 1st October 2019
Library Optimisation		
Memory optimisation and testing		
Fabric/HPC optimisation (for skylake and Intel)		
Demonstration up to 10000 cores		
Initial jet flow simulation		
EVAL 1 – Flow past a cylinder (Re=3900)		
EVAL 2 – Pipe flow (Re=100000)		
MSc projects – Open source pipemesh_generator		Surface-tension model optimisation/integration
Validation – Flow in a pipe		Supporting research activities
O		Compressive-advection model improvement
		Complex flow modelling (free-bubbling flow in laminar and turbulent region)
		Interface-capturing with force-balancing validation (Rayleigh Taylor)
		Academic paper (submitted)
	New library (PETSc) optimisation	PETSc/Momentum solve optimisation (block matrix solve)
	Storage/Mixing tank setup	
	Tank profiling with different models	
	I. Single phase tank model	
	II. Interface Capturing tank model	
	III. Mixing tank model (from checkpoint)	
March/April 2021		

2 CRUDE TANK MODELLING

2.1 MODEL

Refineries often face situations where modelling is required to determine the blending process of crude oil in very large storage tanks. Crude storage tanks have large diameter (typically 50m), often with a floating roof, which allows the liquid height within the tank to vary by up to 15m. In a typical simulation, we will be exploring the interaction of multiple crude blends - typically where the blend properties vary in a mixed linear fashion in terms of density, viscosity, and temperature. These blends are either introduced on the inlets or initialized with stratified layers within the tank.

1. Adaptive meshing technique ideally suited to model type of problem, due to the mix of laminar and turbulent flows.
2. Buoyancy effects are important and as such force-balancing for gravity will be required. Surface tensions is likely less important but will be tested.
3. Typical diffusivity is estimated at 1.0E-7 m^2/s (values from BP).

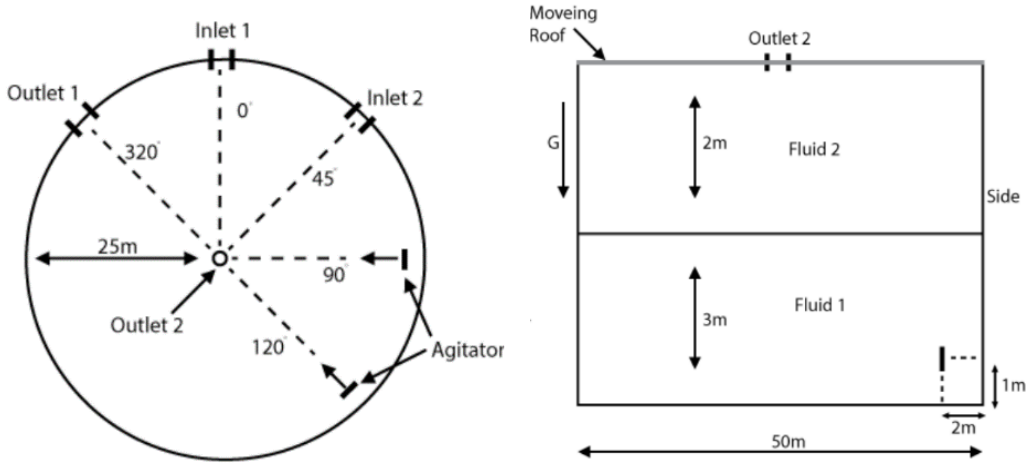


Figure 1 - Schematic of crude mixing tank model. Image from BP initial modelling document.

Operating sequence:

1. At time = 0-3 hours - velocity inlets - 1 & 2 operating, outlets – 1 & 2 operating
2. At time = 3-12 hours as above but outlet 1 flow rate = 0
3. At time = 12-14 hours Velocity inlets and outlets off
4. At time = 14 hours Tank cycling complete

2.2 POST-PROCESSING

The following diagnostics are planned for the simulation.

- Fluid and field properties across segments, lines, and points within the fluid domain.
- Outlet boundary fluid properties and component mass fractions.
- Position of the roof over time (with implementation of the FSI model in this simulation)
- Check-pointing files for IC-FERST to facilitate re-launch of simulations. (exists already)

2.3 MODELLING SEQUENCE

To simulate the tank mixing model in full, the operating sequence stage 1 (0-hours) when all inlets and outlets have initiated and operating, two modelling sequences have been designed.

The initial sequence is designed to capture the sharp interfaces of the stratified fluids (and interface), and the initial inlet jets 1 and 2. Using the compressive advection method (Pavlidis *et al.*, 2017; Obeysekara *et al.*, 2021) as an interface capturing method alongside dynamic mesh adaptivity on the fluid-mass fractions (the interface) and velocity (mesh can be seen in Figure 2 in Sequence I).

The second stage (II) of the modelling sequence is designed to allow mixing between the fluids after resolving the sharp interfaces from the previous sequence by modifying the flux limiters between fluid phases using a checkpoint between 1-5 minutes of simulation time (Figure 3). After this time, the interface and jets are well resolved and established (Figure 4 and 5). In all sequences, adaptive meshing is used at every 20 time-steps, on mass-fraction and velocity, with a

minimum and maximum element edge length constraint of 0.05 m and 1 m, respectively. These parameters have been found to create a balanced mesh on 200 cores, adequately resolve the interface and both jets from inlet 2 and 1 (Figure 4-5).

I. Initial sequence of running a multi-component interface capturing model to resolve the free-surface and jet (40 -100 seconds).

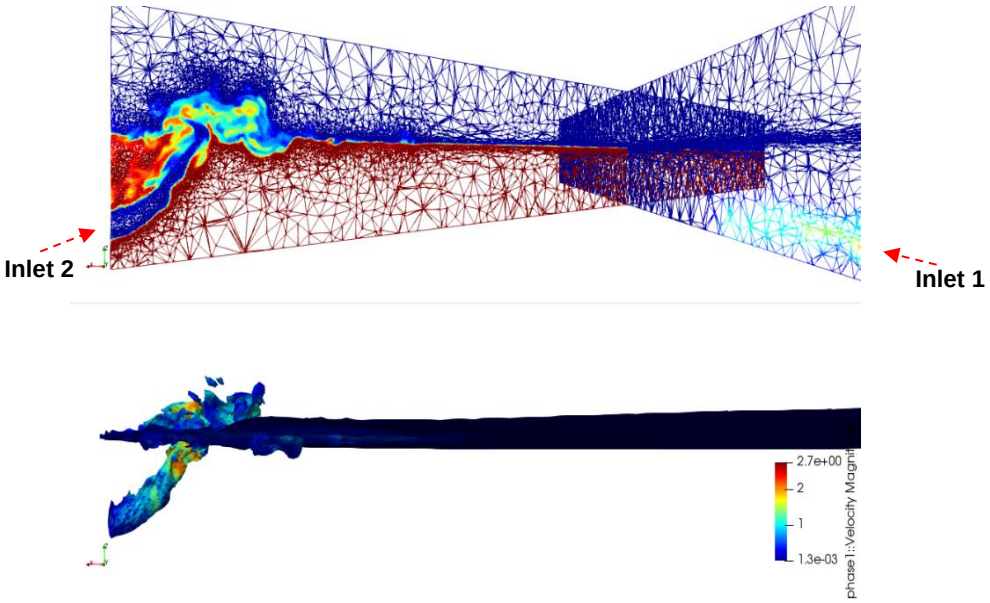


Figure 2 - Slice (top) showing the mesh at inlet 1 and 2, and contours (bottom) of the interface at start of simulation to 40 seconds

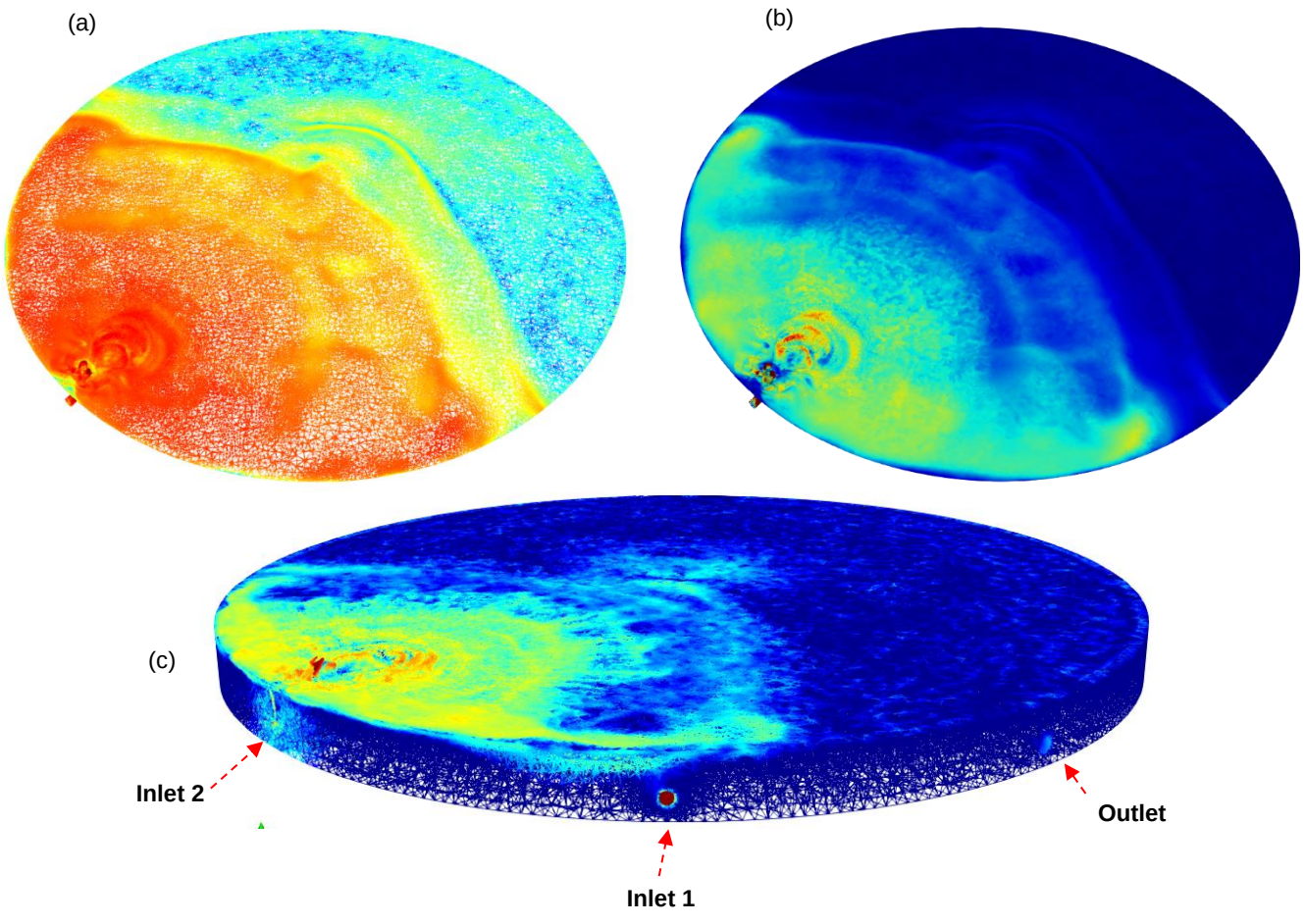


Figure 3 – Side view contour of (a) log-scale velocity profile of simulation; (b) velocity profile showing interface instability, and (c) mesh of heavier fluid and inlet/outlets at 60 seconds

