University of Exeter coursework header sheet

ECMM102 Group Project (Meng) (A, TRM1+2 2017/8)

Extension

054559

1029797

640034919

Coursework: Individual contribution to the group achievement

Submission Deadline: Mon 21st May 2018 12:00

Personal tutor: Dr Isaac Luxmoore

Marker name:	Tabor
Word count:	16234

UNIVERSITY OF EXETER

I2 Report

Investigation into mesh generation using snappyHexMesh for a model sedimentation tank and hydrodynamic vortex separator including volume of fluid simulations

Joshua Lowe

2018

4th year MEng Group Project

I certify that all material in this thesis that is not my own work has been identified and that no material has been included for which a degree has previously been conferred on me.

Signed............................... Jlowe ....

College of Engineering, Mathematics, and Physical Sciences
University of Exeter

I2 Report

ECMM102

Title: Investigation into mesh generation using snappyHexMesh for a model sedimentation tank and hydrodynamic vortex separator including volume of fluid simulations

Word count: 16234
Number of pages: 40

Date of submission: Monday, 21 May 2018

Student Name: Joshua Lowe
Programme: MEng Mechanical Engineering
Student number: 640034919
Candidate number: 054559

Supervisor: Dr Gavin Tabor

Abstract

The work undertaken in this report was part of a group project which had the main aim of validating multiphase simulations performed using the OpenFOAM® solver `driftFluxFoam` against empirical data obtained at the university of Exeter Fluids laboratory for two model primary sedimentation devices; a modified W7 Armfield rectangular sedimentation tank and a hydrodynamic vortex separator (HDVS). The subsequent aim of this work was to provide high quality meshes of these sedimentation devices for use in the `driftFluxFoam` simulations. As the free surface was modelled as a fixed flat horizontal wall, the project also included water-air volume of fluid (VOF) simulations to investigate how well the rigid-lid assumption captured the free surface elevation and topology across the geometries.

The meshes were produced using the native OpenFOAM® meshing utility `snappyHexMesh` and their quality evaluated. A parameter study was performed to gain an understanding of how the software's control parameters affected the quality of the meshes produced. It was found that in most cases, `snappyHexMesh` was able to produce grids that satisfied OpenFOAM®'s mesh quality requirements with suitably low values for aspect ratio, non-orthogonality and skewness. The software was found to also reasonably capture the geometric features of the devices, with a calculated maximum displacement between the mesh boundary and the model surface of 0.6 mm for the HDVS. However, inserting viscous layers into the mesh was found to be too difficult and has been identified as a serious limitation of the `snappyHexMesh` software.

The VOF simulations were performed using the solver `interFoam` and the rectangular tank modelled in 2D to reduce computational time. The simulation was found to accurately predict the surface elevation within 10% above the overflow weir, but more empirical data is required to verify the simulation accuracy across the length of the tank. For the HDVS, less agreement was observed between the experimental data and the numerical results with percentage errors as large as 54.4%. This was attributed to potential differences between the numerical and experimental inlet flow rates and the large 12 mm interface region in the simulation due to inadequate cell resolution above the overflow tray. This was because of limited computational resources and project time.

Keywords: Wastewater, Hydrodynamic vortex separator, snappyHexMesh, Volume of Fluid, free surface, CFD

Acknowledgements

I wish to thank my project supervisor Professor Gavin Tabor and Dr Shenan Grossberg for all the advice, guidance and support they provided during the project. I would also like to thank Dr Recep Kahraman and Mr Matthew Riella, who provided useful insights for the Volume of Fluid simulations undertaken in this report. Finally, special thanks to Mr Robert O'Neale, who was always willing and happy to help with the variety of IT issues encountered during the project.

Table of Contents

1. Introduction and background

In wastewater treatment, sedimentation tanks are used to separate heavy organic solids from the influent. This is known as the primary treatment stage, and takes place after large debris and grit have been removed from the wastewater in a pre-treatment screening process. (Partech, 2018; Water Technology Engineering Ltd, 2018) Assuming that the tanks have been well designed, the sedimentation process can remove between 50 and 70 percent of the suspended solids present in wastewater that otherwise would be discharged back into a receiving water course (Tchobanoglous, et al., 2014, p. 382). Generally, sedimentation tanks are either rectangular or cylindrical in shape, with mechanical scrappers that collect the settled sludge into a containment device (typically situated in the tank floor) from which the sludge is periodically removed. The clarifed effluent exits the tank through overflow weirs, situated on the opposite side to the inlet (Brennan, 2001).

An alternative to rectangular or circular tanks for solid-liquid separation are hydrodynamic vortex separators (HDVSs). HDVSs were originally developed during the early 1960s to remove coarse solids from wastewater exiting combined sewer overflows (CSOs) (Meroney & Sheker, 2016). However, after thirty years of further development, their application has expanded into grit separation (pre-treatment), primary sedimentation, industrial effluent and stormwater treatment (Faram, et al., 2004). An illustration of a HDVS for primary sedimentation (the Swirl-Flo®) is shown in Figure 1. The device consists of a cylindrical chamber with a tangential inlet pipe, a central hopper for solids collection, an underflow pipe for solids removal, and an overflow channel for the clarified effluent to proceed onto the next treatment phase. The trajectories of individual particles throughout the HDVS depend on their size and density, as this affects the balance of the inertial, drag (due to fluid viscosity) and gravity forces experienced. Heavier solids tend to migrate radially towards the outside walls where they continue to settle under the force of gravity, while lighter particles tend to move towards the centre of the device. At the centre, an upward moving spiral of fluid carries the lighter particles towards the overflow (i.e. the non-settleable fraction of solids), whereas secondary currents sweep the settled solids into the central hopper for extraction through the underflow (Faram, et al., 2004).

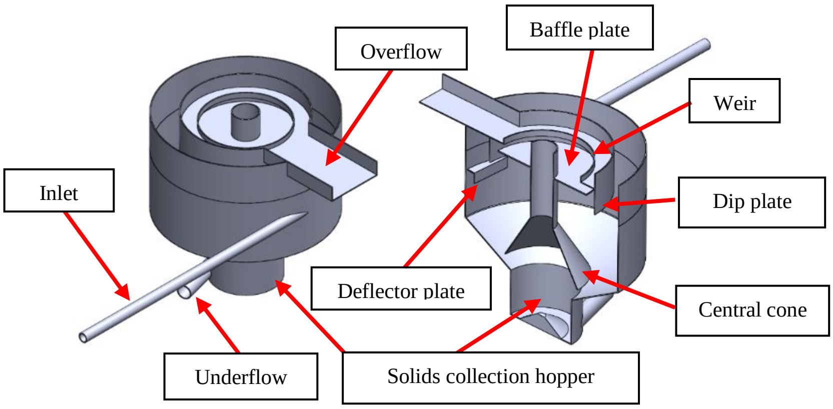


Figure 1: Illustration of the Swirl-Flo® HDVS separator.

HDVSs have several advantages over conventional rectangular and radial sedimentation tanks. Firstly, the long circular flow patterns within HDVSs (due to the rotary motion of the flow) increases the residence time of the solid particles, aiding the gravity settling process. Secondly, larger overflow rates can be applied, meaning that smaller tank volumes can be used, thus reducing the footprint required. Thirdly, they require less maintenance due to the lack of moving components, reducing operating costs (Sullivan, et al., 1978).

In recent years, computational power and its affordability has dramatically increased. As a result, the use of computational fluid dynamics (CFD) to model the different unit processes of water and wastewater treatment (WWT) has also steadily risen. Other factors that have given rise to the increased uptake of CFD in WWT include increasing availability of quality CFD models, as well as the growing need for optimisation of WWT unit processes (Wicklein, et al., 2015). CFD models have been used to further understand the flow conditions present in WWT processes, aiding in their design and optimisation. In comparison to physical models, much less time is required for the same level of analysis of the hydrodynamic behaviour, allowing multiple design options to be explored (Findikakis, 2016).

The quality of the mesh used in CFD simulations has a significant impact on the solution obtained. Using a poor-quality grid can have a large effect on the solution accuracy, the simulation runtime, or whether a solution is obtained at all (i.e. the solver diverges) (Tabor & Fabritius, 2016). The quality of the grid depends primarily on two factors; the local sizing of the cells that divide the computational domain into finite volumes and the cell shapes (Kallinderis & Kontzialis, 2008). In industry, the creation of the domain geometry and generation of a suitable mesh requires over 50% of a CFD project timeframe (Versteeg & Malalasekera, 2007, p. 3), highlighting the importance of this aspect of the CFD process.

The work undertaken in this report was part of a group project in which experiments and multiphase simualtions were performed on two model sedimentation devices; a 1.13 m × 0.4 m × 0.25 m Armfield W7 rectangular sedimentation tank and a 0.6 m diameter HDVS Swirl-Flo® (provided by Hydro International). The solver `driftFluxFoam` (part of the open source

CFD tool box OpenFOAM®) was utilised to perform the multiphase simulations for comparison to empirical results obtained from the experiments; these used an olive stone powder water mixture as a substitue to real wastewater. Mendoza (2018) and Wye (2018) developed rheological and settling velocity models from the empirical data to be used in the `driftFluxFoam` simulations performed by Bentley (2018) and Russell (2018).

Subsequently, the aim of this project was to provide high quality meshes of the sedimentation devices for use in the `driftFluxFoam` simulations. The native OpenFOAM® meshing utility `snappyHexMesh` was selected as it is a fully automated cut-cell meshing software (Greenshields, 2017f) allowing the generated meshes to be compared to the computational grids produced by Scobell (2018) who used the commercial software Pointwise®, which employs a bottom-up meshing approach. A comparison of the meshes produced are provided in Baker et al. (2018).

The utility `snappyhexMesh` is controlled using an OpenFOAM® `C++` dictionary with a substantial number of control parameters. Thus, a further aim of this project was to understand how each of these affected the quality and accuracy of the sedimentation device meshes produced. The third aim of the project was to evaluate the accuracy of modelling the free surface as a fixed, flat and frictionless horizontal wall (known as the rigid-lid assumption) (Beg, 2017) in the `driftFluxFoam` simulations. Thus, two phase water-air volume of fluid (VOF) simulations were also performed using the solver `interFoam` on the sedimentation devices to evaluate the differences between the predicted free surface elevation, the rigid-lid assumption and the experimental results.

The following objectives lead to the completion of these aims:

1) Use the `blockMesh` utility to provide a high quality 2-dimensional mesh of the rectangular tank for VOF `interFoam` simulations.
2) Use the `blockMesh` utility to generate high quality hexahedral background meshes for use with `snappyHexMesh`.
3) Create closed STL files of the sedimentation geometries for use in `snappyHexMesh`.
4) Use the OpenFOAM® utility `checkMesh` to determine the effects of varying individual `snappyHexMesh` settings on the quality of the generated sedimentation device meshes.
5) Use `snappyHexMesh` to generate high quality meshes of the rectangular tank and Swirl-Flo® geometries for use in `driftFluxFoam` and `interFoam` simulations.

2. Literature review

The most extensive study identified in which the quality of generated meshes produced by `snappyHexMesh` were analysed was that undertaken by Tabor and Fabritius (2016). As part of their research, the second-generation non-dominated sorting genetic algorithm (NSGA-II) created by Deb et al. (2002) was used to develop a genetic algorithm code in the language Python for optimising the `snappyHexMesh` grid generation process on three different geometries; these were composed of a bearing (formed by two pipes of different diameters connected by a planar disc), a packed bed consisting of eight spheres, and the well-documented benchmark Ahmed Body case used in car aerodynamics. The total number of cells, mesh quality and the snapping accuracy to the STL surface were identified as the three main optimisation objectives. The quality of the meshes produced were evaluated by combining the quantities for mesh maximum non-orthogonality, maximum skewness and minimum cell size into a single fitness measure, achieved using the fractional biased error of each quantity. This was done to account for the different orders of magnitude between each quality metric.

Tabor and Fabritius (2016) found that, for the bearing case, the parameter setting for the minimum cell volume ratio between adjacent cells had an optimum value of 0.01. As such, it was decided that this value would be used for the meshes generated in this report. They also discovered for all three cases that there was a correlation between the average and maximum surface displacement; hence only the maximum surface displacement has been evaluated in the parameter study undertaken. The same tool developed by Tabor and Fabritius (2016) to quantify the accuracy of the snapping algorithm has also been used in this report to calculate the distance between the STL surface and boundary faces of the meshes produced. Tabor and Fabritius (2016) also discovered that, for certain settings, the total number of cells between different individuals of a population remained very similar; this was attributed to the total number of cells being largely dependent on the resolution of the background mesh. This meant that a detailed investigation into the trade-off between the total number of cells and mesh quality could not be undertaken. As such, in the parameter study undertaken in this report, it was decided that the number of cells in background mesh would be varied to see if this had any effect on mesh quality.

Using CFD, Egarr et al. (2005) compared the discrete phase and the user-defined scalar (UDS) approach for predicting the mean residence time of the fluid at the underflow and overflow of a HDVS. While the main work in the study is not directly related to that in this report, Egarr et al. (2005) provide some useful details about the grid that they used to model the HDVS. The mesh was composed of 745,471 tetrahedral cells, which were refined along the side of the dip plate and in the region between the weir and rigid-lid. A minimum of six cells was also ensured in discretising the underflow and the gaps formed by the proximity of the internal components (such as those between the weir and the baffle plate, the dip plate and the vessel walls). ‘This was to ensure that a crude boundary layer could be represented while also allowing for fluid to pass through the cells in the centre of the channel.' (Egarr, et al., 2005, pp. 58-59) The results using the UDS approach were well validated against the empirical data obtained by Alkhaddar, et al. (1999) for the same device, with a mean error of 11 percent for 20 different operating conditions. Because simulation results are largely dependent on the quality of the mesh used, it was decided to utilise the same refinement practices in this report for the Swirl-Flo® geometry.

Other devices that display swirling behaviour in water engineering are vortex flow control (VFC) devices. These are used to mitigate flooding risk between catchment regions by regulating the flow rate. Jarman et al. (2015) compared five Reynolds-averaged Navier-Stokes (RANS) based turbulence models to determine their suitability modelling turbulence in swirling flows found in VFCs. As HDVSs also exhibit swirling flow, it was deemed that Jarman et al.'s (2015) findings could be useful for this report. Single phase and multiphase simulations were performed using the OpenFOAM® solvers `simpleFoam`, `pisoFoam` and `interFoam`, the results of which were validated against experimental data. Three eddy viscosity turbulence models (the standard k-ε model, realisable k-ε and k-ω SST model) and two differential second-moment turbulence closure (DSM) models were evaluated. It was found that the DSM models were in excellent agreement with experimental data, with the Gibson and Launder (1978) variant having an average percentage error of +0.74% for head loss. However, the eddy viscosity models over-predicted the pressure loss across the chamber, with the standard k-ε model failing to predict the negative axial velocity in the core of the vortex. Negative axial velocities were predicted by the realisable k-ε and k-ω SST models, but did not display symmetric profiles or similar axial velocity magnitudes in comparison to the DSM models. The eddy viscosity models were thus deemed inappropriate for modelling VFCs at high flow rates. For swirl parameters greater than 3.14, the performance of the Gibson and Launder (1978) DSM variant (chosen for its closest correlation to pressure distribution data) was also found to deteriorate, indicating the need for further calibration or the use of the Large Eddy Simulation (LES) approach for higher swirl number geometries.

