Computational investigations and grid refinement study of 3D transient flow in a cylindrical tank using OpenFOAM

F Mohd Sakri*, M S Mat Ali and S A Z Sheikh Salim
Wind Engineering, Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia

*eypasakri@gmail.com

Abstract. The study of physic fluid for a liquid draining inside a tank is easily accessible using numerical simulation. However, numerical simulation is expensive when the liquid draining involves the multi-phase problem. Since an accurate numerical simulation can be obtained if a proper method for error estimation is accomplished, this paper provides systematic assessment of error estimation due to grid convergence error using OpenFOAM. OpenFOAM is an open source CFD-toolbox and it is well-known among the researchers and institutions because of its free applications and ready to use. In this study, three types of grid resolution are used: coarse, medium and fine grids. Grid Convergence Index (GCI) is applied to estimate the error due to the grid sensitivity. A monotonic convergence condition is obtained in this study that shows the grid convergence error has been progressively reduced. The fine grid has the GCI value below 1%. The extrapolated value from Richardson Extrapolation is in the range of the GCI obtained.

1. Introduction
Recently, the use of liquid storage tanks has been necessary for various industries. They can be found in space vehicles and rockets for their propellant storage in the liquid fuel system, water tank for daily use and also in oil and gas industries for long-short term storages, among others. All these applications come with a similar flow phenomenon problem, which is liquid draining that often induces an air-core vortex formation. The air-core is generated when liquid drains through the outlet and dip is developed on the surface as the surface level reaches a certain critical height, Hc. Then, the dip further grows into a vortex with an air core, which lengthens to the bottom of the tank. This air-core vortex will reduce the cross-sectional area and affect the discharge rates.

At this moment, there are various types of passive flow control and air-core vortex suppression mechanisms. The most popular one is the vortex breaker in liquid propellant of launch vehicles [1]. It decreases the critical height, Hc (as shown in Figure 1) by using its circular flat plate with blades or porous wall. In other studies, inclined base [2] and eccentric drain outlet [3] have been used in order to change the pressure at the center of the outlet. Consequently, this will affect and reduce the formation of air-core vortex. Additionally, Ramamurthi and Tharakan [4] stated in their studies that by placing net and turf in the tank also can reduce the development of surface dip and air-core vortex. However, these mechanisms cannot completely suppress the air-core vortex. This problem has been and still continues to be investigated, including in this study. In order to get a comprehensive data, this paper applies numerical simulation in the study of liquid draining inside the tank. Since mesh shows a very important result of numerical simulation [1], it is important to estimate the grid convergence error due to the sensitivity of the resolution.
The purpose of this study is to determine the Grid Convergence Index (GCI) of three types of grid resolution for the liquid draining. It is the most familiar and reliable technique for quantification of numerical uncertainty [5, 6, 7]. The result of this investigation, especially for the finer grid, is very important for further analysis in order to correctly replicate the air-core vortex formation inside the tank. Figure 2 shows the schematic diagram for the case under investigation, which is a cylindrical tank partially filled with water. The diameter of the tank is 90 mm, with 450 mm in length, and the initial height of the water is 350 mm. The diameter of the outlet nozzle is 6 mm, with 15 mm in length. The liquid is drained by gravity, g with open to the atmospheric condition on the top and bottom of the tank [2].

2. Solution methodology

2.1. Navier-Stoke equations

The following continuity and momentum equations are obtained from the Navier-Stoke equations after using the incompressible condition [3, 4]:

\[
\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[ \mu \frac{\partial u_i}{\partial x_j} \right] \tag{1}
\]

\[
\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left[ \mu \frac{\partial u_i}{\partial x_j} \right] \tag{2}
\]

2.2. Turbulence model

In this study, the Reynolds Averaged Navier-Stokes equations (RANS) and two-equation k – \( \varepsilon \) model are employed to simulate the transient flows.