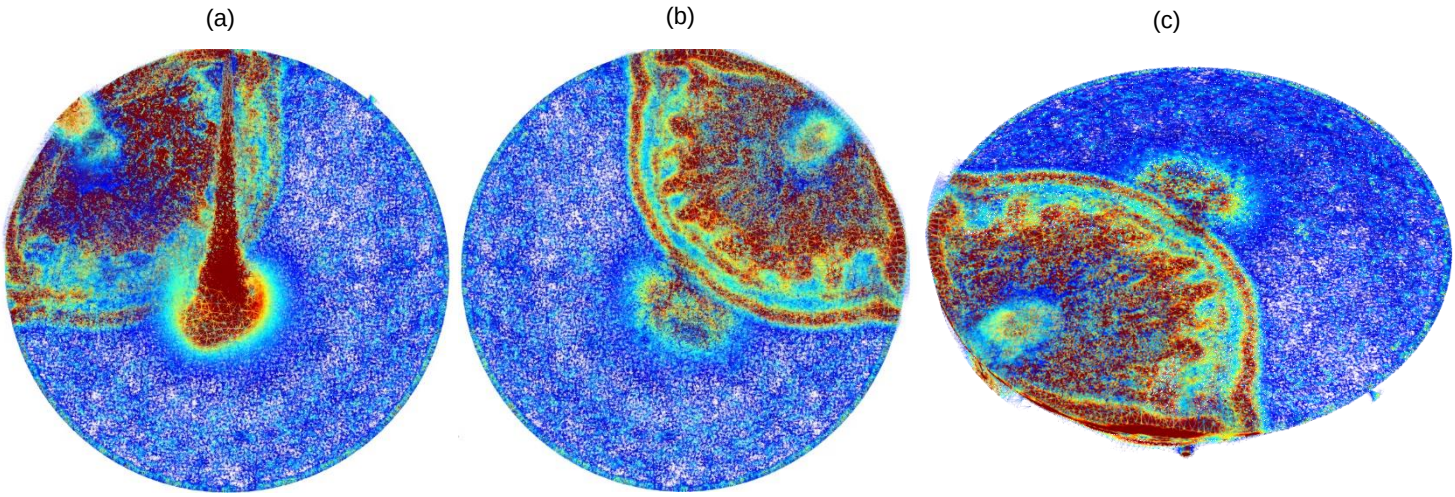


Figure 4 – Velocity scaled wireframe images showing the adaptive mesh from (a) Inlet 1 jet (heavy fluid); (b) Inlet 2 jet and interface perturbation, and (c) side profile at 60 seconds.

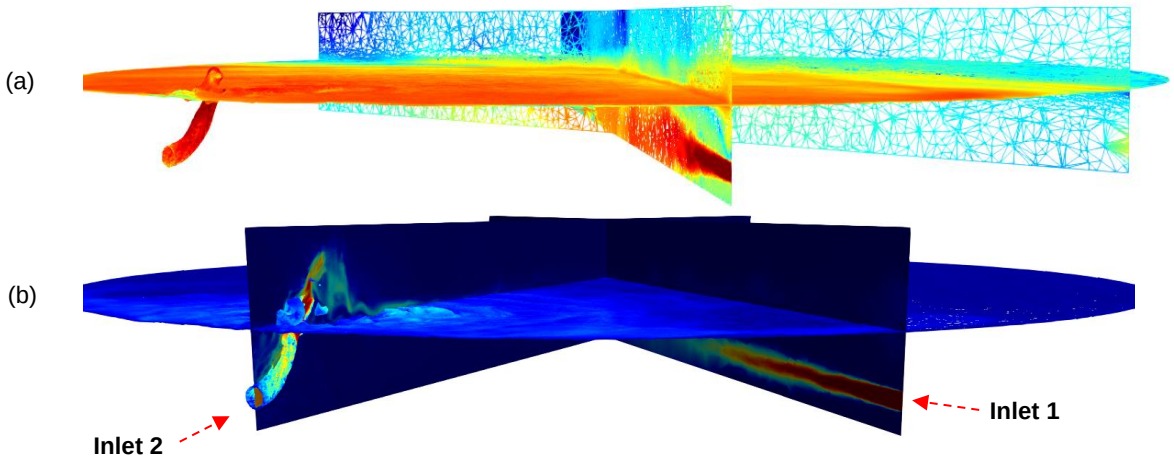


Figure 5 - (a) Velocity profiles of the inlet jets log-scale; (b) relative velocity

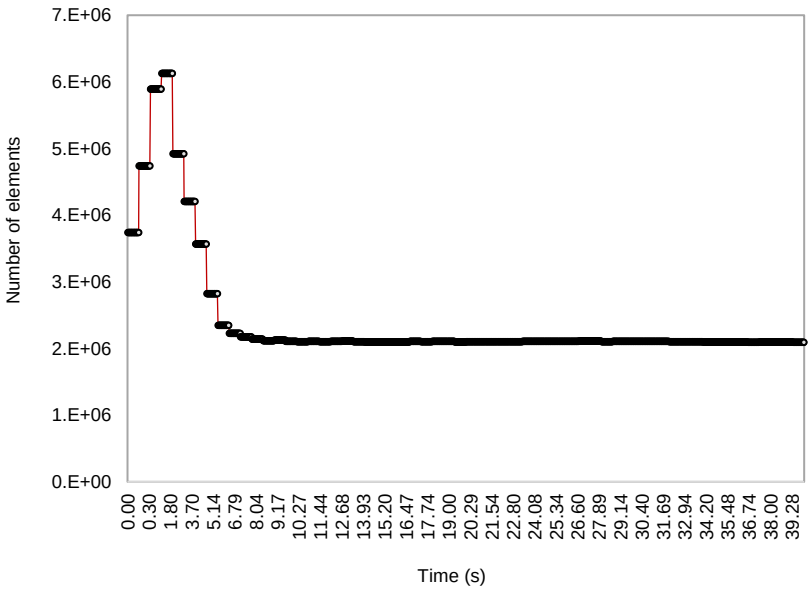
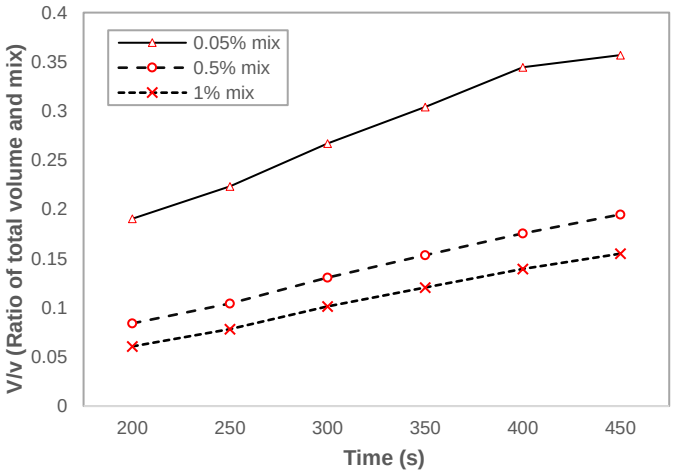


Figure 6 - Dynamic variation of size of mesh (and degrees of freedom) for first 40 seconds of the simulation

II. 2nd stage of running a checkpoint with NVD-limiter (with θ=-0.5) to model mixing for rest of simulation until stage 4 of the sequence.

(a)



(b)

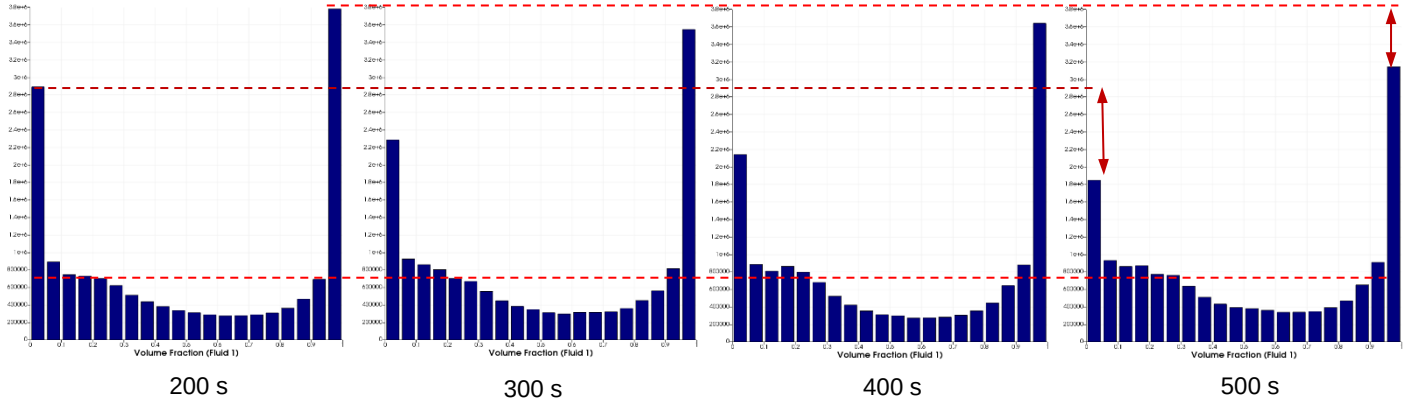


Figure 7 – Plot of (a) volume ratio (volume of mix normalised with volume of domain) of different mix % vs time during the 2nd stage of modelling sequence and (b) histogram of mass fractions of fluid 1 in the tank from 200 s to 500 s of the simulation showing the change in unmixed and mixed fluid (red arrow)

(a)

(b)

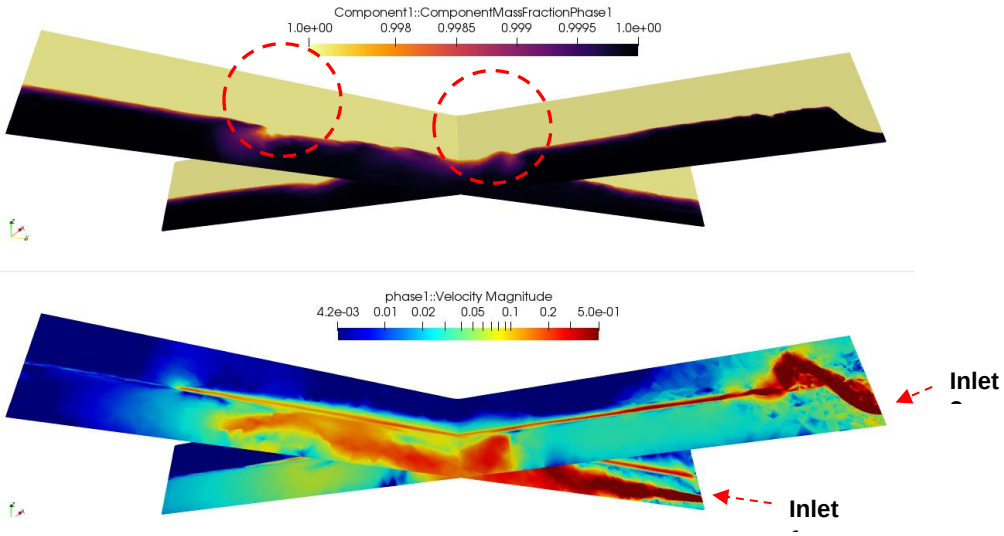


Figure 8 - Slices from inlet 1 and inlet 2 showing (a) mass fraction and (b) velocity at 250 seconds.