While the findings of Jarman et al. (2015) would indicate that using a DSM turbulence model would be preferable, these can require twice the simulation time to that of the eddy viscosity turbulence models (Jarman, et al., 2015). Furthermore, turbulence models such as the Renormalization Group (RNG) k-ε and realizable k-ε have been found to give satisfactory results between simulation and empirical results for HDVSs and similar devices. For example, Lu et al. (2017) found that simulating solid-liquid two phase flow for a vortex-type grit chamber the RNG model had an average error of 7.10% for particles with a diameter less than 100 μm while the realisable k- ε model had the lowest average error of 4.16% for particles greater than 200 μm in diameter. Meroney and Sheker (2016) also verified the realizable k-ε model's suitability by comparing its performance to a low-Reynolds number turbulence model for a HDVS in which internal velocities were sometimes quite low, reporting only minor differences in particle deposition results. Pathapati and Sansalone (2009) used the standard k-ε model when simulating a storm-water HDVS, and found the numerical results to be within 10% of experimental results for particulate matter (PM) concentrations, size distributions, and mass. Ying et al. (2012) also used the standard k- ε model to resolve the flow field for a storm water screened HDVS, similarly finding results to be within 10% of empirical data for PM separation and head-loss. Thus, the performance of the selected turbulence model is case specific. Despite the findings of Pathapati and Sansalone (2009) and Ying et al. (2012) suggesting that the standard k-ε model may be suitable, it was decided to use the realizable k-ε model in the HDVS VOF simulations performed in this report, as it is less computationally expensive than DSM models but has been shown to give better performance than the standard k-ε model in single phase steady-state simulations performed by Jarman et. al (2015).

3. Theoretical background

3.1 Mesh quality

In CFD, the computational domain is divided (or discretised) into a finite set of volumes known as cells. The Navier-Stokes equations which describe the fluid flow are also discretised, both spatially and temporally, with the partial derivatives formulated into a set of linear algebraic equations at each cell (Anderssson, et al., 2012). OpenFOAM® is a 3D unstructured polyhedral code, meaning that cells can have any number of faces that are not limited to a set number of edges (Greenshields, 2017a).

The quality of the mesh can have a significant impact on the accuracy and convergence speed of the numerical solution (or even if one is obtained at all). A number of priori quality metrics have been developed so that a mesh's suitability can be assessed before running a potentially time intensive simulation (Tabor & Fabritius, 2016). Examples of priori mesh quality metrics include cell aspect ratio, non-orthogonality and skewness; these are discussed below.

The ratio of the longest edge length to the shortest edge length is known as the cell aspect ratio (AR) and should ideally be equal to 1 (Bakker, 2008). Solution accuracy and convergence may suffer if meshes contain high aspect ratio cells in important regions inside the interior flow domain (Tu, et al., 2008). Cell aspect ratios below 5 are considered acceptable, but can be up to 10 inside the boundary layer (Anderssson, et al., 2012).

Non-orthogonality occurs when an angle is formed between the vector connecting two adjacent cell centres and the normal vector of the shared face (illustrated in Figure 2 (a)). In OpenFOAM®, this can introduce significant error as diffusive terms 'use the face normal vector to calculate the fluxes between cells' (Tabor & Fabritius, 2016, p. 430). Significant error can occur for non-orthogonality greater than 70° or 80°, but can be mitigated against through correction schemes below these values (Rhoads, 2014). The quality of a mesh is both given by its maximum and average non-orthogonality, as one highly non-orthogonal face can cause the simulation to diverge while the average value gives an indication of the overall mesh quality (Tabor & Fabritius, 2016).

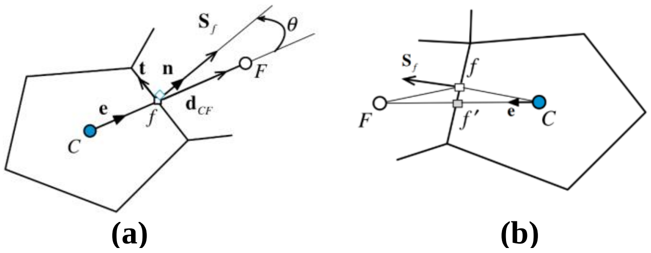


Figure 2: (a) Illustration of mesh non-orthogonality. (b) Illustration of mesh skewness (adapted from Figures 8.9 and 8.15 in Moukalled et al. (2016, pp. 241,254)).

One source of numerical diffusion is cell face skewness. This is depicted in Figure 2(b) and occurs when the centroid connecting vector ($\boldsymbol{c}$) of two adjacent cells does not intersect the centroid of the connecting cell face (f). Face skewness is given by $\xi = \delta/D$ (Rhoads, 2014), where δ is the distance between f and the interpolated face centre (f') and D is length of the vector $\boldsymbol{c}$. This can lead to large errors in OpenFOAM®, as the value of face integrals used to calculate the fluxes of a given quantity ϕ between cells requires the average face value of ϕ at point f. Linear interpolation between the two cell centroids is used to obtain this value. However, if the mesh is skewed, this will give the value of ϕ at point f' which may not be equivalent to the value at point f (Tabor & Fabritius, 2016). Again, corrective methods can be applied to mitigate the effects of skewness, but cannot completely eliminate the error (Rhoads, 2014). In OpenFOAM®, highly skewed faces are those deemed to have a skewness greater than 4 (Wolf Dynamics, 2017b, p. 34).

3.2 Governing numerical equations

The solver `interFoam` uses the RANS equations for modelling two incompressible, isothermal immiscible fluids in which the volume of fluid (VOF) method and an indicator function is used to capture the interface (Greenshields, 2017a). The RANS equations are given in Eq. (1) (the time-averaged mass conservation equation) and Eq. (2) (the time-averaged momentum equation), where $\overline{\boldsymbol{u}}$ is the time (ensemble) average velocity, τ is the mean viscous stress tensor, P is pressure, ρ is fluid density, υ is kinematic viscosity and t is time (Jarman, et al., 2015, p. 21).

$$\nabla.\overline{\boldsymbol{u}} = 0 \tag{1}$$

$$\frac{\partial \overline{\boldsymbol{u}}}{\partial t} + \nabla.\overline{\boldsymbol{u}}\overline{\boldsymbol{u}} + \nabla.\boldsymbol{R} = -\frac{1}{\rho}\nabla P + \nabla.\tau \quad \text{where} \quad \tau = \upsilon(\nabla\overline{\boldsymbol{u}} + \nabla\overline{\boldsymbol{u}}^T) \tag{2}$$

Equation (2) accounts for turbulent effects on the mean flow as it includes the term $\boldsymbol{R}$ (the Reynolds stress, $\overline{\boldsymbol{u}}'\overline{\boldsymbol{u}}'$) which denotes the time-averaged rate of momentum transfer due to turbulence. This represents six additional unknown terms and so the set of equations is not closed (Wilcox, 2006). Thus, a turbulence model is required 'to relate this unknown quantity to the known averaged variables.' (Jarman, et al., 2015, p. 21)

The realisable k-ε turbulence model uses the Boussinesq approximation given by:

$$R = \frac{2}{3}\rho \boldsymbol{I} k - 2\mu_t \overline{\mathbb{s}} \tag{3}$$

where k is the turbulent kinetic energy, μ_t is the eddy viscosity, and $\overline{\mathbb{s}}$ is the mean stress tensor (Jarman, et al., 2015, p. 21). The eddy viscosity is given by Eq. (4) (Wilcox, 2006, p. 130).

$$\mu_t = C_\mu \frac{k^2}{\varepsilon} \tag{4}$$

Multiplying the Navier-Stokes equation by the velocity fluctuating components and time averaging the product, a differential equation for the Reynolds stress tensor can be derived (Wilcox, 2006). From this, a transport equation for k can be formulated, which eliminates many of the unknown quantities (Jarman, et al., 2015).

The realisable k-ε turbulence model has two main differences to the standard k-ε model. Firstly, the derivation of the transport equation for the turbulent dissipation rate, ε, is based on the dynamic equation for fluctuating vorticity (Shih, et al., 1995). Secondly, the term C_μ is not taken to be constant (as in the standard k-ε model), and is calculated using the formulation proposed by Reynolds (1987) and Shih et al. (1994) as cited in Shih et al. (1995). This is done to ensure that R in Eq. (3) cannot become negative and that the Schwarz inequality is satisfied. As a result, the realisable k-ε model has superior performance to the standard k-ε model for flows with variable strain rates such as those which demonstrate swirl (Jarman, et al., 2015).

3.2.1 Volume of fluid method

A number of numerical methods have been developed to track the interface in multiphase flows. One such technique is the VOF method first proposed by Hirt and Nichols (1981) and uses a phase indicator function α (otherwise known as the volume fraction) to track the location of the interface. This is accomplished by defining a control volume as either being filled completely with one phase ($\alpha = 1$), the second phase ($\alpha = 0$), or containing a mixture of the two ($0 < \alpha < 1$). As such, the presence of the interface is denoted by cells containing intermediate values of α between 0 and 1 (Gopala & Wachem, 2008).

Assuming that there is no mass transfer between the two phases, the transport equation given in Eq. (5) governs the evolution of the volume fraction α (Hargreaves, et al., 2007):

$$\frac{\partial \alpha}{\partial t} + \nabla.(\alpha \overline{\boldsymbol{u}}) = 0 \tag{5}$$

However, discretising the advection term using first order schemes introduces numerical diffusion that smears the interface. On the other hand, using second order schemes causes instability and numerical oscillations. To keep the interface sharp, an artificial compressive term is introduced into Eq. (5) giving (Gopala & Wachem, 2008):

$$\frac{\partial \alpha}{\partial t} + \nabla.(\alpha \overline{\boldsymbol{u}}) + \nabla.\big(U^r \alpha(1-\alpha)\big) = 0 \tag{6}$$

The term U^r is a suitable velocity field used to compress the interface, while the term $\alpha(1-\alpha)$ ensures that the artificial compression is only active in the interface region (Gopala & Wachem, 2008). In `interFoam`, the multidimensional universal limiter for explicit solution (MULES) method is employed to ensure boundedness of α, meaning that the term $\nabla.(U^r \alpha)$ can be discretised by a second order accurate scheme (Greenshields, 2017a). However, despite the artificial compressive term, the sharpness of the interface is still heavily dependent on the grid resolution, as the interface region is typically smeared over a few cells (Damián, 2009).

3.3 snappyHexMesh meshing process and control parameters

The overall process of generating a mesh using the `snappyHexMesh` utility is illustrated in Figure 3 (OpenCFD, 2017b). There are three prerequisites before a grid can be generated using `snappyHexMesh`. Firstly, the geometry to be discretised must be provided as one or more triangulated surfaces in the Stereolithography (STL) or Wavefront Object (OBJ) file format. Secondly, a background mesh containing only hexahedral cells should encompass the geometry, with all surfaces intersected by at least one cell edge; this is usually produced using the `blockMesh` utility. Cell aspect ratios should also be approximately equal to 1, otherwise the snapping stage (conforming the background mesh to the surface of the geometry) could fail. Finally, an OpenFOAM® dictionary containing the

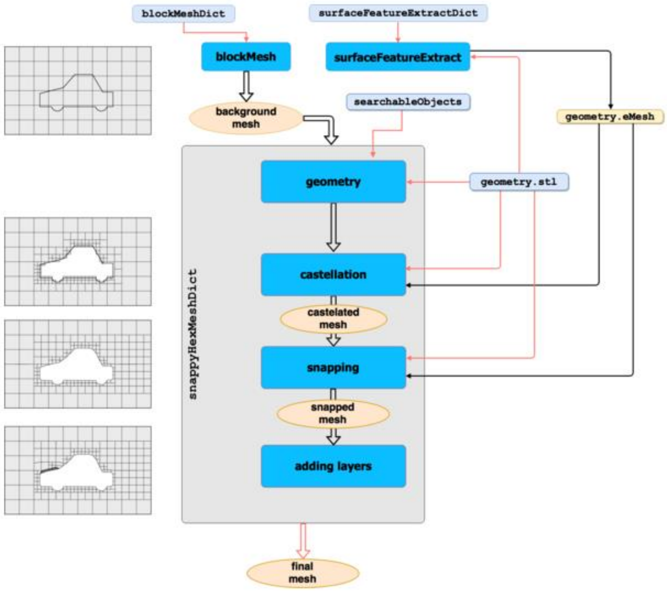


Figure 3: Flow diagram showing the overall meshing process of snappyHexMesh (OpenCFD, 2017b).

relevant keywords and sub-dictionaries used to control the snappyHexMesh meshing stages is required; this is known as the snappyHexMeshDict dictionary (Greenshields, 2017f). Because snappyHexMesh contains more than 70 control settings (Wolf Dynamics, 2017a, p. 19), only those deemed most relevant to each meshing stage will be discussed.

The controls for each stage of the meshing process are contained in five sub-dictionaries: geometry, castellatedMeshControls, snapControls, addLayersControls and meshQualityControls. The castellation, snap and layer addition stages can each be activated or deactivated by assigning the Boolean operator 'true' or 'false' to their respective keywords.

The geometry sub-dictionary contains a list of one or more STL or OBJ files that form the model to be meshed. Primitive shapes (such as boxes, cylinders and spheres) can also be defined; these are used to refine specific regions of the mesh.

The castellatedMeshControls sub-dictionary contains settings for controlling the refinement of the background mesh cells that intersect individual surfaces specified in the geometry sub-dictionary. Cells that intersect surfaces are split according to a defined minimum and maximum refinement level; the former is applied generally across the surface while the latter is applied to cells that observe intersections that form an angle greater than that specified by the user. A refinement level of 0 produces no cell refinement, while values 1 and 2 result in a cell being split into 8 and 64 cells respectively. Edges of the STL file(s) (extracted using the surfaceFeatureExtract utility) and their refinement level can also be defined in a features sub-dictionary; feature edges are used during the snapping stage to assist in conforming the background mesh more accurately to sharper angles. Volume refinement may also be applied using a refinementRegion sub-dictionary, in which cells are divided if they are deemed to be either inside, outside or within a specified distance of a surface or volume (such as a closed STL or a primitive shape). The number of cells between regions with different levels of refinement is controlled using the nCellsBetweenLevels keyword. Once refinement of the background mesh has been completed, cells with 50% of their volume lying outside of the region formed by the STL (specified by a set of coordinates) are removed. Thus, for internal flow problems, cells inside a closed STL should be kept (Greenshields, 2017f).