2.3. Discretization of the transport equation

Transformation from PDE to linearized algebraic equation needs to be done to solve the equations that describe the flow transport. Equation 3 shows the standard form of the transport equation for the field variable, \( \phi \) [4]:

\[
\frac{\partial \rho \phi}{\partial t} + \nabla \cdot \rho \mathbf{U} \phi - \nabla \cdot (\rho \nabla \phi) = S_{\phi}(\phi) \tag{3}
\]

All terms in Equation 3 are integrated over time step ranging from \( t \to t + \Delta t \) and control volume, \( V_p \) as shown in Equation 4 [4].

\[
\int_t^{t+\Delta t} \left[ \frac{\partial}{\partial t} \int_{V_p} \rho dV + \int_{V_p} \nabla \cdot (\rho \mathbf{U} \phi) dV - \int_{V_p} \nabla \cdot (\rho \nabla \phi) dV \right] dt = \int_t^{t+\Delta t} \left[ \int_{V_p} S(\phi) dV \right] dt \tag{4}
\]

Here, \( \mathbf{U} \) is the velocity field, \( \rho \) is the density, \( \Gamma \) is the diffusion coefficient, \( S \) is all source terms and index \( p \) describes the midpoint of the control volume. Table 1 describes numerical schemes for non-swirl case.
Table 1. Order of convergence and numerical scheme of differential operator in non-swirl case [5]

| Term          | Scheme     | Order of Convergence | Interpolation Scheme |
|---------------|------------|----------------------|----------------------|
| \(\frac{\partial}{\partial t} \frac{\partial^2}{\partial x^2}t\) | Euler      | \(O(h)\)             |                      |
| \(\nabla\)    | Gauss      | \(O(h^2)\)           | Linear               |
| \(\nabla (\rho \phi, U)\) | Gauss      | \(O(h^2)\)           | Van Leer             |
| \(\nabla (\phi, \alpha)\) | Gauss      | \(O(h)\)             | upwind               |
| \(\nabla (\phi, k)\) | Gauss      | \(O(h)\)             | linear corrected     |
| \(\nabla^2\)  | Gauss      | \(O(h^2)\)           |                      |

2.4. Discretization of governing equations

2.4.1. Convection scheme. Using Gauss’ theorem, the generalized linearization form of the convective term in Equation 4 is given in Equation 5 [4]:

\[
\int_{V} \nabla (\rho U \phi) dV = \sum_{f} S. (\rho U \phi)_{f} = \sum_{f} S. (\rho U)_{f} \phi_{f} = \sum_{f} F \phi_{f}
\]  

(5)

where \(F\) is the mass flux through the cell’s faces, as given in Equation 6.

\[
F = S. (\rho U)_{f} = S. u_{f}
\]  

(6)

Here, \(S\) is the surface area element pointing outward, \(u\) is the local velocity in three spatial component \((x, y, \text{and} z)\) and index \(f\) describes the value at the surface control volume. It can be assumed that the mass flux from the Equation 6 can be calculated from the interpolated values of \(U\) and \(\rho\). In this study, Second Order Upwind (SOU) differencing scheme is applied. This scheme is conditionally bounded and is more advanced compared to Central Differencing Scheme (CDS) [6, 7, 8]. For SOU, the critical cell Reynolds number of the result of purely convective transport of a scalar field is free from \(\infty\) and it is infinite [6, 9] when the cell Reynolds number is over two. The face value is determined according to the polynomial in Equation 7 [10]:

\[
u_{e} = \begin{cases} 
    u_{p} + (u_{p} - u_{w}) \Delta x_{pN} \text{for} F \geq 0; \\
    u_{N} + (u_{N} - u_{NN}) \Delta x_{NN} \text{for} F < 0,
\end{cases}
\]  

(7)

\[
\Delta x_{pN} = \frac{x_{N} - x_{p}}{x_{p} - x_{w}}
\]  

(8)

\[
\Delta x_{NN} = \frac{x_{NN} - x_{n}}{x_{NN} - x_{n}}
\]  

(9)

2.4.2. Diffusion scheme. Using Gauss’ theorem, a generalized linearization form of the diffusion term in Equation 4 is given as Equation 10 [4]:

\[
\int_{V} \nabla (\rho \Gamma_{\phi} \nabla \phi) dV = \sum_{f} S. (\rho \Gamma_{\phi} \nabla \phi)_{f} = \sum_{f} (\rho \Gamma_{\phi})_{f} S. (\nabla \phi)_{f}
\]  

(10)

The following Equation 11 is possible to be used for the orthogonal mesh, such that the vectors \(S\) and \(d\) are parallel [4]:

\[
S. (\nabla \phi)_{f} = |S| \frac{\phi_{N} - \phi_{P}}{|d|}
\]  

(11)

According to [4], the face gradient of \(\phi\) can be collected from two values around the face, which are:

- Cell-centred gradient for the cells sharing the face: \((\nabla \phi)_{p} = \frac{1}{V_{p}} \sum_{f} S \phi_{f}\)
- Cell-centred gradient for interpolate it to the face: \((\nabla \phi)_{f} = f_{x}(\nabla \phi)_{p} + (1 - f_{x})(\nabla \phi)_{N}\)
In this study, the diffusion terms are discretized using the 2nd order deferred correction scheme. As Equation 11 is the 1st order accurate uncorrected, the diffusion term is corrected by using Equation 14 when the vectors N and P are not coordinated [11]. Thus the term is separated into non-orthogonal and orthogonal parts as follows [12]:

\[
S. (\nabla \phi_f) = \delta \left( \frac{\Phi_E - \Phi_W}{|d|} \right)_f + k \left( \frac{\Phi_E - \Phi_W}{|d|} \right)_f
\]  

(14)

The first part of Equation 14 is for the orthogonal whereas the second part is for the non-orthogonal. Vector \( \delta \) is parallel to the distance vector, d and vector \( k \) is aligned to the face of the gradient, \( \nabla \phi_f \). In addition, Equation 14 must satisfy the following requirement [4]:

\[
S = \delta + k
\]  

(15)

The limiting Equation 16 is recommended for stability as the non-orthogonal part is much bigger than the orthogonal part [12]:

\[
K_L \left[ k \left( \frac{\Phi_E - \Phi_W}{|d|} \right)_f \right] = S. (\nabla \phi_f) - \delta \left( \frac{\Phi_E - \Phi_W}{|d|} \right)_f
\]  

(16)

where \( K_L \) is user-specified parameter with \( 0 < K_L < 1 \).

2.4.3. Gradient term. The gradient term is discretized using bounded central differencing scheme that is a 2nd order Gaussian method. As in Equation 17, the gradient is calculated using integrals over faces [4]:

\[
\int_{V_f} \nabla \phi dV = \hat{\Phi}_{V_f} = \int_{S_f} \phi = \Sigma_f S_f \phi_f
\]  

(17)

The face value can be evaluated from the cell center values as follows:

\[
\phi_f = f_x \phi_p + (1 - f_x) \phi_N
\]  

(18)

\[
f_x = \frac{\mathcal{P}}{\mathcal{N}}
\]  

(19)

Figure 3 shows the interpolation factor of \( f_x \) is the ratio between the distances \( \mathcal{P} \) and \( \mathcal{N} \).

2.4.4. Temporal/time scheme. For transient flow, the physical properties of the flow are changing over time. Thus volume integrals from Equation 4 can be converted into surface integrals by applying the Gauss Theorem. They can then be written as the sum over the regarded control volume [4, 13] and the expression is called as “semi-discretized” form of the transport equation as shown in Equation 20 [14]:

\[
\int_t^{t+\Delta t} \left( \frac{\partial \phi_f}{\partial t} \right)_p + \sum F \phi_f - \sum (\rho \Gamma_f) S. (\nabla \phi_f) \right) dt = \int_t^{t+\Delta t} \left[ S_u \phi_p + S_p \phi_N \right] dt
\]  

This time discretization can be split into two parts: explicit and implicit methods. The explicit method needs a low computational effort as it only uses the previous time level in the discretization. However, when the Courant-Friedrichs-Lewy number is bigger than \( \infty \), there is a problem with the stability [8]. On the other hand, the implicit method needs more computational effort as it uses the new time level in discretization. Robust solutions are normally retrieved when the coupling between flow properties is bigger than in the explicit method [8].

The Euler implicit scheme is 1st order accurate linear approximation and it guarantees boundedness. The coupling in the system is more stable than the explicit method, even if the Courant number limit is violated [14]. As shown in Equation 21 and Equation 22, this method considers the face values in the new time level cell [4]:

\[
\phi_f = f_x \phi_p^n + (1 - f_x) \phi^N
\]  

(21)

\[
S. (\nabla \phi_f) = \left| \mathcal{D} \right| \left( \frac{\phi^N - \Phi^p}{|d|} \right) + k. (\nabla \phi_f)
\]  

(22)
2.5. Boundary conditions

Boundary conditions set down the series of faces in the computational mesh, which correspond to the boundaries of physical domain. They are separated into numerical and physical boundary conditions. Numerical boundary conditions are divided into the Von Neumann and Dirichlet boundary conditions. These boundary conditions stipulate the gradient of the variable normal to the boundary and the value of the variable on the boundary (or fixed/or constant value), respectively [4]. Meanwhile, the physical boundary conditions for the incompressible flow are explained as follows:

- **Inlet**: the velocity value is determined and the pressure condition is specified as zero gradient
- **Outlet**: the outlet boundary is prescribed similar to total mass balance. This can be performed in two ways:
  a) when the velocity dissemination is projected from the inside of the domain, the boundary condition on pressure is set as zeroGradient
  b) when the pressure dissemination is stated, boundary condition on pressure and velocity is set as fixedValue and zeroGradient, respectively.
- **Impermeable no-slip wall**: the flow velocity on the wall is the same as the wall. Thus the flux through the wall is zero, so that the pressure gradient is set as zeroGradient
- **Slip wall**: in scalar case, the variable is set as zeroGradient, but in vector magnitude case, the variable that normal and tangential to the wall is set as zero and zeroGradient, respectively.

In this study, three different patches have been constructed in the computational domain. The top surface is represented as an inlet and the bottom surface is represented as an outlet. Moreover, the side surface is represented as a wall. A fixedValue boundary condition is set up at the outlet patch while that at the wall patch is zeroGradient. In this case, the liquid is drained normally by the gravity. Other details of boundary conditions are listed in Table 2 and the description of each boundary is explained in Table 3.

### Table 2. Boundary conditions for the non-swirl case

| Flow Properties | Type of Patches                  | Outlet      | Inlet                | Walls              |
|-----------------|---------------------------------|-------------|---------------------|--------------------|
| α               | zeroGradient                    | inletOutlet| zeroGradient        |                    |
| U               | zeroGradient                    | pressureInletOutletVelocity | fixedValue         |                    |
| ρ g/  | fixedValue                     | totalPressure| bouyantPressure    |                    |
| k               | zeroGradient                    | inletOutlet| kqRWallFunction     |                    |
| nuT             | calculated                      | calculated | nutkWallFunction    |                    |
| nuTilda         | zeroGradient                    | inletOutlet| zeroGradient        |                    |
| ε               | zeroGradient                    | inletOutlet| epsilonWallFunction |                    |

### Table 3. Explanation of boundary conditions

| Type                        | Description of boundary conditions                                                                 |
|-----------------------------|-----------------------------------------------------------------------------------------------------|
| zeroGradient                | Normal gradient of \( \phi \) is zero                                                                 |
| fixedValue                  | Value of \( \phi \) is specified                                                                    |
| pressureInletOutletVelocity | Combination of pressureInletVelocity and inletOutlet(pressureInletVelocity : When \( P \) is known at inlet, \( U \) is evaluated from the flux, normal to the patch) |
| totalPressure               | Total pressure \( P_0 = P + \frac{1}{2} \rho |U|^2 \) is fixed; when \( U \) changes, \( P \) is adjusted accordingly                             |
| inletOutlet                 | Switches \( U \) and between fixedValue and zeroGradient depending on direction of \( U \).          |
| bouyantPressure             | Sets fixedGradient pressure based on the atmospheric pressure gradient                               |
| calculated                  | Boundary field \( \phi \) derived from other fields                                                 |
| nutkWallFunction            | On corresponding patches in the turbulent fields \( k \) and \( \mu \)                              |
| kqRWallFunction             | On corresponding patches in the turbulent fields \( k \), \( q \) and \( R \)                         |
| epsilonWallFunction         | On corresponding patches in the epsilon field                                                        |
2.6. Multiphase solvers

Multiphase solvers are available in OpenFOAM. They are listed as follows [13]:
- interFoam, LTSinterFoam and InterDyMFoam: these are based on Volume of Fluid (VOF) method
- twoPhaseEulerFoam, multiphaseEulerFoam and multiphaseInterFoam: these are based on the Eulerian-Eulerian method [17]

2.6.1. InterFoam. In this study, the interFoam solver is applied because it is capable to simulate flow of two fluids (liquid and air). In this solver, only one momentum and one mass conservation equation is determined for both fluids. Therefore, the viscosity and density of both fluids are averaged based on the volume fractions in the cell. In this condition, momentum and mass transfer are inconsiderate [13].