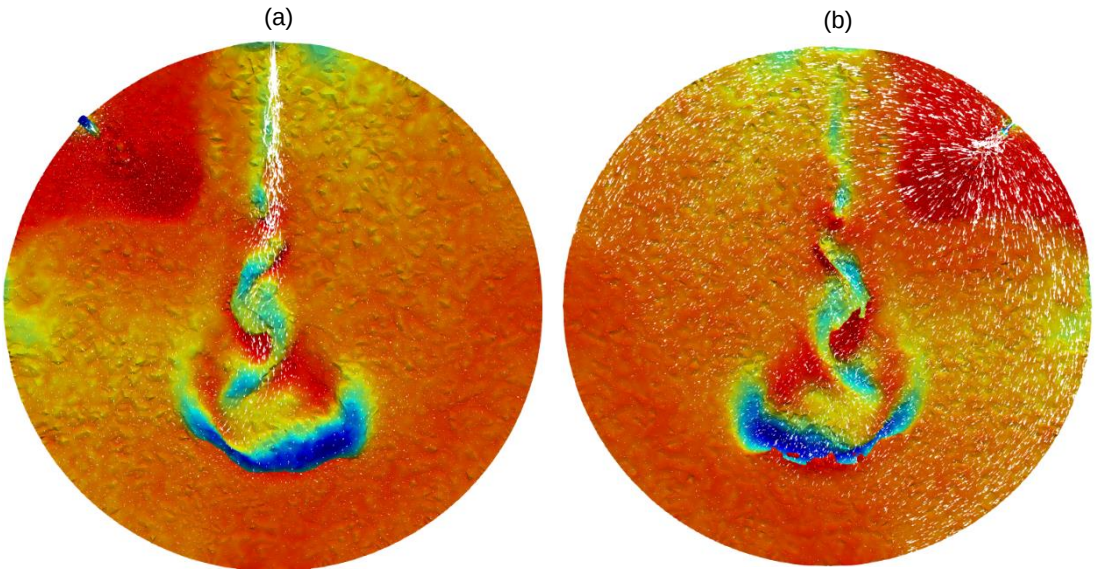


Figure 9 - Interface perturbation as, here showing the jet from inlet 1 and velocity vector glyphs as a view from (a) underneath the domain (-z-direction) and (b) from the top of the domain (+z-direction)

2.4 HPC INFORMATION

- Tested up to 2000 cores.
- Current model is optimised to run with 200 cores, this is mainly due to ideal load-balancing the degrees of freedoms of the partitions/cores.
- Typical simulation times (at 200 cores) using the P1DG method for velocity are:
    - <6 days for 40 seconds at high-resolution CFL1 setting.
    - <2 days for 40 seconds at CFL5 setting.
    - <1 day for 40 seconds at CFL10 setting.

2.5 P0DG FOR VELOCITY

The **P0**DG method for velocity has been developed recently by Pablo and Chris to improve performance, namely related to speed, of simulations in IC-FERST. Initial testing by Pablo for 2D and air-flow modelling has shown speed-ups of up to 10x the P1DG method (default). Applied to the multi-regime, simulating large mesh tank mixing model for 40 seconds, the following performance improvements were observed:

- **Without interface capturing scheme/limiters** - speed increase: 2x
- **With interface capturing scheme/limiters** - speed increase: 2.5x

Comparing the velocity and flow profile (see Figure 10), we can see a close match between results from the new P0DG method and the default method using P1DG at 40 seconds of the tank mixing simulation. Profiling the simulation using PETSc profiling tools, the improvements in the computational time and floating-point operations (FLOPS) is seen largely in solving for velocity (a reduction of 87.7% to 67% of the the total computational time speant at this stage, see Figure 11).

Using the Jacobi solver for velocity, P1DG was ~20x and ~3x more expensive (in terms of time) for the matrix multiplication (MatMul) and linear solve (KSPsolve) functions using PETSc.

Simulation options:
P0 flux limiter
Pressure solver options: HYPRE BoomerAMG solver
Velocity solver options:

- Jacobi with rowsum and absolute values **(2.5x faster)**
- HYPRE BoomerAMG **(1.5x faster)**

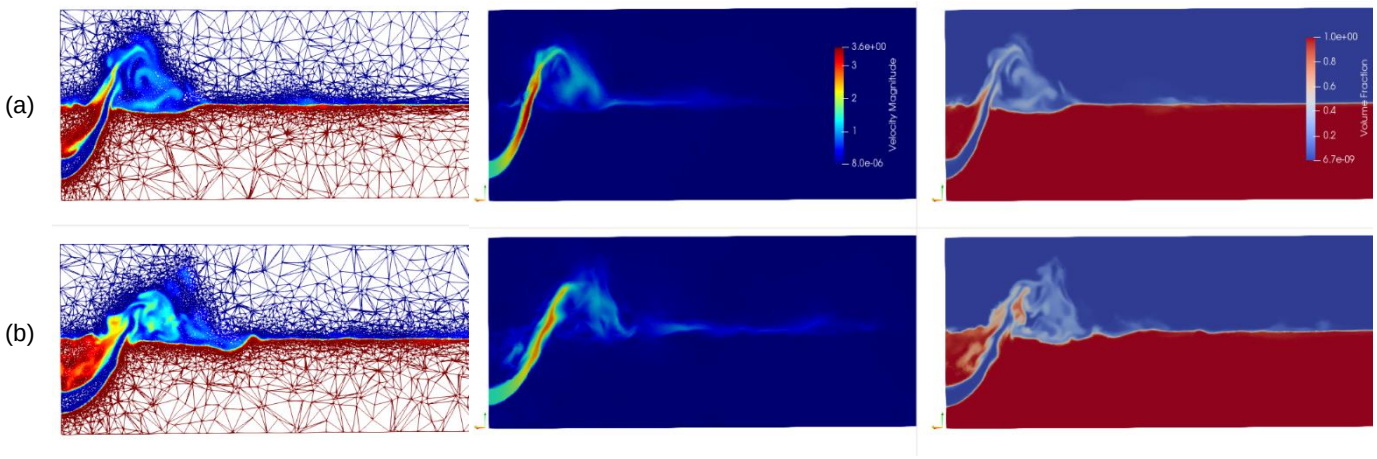


Figure 10 – Qualitative comparison of (left)mesh, (centre) velocity profile and (right) mass fraction of inlet 2 (buoyancy dominated jet) between (a) P0DG method and (b) P1DG method for velocity.

(a)

Summary of Stages:	----- Time ------		----- Flop ------	
	Avg	%Total	Avg	%Total
0: Main Stage:	2.0057e+00	0.0%	8.0484e+06	0.0%
1: WRAP:	2.2719e+02	0.4%	1.4670e+08	0.0%
2: PRELIM:	2.1865e-02	0.0%	0.0000e+00	0.0%
3: FORCE:	4.3630e+04	67.2%	7.7016e+13	90.6%
4: SATURATION:	9.3647e-02	0.0%	0.0000e+00	0.0%
5: TEMP:	3.5482e+01	0.1%	0.0000e+00	0.0%
6: COMP:	1.1148e+04	17.2%	5.2434e+12	6.2%
7: DT+VTU:	2.0157e+03	3.1%	8.1047e+11	1.0%
8: ADAPT:	7.6980e+03	11.9%	1.9833e+12	2.3%
9: REST:	1.2181e+02	0.2%	0.0000e+00	0.0%

(b)

Summary of Stages:	----- Time ------		----- Flop ------	
	Avg	%Total	Avg	%Total
0: Main Stage:	1.9100e+00	0.0%	8.0484e+06	0.0%
1: WRAP:	2.5369e+02	0.2%	1.4670e+08	0.0%
2: PRELIM:	2.8538e-02	0.0%	0.0000e+00	0.0%
3: FORCE:	1.3071e+05	87.7%	1.2442e+15	99.3%
4: SATURATION:	1.3538e-01	0.0%	0.0000e+00	0.0%
5: TEMP:	4.6916e+01	0.0%	0.0000e+00	0.0%
6: COMP:	8.5532e+03	5.7%	6.1546e+12	0.5%
7: DT+VTU:	2.2611e+03	1.5%	9.7729e+11	0.1%
8: ADAPT:	7.0522e+03	4.7%	2.1789e+12	0.2%
9: REST:	1.4186e+02	0.1%	0.0000e+00	0.0%

Figure 11 - Profiling of (a) P0DG method and (b) P1DG method for velocity show a big reduction in the computational overhead of velocity solve

3 GENERAL OIL & GAS MODELLINGS

This section will highlight some of the oil and gas modelling specific outputs and developments that are a result of collaborations and further research that was undertaken during the period.

I. MUFFINs and PREMIERE project – Modelling capability to capture complex flow such as slug-flow (e.g. surface tension dominated flows) has been compared with non-adaptive mesh simulations (Figure 12) and validated (Figure 13) using IC-FERST, before being used to model turbulent free bubbling flow (Figure 14).

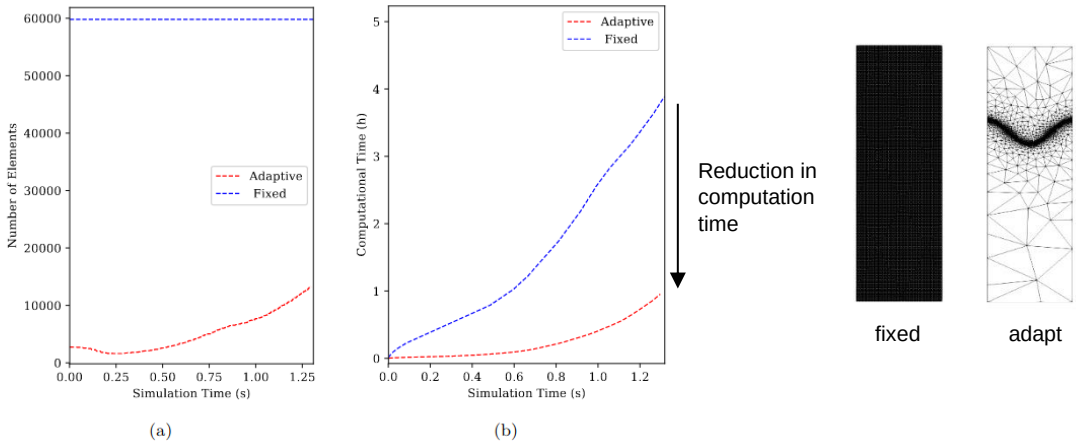


Figure 12 - Comparison of total number of elements (a) and computational time (b) between adaptive (red) and fixed(blue) mesh simulations with equivalent minimum element edge-length resolution e^{min} of 0.01