The snapControls sub-dictionary contains the settings used to control the modification and morphing of the castellated mesh to the geometry. The mesh is first smoothed on the patches and in the volume, followed by displacement of the castellated boundary cell vertices onto the surface of the STL. Only those vertices within a certain distance of the STL are snapped to the surface, controlled using the tolerance keyword entry (typically set to 2) (OpenCFD, 2017a). An nSmoothPatch keyword defines the number of patch smoothing operations and is typically set to 3 (Greenshields, 2017f). After displacement of the boundary cell vertices, the internal mesh is adjusted (or 'relaxed') according to the number of iterations specified using the nSolveIter keyword. Cell vertices that are then found to invalidate the required mesh quality (specified in the meshQualityControls sub-dictionary) are scaled back until the constraints are satisfied or the maximum number of iterations is reached. The latter is defined using the nRelaxIter keyword. (Engys, 2012; OpenCFD, 2017)

After the snapping stage is complete, boundary layers may be inserted on patches specified by the user in the layers sub-dictionary under addLayersControls. The number and total thickness of the layers on each patch can be set, either relative to the size of the undistorted cells on the boundaries or as an absolute value. The mesh is then shrunk back from the surface, relaxed, and the specified number of layers (consisting of hexahedral cells) inserted. If at any stage the mesh quality constraints are invalidated, the layers are removed, the displacement is scaled back, and the process repeated (Greenshields, 2017f). The addLayersControls sub-dictionary contains a substantial number of control parameters, which will not be discussed.

The last sub-dictionary, meshQualityControls, defines upper or lower bounds for mesh quality metrics that the grid generated by snappyHexMesh must satisfy. If a mesh quality metric is invalidated during the meshing process, the iteration is undone and the mesh restored to an error free state. Mesh quality controls include (but are not limited to) maximum non-orthogonality, maximum boundary face skewness, maximum internal face skewness and minimum cell volume ratio (Engys, 2012).

4. Mesh generation and VOF OpenFOAM® case set-up

The following sub-sections detail the processes used to generate the grids for the model rectangular sedimentation tank and the HDVS Swirl-Flo® devices. A 2D structured hexahedral mesh of the rectangular tank was created using blockMesh due to its relatively simple geometry. This was then used to simulate and compare the free surface heights for the same experimental conditions recorded by Wye (2018). The results from the 2D VOF simulations (performed using the solver interFoam) were then used to influence the geometry and mesh created for the 3D rectangular tank. This is discussed in more detail in Section 4.1.2.5. VOF 3D simulations were also performed for the Swirl-Flo® geometry. The final meshes produced for both the rectangular tank and Swirl-Flo® (including relevant mesh quality metrics) are given in Sections 4.1.7 and 4.2.1 respectively.

4.1 Rectangular sedimentation tank

Initial experiments were performed using an Armfield W7 model sedimentation tank by Baker (2018), Mendoza (2018) and Wye (2018). However, upon the request of the project client, the tank geometry was modified by raising the base and including a hopper, reducing the depth of the tank. The final 3D model of the final experimental geometry used in the CFD simulations is shown in Figure 4.

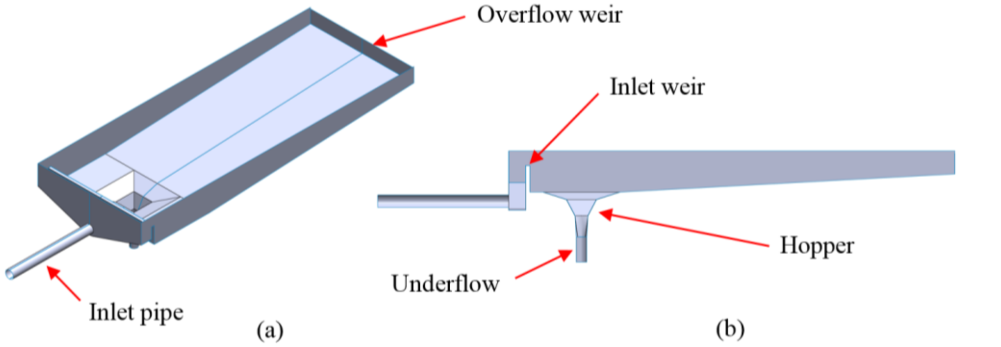


Figure 4: 3D SOLIDWORKS® model of the final rectangular sedimentation tank. (a) Isometric view (b) Section view.

The dimensions of the original Armfield tank were used to create the 2D rectangular tank mesh as it was made prior to the modifications. Thus, the 2D mesh presented in this report is not representative of the final geometry used during the experiments. However, it is believed that the modifications had no effect on the free surface heights measured over the overflow weir for a given flow rate and so remains relevant to this report.

4.1.1 2D hexahedral structured mesh

In OpenFOAM®, it is not possible to perform simulations on purely 2D grids as it is a 3D polyhedral CFD code. However, by creating a mesh with a width of one cell, and by assigning special empty boundary conditions, 2D cases can be simulated (Greenshields, 2017d). Thus, to reduce the computational time required for the VOF simulations, a 2D structured conformal hexahedral mesh was created of the Armfield W7 model sedimentation tank. This was accomplished using the OpenFOAM® `blockMesh` utility.

To generate grids using `blockMesh`, an OpenFOAM® dictionary called `blockMeshDict` is required. This contains the relevant keywords and sub-dictionaries that specify the number of blocks, their connectivity, and the number of cells in the X, Y and Z directions. The utility works by decomposing a geometry as one or more hexahedral blocks, each defined by eight vertices. Edges can either be straight or curved, with the latter requiring the type (for example, an arc, spline or polyline) and appropriate interpolation points to be specified (Greenshields, 2017e; Wolf Dynamics, 2017b).

The block structure of the 2D mesh generated is shown in Figure 5. The multi-block structure and associated dimensions were based on a section cut taken at the mid-plane along the tank's length, and included both a clean water and slurry inlet. The overflow weir was also incorporated so that changes in free surface height over its length could be determined in the

VOF simulations (see section 4.1.2). Table 1 gives the dimensions of each block that form the multi-block structure.

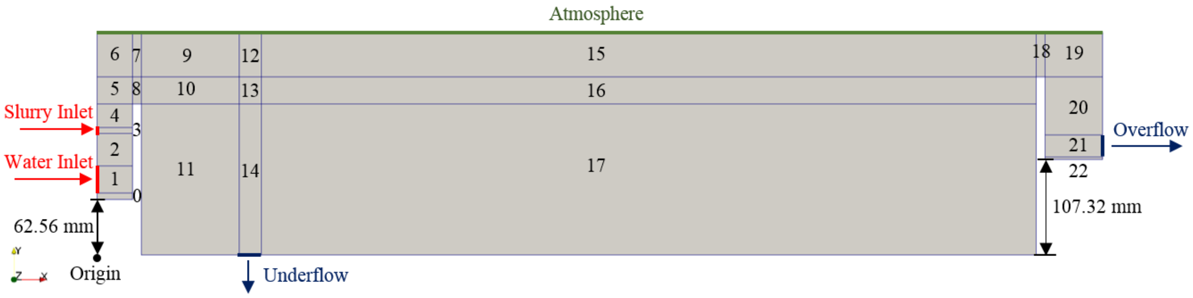


Figure 5: Block structure of 2D hexahedral mesh of original rectangular tank.

Table 1: Dimensions of each block.

Block	x (mm)	y (mm)	Block	x (mm)	y (mm)	Block	x (mm)	y (mm)	Block	x (mm)	y (mm)
0	40	7.44	6	40	50	12	25	50	18	10	50
1	40	30	7	10	50	13	25	30	19	65	50
2	40	36.5	8	10	30	14	25	170	20	65	65
3	40	7	9	109.65	50	15	870.35	50	21	65	25
4	40	26.5	10	109.65	30	16	870.35	30	22	65	2.68
5	40	30	11	109.65	170	17	870.35	170			

It can be seen from Figure 5 that, for a conformal mesh to be generated, blocks that form columns require the same number of cells in the X direction. Similarly, blocks that form rows must have the same number of cells in the Y direction. As this was a mesh for 2D simulations, all blocks had a width of 1 mm and one cell in the Z direction.

As `blockMesh` allows complete control of the cell spacing in each block, it was possible to generate cells with the ideal aspect ratios of 1. For 2D cases, an aspect ratio of 1 is achieved when the cell spacing in the X direction, Δx, is equal to the cell spacing in the Y direction, Δy. Thus, by specifying that each block have the same cell spacing, the number of cells in the X (N_x) and Y direction (N_y) of each block could be calculated by dividing their corresponding length, X, and height, Y, by the required cell width, Δx. Depending on Δx, this may produce blocks containing a rational number of cells. To prevent this, N_x and N_y of each block were rounded to the nearest whole number. An Excel spreadsheet was utilised in which Δx could be specified and N_x and N_y calculated for each block. Rather than manually changing the number of cells in the X and Y direction for each of the twenty-three blocks in the `blockMesh` dictionary, macro syntax was employed so that only N_x and N_y of the first blocks of each column and row in Figure 5 needed to be specified. A total of 84 vertices and their corresponding coordinates had to be defined in order to generate the multi-block structure illustrated in Figure 5.

4.1.2 2D VOF simulations

To simulate the free surface elevation for the same experimental conditions, the solver `interFoam` was used. The case set-up is described in the following sub-sections.

4.1.2.1 Mesh density

The multi-block structure illustrated in Figure 5 contained 825,016 cells with blocks 11, 14 and 17 having a Δx and Δy equal to 1 mm. To accurately capture the interface region, cells in the top multi-block row structure were further refined to have a Δy of 0.1 mm, giving a maximum acceptable aspect ratio of 10. To prevent a large increase in Δy between the two regions, cell grading was utilised in the middle multi-block structure (row beginning with block 5). This is illustrated in Figure 6.

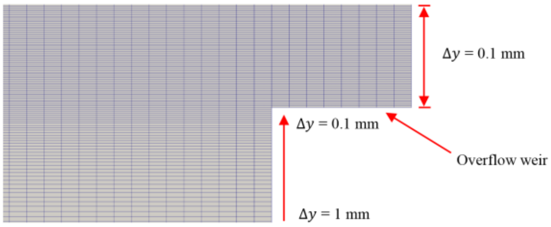


Figure 6: 825,016 cell mesh with cell grading.

4.1.2.2 Boundary conditions

Unless highlighted in Figure 5, all block faces that exist on a boundary were classified as walls. The boundary conditions used and their respective value for each patch are given in Table 2.

Table 2: Patch types and their respective boundary conditions used in the 2D VOF simulation.

		alpha.water		p_rgh		U	
Boundary Name	**Patch Type**	**Applied Boundary Conditions**	**Value**	**Applied Boundary Conditions**	**Value (kg.m^{-1}.s^{-2})**	**Applied Boundary Conditions**	**Value (m.s^{-1})**
atmosphere	patch	inletOutlet	0	totalPressure	p0 = 0	inletOutlet	0
waterInlet	patch	fixedValue	1	fixedFluxPressure	0	fixedValue	0.034
slurryInlet	patch	fixedValue	1	fixedFluxPressure	0	fixedValue	0.012
walls	wall	zeroGradient	N/A	fixedFluxPressure	0	fixedValue	0
underflow	patch	zeroGradient	N/A	fixedFluxPressure	0	fixedValue	0.002
overflow	patch	zeroGradient	N/A	totalPressure	p0 = 0	zeroGradient	N/A
frontAndBack	empty	empty	N/A	empty	N/A	empty	N/A

For this case, surface tension effects at the walls were ignored by setting the volume fraction boundary condition (alpha.water) to *zeroGradient* (Greenshields, 2017a). Both the water inlet and slurry inlet were specified as injecting water into the domain by using a *fixedValue* condition of 1 for *alpha.water*. The no-slip condition was achieved at wall boundaries by specifying a fixed value of 0 m/s for the velocity (U). The dynamic pressure (p_rgh) was used rather than static pressure p as it made implementation of the boundary conditions easier.

Furthermore, using p_rgh ensures a more accurate calculation of the balance between pressure and buoyancy forces on non-orthogonal and skewed meshes (though this was not a problem in this case) (Queguineur, et al., 2015). Atmospheric pressure was specified at the atmosphere and overflow patches using the *totalPressure* boundary condition, while *fixedFluxPressure* was used at the wall boundaries. The latter is used to ensure that the boundary flux matches the velocity boundary condition by adjusting the pressure gradient for solvers that include body forces (such as `interFoam`) Greenshields (2017a). The front and back planes of the mesh were specified as having the empty type boundary condition as no solution was required in the 3rd dimension. For the atmosphere patch, the *inletOutlet* boundary condition was used to apply a *zeroGradient* condition when flow leaves the domain and a *fixedValue* of 0 m/s when flow enters to prevent backflow.

4.1.2.3 Initial conditions and fluid properties

To ensure that the 2D simulations were comparable to the 3D tank experiments, the inlet velocities of the fluid were scaled based on the theoretical residence time, $T_R = V/Q$ (where V is the volume of the tank and Q is the inlet flow rate). By equating T_R of the 2D and 3D tank, using the relationship $Q = UA$ (where U is velocity and A is the inlet area) and rearranging, the equivalent inlet velocity for the 2D case can be given by:

$$U_{2D} = \frac{V_{2D}}{V_{3D}} \cdot \frac{Q_{3D}}{Y_i \cdot W} \tag{7}$$

where Y_i is the inlet height and W is the inlet width for the 2D mesh.

Given the width of the 3D tank (0.4 m), the dimensions of blocks 1, 2 and 3 in Table 1, and the measured experimental flow rates of the clean water (4.13E-04 m^3/s), slurry inlet (3.33E-05 m^3/s) and underflow (2.17E-05 m^3/s), the equivalent velocities for the 2D case were calculated to be 0.034 m/s, 0.012 m/s and 0.002 m/s respectively. The properties of the fluids were set to correspond with the experimental conditions for atmospheric pressure and a temperature of 14.5 °C.

To reduce the time required to perform the simulation, the `setFields` utility was employed to set cells below the overflow weir as containing water ($\alpha = 1$). This meant that a stable free surface height could be achieved more quickly as it was not necessary to simulate the complete filling of the tank.

4.1.2.4 Time step control and discretisation schemes (solver set-up)

To verify whether the flow was laminar or turbulent for the 2D case, the Reynolds number was calculated using the hydraulic diameter for a closed rectangular channel. For the 2D inlet and outlet velocities calculated in Section 4.1.2.3, the maximum Reynolds number was calculated to be 58 for the 2D water inlet. Thus, the 2D case was simulated for laminar flow.

To produce good accuracy, the convection term $\nabla \cdot (\overline{\boldsymbol{u}}\overline{\boldsymbol{u}})$ in Eq. (2) (momentum equation) was discretised using the `Gauss linearUpwind grad(U)` scheme (Greenshields, 2017a) as it is second order accurate, but less unbounded than the `linear` (central differencing) scheme (Greenshields, 2017c). The convection term $\nabla \cdot (\alpha\overline{\boldsymbol{u}})$ was discretised using the `vanLeer` scheme, as it 'ensures that the calculated values of the volume fraction are limited to $0 \leq \alpha \leq 1$' (Queguineur, et al., 2015, p. 215). The compression term $\nabla \cdot (U^r\alpha)$ was discretised using the `interfaceCompression` scheme to produce a reasonably sharp interface resolution (Queguineur, et al., 2015). The second order accurate `linear` differencing scheme was used to discretise the gradient terms, while the first order `Euler` differencing scheme was employed to discretise the first time derivative.

To determine when the free surface height reached a steady-state, a preliminary simulation was performed on a coarser mesh of 146,000 cells to reduce computational time. To ensure stability, the `adjustTimeStep` option was enabled so that the simulation did not exceed a maximum set Courant number of 1. It was discovered that the free surface height became stable after 30 seconds. Thus, the simulation was set to run for 30 seconds on the higher resolution mesh detailed in Section 4.1.2.1. The simulation was performed in parallel and required 48 hours to complete on 14 Intel® Xeon® E5-2637 3.50 GHz cores.