2.6.2. Volume of fluid (VOF). Volume of Fluid (VOF) method, a Eulerian-Eulerian model, is applied in averaging procedure for the liquid and air (multiphase) flow system. The two phases of fluids are modelled by adopting RANS equations. These governing equations are calculated numerically based on the OpenFOAM numerical simulation system. The parameter of \( \alpha \) in the Volume of Fluid (VOF) method is a function that indicates the relative fraction between liquid and gas in each cell of physical domain. \( \alpha \) equals to one when the fluid is in the liquid phase and \( \alpha \) equals to 0 when the fluid is in the gas phase. The change of their properties in each cell is calculated in the Equation 23 [18].

\[
\rho = \alpha \rho_l + (1 - \alpha) \rho_g, \quad \mu = \alpha \mu_l + (1 - \alpha) \mu_g, \quad \vec{U} = \alpha \vec{U}_l + (1 - \alpha) \vec{U}_g
\]  

(23)

Here, the subscripts \( l \) and \( g \) indicate the liquid and gas, while \( \rho \) is the density and \( \mu \) is the dynamic viscosity. \( \vec{U} \) is the mean velocity of fluid and it is transported with the function of \( \alpha \).

The VOF method can also be recognised as an isothermal fluid system as shown in Equation 24. Only a single set of governing equations is needed in this method and it is cheaper than the Eulerian-Lagrangian model, which has a number of governing equations and relying on the number of phases.

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{U}) = 0 \ ; \ \text{Incomp.} \Rightarrow \frac{\partial \vec{U}_i}{\partial x_i} = 0 \ ; \ i = x, y, z
\]

\[
\frac{\partial \vec{U}_i}{\partial t} + \vec{U}_j \frac{\partial \vec{U}_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_j} \left( \nu \frac{\partial \vec{U}_i}{\partial x_j} - \frac{\partial \vec{U}_j}{\partial x_i} \right) + f_i + \alpha \xi \nabla \alpha
\]

\[
\frac{\partial \xi}{\partial t} + \vec{U}_j \frac{\partial \xi}{\partial x_j} = -C_{1\varepsilon} \frac{\varepsilon}{k} \left( \vec{U}_j \frac{\partial \vec{U}_i}{\partial x_j} - \frac{\partial \vec{U}_i}{\partial x_j} \right) - C_{2\varepsilon} \frac{\varepsilon^2}{k} + \frac{\partial}{\partial x_j} \left[ \left( \nu + \nu_T \right) \frac{\partial \varepsilon}{\partial x_j} \right] - \frac{\partial \alpha}{\partial t} + \nabla \cdot (\alpha \vec{U}) + \nabla \cdot (\alpha (1 - \alpha) (\vec{U}_i - \vec{U}_g)) = 0
\]

(24)

where \( k \) is the turbulent kinematic energy, \( \nu_T \) is the turbulent kinematic viscosity, \( \nu \) is the kinematic viscosity, \( \sigma \) is the surface tension, \( \xi \) is the curvature of free surface, \( \varepsilon \) is the dissipation rate of the turbulent kinetic energy and \( C_{1\varepsilon}, C_{2\varepsilon} \) and \( \gamma_0 \) are characteristic constants for the \( k - \varepsilon \) model. In this study, following Ref. [18], the constants \( C_{1\varepsilon}, C_{2\varepsilon} \) and \( \gamma_0 \) are set as 1.44, 1.92 and 1.30, respectively.

2.7. Algebraic equations for the Navier-Stokes equations

The discretization of the transport equations creates a system of algebraic equations that is as indicated by Equation 25.

\[
[A] \phi = [R]
\]  

(25)
Here, \([A]\) is a sparse matrix, \([\phi]\) is the vector of \(\phi\)–s for all control volumes and \([R]\) is the source term vector. Matrix \([A]\) consists of all coefficients of the variables that are saved in the column vector \([\phi]\) and \([R]\) controls all right hand side terms [4]. The final form of the discretization process is shown by the equations below [4].

\[
a_p U_p = H(U) - \sum_f S(p)_f \quad (26)
\]

\[
\sum_f S \left[ \frac{1}{a_p} \right] \nabla p f = \sum_f S \left[ \frac{H(U)}{a_p} \right] \quad (27)
\]

where the flux is given by Equation 28 as follows:

\[
F = S \cdot U_f = S \left[ \frac{H(U)}{a_p} \right] - \left( \frac{1}{a_p} \right) \nabla p f \quad (28)
\]

The discretization of Equation 26 and Equation 27 needs a pressure-velocity coupling treatment to solve the equations [4]. In this study, Pressure Implicit Split Operator (PISO) is applied in order to solve the governing equations. PISO is a pressure-velocity treatment for the transient flow and it was introduced by Isaa [19] to improve Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) scheme [20]. In particular, when standard PISO algorithm is set, the complete procedure of interFOAM solver contains of the following steps [13, 21] (see Figure 3 for annotations):