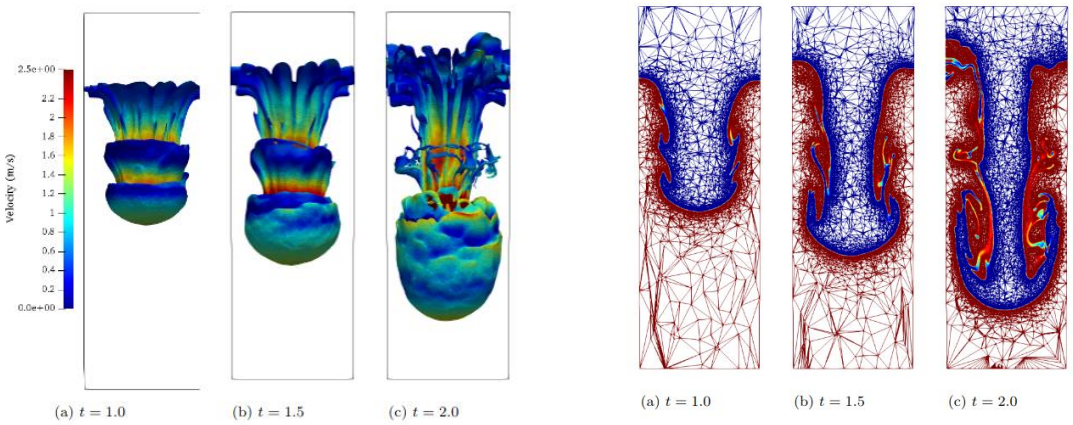


Figure 14 - 2D slice (zx-plane) of the Rayleigh-Taylor simulation showing the mesh at different times, with low-density and high-density phase volume fractions as red and blue

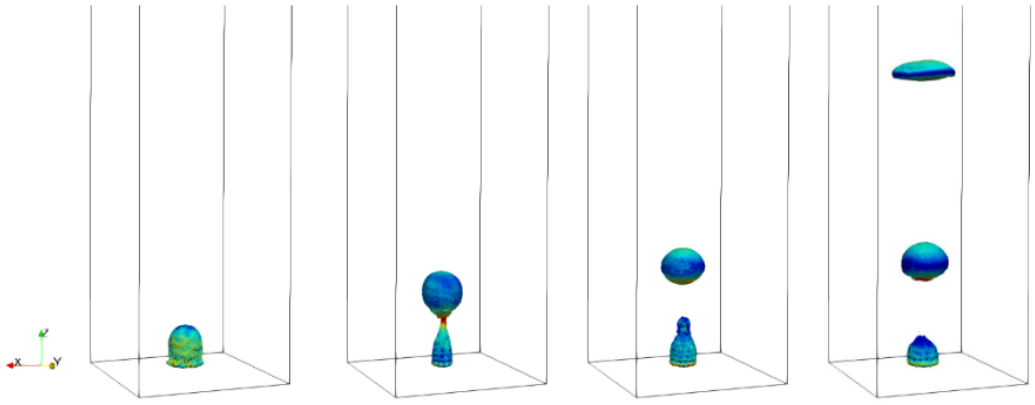


Figure 13 Snapshots of four different simulation stages showing contours of free-bubbling air phase in a vertical pipe, zoomed into the initial bubble generation region

II. Open-source pipe-network builder (MSc project)
   - Highlighted in previous report (attached to this document)

III. Other modelling problems:
   - **Flexible risers** (combined with FSI)
   - **Pipework Vibration** (combined with FSI)
     The project looked at forces of a slug, generated down a long 6 inch pipe on a 90° bend - experimentally and numerically. This is a multiphase project with a high density ratio between the gas and liquid flow medium.
   - **Numerical Wave Tank**
     It is hoped that the substantial scale challenges within a numerical wave tank can be better addressed with an adaptive meshing technique. This can better refine sharp interfaces in the free surface region, as well as capture important flow features in critical areas. We would propose using real scale ship performance metrics, generated by Lloyds Register.

4 CODE STATUS (IC-FERST INERTIA)

4.1 GENERAL STATUS

Main general developers: Chris Pain, Pablo Salinas
Other developers: Jianseng Xiang (FSI)
Student developers: Linfeng Li, Morgan Kerhouant, Amin Nadimy
Version control is handled through GITHUB, with every branch tested regularly.

4.2 MEMORY

4.2.1 MEMORY OVERHEAD

Early in the project we identified and reduced the memory overhead (non-leaking memory) of the modelling framework by running several profiling simulations that overloaded the memory available per node. This has included modifying how fields are allocated between donor target fields (from straight assignment to Do Loops), a Fortran issue that assigns additional memory for the allocation.

4.2.2 LEAKS

Initial memory leaks were identified in Phase 1 with external libraries (Open MPI 3.x) and surface tension model. These were fixed and the code has largely performed as expected with most simulations. Recently, further leaks have been identified that seem to be model specific (i.e. they are noticeable for adaptive meshing slug flow simulations, but are not present in the tank mixing simulation with the same numerical algorithms). After an extensive Valgrind profiler sprint combined with Morgan and the FSI development team (J Xiang and L Li), we identified the following issues:

- **VTK interface** (unknown cause, seems external)
- **HYPRE** (unknown cause, seems external)
- **PETSc version** (<3.14, only noticed by FSI team, fixed by upgrading to newest version)
- **Diagnostic tools** (some issues noticed, does not seem to affect performancea).
- **State/Memory allocation** (internal and known issue, suspected compiler-based issue)
- **Adaptive sweeps** (issue resolved by removing the option)

4.3 PROFILING

4.3.1 STORAGE TANKS

- Low overhead of adaptivity routine in the modelling framework.
- The PETSc solve itself is about ~10% of the total simulation for momentum.
- Valgrind/Callgrind (calls and memory) points to a cost of 5% for Momentum_solve by KSPSolve and 10-12% for Pressure _solve.

(a) CFL1

Summary of Stages: ----- Time ------ ----- Flop ------

		Avg	%Total	Avg	%Total
0:	Main Stage:	2.3431e+00	0.0%	8.0484e+06	0.0%
1:	WRAP:	2.6190e+02	0.0%	1.4670e+08	0.0%
2:	PRELIM:	2.8434e-02	0.0%	0.0000e+00	0.0%
3:	FORCE:	4.8994e+05	90.4%	1.4517e+15	98.3%
4:	SATURATION:	5.5815e-01	0.0%	0.0000e+00	0.0%
5:	TEMP:	1.5898e+02	0.0%	0.0000e+00	0.0%
6:	COMP:	2.7292e+04	5.0%	2.0127e+13	1.4%
7:	DT+VTU:	6.0495e+03	1.1%	2.9165e+12	0.2%
8:	ADAPT:	1.7760e+04	3.3%	2.6007e+12	0.2%
9:	REST:	5.3301e+02	0.1%	0.0000e+00	0.0%

(b) CFL5

Summary of Stages: ----- Time ------ ----- Flop ------

		Avg	%Total	Avg	%Total
0:	Main Stage:	2.5522e+00	0.0%	8.0484e+06	0.0%
1:	WRAP:	2.5194e+02	0.2%	1.4670e+08	0.0%
2:	PRELIM:	2.8439e-02	0.0%	0.0000e+00	0.0%
3:	FORCE:	1.2299e+05	88.0%	1.1716e+15	99.3%
4:	SATURATION:	1.2522e-01	0.0%	0.0000e+00	0.0%
5:	TEMP:	4.2505e+01	0.0%	0.0000e+00	0.0%
6:	COMP:	7.7133e+03	5.5%	5.4864e+12	0.5%
7:	DT+VTU:	2.1443e+03	1.5%	8.7294e+11	0.1%
8:	ADAPT:	6.5169e+03	4.7%	1.9054e+12	0.2%
9:	REST:	1.3441e+02	0.1%	0.0000e+00	0.0%

4.4 MESHING

Several meshing issues had arisen in the initial testing of the Tank Mixing simulation which were quickly resolved by re-drawing the inlet geometries and increasing the number of iterations of adaptivity per adaptive time-step. These examples are rare and have not presented issues in most other simulations. The three examples of these issues encountered at the initial setup/pre-processing stage of the Tank Mixing model geometry are presented below in Figure 16-17.

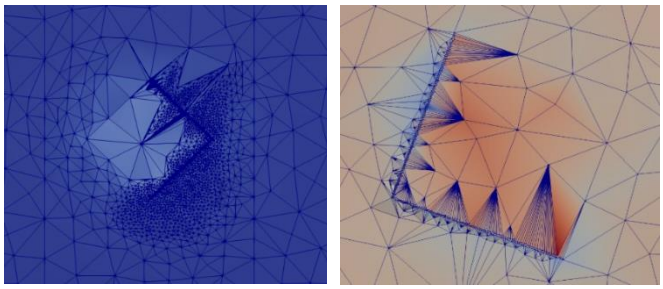


Figure 15 - High-aspect ratio surface mesh elements

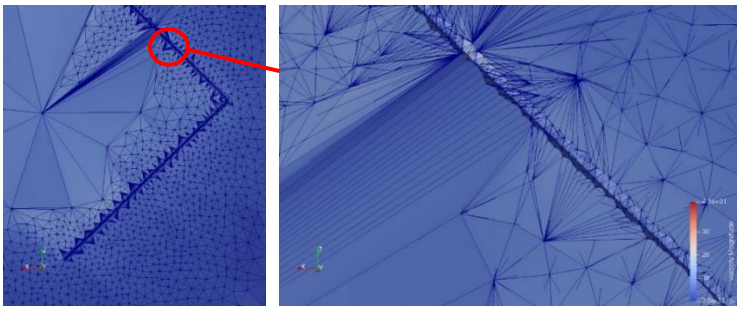


Figure 16 - Small element size on surface lines (resolve using different geometry for inlet)

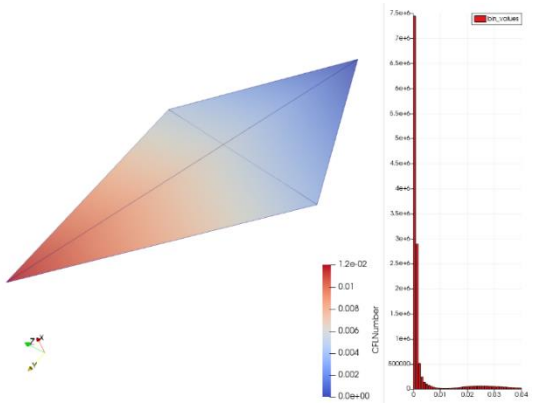


Figure 17 - Small volume elements (resolved using more adaptivity iterations

Re-drawing the geometry at the inlet, the meshing issues were resolved for the adapt at first timestep (see Figure 18a) and the subsequently adapts at the mesh inlet (see Figure 18b). No other meshing issues were encountered using a new mesh diagnostic method, which includes limiting the gradation, aspect ratio of the adaptive mesh criteria and increasing the iterations of the adaptive routine.