4.1.2.5 Results and discussion

As the free surface height results from the 2D VOF simulation influenced the 3D mesh generation of the rectangular tank, they have been discussed here. Free surface heights were taken at $t = 30$ seconds using the `sample` utility, corresponding to the experimental water height measurement locations illustrated in Figure 7.

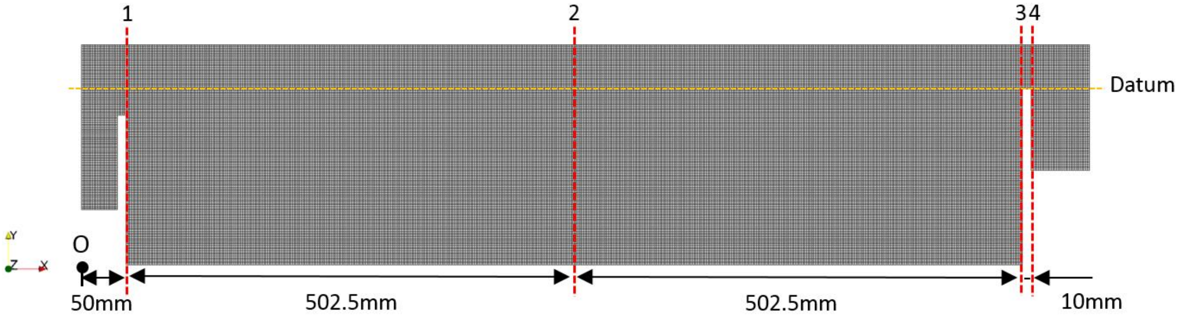


Figure 7: Free surface height measurement locations.

Figure 8 shows a graph of the free surface heights plotted at $t = 30$ seconds along the measurement lines illustrated in Figure 7.

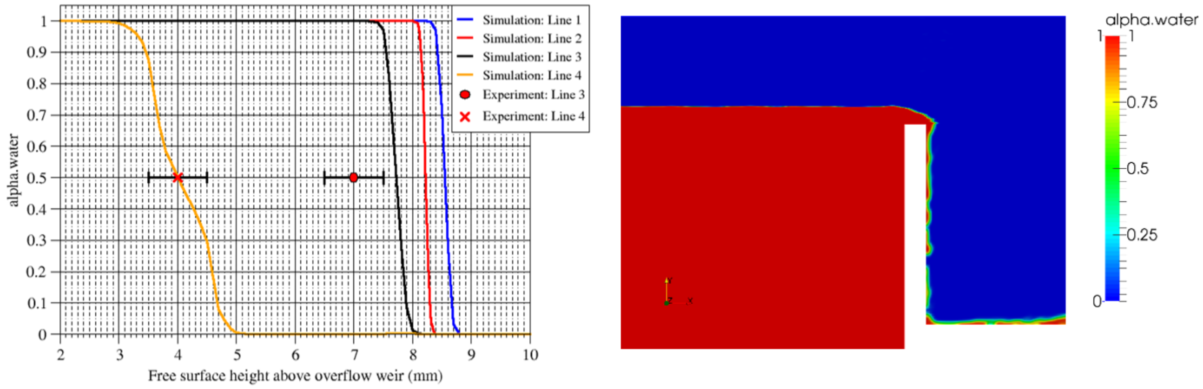


Figure 8: Left: Graph of experimental and simulation free surface heights above the overflow weir at $t = 30$ seconds. Right: Contour plot of alpha.water at $t = 30$ seconds.

Taking the free surface height to be at $\alpha = 0.5$ (Queguineur, et al., 2015), the free surface height above the overflow weir in the simulation was 8.55 mm, 8.2 mm, 7.7 mm and 4 mm for measurement lines 1, 2, 3 and 4 respectively. The smallest and largest interface region thickness of 0.4 mm and 2.2 mm was observed for measurement lines 2 and 4 respectively. The larger interface thickness for line 4 is due to the free surface not being completely horizontal; this is because the water is falling over the edge of the overflow weir, causing the interface region to smear across more cells.

Experimental measurements corresponding to lines 3 and 4 are also plotted with free surface heights of 7 mm and 4 mm respectively. Despite the smearing of the interface, there is very good agreement between the simulation and experimental results for measurement line 4 (at $\alpha = 0.5$ the simulation has the same result as that of the experiment). However, there is more discrepancy between the simulation and experimental results for measurement line 3, despite the interface region being only 0.8 mm thick. For $\alpha = 0.5$, an error of 10% between the results is observed. While this is still reasonably accurate, the error can be as large as 18.5% when considering the experimental error.

Both the simulation and experimental results indicate a change in free surface elevation along the length of the tank. According to the latter, the largest free surface height (8.55 mm) occurs along line 1. This is due to the presence of the inlet weir wall, which directs the flow from the inlets upwards towards the free surface. However, the change in surface elevation between lines 1 and 3 is minimal, differing only by 0.85 mm. Thus, a horizontal slip wall 8.1 mm above the overflow weir could be deemed to accurately model the free surface, as the maximum free surface elevation error would only be 5.26%. However, a percentage error of 102.5 % would be observed between the rigid-lid height and the free surface elevation results for line 4. Thus, it was decided to cut the computational domain at the start of the overflow weir (line 3) for meshing the 3D geometries for `driftFluxFoam`.

4.1.3 3D rectangular tank mesh generation – snappyHexMesh

The following sub-sections detail the `snappyHexMesh` methodology for producing 3D meshes of the final rectangular tank geometry. A parameter study was conducted to evaluate the effect of varying individual settings on the quality and geometric accuracy of the meshes produced. This was only conducted on the rectangular tank to save time.

4.1.4 CAD modification and clean-up

The final rectangular tank geometry can be seen in Figure 4. To increase computational efficiency, it was decided to split the geometry in half, thus reducing the number of cells required (the model shown in Figure 4 is complete only for illustrative purposes).

There are several differences to the CAD model and the experimental set-up. In the latter, flexible tubing is used for both the clean water and slurry inlet, with internal diameters of 30 mm and 7 mm respectively. While the 7 mm slurry inlet was included in the 2D case, it was removed from the 3D model to improve computational efficiency and simplify the meshing process; it would otherwise require a much higher cell resolution. Furthermore, the length of the pipe was chosen to be 300 mm to allow the flow to fully develop before entering the tank. The pipe length in the experimental set-up was 180 mm (after bends) and so was deemed acceptable.

The computational domain height was also set to be 7 mm above the overflow weir. In reality, the tank extends a further 43 mm in the Y direction. However, as the free surface is not simulated in `driftFluxFoam`, the computational domain was lowered to the free surface height observed in initial experiments for typical inlet flow rates. While the 2D VOF simulation results in Section 4.1.2.5 suggest a height of 8.1 mm, it was decided to use the results from the experiments as these were the validation objective.

4.1.5 Stereolithography file preparation

The meshing utility `snappyHexMesh` requires that the geometry is provided as one or more Stereolithography (STL) files (Greenshields, 2017f). As only the internal faces were required (those that form the boundaries of the computational domain), the outside faces were removed from the SOLIDWORKS® model. Faces that make up the walls, rigid-lid, inlet and outlets were then exported separately as high resolution STL files. These were then combined together using the Linux concatenate command (`cat`) to form an STL file with specified regions. This was done so that patch types, and thus boundary conditions, could be assigned to different regions of the model. However, it was discovered that the STL files did not form a closed boundary as `snappyHexMesh` did not remove cells deemed to be outside of the geometry (i.e. both the inside and outside of the tank was meshed).

To produce closed STL surfaces, the open source CAD software SALOME (OPEN CASCADE, 2018) was used. The SOLIDWORKS® geometries were imported into the program in the STEP file format. Intersecting edges were removed using the `sewing` geometric objects repair tool so that a triangulated surface mesh on each of the model's surfaces could be generated. This was accomplished using the Netgen-2D algorithm and was refined in areas with high curvature (for example, the inlet) to accurately capture the topology of the model. Sub-meshes between regions with shared edges were also created to ensure that the overall surface mesh formed a closed boundary. Each surface mesh was then exported as a separate STL file and concatenated. This formed a closed STL file which could be used by `snappyHexMesh`, and was verified as being watertight using the OpenFOAM® `surfaceCheck` utility.

4.1.6 snappyHexMesh parameter study

Using the rectangular tank model, a parameter study was undertaken to evaluate how settings in `snappyHexMesh` affected the quality and geometric accuracy of the grids produced. To reduce the number of independent variables, the addition of boundary layers was deactivated during the study. The OpenFOAM® `checkMesh` utility and the mesh quality report outputted from `snappyHexMesh` were used to evaluate mesh quality metrics; the former reported the maximum and average values for aspect ratio, non-orthogonality and cell face skewness while the latter specified how many cells had invalidated the set mesh quality constraints. To provide a quantitative measure of the meshing accuracy, a custom utility written by Tabor and Fabritius (2016) was used to determine the minimum, average and maximum distance of the mesh boundary to the surface of the STL file.

A control case with default settings for the `castellatedMeshControls`, `snapControls` and `meshQualityControls` sub-dictionaries was used, with no refinement applied to surfaces or feature edges. The mesh was adapted from a background grid with 1,000,890 cells (cell spacing of 4.1 mm). To reduce the time required to generate the meshes, `snappyHexMesh` was run in parallel across 8 Intel® Xeon® E5-2637 3.50 GHz CPU cores. The utility was also executed so that a mesh from each stage would be produced, facilitating trouble shooting. The OpenFOAM® command `reconstructParMesh –constant` was used to reconstruct the meshes after parallel execution for mesh quality evaluation and visualisation. The open source software `paraview` was used to display the meshes, with polyhedral decomposition disabled to prevent polyhedral cell faces from being triangulated.

Over 90 meshes using different settings were generated with the maximum aspect ratio, maximum and average non-orthogonality, maximum face skewness and the maximum distance of the mesh boundary to the surface of the STL file evaluated. Unless otherwise stated, the cases used the documentation default values for `nCellsBetweenLevels` (3), nSmoothPatch (3), tolerance (2.0), nSolveIter (30), nRelaxIter (5) and nFeatureSnapIter (10) (Greenshields, 2017f).

The key of the cases is as follows: *SR*, *FR* and *V* denotes the maximum and minimum level of surface, feature edge and closed STL volume refinement respectively with the associated integer being the refinement level applied; *BD1*, *BD2*, and *BD3* cases have a background mesh density of 2,023,000, 4,009,280 and 8,016,875 cells respectively; *nCBL* denotes the value of nCellsBetweenLevels used; and *Q* denotes changes to the default quality constraints set (values displayed in Table 3). These were selected based on some of the maximum and minimum values employed in Tabor & Fabritius (2016, p. 431).

Table 3: Mesh quality constraint settings used for control, SR1.Q1, SR1.Q2 and SR1.Q3 cases.

Mesh Quality Constraint	Control	SR1.Q1	SR1.Q2	SR1.Q3
Max allowable non-orthogonality (°)	65	65	65	65
Max allowable boundary skewness	20	2.4	1.1	2.4
Max allowable internal skewness	4	1	1	0.5

4.1.6.1 snappyHexMesh parameter study results and discussion

The results from the snappyHexMesh parameter study are shown in Figure 9 and were used to influence the settings defined for generating the final rectangular tank and Swirl-Flo® grids. Only 23 out of 90 cases have been shown due to the large quantity of data.

It can be seen in Figure 9 that applying a surface, feature edge and volume refinement level of 1 improved the maximum aspect ratio (AR) and maximum non-orthogonality when compared to the *control* case and greater refinement levels. For example, a maximum AR of 5.6 was observed for *SR1*, compared to ARs of 7.4 and 8.1 obtained for the *control* case, *SR2* and above refinement levels respectively.

However, increasing the surface, features and volume discretisation was found to increase the face skewness from 2.2 (for the *control* case) to 2.5. This increase was deemed to be more than acceptable when considering the large reduction in the maximum distance of the mesh boundary to the surface of the STL file. For example, a 90% decrease in the maximum distance of the mesh boundary to the STL surface was observed when comparing *SR1* and *V1* to the *control* case. Applying further refinement did not improve the result, suggesting that snapping accuracy cannot be continuously increased using surface and volume refinement. Thus, it is better to use a lower refinement level for this geometry, as less cells are needed to achieve the same result (the *SR1* mesh had 690,179 cells, 70.8% less than that for the *SR2* case). Using refinement levels greater than 1 for *FR* only decreased the maximum distance by 5% when compared to the *control* case. This indicates that edge refinement is limited when improving snapping accuracy in simple geometries (i.e. the rectangular tank) as once the edges are refined, the maximum boundary displacement will occur on the surface.

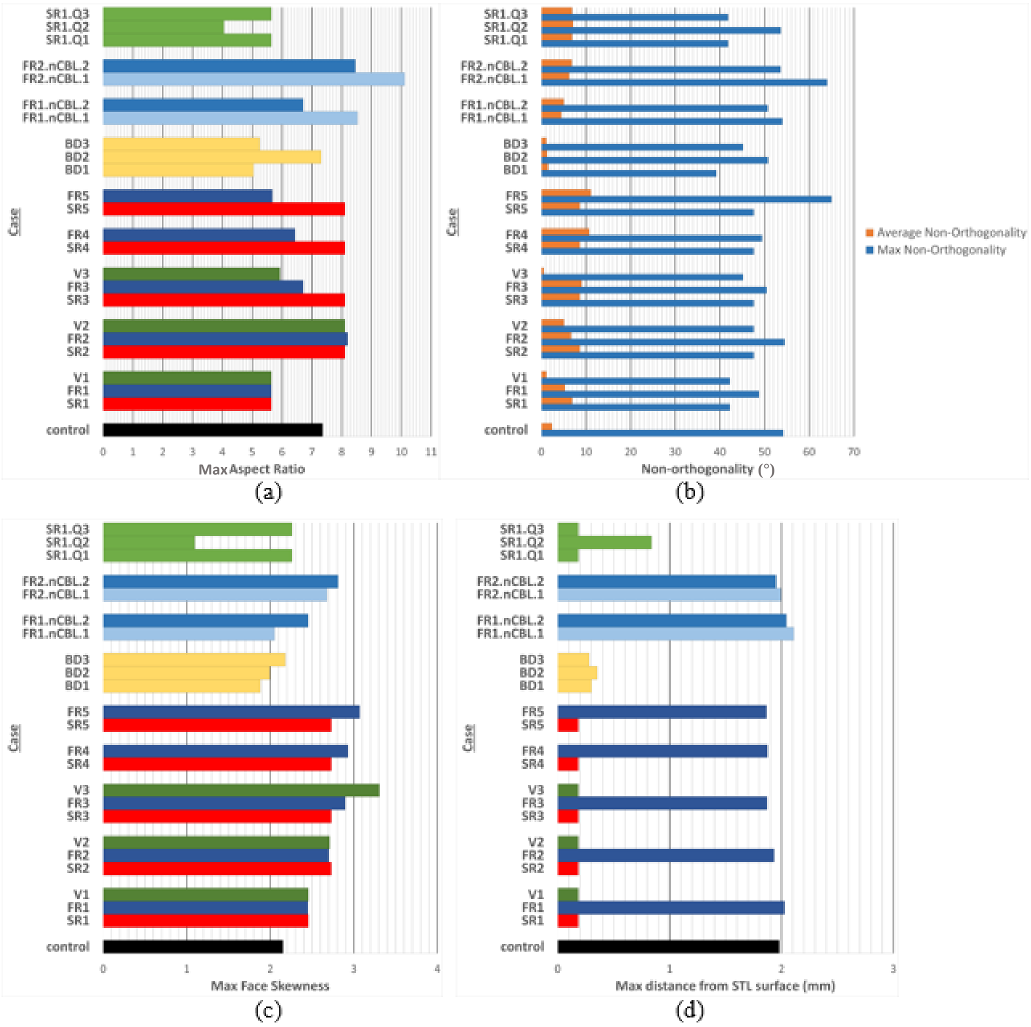


Figure 9: Mesh quality metric results for (a) Maximum aspect ratio; (b) Non-orthogonality; (c) Maximum face skewness; (d) Maximum distance of the mesh boundary to the STL surface.