(a) (b)

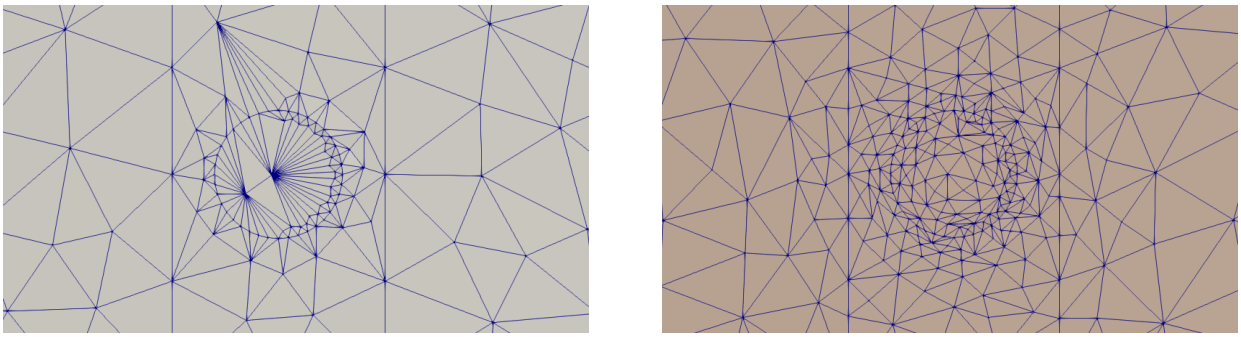


Figure 18 - Mesh near the inlet for (a) the first time-step and (b) at 200 seconds

5 ACCUMULATED KNOWLEDGE

5.1 HPC OPTIMISATION

- **Version control** through GITHUB is given by the group lead developers (myself and Pablo are admins at the moment). This will be revoked when I leave. However, other users (Andre and Morgan) also have normal access.
- **Installation**: Optimum installation settings for IC-FERST on the HPC are controlled by the "userbasrc" file, which is summarised in the Appendix.
- Specific **run-time** optimisations are unnecessary for the CFD itself, but the SGE script is optimised for OpenMPI to work with specific fabric options in Ivy-Bridge as below (OpenMPI 4.x Bash version). UCX/OpenIB is the specific fabric libraries that are used by OpenMPI and available in Ivybridge (in red).

```
#!/bin/bash

#$ -S /bin/bash
#$ -j y
#$ -cwd
#$ -N <name>
#$ -P amt-ext
#$ -pe ivybridge-ib-ctrl-slaves <ncores>

cd <path to simulation>

# Create the mpi hosts file
export MACHINES=hostfile
cat $PE_HOSTFILE | awk '{printf("%s slots=%s\n",$1,$2)}' >$MACHINES

export OMPI_MCA_btl_openib_allow_ib=1
export OMPI_MCA_btl_openib_if_include="mlx4_0:1"

mpirun --mca pml ucx -np <ncores> -machinefile hostfile <path to code> <simulation> >> <log>
```

- Node to be used with OpenMPI for are:
  - ivybridge-ib-ctrl-slaves
  - cascadelake-high-ipc-ctrl-slaves

5.2 JACOBI SOLVER FOR VELOCITY (STABILITY AND SPEED)

For diagonal dominant systems such Momentum matrix in our turbulent flow simulations, a Jacobi solve has been shown to provide better stability at runtime. This is specified as a "Linear_Solver" option for Velocity in the Diamond interface; the default is the *rowmax* operation.

5.3 PETSC

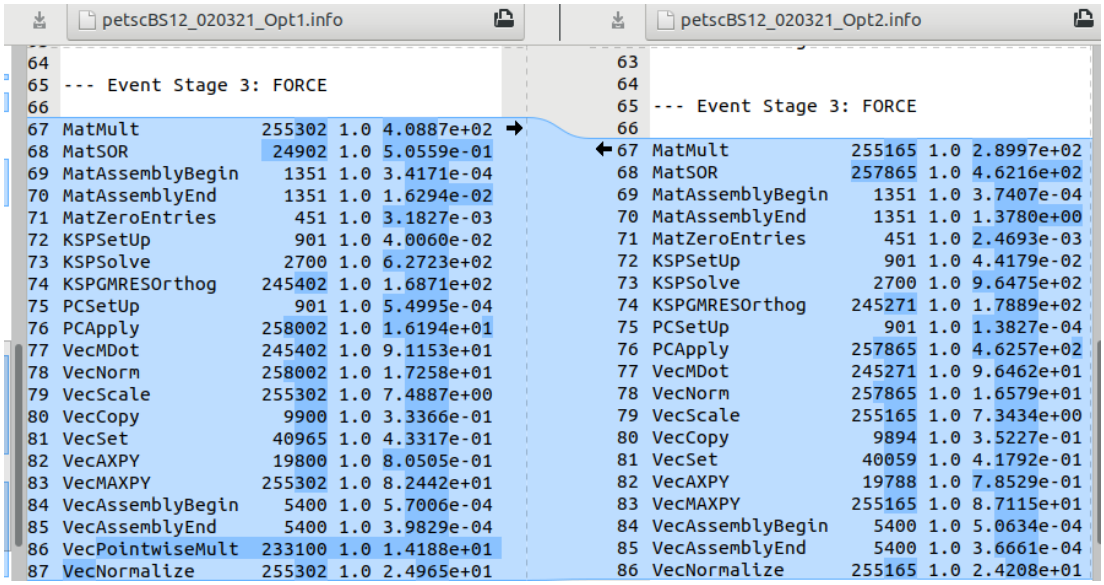
- PETSc version 3.14.x has been shown to be better at dealing with memory issues generally (validated by myself, Linfeng and Morgan). This can be a result of an external library such as HYPRE rather than PETSc itself.
- Current MASTER branch of IC-FERST repository supports 3.14.x and older dependent on the system (i.e. backward compatibility is ensured).
- Shell scripts to build new PETSc and Zoltan for a user/developer (as an Environment Module) have been written and distributed for testing.

5.4 BLOCK AIJ MATRIX SOLVE (10-100% FASTER)

BRANCH:

- **2D status:**
  - 6x6 block size is working fully in serial with current stable version of petsc (3.14.x) and is optimised
- **3D status:**
  - 12x12 block size is working fully in serial but is not optimised with current working stable version (3.14.x).
  - Recently, 12x12 block size has been optimised (MatMul) from Petsc in a development version of PETSc by Barry Smith (https://gitlab.com/petsc/petsc/-/issues/825 ; branch: barry/2021-02-12/add-baij-12)
  - PETSc options available are: -mat_baij_mult_version <0,1,2,3> set in Fluiditiy/IC-FERST using the environment variable "PETSC_OPTIONS" (i.e. export PETSC_OPTIONS='-mat_baij_mult_version 1')
  - Reported profiling by PETSc devs are ~20%
  - Initial profiling shows that 3<2<1, but more investigation is needed.

- Option 2 and 3 are slower due to PCApply and MatAssembly taking more time but is faster for the KSPSolve and MatMul.



petscBS12_020321_Opt1.info

```
64
65 --- Event Stage 3: FORCE
66
67 MatMult              255302 1.0 4.0887e+02
68 MatSOR               24902 1.0 5.0559e-01
69 MatAssemblyBegin      1351 1.0 3.4171e-04
70 MatAssemblyEnd        1351 1.0 1.6294e-02
71 MatZeroEntries         451 1.0 3.1827e-03
72 KSPSetUp               901 1.0 4.0060e-02
73 KSPSolve              2700 1.0 6.2723e+02
74 KSPGMRESOrthog      245402 1.0 1.6871e+02
75 PCSetUp               901 1.0 5.4995e-04
76 PCApply             258002 1.0 1.6194e+01
77 VecMDot             245402 1.0 9.1153e+01
78 VecNorm             258002 1.0 1.7258e+01
79 VecScale            255302 1.0 7.4887e+00
80 VecCopy               9900 1.0 3.3366e-01
81 VecSet               40965 1.0 4.3317e-01
82 VecAXPY              19800 1.0 8.0505e-01
83 VecMAXPY            255302 1.0 8.2442e+01
84 VecAssemblyBegin      5400 1.0 5.7006e-04
85 VecAssemblyEnd        5400 1.0 3.9829e-04
86 VecPointwiseMult    233100 1.0 1.4188e+01
87 VecNormalize        255302 1.0 2.4965e+01
```

petscBS12_020321_Opt2.info

```
63
64
65 --- Event Stage 3: FORCE
66
67 MatMult              255165 1.0 2.8997e+02
68 MatSOR               257865 1.0 4.6216e+02
69 MatAssemblyBegin      1351 1.0 3.7407e-04
70 MatAssemblyEnd        1351 1.0 1.3780e+00
71 MatZeroEntries         451 1.0 2.4693e-03
72 KSPSetUp               901 1.0 4.4179e-02
73 KSPSolve              2700 1.0 9.6475e+02
74 KSPGMRESOrthog      245271 1.0 1.7889e+02
75 PCSetUp               901 1.0 1.3827e-04
76 PCApply             257865 1.0 4.6257e+02
77 VecMDot             245271 1.0 9.6462e+01
78 VecNorm             257865 1.0 1.6579e+01
79 VecScale            255165 1.0 7.3434e+00
80 VecCopy               9894 1.0 3.5227e-01
81 VecSet               40059 1.0 4.1792e-01
82 VecAXPY              19788 1.0 7.8529e-01
83 VecMAXPY            255165 1.0 8.7115e+01
84 VecAssemblyBegin      5400 1.0 5.0634e-04
85 VecAssemblyEnd        5400 1.0 3.6661e-04
86 VecNormalize        255165 1.0 2.4208e+01
```

Figure 19 - Comparison of profiling results between Option 1 and Option 2 of new development, to better utilise matrix multiplication operations

5.5 ZOLTAN VS PARMETIS

Common usage and multiple users have now reported that Parmetis is much better at load balancing partitions with very large simulations on the HPC, whereas Zoltan can produce empty partitions while load balancing. Therefore Parmetis is the partitioner of choice for massively parallel simulations, whereas on small simulations, Zoltan's ZHG is used.

6 APPENDIX

6.1 BP-CHPC DIRECTORIES AND ENVIRONMENT

SOFTWARE

>> /hpcdata/obatj8/software

- Code location for Intel/gcc and OpenMPI/gcc
- Installation scripts
- Libraries (including PETSc and Zoltan)
- Run scripts and templates

SIMULATIONS

>> /lustre05/vol0/obatj8/

- Full simulations (Storage tank, Phase 1 cases Eval1 and Eval2, slug-flow and free-bubbling flow)
- Profiling simulations

ENVIRONMENT

>> /home/obatj8/userbashrc

Environment variables are set up using environment flags for:

- Diamond and GMSH

```
PATH=$PATH:/hpcdata/obatj8/software/live/builds/gmsh/gmsh-3.0.6-Linux64/bin
PATH=$PATH:/hpcdata/obatj8software/live/builds/diamond/usr/local/bin
```

- Compiler: gcc7.x on the HPC (usually gcc4.8.5 on local Ubuntu 18.x)

- Open MPI (OMPI)
  - Version 4.x (Petsc 3.8)
  - Version 4.x debug (Petsc 3.8)
  - Version 2.x Skylake Fabric optimised (Cindy installed)
  - Version 4.x with newest PETSc build (3.14)

Example:

```
module load openmpi/4.0.1
PATH=$PATH:/hpc/apps/RHEL/3.10/x86_64/openmpi/4.0.1/bin

# PETSC and ZOLTAN
export PETSC_DIR=/hpcdata/obatj8/software/live314/builds/petsc/3.14.1/petsc-3.14.1
export PETSC_ARCH=linux-gnu-c-opt
PATH=$PATH:/hpcdata/obatj8/software/live314/builds/zoltan/v3.83/zoltan-v3.83-build
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/hpcdata/obatj8/software/live314/builds/zoltan/v3.83/zoltan-v3.83-
build/lib

LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/hpc/apps/RHEL/3.10/x86_64/openmpi/4.0.1/lib/openmpi
LD_RUN_PATH=$LD_RUN_PATH:/hpc/apps/RHEL/3.10/x86_64/openmpi/4.0.1/lib/openmpi
PATH=$PATH:/hpcdata/obatj8/software/ICFERST_ompi314/bin
export PYTHONPATH="${PYTHONPATH}:/hpcdata/obatj8/software/ICFERST_ompi314/python"
PATH=$PATH:/hpcdata/obatj8/software/vtk/VTK_build

export PATH
export LD_LIBRARY_PATH
export LD_RUN_PATH
```

- Intel MPI (IMPI)
  - Ivybridge
  - CascadeLake (different intel version to Ivybridge)
  - IFort setup

Phase 1 Report

Distributed in December 2019

1 SUMMARY

The inertial/turbulence modelling code has been validated for multiple scenarios, from compressible shock problem (2D and 3D sod shock tube problem) to simple flow between parallel plates in both the laminar and turbulence regime. These simple and fundamental problems demonstrate and test a range of modelling capabilities of the code and its state-of-the-art numerical methods/schemes. An example of a simple validation case is presented in Section 3, and has been designed recently to validate/test the code and also benchmark against FLUIDITY (CFD code developed in AMCG, Imperial College, and precursor to IC-FERST).

Results from Section 4.1 and 4.2 show significant progress of Phase 1 of the ICAM62 project, the flow past a cylinder and pipe flow of Reynolds number 3900 and 100000, respectively. Th flow past a cylinder is presented here shows strong agreement with experimental results taken from literature and is a novel demonstration of 3D flow past a cylinder with adaptive meshing. Considerable accuracy can be obtained when using adaptive meshing technology with linear-elements and Discontinuous-Galerkin methods (referred to as P1DG here) for solutions of velocity/momentum in IC-FERST.

In terms of meeting ICAM62 project deliverables, final Phase 1 deliverables have been addressed with the completion of verifying the flow past a cylinder test against literature ('Case 1'). Additionally, 'Case 2' deliverable, a transient, single-phase flow in pipe simulation with adaptive meshing has been completed for a several pipe configurations; a strong scaling performance evaluation of a straight pipe simulation is also presented in this report in Section 2.2. The parallel performance of IC-FERST turbulence modelling shows very good scaling when ensuring that the number of degrees of freedom per core is kept at ~30,000. We think that the main bottle-neck could be the parallel efficiency of the PETSc linear solvers (in this case GAMG for pressure) at large numbers of cores and therefore, accounting for solver timings might give better representation (and improved results) for the parallel efficiency of the code itself.

As part of an earlier deliverable to 'develop a tool workflow for complex industrial geometry', the development of a complex pipe network generator tool as part of a summer research project is highlighted in Section 6. This tool, developed by an MSc student working with Asiri, has been tests on the ICL HPC, and is released as an open-source tool based on Python scripting. Initial progress has been made on Phase 2 of the project a part of overall research activities and involvements in other projects, including improvements in the robustness in the parallel mesh adaptivity method, mesh-movements methods (implicit solid modelling). The MUFFINS/PREMIER project focuses on CFD modelling using IC-FERST or slug-flow and other multi-phase applications. The development and studies deliverable as part of both projects are relevant to the delivery of the ICAM2 project improvements to the numerical modelling framework such as addition of new PETSc linear solvers (such as the point-Jacobi for solving velocity) has shown to significantly improve the robustness of simulations in parallel. The implicit fluid-solid interaction (FSI) method developed as part of MUFFINS project could be potentially important to modelling different structural aspects of the Phase 2 Mixing tank (the moving roof, the impellers, etc). This method can be parallelised using the same framework as the fluid-modelling part of IC-FERST, and therefore, is 'ready-to-go' with regards to installation and testing in the HPC.

Overall, the research activities, both within the ICAM62 project and the involvement in MUFFINS/PREMIER, will enable Phase 2 of the project (i.e. study of a crude strong tank) to be successfully completed when work re-commences by Asiri. Additionally, the work by the research group as a whole in such projects as MUFFINS/PREMIER has also made significant contributions to additional projects of interest to BP and the wider industry (i.e. pipework vibration study through fluid-solid interaction modelling in MUFFINS by Chris Pain and Jiansheng Xiang and deep learning methods for predictive fluid dynamics and quantification of multiphase systems in PREMIER).

7 HPC

7.1 SUMMARY

- Built and tested the latest release of IC-FERST with new OpenMPI 4.x and latest BP-HPC system Intel-MPI
- IC-FERST and OpenMPI to be tested with new HPC operating system update CentOS 7.7 in the coming weeks in the HASKELL test sever.

7.2 HPC SCALING RESULTS

Strong scaling testing has been re-considered and has shown good scaling (see Figure 1) when ensuring that each core is not underloaded (i.e. the number of elements per core is above a certain threshold, around 30,000 elements per core).

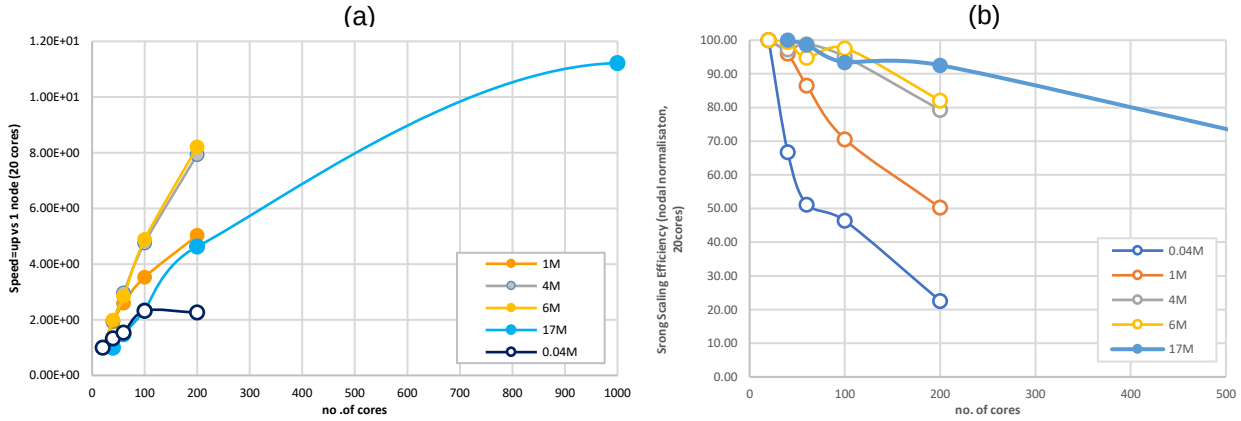


Figure 20 - Strong scaling test on IVY-BRIDGE for four different mesh sizes showing (a) speed-up ad (b) strong-parallel efficiency when comparing to a base-case scenario of 1 node (20 cores). Simulation is a single phase injection into a straight pipe

8 PERFORMANCE AND VALIDATION

8.1 FLOW BETWEEN PARALLEL PLATES

Laetitia Mottet, Asiri Obeysekara, Christopher Pain *et al* (Summer 2019)

Validation of simple flow between parallel plates was undertaken by Laetitia Mottet (AMCG), comparing theoretical results (Cubic flow, Snow et al (1965)) and with Fluidity (Continuous- and Discontinuous-Galerkin, referred to as CG and DG, respectively) and IC-FERST (DG). Laetitia was provided the setup files and geometries by Asiri (for IC-FERST) and conducted the validation and benchmarking test. The results shown below in Figures 2 and 3 highlight validation of the velocity profile and shear stress measured along a cross section of a domain representing two-parallel plates in the laminar and turbulent regime.

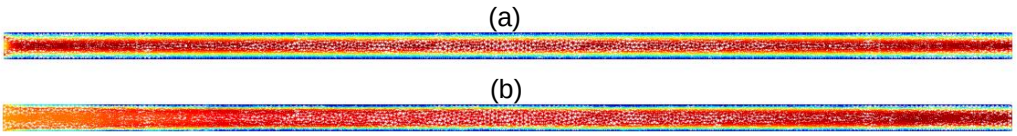


Figure 21 - Velocity field for flow between parallel plates on a fixed mesh in the (a) laminar and (b) turbulent flow regime

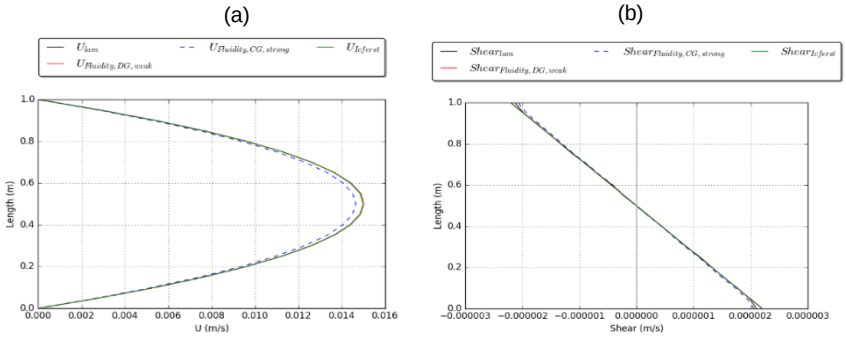


Figure 22 – Comparison between theoretical, fluidity and ic-ferst for (left) cross-stream velocity profile and (right) shear-stress profile in the laminar regime.

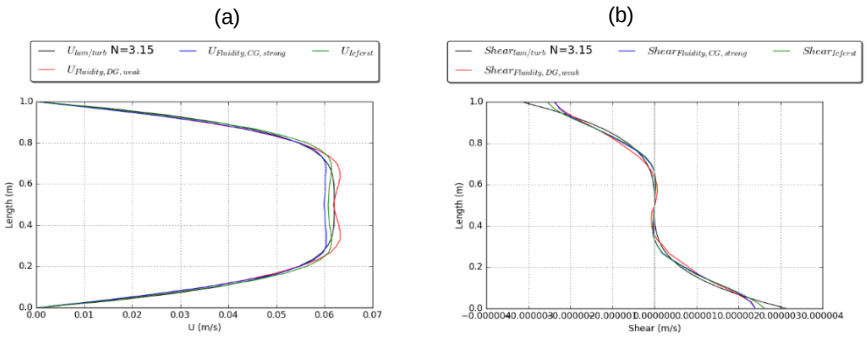


Figure 23 - Comparison between theoretical, fluidity and ic-ferst for (left) cross-stream velocity profile and (right) shear-stress profile in the turbulence regime

9 SINGLE-PHASE TURBULENT FLOW

9.1 FLOW PAST A CYLINDER – EVAL CASE 1

9.1.1 DOMAIN

Validation of 3D flow past a cylinder validation using dynamic mesh optimization.