Using a higher resolution background mesh was found to also improve maximum AR, non-orthogonality, skewness and mesh boundary distance to the STL surface. It appears that the optimum background mesh density was 2 million cells (*BD1*) for the cases tested as it had the lowest values for maximum AR (5.0) and maximum non-orthogonality (39.2°), with similar values for average non-orthogonality (1.5°) and snapping accuracy to the other *BD* cases. Thus, *BD1* was used when creating the final meshes of the rectangular tank, as it facilitated producing higher quality meshes than the *control* case while using significantly fewer cells than the meshes produced for the *BD2* and *BD3* cases respectively.

Lowering the value of `nCellsBetweenLevels` from the default value of 3 was found to increase the maximum AR and maximum non-orthogonality for the *FR1* and *FR2* cases. The only improvement was observed for maximum skewness for the *FR1.nCBL1* case (though only by 0.4 when compared to the skewness value of 2.4 for *FR1*). However, while this would suggest leaving it at its default value of 3 would be most suitable, its effect on the mesh was dependent on the other control settings used. For example, the only difference observed in mesh quality metrics for *SR2.nCBL1* (not displayed in Figure 9) was an increase in average non-orthogonality and skewness by 14.7% and 0.2% respectively. However, reducing `nCellsBetweenLevels` to 1 decreased the number of cells by 23.3% when compared to the equivalent case using the default value of 3. Thus, this value was chosen on a case by case basis, and was lowered when a reduction in cells was required without greatly decreasing mesh quality. For meshes with large changes in cell refinement levels, this value was increased to ensure a smoother cell spacing transition between different regions of the mesh.

Maximum skewness was found to be best controlled by defining the appropriate mesh quality constraints. From the parameter study, it was found that lowering the maximum allowable boundary skewness to 1.1 reduced the maximum AR and face skewness by 44.9% and 48.9 % respectively. However, this also lowered the snapping accuracy to some degree as the maximum distance of the mesh boundary to the STL surface increased from 0.2 mm for the *SR1.Q1* and *SR1.Q3* cases to 0.8 mm. It also had the worst maximum non-orthogonality (53.7°) of the quality constraint cases despite no alteration being made to the maximum allowable non-orthogonality constraint. Thus, while lowering the maximum boundary skewness improves the skewness quality metric, it can cause mesh non-orthogonality to increase. Furthermore, decreasing the maximum allowable boundary skewness further would sometimes cause the snapping stage to fail. Because of this, it was decided that the final rectangular tank and HDVS computational grids would be generated using the mesh quality constraints defined for *Q3*, as it gave acceptably low values of maximum mesh boundary displacement while also ensuring excellent values of internal and boundary skewness. The maximum allowable non-orthogonality was left at the default of 65° as it was found to not be too restrictive on the snapping stage and also ensured excellent mesh quality.

4.1.7 Final rectangular tank mesh

The final mesh produced using `snappyHexMesh` is shown in Figure 10. A surface and edge refinement of 1 was used, as well as the background density *BD1* and quality constraints *Q3* identified in the parameter study.

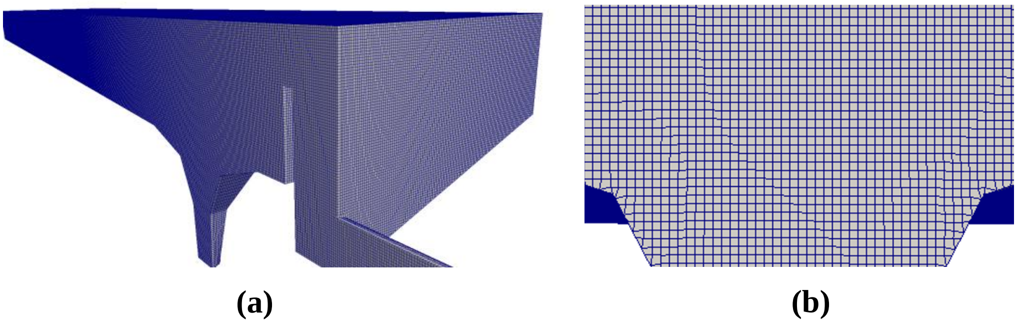


Figure 10: Final rectangular tank mesh produced (a) 3D view (b) close up of cell structure.

The utility checkMesh reported that the mesh contains a total of 1,364,254 cells, with a maximum aspect ratio, maximum non-orthogonality, average non-orthogonality and maximum skewness of 5.29, 42.8°, 6.30° and 2.40 respectively. The maximum distance between the mesh boundary and the STL surface was calculated to be 0.3 mm.

4.2 Hydrodynamic vortex separator

The following sub-sections detail the mesh generation process of the Swirl-Flo® HDVS for the solid-liquid driftFluxFoam simulations performed by Russell (2018) and the water-air interFoam simulations completed for this report.

4.2.1 Mesh generation

The same process outlined in section 4.1.5 was used to create STL files of the HDVS geometry for use with snappyHexMesh. The geometry was split into sections so that different levels of refinement and boundary conditions could be applied. These are illustrated in Figure 11.

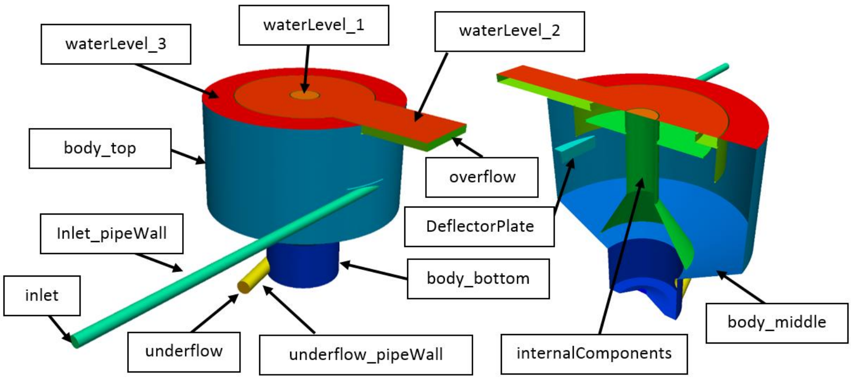


Figure 11: HDVS geometry named sections.

For the geometry to be used in driftFluxFoam simulations, the water height was set to 23 mm above the overflow tray upon the recommendation of the project client (experimental water height data was not available at the time). For the VOF simulations, the upper computational domain boundary was set to 50 mm above the overflow tray. It was believed that this would improve the simulation runtime as less cells would be required, while also ensuring a large enough gap between the water-air interface and the atmosphere boundary. However, later simulations revealed that this height should have been extended further.

The background grid was generated using the blockMesh utility. However, using a single block to encompass the geometry meant that the base mesh had an unnecessary number of cells as there was a large region of empty space surrounding the inlet pipe (see Figure 12 (a)). Thus, a multi-block mesh consisting of ten blocks was created (shown in Figure 12 (b)). For the same cell spacing, the single block mesh required 2.7 times more cells when compared to the mesh generated using the multi-block structure. Thus, the latter was used as it significantly improved meshing efficiency.

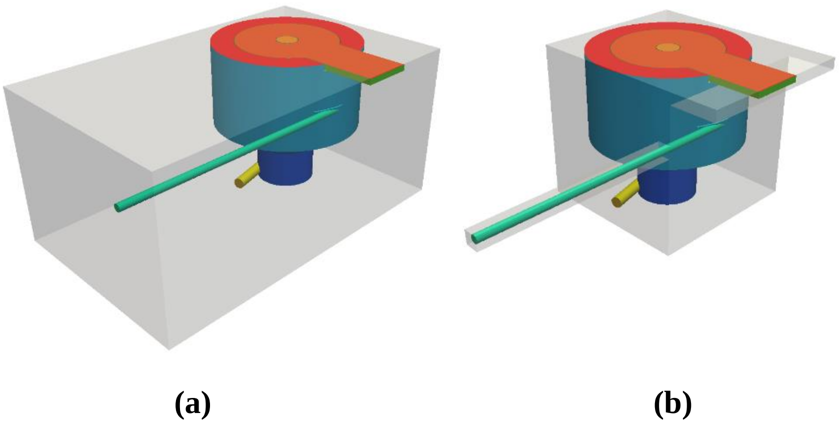


Figure 12: (a) Boundary of single block background mesh surrounding HDVS geometry. (b) Boundary of multi-block background mesh surrounding HDVS geometry.

An initial mesh of the Swirl-Flo® was produced using the process outlined in Section 3.3. Unfortunately, layer addition was unsuccessful and so this stage was deactivated in the `snappyHexMesh` dictionary. More details about this are provided in Section 4.2.1.1. The initial mesh was produced using a background grid containing 2.9 million cells, equating to a cell spacing of 4.3 mm. A surface refinement level of 2 was defined for all surfaces (except *body_middle*, *body_top* and the *waterLevel* surfaces) to ensure that the topology of the geometry was accurately captured. For the *body_middle*, *body_top* and *waterLevel* surfaces, a surface refinement level of 1 was applied to reduce the number of cells in the computational domain, while still providing sufficient snapping accuracy. The quality constraints in Table 3 for case *SR1.Q3* were used to ensure that the mesh produced was of excellent quality.

The initial mesh resulted in 4.4 million cells and is shown in Figure 13. The average and maximum distance of the mesh boundary to the surface of the STL was found to be 0.07 mm and 0.8 mm respectively. The utility `checkMesh` reported the maximum AR, maximum non-orthogonality, average non-orthogonality and maximum skewness to be 8.2, 59.9°, 11.1° and 2.4 respectively.

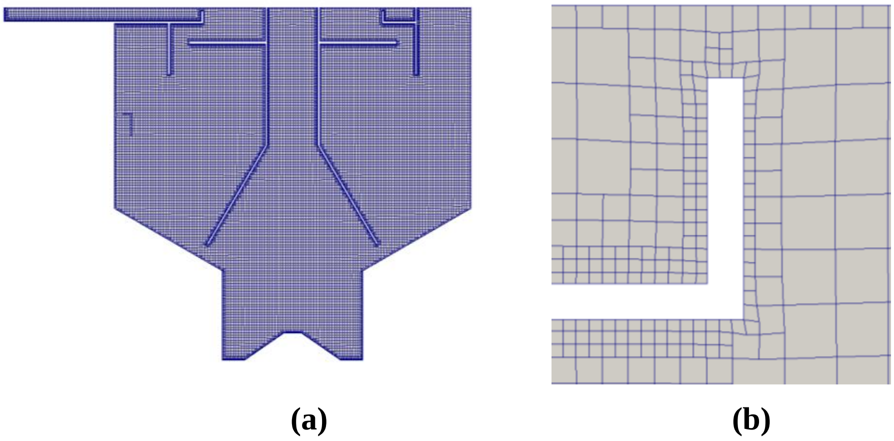


Figure 13: 4.4 million HDVS mesh. (a) section-cut view (b) Close up of cell discretisation.

However, despite the grid having favourable priori mesh quality metrics, it has several issues. Firstly, the gap between the frictionless wall and overflow weir (see Figure 13 (b)) was only discretised by 4 cells. For similar channels, Egarr et al. (2005) ensured that a minimum of six cells were present. Furthermore, initial `driftFluxFoam` simulations performed by Russell (2018) at the current mesh resolution took 19 days to complete for a simulation time of 132 seconds. Thus, to reduce the computational time required, a new mesh was generated with almost half the number of cells. This was accomplished by increasing the cell spacing of the base mesh to 8 mm. A refinement region was also applied to ensure that at least six cells were present between the channel. The Semi-Implicit Method for Pressure Linked Equations steady-state solver `simpleFoam` was also used to quickly identify regions containing high velocity gradients. To increase computational efficiency, only cells present in these regions were refined. Figure 14 shows a velocity contour plot from the `simpleFoam` simulation, taken normal to the Y axis and at the centre of the inlet pipe.

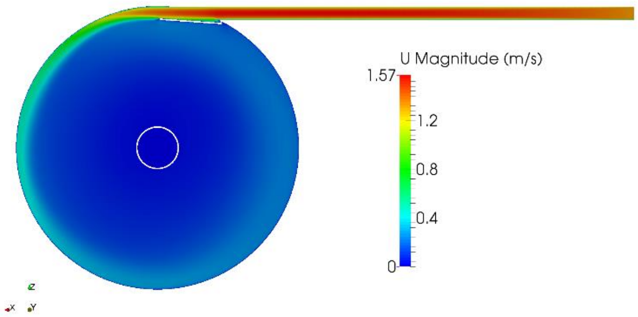


Figure 14: Velocity contour plot taken at the centre of the inlet pipe.

It can be seen that the greatest velocities occur along the outside chamber walls, roughly equivalent to the width of the deflector plate. Thus, cells within 26 mm of the STL surfaces were refined. The overflow region was also refined to ensure that there were six cells between the weir and no slip wall through which the flow passes (see Figure 15 (b)). The final mesh is shown in Figure 15.

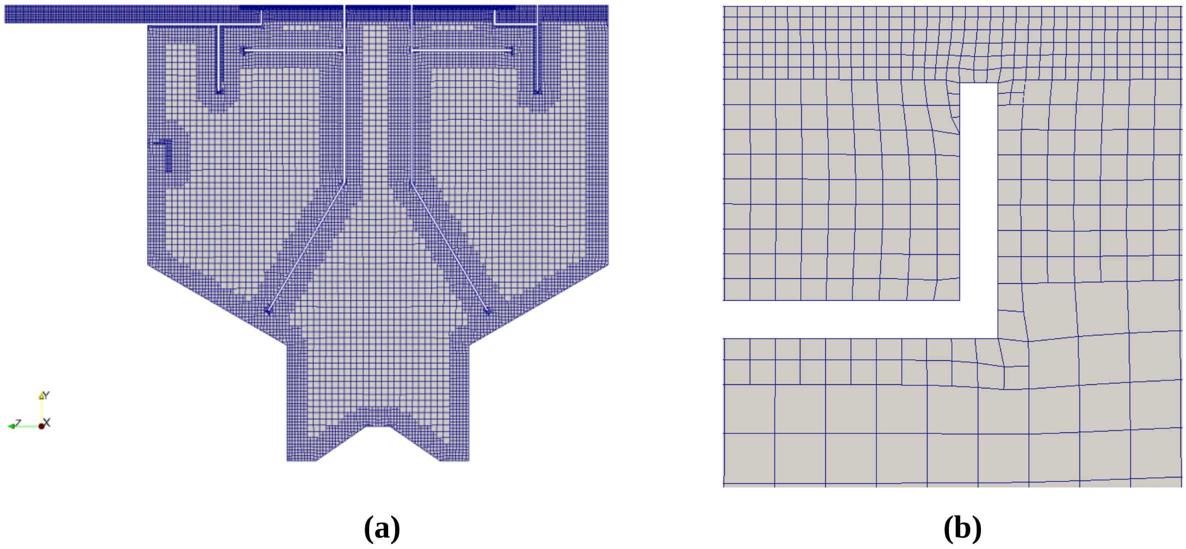


Figure 15: (a) Section-cut taken of the final rigid-lid HDVS mesh. (b) Increased refinement between wall and waterLevel slip wall.