Results (section 3.1.2) are verified against numerous experimental studies for Reynolds number 3900. The schematic and initial mesh for the simulation is shown in Figure 5 and Figure 6, respectively. The simulation is based on the paper by Rajani *et al* (2016)

Comparisons are made for adaptive meshing minimum edge length limits of 1 mm and 0.8 mm (Figure 7).

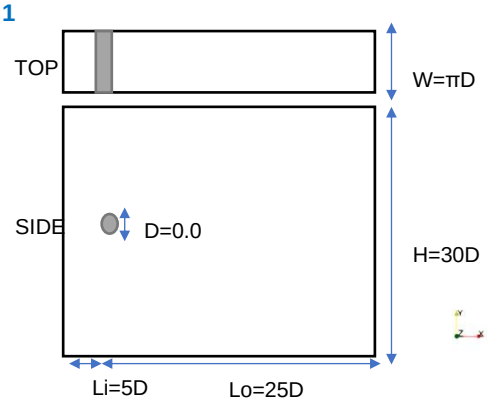


Figure 24 - Schematic of 3D flow past a cylinder investigation from a 30D x 30D x πD domain from Rajani et al (2016)

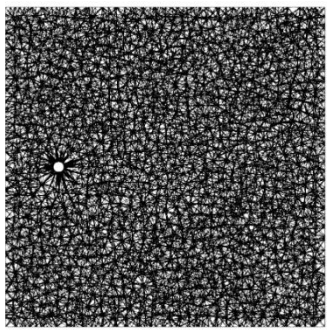


Figure 25 - Mesh of 3D flow past a cylinder investigation from a 30D x 30D x πD domain from Rajani et al (2016)

9.1.2 RESULTS

Very close agreement can be seen in Figure 8 for both cases when compared with the mean velocity profiles along the wake-line of the cylinder, and transverse to the cylinder as measured in five different sets of experimental results in literature. As seen in Figure 8(a) and 8(b), even the 'coarse' simulation using adaptive meshing was able to get a very close agreement between the simulation and experiments on 5 cores and maximum total elements of 350,000 using minimum element edge length of 1 mm, much lower than the LES model in Rajani et al (2016) which used fixed meshes of 550,000 to 5.5M elements.

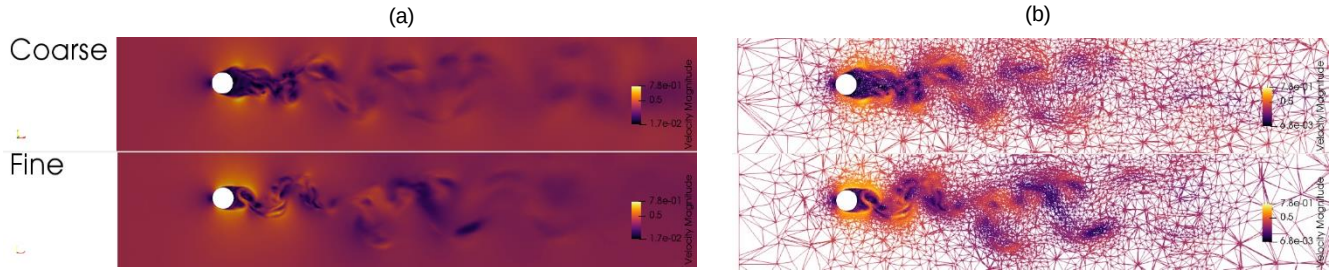


Figure 26 - Results for flow past a cylinder numerical simulation using IC-FERST visualising on (a) surface cut-plane and (b) the adaptive mesh for coarse (top) and (fine) results with 1 mm and 0.8 mm minimum edge length, respectively.

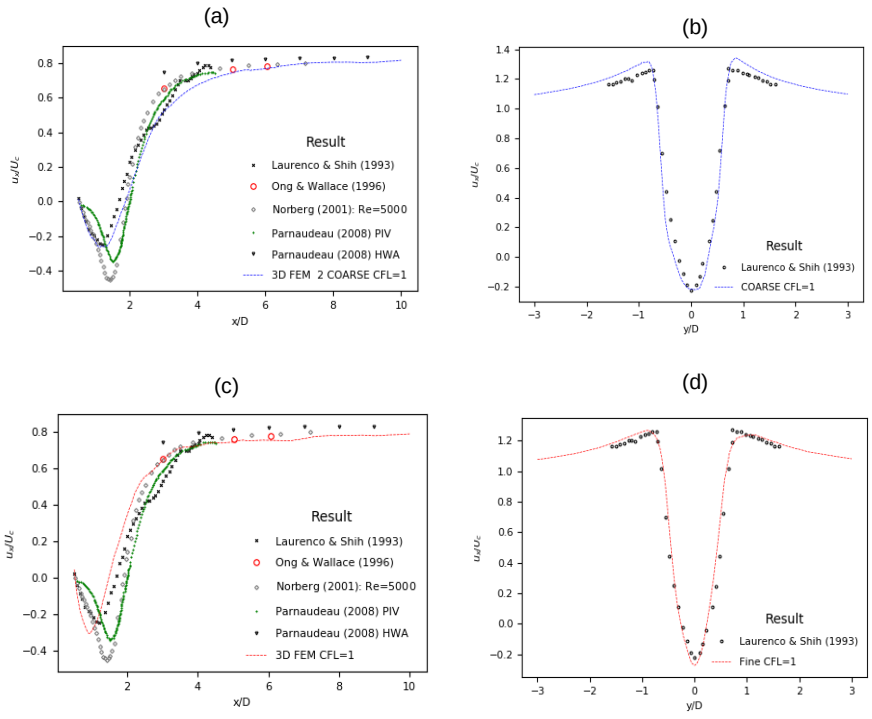


Figure 27 - Verification of streamwise velocity (a) along the wake-line for coarse mesh parameters, (b) transverse for coarse mesh parameters, (c) along the wake-line for fine mesh parameters and (d) transverse

Experimental studies derived from:

1. Rajani, B. N., Kandasamy, A., & Majumdar, S. (2016). LES of flow past circular cylinder at Re = 3900. Journal of Applied Fluid Mechanics, 9(3), 1421–1435.
2. Parnaudeau, P., Carlier, J., Heitz, D., & Lamballais, E. (2011). Experimental and numerical studies of the flow over a circular cylinder at Reynolds number 3900. 085101(3900).
3. Franke, J., & Frank, W. (2002). Large eddy simulation of the flow past a circular cylinder at ReD = 3900. Journal of Wind Engineering and Industrial Aerodynamics, 90(10), 1191–1206.
4. L.M. Lourenco, C. Shih, Characteristics of the plane turbulent near wake of a circular cylinder. A particle image velocimetry study. 1993, (data taken from Beaudan, and Moin(1994)).
5. L. Ong, J. Wallace, The velocity field of the turbulent very near wake of a circular cylinder, Exp. Fluids 20 (1996) 441–453.

9.2 SINGLE PHASE PIPE FLOW – EVAL CASE 2

The preliminarily results from modelling Re=100000 flow in a pipe using IC-FERST are presented below for 13 seconds of simulation time, as turbulent eddies start developing at the two 90° bends. The meshes from the results highlight dynamic mesh optimisation resolving well on velocity starting from an initial coarse mesh (see Figures 9 and 10). The pipe-with-bend flow simulations at high Reynolds number shave been run on the HPC up to 1000 cores on IVY-BRIDGE using OpenMPI and Intel MPI.

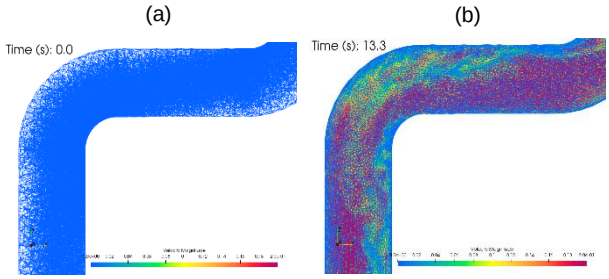


Figure 29 - Visualising the adaptive mesh and velocity at time (a) initial T=0 s and (b) T=13 seconds for Re=100000

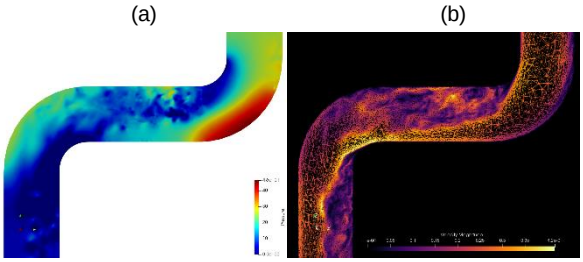


Figure 28 – Visualisation of a cut-plane showing (a) Pressure and (b) velocity of single-phase flow in a pipe at T=13 seconds

10 MULTI-PHASE FLOW MODELLING (MUFFINS/PREMIER PROJECT + ICAM62)

Experimental studies derived from:

1. Mayor, T. S., Pinto, A. M. F. R., & Campos, J. B. L. M. (2007). Hydrodynamics of Gas - Liquid Slug Flow along Vertical Pipes in the Laminar Regime s Experimental and Simulation Study. 3794–3809. https://doi.org/10.1021/ie0609923

10.1 INTRODUCTION

The MUFFINS research is an EPSRC-funded project involving Imperial College London and other project partners including research organizations and oil & gas industrial companies (http://www.muffinsproject.org.uk/). Through the development of IC-FERST, alongside machine learning and FSI, it aims to develop the “next-generation” of tools for research and industry. As part of research activities for the project, recent improvements to the viscosity scheme and compressive advection scheme for interface capturing, alongside the addition of a point-Jacobi preconditioner for solving velocity (using linear solver PETSc GMRES) has improved the multiphase modelling capabilities of IC-FERST. A current validation case for the MUFFINS project is described below, and demonstrates the ability of IC-FERST to capture free-bubbling flow along a vertical pipe. The simulations are run in parallel and have been tested on the HPC up to a 100 cores (OpenMPI 4.1, IVY-BRIDGE).

10.2 GAS/LIQUID FREE-BUBBLING FLOW ALONG A VERTICAL PIPE (MAYOR *ET AL*, 2007)

The geometrical parameters of the modelling domain and phase properties are shown in Table 1, they are based on lab experiments conducted by Mayor *et al* in 2007 for a range air/liquid superficial velocities. The results in this document are for air and liquid superficial velocities of 0.38 ms^{-1} and 0.21 ms^{-1}, respectively. The liquid used in the experiments is an 85% aqueous glycerol solution.

Table 2 - Parameters for vertical multiphase flow experiment (based on experiments by Mayor et al (2007))

Main Geometrical Parameter	Dimensions	
Internal pipe diameter (d) m	0.003	
External pipe diameter (D) m	0.034	
Length (L)	150*D	
Main Fluid Parameters	**Air**	**Glycerol**
Density (kg/m^3)	1.125	1100
Viscosity (Pa.s)	1.81×10^{-5}	0.114

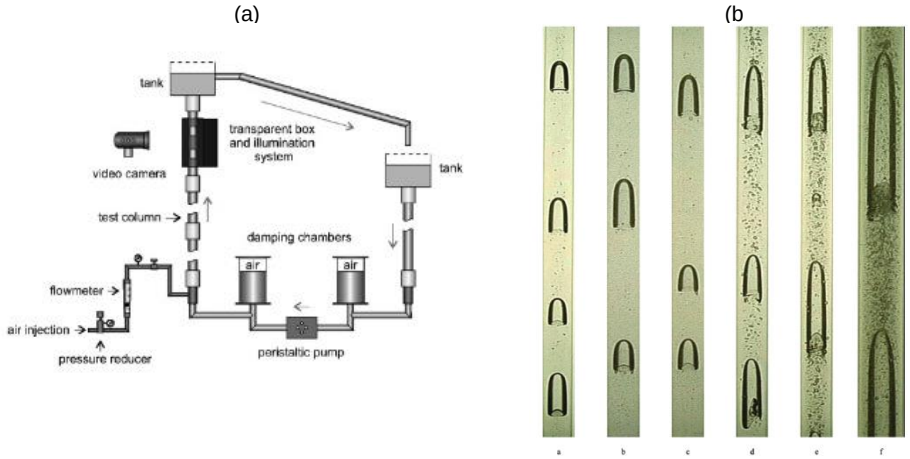


Figure 30 - Free-bubbling slug flow experiment conducted by Mayor et al (2007) using an experimental setup as in (a) for a vertical pipe injected with different liquid/ai superficial velocities producing different slug and Taylor bubble properties as imaged in (b)

10.3 PRELMINARY RESULTS

10.3.1 SURFACE TENSION

Preliminary show IC-FERST effectively captures free-bubbling flow with the surface tension solver implemented in the code (Zhi *et al*, 2015). The dynamic adaptive mesh optimization has a max/min element edge length aspect ratio of 250, with both the velocity and phase volume fraction (interface) as the adaptive meshing metric.

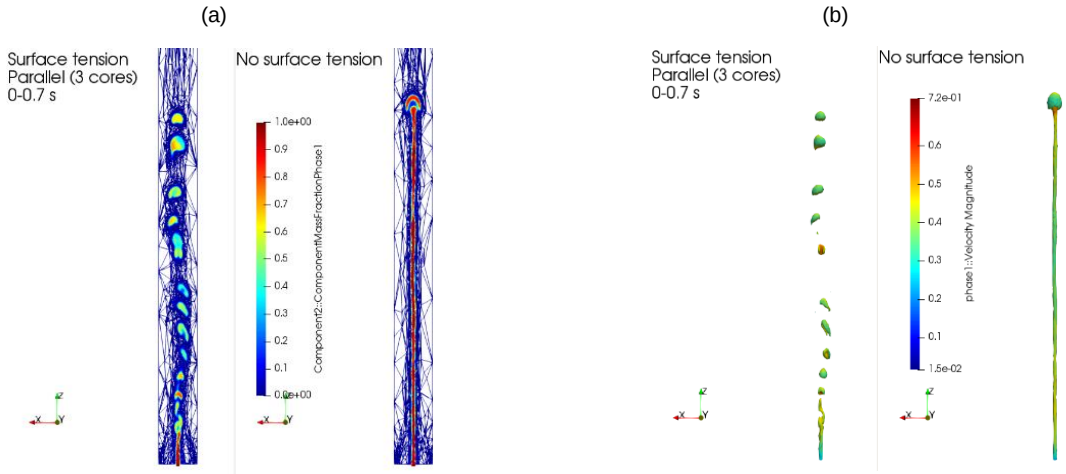


Figure 31 - Comparisons of (left) mesh and (right) gas-phase shape when considering (a) surface tension and (b) not considering surface tension

10.3.2 SLUG FLOW MODELLING

The initial results from the study demonstrate that the numerical methods and schemes in IC-FERST is able to capture free-bubbling vertical slug flow as in the experimental studies by Mayor *et al* (2007) (see Figure 13). Although the study is in the laminar flow regime (due to the high viscosity of the liquid phase), IC-FERST is able to capture the bullet-shape bubbles evolving in the vertical pipe. The adaptive meshing, interface-capturing scheme, alongside the hydrostatic pressure and surface tensions solvers can model the complex interaction and coalescence of the Taylor bubbles and the liquid “slug” regions between the bubbles. Further investigation of longer pipe lengths, different gas and liquid superficial velocities, and improvements to the interface-capturing method aims to improve the multi-phase modelling capabilities of IC-FERST and validate the code for modelling these complex flows.

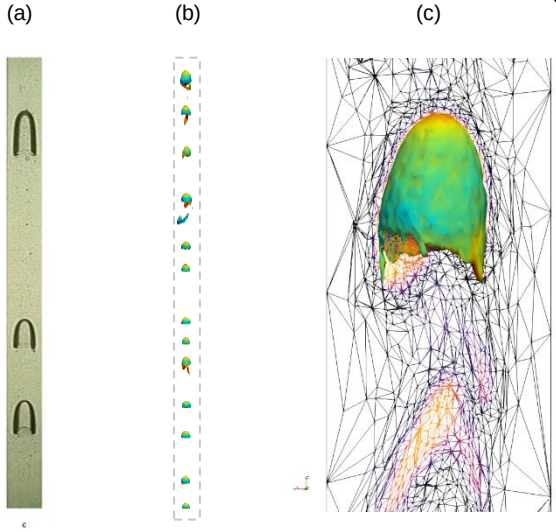


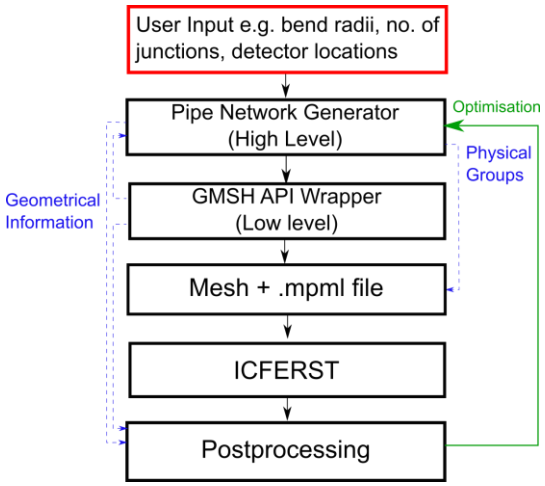
Figure 32 - (a) Experimental results for superficial velocities studies in this simulation (b), capturing the "bullet-shaped" taylor bubbles as in (c) with the mesh and contour of a bubble with velocity profile

11 STUDENT RESEARCH PROJECTS

Development of a Complex Pipe-network/Mesh Generator
PROJECT: “Integration of CFD Modelling Framework IC-FERST for Industrial Application in BP: Automation of pre- and postprocessing using PythoN”
STUDENT: Duncan Hunter (Applied Computation And Engineering Masters Course)

As part of a summer MSc research project supervised by Asiri, Chis and Andre, the ***PIPEMESH*** pipe network generator was designed and developed by Duncan Hunter. This tool allows the creation complex single pipe and networks to be created easily, leveraging on existing GMSH API additional tools created by the student. The work-flow of the tool, including integration with IC-FERST and examples are given below.

11.1.1 WORKFLOW/SCHEMATIC



- Scriptable generation of pipe network meshes.
- Batch IC-FERST pipe network flow simulations.
- HPC capable.
- Can integrate with post-processing software.
- Overall a smoother/more automated process.

Download via:
(python package) https://pypi.org/project/pipemesh/
(github) https://github.com/Duncan-Hunter/pipemesh

11.1.2 PIPE-PIECES GENERATOR

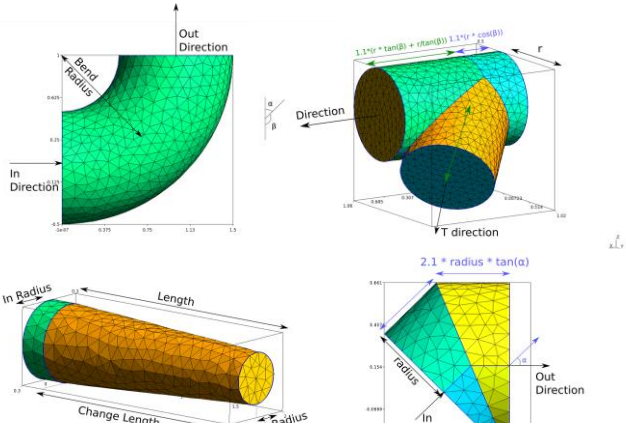


Figure 34 - Examples of pipe geometries easily produced using the PipeMesh tool

- Use GMSH API to create specific objects and meshes.
- Store object information in Python classes.
- Use these classes to build pipe meshes with Python (join the pieces together).
- Use geometrical information to set up automatic post-processing and simulation files.

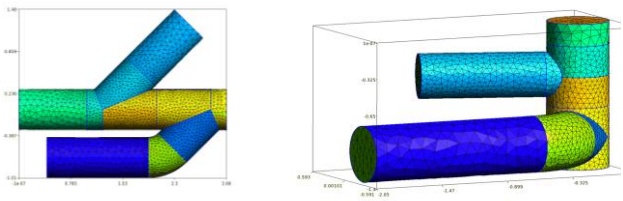


Figure 35 - Examples of pipe network attachments generated using the tool

11.1.3

- Add pieces in modular, sequential fashion.
- User or program creates geometry with length, radii, directions, mesh sizes.

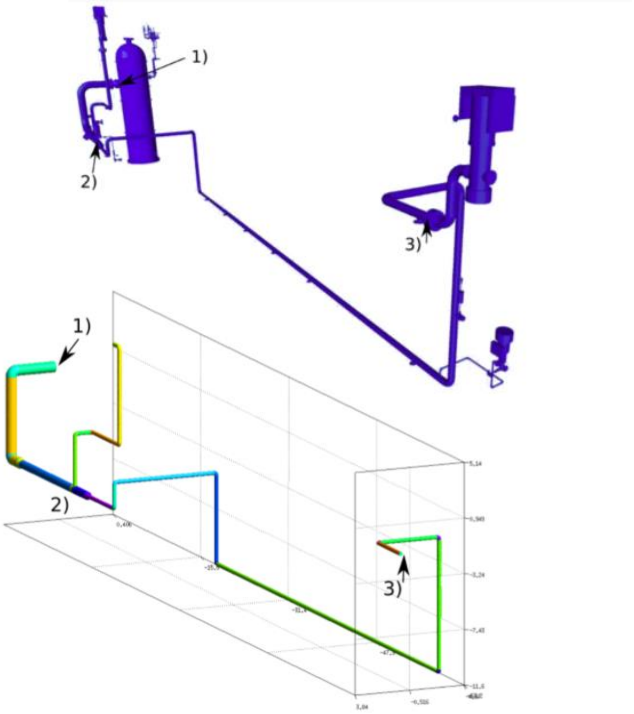


Figure 33 - Example of re-producing a realistic pipe network (top) using the PipeMesh tool which is ready for parallel, adaptive meshing simulation (bottom)

END