The final rigid-lid Swirl-Flo® mesh produced had 2.5 million cells, with a maximum AR, maximum non-orthogonality, average non-orthogonality and maximum skewness of 7.8, 64.5°, 7.5° and 2.4 respectively. The average and maximum displacement of the mesh boundary from the STL surface was found to be 0.36 µm and 0.6 mm respectively.

4.2.1.1 Addition of boundary layers

Multiple attempts were made to generate boundary layers on the walls of the HDVS. However, after the stage had completed, it was found that in most cases that no layers had been inserted, and the mesh quality had significantly decreased, with several faces having a non-orthogonality greater than 75°. After much trial and error, altering the `addLayers` settings so that layers below 180° were not automatically collapsed (Engys, 2012), providing relaxed quality constraints, and specifying the generation of three layers, `snappyHexMesh` reported that 90.8% of the walls had layer coverage. The first cell height was set to 0.5 mm according the Pointwise® Y+ calculator (Pointwise Inc., 2018) for a Y+ (non-dimensional distance) value of 30 and the inlet velocity. However, upon running `checkMesh`, it was discovered that there were 35 severe non-orthogonal faces, with a maximum value of 73.5°. Even though `checkMesh` reported that the mesh was still okay, the cell count had also dramatically increased from 4.4 million to 7.8 million cells. Due to the time required to run the `driftFluxFoam` solver on the 4.4 million cell mesh, this was deemed not to be viable.

4.2.2 3D VOF HDVS simulations

4.2.2.1 Mesh generation

To capture the water-air interface region, the rigid-lid geometry was extended to 50 mm above the overflow tray. The subsequent mesh generated used the same refinement regions as the final rigid-lid Swirl-Flo® grid illustrated in Figure 15, but employed a background mesh with a greater cell spacing (10 mm) to reduce the number of cells and the computational time required. The resulting mesh had 1.6 million cells, with those above the overflow tray refined to have a cell spacing of 2.5 mm; this was done so that the water-air interface region would be captured more accurately. Increasing the refinement level would reduce the mesh step size in the overflow region to 1.25 mm, but inflate the overall cell count to 8.7 million cells. To reduce computational time, it was decided to perform the `interFoam` simulations to steady-state using the 1.6 million cell mesh, then transfer the solution to the higher resolution grid using the OpenFOAM® `mapFields` utility.

4.2.2.2 Simulation case set-up

The inlet and underflow flow rates for the free surface elevation heights recorded by Wye (2018) were 0.785 L/s and 0.1 L/s respectively. The water temperature was measured to be 19°C and the corresponding water and air properties at atmospheric pressure used in the simulations.

For an inlet diameter (D_i) of 28 mm, the Reynolds number was calculated to be 34,751, using the equation $Re = 4Q/(\upsilon\pi D_i)$ (Queguineur, et al., 2015, p. 215). Thus, turbulence was modelled for the Swirl-Flo® simulations using the realisable k-ϵ model. Initial values for k and ε were calculated using Eq.s (8), (9) and (10). The turbulence intensity, I, was calculated using the relationship $I = 0.16 \times Re^{-1/8}$ (Queguineur, et al., 2015, p. 215).

$$k = \frac{3}{2}(UI)^2 \qquad (8) \qquad \varepsilon = C_\mu{}^{3/4}\frac{k^{3/2}}{l} \qquad (9) \qquad l = 0.07D_i \qquad (10)$$

The initial values of k and ε were calculated to be 0.00457 m^2/s^2 and 0.0259 m^2/s^3 respectively. The same time step control and discretisation settings detailed in Section 4.1.2.4 were used, with the additional k and ε terms being discretised using the first-order `upwind` scheme.

The boundary condition types in this case were similar to those detailed in Section 4.1.2.2. Notable exceptions include the addition of the turbulent intensity k and turbulent dissipation rate ε terms - these were modelled at the domain walls using standard wall functions. An outflow of 0.833 m/s was also applied at the overflow as it was discovered in initial simulations that the computational domain completely filled with water when using the `zeroGradient` boundary condition. This was calculated based on a measured water outflow of 0.685 L/s and an effective outflow area 776E-06 m^3.

As in the 2D rectangular tank case, the OpenFOAM® `setFields` utility was used to specify the cells below the overflow tray as containing water to reduce the time required to reach stable free-surface heights. The required simulation time was calculated based on the theoretical residence time of the Swirl-Flo®, $T_R = V/Q$ (Alkhaddar, et al., 2002). Using the material properties tool in SOLIDWORKS®, the volume of the Swirl-Flo® geometry was determined to be 0.114 m^3. Thus, for an inlet flow rate of 0.785 L/s, the theoretical residence time of the Swirl-Flo® was calculated to be 2.4 minutes. However, as the flow paths within a HDVS are helical, a simulation time of 5 minutes was chosen to ensure that the equivalent volume of fluid passed through the computational domain, and to allow transient effects to dissipate.

To reduce computational time, the simulation was performed in parallel using 28 Intel® Xeon® E5-2680 2.40 GHz CPU cores. The mesh and fields were distributed using the decomposition method `scotch` as this required no geometric input and minimised the number of processor boundaries (Greenshields, 2017b). The `renumberMesh` utility was also used to reduce bandwidth and speed up the computation on the generated mesh (Wolf Dynamics, 2017c).

4.2.2.3 Unsuccessful simulations

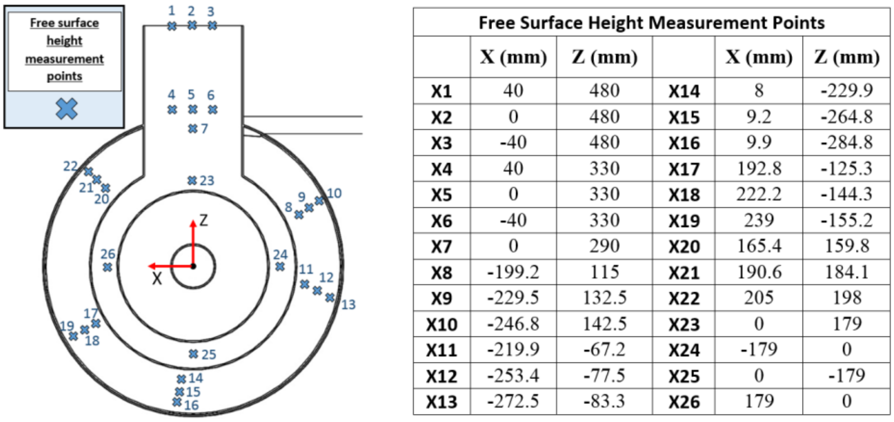
For the 2D case simulation, surface tension effects between the walls and interface were not modelled. However, as several free surface height measurement points were in the vicinity of the walls in the 3D HDVS model, it was decided that a simulation would be performed in which these affects were accounted for. The volume fraction boundary condition type was changed from `zeroGradient` to `dynamicAlphaContactAngle`, which required values for advancing, receding and static contact angles (CAs) made by the water with the walls. As the Swirl-Flo® is formed from the material stainless steel grade 316 or 304 (Jarman 2018, pers.comm., 19 April), static and dynamic CAs determined experimentally by Románszki, et al. (2013) of stainless steel 304 samples were used. These were 56°, 100° and 36° for static, advancing and receding water-steel CAs respectively (Románszki, et al., 2013, p. 2).

However, it was discovered that the obtained solution displayed unphysical behaviour, with water present in the atmosphere region above the interface and large air pockets trapped in the main chamber of the Swirl-Flo®. Furthermore, the simulation took 1.5 times longer to complete for the same simulation time. Because of this and project time constraints, it was decided that surface tension effects at the walls would not be simulated.

As mentioned in Section 4.2.2.1, it was decided that the `mapFields` utility would be used to transfer the steady-state solution obtained using the 1.6 million cell mesh to a grid with a smaller cell step spacing above the overflow tray so that the interface region could be more accurately captured. However, after transferring the solution to the higher resolution mesh and attempting to continue the simulation, a floating-point exception error was encountered. This was due to the `mapFields` utility assigning a value of 0 m^2/s^3 to some cells in the mesh for ε, leading to a divide-by-zero error in the turbulent viscosity equation (see Eq. (4)). Thus, the simulation was continued for one second for a laminar flow regime. However, upon examining the solution, it was discovered that the free surface profile had been affected, with ripples forming due to increased motion of the free surface. As a result, at some sampled locations the interface region was larger than that found in the turbulent simulation, despite using a higher cell resolution. Thus, the results obtained for the coarser mesh have been used instead.

4.2.2.4 HDVS VOF simulation results

A total of 26 free surface measurement points were taken during the HDVS experiments. The positions of these are illustrated in Figure 16.



Free Surface Height Measurement Points					
	X (mm)	Z (mm)		X (mm)	Z (mm)
X1	40	480	X14	8	-229.9
X2	0	480	X15	9.2	-264.8
X3	-40	480	X16	9.9	-284.8
X4	40	330	X17	192.8	-125.3
X5	0	330	X18	222.2	-144.3
X6	-40	330	X19	239	-155.2
X7	0	290	X20	165.4	159.8
X8	-199.2	115	X21	190.6	184.1
X9	-229.5	132.5	X22	205	198
X10	-246.8	142.5	X23	0	179
X11	-219.9	-67.2	X24	-179	0
X12	-253.4	-77.5	X25	0	-179
X13	-272.5	-83.3	X26	179	0

Figure 16: Illustration of free surface height measurement points (not to scale).

To verify that the free surface water height had stabilised after 5 minutes of simulation time, a measurement location (X8) was chosen and the volume fraction α integrated along a line parallel to the Y axis from the base of the overflow tray to the atmosphere patch. Figure 17 shows the integrated values of α plotted over time. It can be seen that the integrated values (and thus the free surface elevation) reach a steady state at approximately 120 seconds.

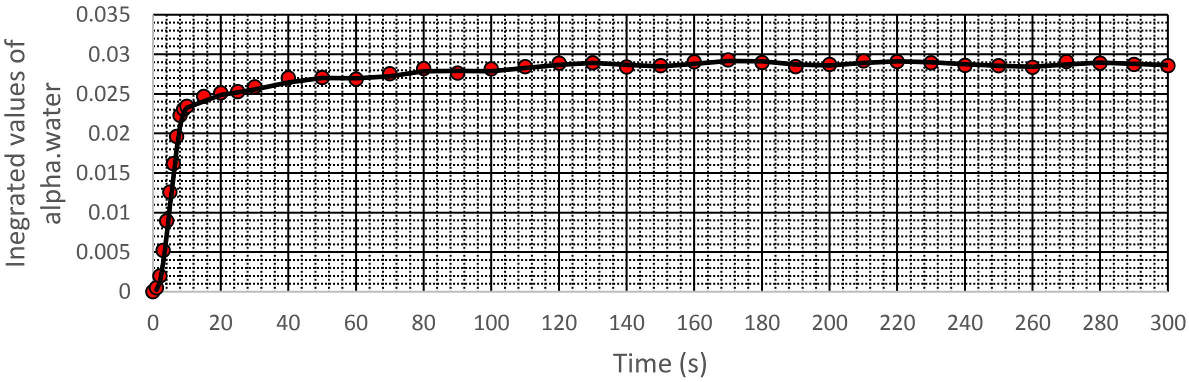


Figure 17: Plot of integrated values of alpha.water along a line parallel to Y axis at measurement point X8 against time.

Figure 18 shows a volume fraction contour (a) and a plot of the volume fraction against the distance above the overflow tray (b) for several measurement points. For clarity, all simulation and experimental results are given in Table 4. The free surface height was taken to be at $\alpha = 0.5$ for the computational results (Queguineur, et al., 2015).

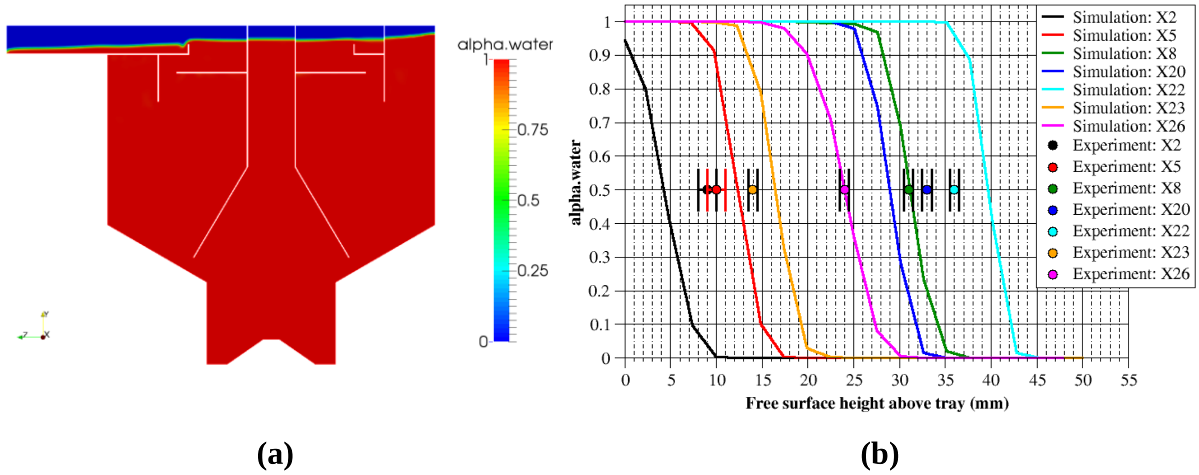


Figure 18: (a) Phase fraction contour section-cut taken at centre of HDVS normal to X axis at $t = 300\ s$. (b) Phase fraction against the distance from the base of the overflow tray at $t = 300\ s$.

The average free surface height obtained from the experiments and simulation for points X1 to X26 was 23.3 mm and 24.8 mm respectively. This would suggest that the rigid-lid height of 23 mm used in the `driftFluxFoam` simulations was a reasonable assumption. However, the standard deviation of the experiment and simulation results was calculated to be 9.5 mm and 11.2 mm, indicating large variations in free surface elevation across the HDVS overflow tray. Thus, the free surface height measured for the Swirl-Flo® differs too greatly for its topology simply to be modelled by a flat wall.

Comparing the computational and empirical results, the greatest percentage error of 54.4% was observed for measurement points X1 and X2. This could be due to fixing the overflow rate in the simulation, or the presence of a systematic experimental error of 5 ± 0.5 mm for X1 to X7 (this has been applied to the data given in Table 4). Alternatively, very good agreement is observed for measurement points X8 and X26, which both had the same values for free surface height of 31 mm and 24 mm respectively. It was found that 73% of the simulation results over predicted the free surface elevation. This could be due to simulated flow rate being too large, as the experimental flow rates would have fluctuated to some degree.

The interface region for 3D simulation is relatively large (12 mm) in comparison to the surface heights being measured. While cell refinement was applied in the overflow region, this was limited by the computational resources available. The simulation took 6 days to complete when performed in parallel across 28 cores, and required 70 gigabytes of memory. While 6 days is relatively small simulation time in CFD, prior problems with boundary conditions and unphysical effects meant that there was insufficient time to re-run the simulation.

Table 4: Experimental and simulation results for the measurement points denoted in Figure 16.

Measurement Point	Experimental measurements (mm)	Simulation results at $\alpha = 0.5$ (mm)	Percentage error between experiment and simulation results (%)
X1	9.0 ± 1.0	4.1	54.4
X2	9.0 ± 1.0	4.1	54.4
X3	9.0 ± 1.0	5.0	44.4
X4	9.0 ± 1.0	13.1	45.6
X5	10.0 ± 1.0	12.4	24.0
X6	12.0 ± 1.0	8.1	33.3
X7	11.0 ± 1.0	13	18.2
X8	31.0 ± 0.5	31.0	0.0
X9	32.0 ± 0.5	33.3	4.1
X10	34.0 ± 0.5	35.4	4.1
X11	28.0 ± 0.5	29.6	5.7
X12	30.0 ± 0.5	32.5	8.3
X13	32.0 ± 0.5	34.8	8.7
X14	26.0 ± 0.5	29.4	13.1
X15	27.0 ± 0.5	32.5	20.4
X16	30.0 ± 0.5	35.5	18.3
X17	28.0 ± 0.5	29.5	5.4
X18	28.0 ± 0.5	32.5	16.1
X19	30.0 ± 0.5	34.5	15.0
X20	33.0 ± 0.5	29	12.1
X21	33.0 ± 0.5	35.6	7.9
X22	36.0 ± 0.5	39.7	10.3
X23	14.0 ± 0.5	16.4	17.1
X24	20.0 ± 0.5	25	25.0
X25	21.0 ± 0.5	25.4	21.0
X26	24.0 ± 0.5	24	0.0

5. Discussion and conclusions

The OpenFOAM® utility `snappyHexMesh` was found to generate high quality grids of the rectangular tank and Swirl-Flo® geometries. Both had a maximum non-orthogonality less than 70° and a maximum skewness less than 4. This was due to the mesh quality constraints used during the grid generation process, for which a maximum non-orthogonality and skewness of 65° and 2.4 was imposed. The default allowable non-orthogonality (65°) was found to be appropriate, as meshes with this constraint were generated without snapping problems and had no severely non-orthogonal faces. Furthermore, decreasing the maximum allowable boundary skewness from 20 (default) to 2.4 still provided reasonable snapping accuracy, while ensuring that the maximum skewness was below 4. Low values of average non-orthogonality (6.30° and 7.50° for the tank and HDVS respectively) also indicate that the meshes are of a high quality. The rectangular tank had lower values for all mesh quality metrics (apart from skewness) than the Swirl-Flo® grid with a maximum displacement between the mesh boundary and STL surface half that calculated for the HDVS mesh. This could be due to the rectangular tank having a simpler geometry, simplifying the mesh generation process. Whilst both the rectangular and HDVS grids exceeded the maximum allowable aspect ratio of 5, these occurrences were limited to cells on the boundaries, resulting from modification of the background mesh to conform it to the surface of the geometry. This was deemed acceptable due to the isolated nature of these cells and because the maximum AR remained below 10.

While the rectangular tank may have better mesh quality metric values than the HDVS grid, the number of cells used is quite inefficient as refinement regions were not utilised as in the Swirl-Flo® case. Unfortunately, due to project time constraints, a mesh independency study using the `driftFluxFoam` solver could not be undertaken. However, Scobell (2018) performed steady-state simulations using the `simpleFoam` solver to estimate the number of cells required for mesh independency to be achieved. From his results, it was discovered that a hexahedral rectangular tank mesh converged at 670,000 cells (cell spacing of 5 mm) and a tetrahedral HDVS mesh converged at approximately 5.5 million cells (cell spacing of 5.25 mm). The background mesh spacing used in the rectangular and Swirl-Flo® case was 3.25 mm and 8 mm respectively. Thus, increasing the background mesh spacing to 10 mm and applying a refinement region of level 1 for the inlet, inlet weir and overflow regions of the rectangular tank would produce a more computationally efficient mesh. While the Swirl-Flo® case used a greater cell spacing than Scobell's findings, the mesh was refined in the inlet, underflow, overflow, and vessel wall regions. The lowest level of refinement applied resulted in a cell spacing of 4 mm, and so the mesh is deemed to have an appropriate cell spacing whilst improving computational efficiency.

Because `snappyHexMesh` ensures that the final mesh produced does not invalidate specified mesh quality metrics, the grids created were found to in most cases always satisfy the quality constraints defined by OpenFOAM®. Furthermore, because the starting background mesh contains hexahedral cells with aspect ratios close to 1, the maximum aspect ratios reported by `checkMesh` were always relatively low (5.29 for the rectangular tank and 7.81 for the Swirl-Flo®). While decreasing the maximum allowable boundary skewness below 2.4 meant that the skewness of the mesh could be reduced greater still, it was found that it caused the maximum displacement of the mesh boundary to the STL surface to significantly increase; in the rectangular tank parameter study, decreasing the maximum allowable boundary skewness from 2.4 to 1.1 caused the maximum mesh/STL distance to increase by a factor of 4. A 28.2% increase in maximum non-orthogonality was also observed (41.9° to 53.7°). Thus, the maximum boundary skewness limit of 2.4 was chosen as it provided sufficient snapping accuracy and good mesh quality.

The number of cells generated by `snappyHexMesh` was found to be best controlled by altering the cell density of the background grid. Applying increasing levels of refinement caused the cell count to dramatically increase due to the cell spacing being divided by 2 to the power of the specified refinement level. Furthermore, increasing surface and edge refinement was found not to improve snapping accuracy after level 1, and actually increased the maximum aspect ratio, non-orthogonality and face skewness of the mesh. However, it is assumed that snapping accuracy and mesh quality will have improved for greater refinement levels if a coarser background grid had been used. This is because a similar trend was observed when altering the background mesh density; refining the background mesh from approximately 1 million to 2 million cells improved mesh quality and snapping, but then deteriorated after refining the background grid to approximately 4 million cells. This suggests that there is an optimum cell step spacing when conforming the mesh to the geometry. However, it is worth noting that these trends were observed for the parameter study performed on the rectangular tank, and may not necessarily be the same for the Swirl-Flo® or more complex geometries.

One major advantage of `snappyHexMesh` was its fast meshing ability. The mesh generation times for the final rectangular tank and Swirl-Flo® geometries were 65 seconds and 4.7 minutes respectively. However, the Swirl-Flo® mesh had approximately 1.8 times more cells than the rectangular tank yet took approximately 4.3 times longer to generate. Some discrepancy will be due to using differing number of cores (10 for the Swirl-Flo® and 8 for the rectangular tank), but it can be concluded that meshing time can increase significantly when the complexity of the geometry increases. Furthermore, these meshing times are without layer addition. For the layers addition attempt to the 4.4 million Swirl-Flo® mesh detailed in Section 4.2.1.1, the layer addition step alone took 17.2 minutes, 2.3 times longer than the castellation and snapping stages combined. However, these meshing times are still relatively quick. Thus, `snappyHexMesh` is suitable for applications in which the geometries to be meshed are always similar, as once the control parameters have been selected, they can be used continuously to generate good quality meshes quickly after provision of the STL files (Fahner & Broers, 2017).

Unfortunately, it was discovered that layers could not be inserted into the rectangular tank and Swirl-Flo® meshes. The difficulty in using `snappyHexMesh` to generate viscous layers has been noted in several studies and meshing guides such as Tapia (2009), Cavar, et al. (2016) and Fahner & Broers (2017). In OpenFOAM®, when simulating turbulence using high Reynolds number RAS models, it is required that the first cell have a Y+ value (non-dimensional distance) between 30 and 200 (OpenCFD, 2017c). The Y+ values for the interFoam HDVS simulation were calculated using the `yPlusRAS` utility for the latest time step, and were found to vary significantly (between 0.5 and 469.1) for different regions of the model. Thus, turbulence will not have been simulated correctly at the walls for most of the computational domain. This is a serious limitation of the `snappyHexMesh` utility.

The free surface elevation simulation results for the 2D rectangular tank case showed very good agreement with experimental results. When taking the free surface elevation to be at $\alpha = 0.5$ (Queguineur, et al., 2015), the largest percentage error was 10%. However, only two free surface heights were measured for the rectangular tank during the experiments. While the numerical results show excellent agreement with empirical results, more data should have been collected to verify agreement for the free surface elevation across the length of the tank.

Contrary to the 2D rectangular tank case, both the empirical and numerical results for the HDVS indicate that the free surface cannot be modelled accurately by a flat wall. This is because the free surface elevation varies greatly across the Swirl-Flo® geometry, with a simulation and experimental standard deviation of 11.2 mm and 9.5 mm respectively. However, examining the data in Table 4, the greatest free surface heights tend to occur in the outer annulus of the geometry (measurement points X8 to X22). As the rigid-lid in the HDVS is modelled as three separate walls (*waterLevel_1*, *2* and *3* in Figure 11), it would be possible to set these regions to different water level heights. Thus, the rigid-lid height of 23 mm could be improved by setting the outer annulus wall height to be 30.5 mm above the overflow tray - the average experimental result for the measurement points in this region. This also would reduce the difference in distance between the rigid-lid and the results for the 15 measurement points, with a lower standard deviation of 2.8 mm.

There were several issues with the HDVS VOF simulation. Firstly, fixing the overflow rate may have been responsible for the large percentage error of 54.4% observed for measurement points X1 and X2 positioned at the outlet. Another simulation was performed with the domain extended and a `zeroGradient` boundary condition used for the overflow velocity. From this, it was discovered that the stable free surface height varied between 102 mm and 126 mm above the overflow tray. As the overflow tray wall is only 97 mm high in reality, it was decided to fix the velocity at the overflow, as this gave results more comparable and realistic to that of the experiment. However, this does suggest that the inlet velocity in the simulation was too high, as the experimental conditions would not have been constant and would have fluctuated to some degree. This is supported by the observation that over 70% of the numerical measurement points over-predicted the free surface elevation.

Another issue with the numerical simulation is that the interface thickness was found to be too large (12 mm) in comparison to the surface elevation heights measured. Unfortunately, a greater cell discretisation could not be applied in the overflow region due to limited project time and computational resources; the 1.6 million cell solution alone required 70 gigabytes of memory for a simulated time of 300 seconds and a write interval of 2 seconds. However, the simulation time can be reduced by 60% as it was found that the free surface elevation results reached steady-state after 120 seconds. The memory requirement can also be further reduced by increasing the write interval. The `mapFields` utility was used in an attempt to continue the simulation for several more iterations on a higher resolution mesh to mitigate the computational time required but was unsuccessful due to the utility assigning zero cell field values for ε. A potential solution to this could be writing a python script to read through the turbulent dissipation rate cell fields and replace the zero values with the lowest value observed in the original solution.

While the rigid-lid assumption does not accurately capture the free surface shape for the Swirl-Flo®, this does not necessarily indicate that it is an invalid assumption. For instance, if the simulated velocity profiles are in excellent agreement to those obtained experimentally, then it could be argued that accurately capturing the free surface shape and elevation is not of great concern. This has not been examined in this report and thus is a recommendation for further work.

5.1 Conclusions

To summarise, the internal domains of a modified Armfield W7 rectangular sedimentation tank and a model hydrodynamic vortex separator (the Swirl-Flo®) have been meshed using the native OpenFOAM utility `snappyHexMesh`. It was found that `snappyHexMesh` was able to generate hexahedral dominant grids quickly that satisfied OpenFOAM®'s mesh quality requirements. However, its inability to successfully insert boundary layers is a significant drawback of the software, making it unsuitable for some CFD applications.

Water-air volume of fluid (VOF) simulations were also performed on the devices using the solver `interFoam` to evaluate the validity of modelling the free surface as a horizontal slip wall. Comparing numerical and empirical results indicate that the free surface for the rectangular tank can be modelled accurately using a rigid-lid, provided that the computational domain is cut-off at the start of the overflow weir. However, more experimental data needs to be provided to affirm the accuracy of the simulation results. Numerical results for the VOF Swirl-Flo® simulation show significantly larger differences between the free surface elevation and those measured in the experiment. Thus, a rigid-lid assumption does not accurately capture the free surface elevation across all of the geometry. However, as the slip wall on the HDVS is split into three parts, the heights of these can be changed independently to reduce the difference between the rigid-lid height and the experimental results. The accuracy of the simulation results was reduced by the large interface thickness, and so the simulation should be re-run with a greater cell resolution in the overflow region.

5.2 Recommendations for further work

Further trial and error of the `snappyHexMesh` layer addition stage should be undertaken, as this is controlled by a substantial number of parameters that were not able to be fully investigated during the project timeframe. Meshes of the primary sedimentation devices should also be generated using another open source software, such as `cfMesh`, and compared to those generated by `snappyHexMesh`. This would allow meshing efficiencies of the different software to be compared, as well as evaluating their success at inserting boundary layer cells for the given geometries. A mesh independency study using the `driftFluxFoam` solver should also be performed to verify the required cell discretisation. This was unable to be accomplished during this project due to time constraints. VOF simulations for the HDVS should also be performed with a higher cell resolution in the overflow region to reduce the water-air interface thickness.

6. Project management

Procedures for managing the project included keeping a record of the work done each day in a log book. Separate Microsoft® Word documents were also created to store research source locations and web hyperlinks for specific literature topics (including but not limited to mesh generation, VOF simulations, and turbulence modelling). This meant that references could be found more easily if required later. The record of literature was stored in the Cloud to allow access from anywhere with an internet connection, and the files regularly backed-up to an external hard drive to mitigate the risk of data loss. A workflow chart created by Wye (2018) was used to visualise the overall progress of the project alongside a personal Gantt chart detailing the activities required to be completed. Each activity had float time assigned, which ultimately was fully utilised due to remodelling of the rectangular tank, problems with `snappyHexMesh` layer addition and the requirement to re-run VOF HDVS simulations due to incorrect assignment of boundary conditions.

During the project, a large number of meshes were created in order to evaluate the effect that varying individual control parameters in `snappyHexMesh` had on the quality of the grids produced. To keep track of which parameters were being changed, an Microsoft® Excel spreadsheet was created to note the settings information and results for each case. Log files were also included in each case directory, which noted the settings used, changes made from previous cases, and the corresponding effects on the mesh. A similar system was also adopted for recording what simulations had been performed, with information detailing the solver, mesh and initial conditions used.

A factor which impacted on project progression was the changes made to the rectangular tank geometry (the raising of the bed and the inclusion of a hopper). This meant that a new SOLIDWORKS® CAD geometry of the rectangular tank had to be created and the SALOME STL creation process repeated. New `snappyHexMesh` dictionary files also had to be set-up to account for the additional STL files and the new rectangular tank geometry features (i.e. the walls which formed the hopper). To save time, the 2D grid generated using `blockMesh` was not altered as it was deemed that the free surface height over the weir for a given flow rate would not be greatly affected by the change in the bed height.

An opportunity to modify the genetic algorithm code developed by Tabor and Fabritius (2016) was presented during the project for optimising the `snappyHexMesh` process on the 8th of

March. The Python script was originally coded for use with pyFoam on OpenFOAM® version 2.1.1. It was discovered that, for the code to work with OpenFOAM® version 2.4, it would have to be largely rewritten due to the large development jump between the two OpenFOAM® versions. Thus, the time required to port the genetic algorithm code for use on version 2.4 was deemed to not be worthwhile. Instead, OpenFOAM® version 2.1.1 (including pyFoam) was downloaded and compiled in an attempt to run the code on the earlier version. However, errors were still encountered when trying to use the code. Because of the author's limited working knowledge of Python, and because several attempts had been made to fix the code over the ongoing weeks, it was deemed that it would be best to utilise the remaining time on another aspect of the project. Thus, instead it was decided that simulating the free surface of the HDVS would be beneficial, due to lower occurrences of HDVS free surface modelling in the literature (Jarman, et al., 2008).

7. Contribution to group functioning

Contributions to the group functioning include providing rectangular tank and HDVS meshes for use in the driftFluxFoam simulations described in Bentley (2018) and Russell (2018). Another example of contribution to group functioning includes taking and preparing minutes and agendas for group meetings. This was done as part of a rotation, in which members of the group alternated the roles. The author also assisted Mendoza (2018) in performing their initial viscosity experiments at the Hydro International office located in Clevedon Hall Estate. Assistance was also providing in modifying the SOLIDWORKS® CAD geometries of the rectangular tank and the HDVS prepared by Scobell (2018).

Examples of how members of the group contributed to the work done in this individual report include Baker (2018), Mendoza (2018) and Wye (2018) providing free surface height data for given inlet and underflow velocities for both the rectangular tank and the HDVS. This data was used to set up VOF simulations with the appropriate initial conditions, and allowed the VOF simulation results to be compared to that of the experiments.

8. References

Alkhaddar, R. M., Andoh, R. Y. G. & Phipps, D., 2002. *The Experimental and Residence Time Distribution Estimation of the Decomposition of Hydrogen Peroxide within a Hydrodynamic Vortex Separator.* Portland, 9th International Conference on Urban Drainage.

Alkhaddar, R. M., Higgins, P. R., Phipps, D. A. & Andoh, R. Y. G., 1999. *The residence time distribution of prototype hydrodynamic vortex separator operating with a baseflow component.* Sydney, 8th International Conference on Urban Storm Drainage, pp. 18-25.

Anderssson, B. et al., 2012. *Computational Fluid Dynamics for Engineers.* 1st ed. New York: Cambridge University Press.

Baker, A., 2018. *Validation of Sedimentation Tanks through Empirical Studies,* Exeter: University of Exeter.

Baker, A. et al., 2018. *Experimental and Numerical Investigation Into the Use of Olive Stone Powder as a Substitute for Primary Sludge Modelling,* Exeter: University of Exeter.

Bakker, A., 2008. *Lecture 7 - Meshing: Applied Computational Fluid Dynamics.* [Online]
Available at: http://www.bakker.org/dartmouth06/engs150/07-mesh.pdf
[Accessed 17 May 2018].

Beg, M. N. A., 2017. *Application of CFD modelling in Water Resources Engineering.* [Online]
Available at: https://www.sheffield.ac.uk/polopoly_fs/1.711730!/file/Coimbra_Outreach_final.pdf
[Accessed 13 May 2018].

Bentley, R., 2018. *An Investigation into MULES and its Application into Simulating Settling Behaviour in an Armfield Rectangular Settling Tank,* Exeter: University of Exeter.

Brennan, D., 2001. *The Numerical Simulation of Two-Phase Flows in Settling Tanks,* London: Imperial College of Science, Technology and Medicine.

Cavar, D. et al., 2016. Comparison of OpenFOAM and EllipSys3D for neutral atmospheric flow over complex terrain. *Wind Energy Science,* 1(1), pp. 55-70.

Damián, S. M., 2009. *Description and utilization of interFoam multiphase solver.* [Online]
Available at: http://infofich.unl.edu.ar/upload/3be0e16065026527477b4b948c4caa7523c8ea52.pdf
[Accessed 19 May 2018].

Deb, K., Anand, A. & Joshi, D., 2002. A Computationally Efficient Evolutionary Algorithm for Real-Parameter Optimization. *Evolutionary Computation,* 10(4), pp. 371-395.

Douglas, J. F., Gaziorek, J. M., Swaffield, J. A. & Jack, L. B., 2005. *Fluid Mechanics.* 5th ed. Harlow: Pearson .

Egarr, D. A. et al., 2005. Computational fluid dynamic prediction of the residence time distribution of a prototype hydrodynamic vortex separator operating with a base flow component. *Proceedings of the Institute of Mechanical Engineers Part E: Journal of Process Mechanical Engineering,* 219(1), pp. 53-67.

Engys, 2012. *A Comprehensive Tour of snappyHexMesh - 7th OpenFOAM workshop.* [Online]
Available at: https://openfoamwiki.net/images/f/f0/Final-AndrewJacksonSlidesOFW7.pdf
[Accessed 1 May 2018].

Fahner, T. & Broers, R., 2017. *The challenge of many different applications solved by Ansa.* [Online]
Available at: https://www.beta-cae.com/events/c7pdf/3C_1_FAHNER.pdf
[Accessed 12 May 2018].

Faram, M. G., James, M. D. & Williams, C. A., 2004. *Wastewater Treatment Using Hydrodynamic Vortex Separators.* Wakefield, CIWEM.

Findikakis, A. N., 2016. The Contribution of Advanced Simulation Tools to the Hydraulic Design of Water and Wastewater Treatment Facilities. *Hydrolink: Hydraulics of Wastewater Treatment,* 6 July, Issue 2, p. 34.

Gopala, V. R. & Wachem, B. G. v., 2008. Volume of fluid methods for immiscible-fluid and free-surface flows. *Chemical Engineering Journal,* 141(1-3), pp. 204-221.

Greenshields, C., 2017b. *OpenFOAM v5 User Guide: 3.4 Running applications in parallel.* [Online]
Available at: https://cfd.direct/openfoam/user-guide/running-applications-parallel/
[Accessed 10 May 2018].

Greenshields, C., 2017c. *OpenFOAM v5 User Guide: 4.4 Numerical schemes.* [Online]
Available at: https://cfd.direct/openfoam/user-guide/fvschemes/
[Accessed 29 April 2018].

Greenshields, C., 2017d. *OpenFOAM v5 User Guide: 5.1 Mesh description.* [Online]
Available at: https://cfd.direct/openfoam/user-guide/mesh-description/#x23-1620005.1
[Accessed 24 April 2018].

Greenshields, C., 2017e. *OpenFOAM v5 User Guide: 5.3 Mesh generation with blockMesh.* [Online]
Available at: https://cfd.direct/openfoam/user-guide/blockmesh/#x25-1790005.3
[Accessed 24 April 2018].

Greenshields, C., 2017f. *OpenFOAM v5 User Guide: 5.4 Mesh generation with snappyHexMesh.* [Online]
Available at: https://cfd.direct/openfoam/user-guide/snappyHexMesh/#x26-1910005.4
[Accessed 15 April 2018].

Greenshields, . C. J., 2017a. *User Guide version 5.0.* [Online]
Available at: http://foam.sourceforge.net/docs/Guides-a4/OpenFOAMUserGuide-A4.pdf
[Accessed 29 April 2018].

Hargreaves, M. D., Morvan, H. P. & Wright, N. G., 2007. Validation of the Volume of Fluid Method for Free Surface Calculation: The Broad-Crested Weir. *Engineering Applications of Computational Fluid Mechanics,* 1(2), pp. 136-146.

Hirt, C. W. & Nichols, B. D., 1981. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics,* 39(1), pp. 201-225.

Jarman, D. S., Butler, D., Tabor, G. & Andoh, R., 2015. Modelling of vortex flow controls at high drainage flow rates. *Proceedings of the ICE - Engineering and Computational Mechanics,* 168(1), pp. 17-34.

Jarman, D. S. et al., 2008. *Computational fluid dynamics as a tool for urban drainage system analysis: A review of applications and best practice.* Edinburgh, 11th International Conference on Urban Drainage.

Kallinderis , Y. & Kontzialis, C., 2008. A priori mesh quality estimation via direct relation between truncation error and mesh distortion. *Journal of Computational Physics,* 228(3), pp. 881-902.

Lu, Z. et al., 2017. Comparative study on turbulence models simulating inorganic particle removal in a Pista grit chamber. *Environmental Technology,* pp. 1-12.

Mendoza, M., 2018. *Determination of a Viscosity Model for an Olive Stone Powder and Water Suspension for Primary Sedimentation Tank Modelling,* Exeter: University of Exeter.

Meroney, R. N. & Sheker, R. E., 2016. Removing Grit During Wastewater Treatment: CFD Analysis of HDVS Performance. *Water Environment Research,* 88(5), pp. 438-448.

Meroney, R. N. & Sheker, R. E., 2016. Removing Grit During Wastewater Treatment:CFD Analysis of HDVS Performance. *Water Environment Research,* 88(5), pp. 438-448.

Moukalled, F., Mangani, L. & Darwish, M., 2016. *The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab®.* 1st ed. Cham: Springer International Publishing.

OPEN CASCADE, 2018. *SALOME - The Open Source Integration Platform for Numerical Simulation.* [Online]
Available at: https://www.salome-platform.org/
[Accessed 30 April 2018].

OpenCFD, 2017a. *OpenFOAM v1712 - The open source CFD toolbox extended code guide - Snapping.* [Online]
Available at: https://www.openfoam.com/documentation/cpp-guide/html/guide-meshing-snappyhexmesh-snapping.html
[Accessed 1 May 2018].

OpenCFD, 2017b. *OpenFOAM v1712 - The open source CFD toolbox extended code guide - snappyHexMesh.* [Online]
Available at: https://www.openfoam.com/documentation/cpp-guide/html/guide-meshing-snappyhexmesh.html
[Accessed 1 May 2018].

OpenCFD, 2017c. *OpenFOAM v1712 - The open source CFD toolbox extended code guide - Turbulence.* [Online]
Available at: https://www.openfoam.com/documentation/cpp-guide/html/guide-turbulence.html
[Accessed 12 May 2018].

Partech, 2018. *Primary Settlement Tank (PST).* [Online]
Available at: https://www.partech.co.uk/application/primary-settlement-tanks/
[Accessed 7 April 2018].

Pathapati, S.-S. & Sansalone, J. J., 2009. CFD Modeling of a Storm-Water Hydrodynamic Separator. *Journal of Environmental Engineering,* 135(4), pp. 191-202.

Pointwise Inc., 2018. *Compute Grid Spacing for a Given Y+.* [Online]
Available at: http://www.pointwise.com/yplus/
[Accessed 10 May 2018].

Queguineur, G., Jarman, D., Paterson, E. & Tabor, G., 2015. Computational fluid dynamics of vortex flow controls at low flow rates. *Proceedings of the Institution of Civil Engineers - Engineering and Computational Mechanics,* 166(4), p. 211–221.

Reynolds, W. C., 1987. *Fundamentals of turbulence for turbulence modeling and simulation,* s.l.: s.n.

Rhoads, J., 2014. *OpenFOAM Workshop 2014: Effects of grid quality on solution accuracy.* [Online]
Available at: https://afinemesh.files.wordpress.com/2014/07/ofw20141.pdf
[Accessed 5 May 2018].

Románszki, L., Mohos, M., Telegdi, J. & Nyikos, L., 2013. *Contact Angle Measurement is an Efficient Tool for the Characterization of Corrosion Pro- tection Nanolayers on Copper Alloys and Stainless Steel.* Sumy, Sumy State University.

Russell, T., 2018. *The Validation of a Vortex Separator with the use of OpenFOAM,* Exeter: University of Exeter.

Scobell, T., 2018. *Investigation into Mesh Generation Techniques Using Pointwise on Primary Sewage Sedimentation Devices,* Exeter: University of Exeter.

Shih, T.-H.et al., 1995. A new k-ϵ eddy viscosity model for high reynolds number turbulent flows. *Computers and Fluids,* 24(3), pp. 227-238.

Shih, T.-H., Zhu, J. & Lumley, J. L., 1994. *A new Reynolds stress algebraic equation model,* Cleveland: NASA.

Sullivan, R. H. et al., 1978. *The Swirl Primary Separator: Development and Pilot Demonstration,* Cincinnati: United States Environmental Protection Agency.

Tabor, G. & Fabritius, B., 2016. Improving the quality of finite volume meshes through genetic optimisation. *Engineering with Computers,* 32(3), pp. 425-440.

Tapia, X. P., 2009. *Modelling of wind flow over complex terrain using OpenFoam,* Gävle: University of Gävle.

Tchobanoglous, G. et al., 2014. *Wastewater Engineering: Treatment and Resource Recovery.* 5th ed. New York: McGraw-Hill Education.

Tu, J., Yeoh, G. H. & Liu, C., 2008. Practical Guidelines for CFD Simulation and Analysis. In: *Compuational Fluid Dynamics: A practical Approach.* Oxford: Elsevier Inc., pp. 224-276.

Versteeg, H. K. & Malalasekera, W., 2007. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method.* 2nd ed. London: Pearson Education Limited.

Water Technology Engineering Ltd, 2018. *Sewage Treatment Processes Explained.* [Online]
Available at: https://www.wte-ltd.co.uk/sewage_treatment_explained.html
[Accessed 7 April 2018].

Wicklein, E. et al., 2015. Good modelling practice in applying computational fluid dynamics for WWTP modelling. *Water Science & Technology,* 73(5), pp. 969-982.

Wilcox, D. C., 2006. *Turbulence Modeling for CFD.* 3rd ed. La Cañada Flintridge: DCW Industries Inc..

Wolf Dynamics, 2017a. *Mesh generation using snappyHexMesh.* [Online]
Available at: http://www.wolfdynamics.com/wiki/meshing_OF_SHM.pdf
[Accessed 1 May 2018].

Wolf Dynamics, 2017b. *OpenFOAM Introductory Course Training Material.* [Online]
Available at: http://www.wolfdynamics.com/images/OF_intro_training/module3.pdf
[Accessed 24 April 2018].

Wolf Dynamics, 2017c. *Tips and tricks in OpenFOAM®.* [Online]
Available at: http://www.wolfdynamics.com/wiki/OFtipsandtricks.pdf
[Accessed 10 May 2018].

Wye, A., 2018. *Investigation and Development of a Model to Describe the Settling Characteristics of Olive Stone Powder,* Exeter: University of Exeter.

Ying, G. et al., 2012. Stormwater treatment: examples of computational fluid dynamics modeling. *Frontiers of Environmental Science and Engineering,* 6(5), pp. 638-648.