- VOF equations are solved on basis of the old velocity field from the previous time step. Thus, this produces new values for the density and the volumetric of each cell.
- The PISO algorithm starts with the initial values. On the momentum predictor stage, previous time step is used to produce a new velocity field by the Equation 26.
- Then to satisfy the continuity equation, the value of the pressure field is corrected.
- From momentum equation, continuity equation is reconstructed to combine the new velocity field \(U^*\):

\[
U_f = S \left[ \frac{H(U^*)}{a_p} \right] - \left( \frac{1}{a_p} \right) \nabla p^* f \quad (29)
\]

\[
\sum_f S U^* f = 0 \quad (30)
\]

\[
\sum_f S \left[ \frac{H(U^*)}{a_p} \right] - \left( \frac{1}{a_p} \right) \nabla p^* f = 0 \quad (31)
\]

In the explicit velocity correction stage, the new corrected pressure \(p^*\) is used to produce the new corrected velocity field, \(U^{**}\). This step continues to obtain two corrected processes \((n = 2)\) by applying the new operator \(H(U^{**})\). These two corrected processes are adequate to attain the robust solution [8, 19].

3. Result and discussion

3.1. Grid refinement

The grid refinement study is applied to the three grid resolutions. Case A has the finest grid, Case B has the medium resolution and Case C has the coarsest resolution. These cases are shown in Table 4 for its specification and Figure 4 for its visualization. Richardson Extrapolation is acknowledged as “the deferred approach to the limit \((h \to 0)\)”. It determines a higher-order estimate of flow fields from a series of lower-order discrete values \((f_1, f_2, \ldots, f_n)\). The method of Richardson Extrapolation is shown in Equation 32 [22].

\[
f = f [\text{exact}] + g_1 h + g_2 h^2 + g_3 h^3 + \cdots \quad (32)
\]

where \(f\) is discrete solutions, \(h\) is grid spacing, and \(g_1\) and \(g_2\) are functions that are illustrated in the continuum and do not lean on any discretization. The quantity \(f\) is considered “second-order” when \(g_1 \neq 0\). The \(f_{h=0}\) is the continuum value at zero grid spacing.
Figure 3. PISO algorithm [8]
Without considering the non-appearance of odd power in Equation 32, Richardson Extrapolation can be generalized to $p$th order methods and $r$-value of grid ratio as follows:

$$f_{\text{exact}} \approx f_1 + \frac{(f_1-f_2)}{r^p - 1}$$

(33)

In this study, the grid refinement ratio, $r$ for the uniformed meshed is described as:

$$r = \frac{\text{Total no.of grid (Fine)}}{\text{Total no.of grid (Medium)}} = \frac{\text{Total no.of grid (Medium)}}{\text{Total no.of grid (Coarse)}} = 1.54$$

(34)

The refinement ratio is higher than the minimum value of 1.3. Equation 32 can be measured for order-of-accuracy by applying Equation 35.

$$p = \frac{\ln\left(\frac{\epsilon_{31}}{\epsilon_{32}}\right)}{\ln(r)}$$

(35)

$$\epsilon_{i+1} = f_{i+1} - f_i$$

(36)

Convergence conditions of this system must be clarified first in order to assess the extrapolated value from the equations above. The convergence conditions are listed as follow:

- Monotonic convergence: $0 < R < 1$
- Oscillatory convergence: $R < 1$
- Divergence: $R > 1$

where $R$ is the convergence ratio: $R = \frac{\epsilon_{31}}{\epsilon_{32}}$ (37)

For monotonic convergence, generalized Richardson Extrapolation is applied to estimate the errors and uncertainties, which are explained in Equation 37. For oscillatory convergence, the results show to exhibit some oscillations. Lastly, for divergence, the results diverge while errors and uncertainties are impossible to be determined [23]. Table 5 shows the results for order of accuracy for draining time, $t_{\text{drain}}$ and flow rate, $Q$. The results are calculated from three different types of mesh as defined in the previous Table 3.
Table 5. Grid Convergence Index (GCI) and order of accuracy for parameter: draining time, \( t_{\text{drain}} \) and flow rate, \( Q \)

| \( \varepsilon_{32} \) | \( \varepsilon_{21} \) | \( R \) | \( p \) | GCI \(_{32} \) [%] | GCI \(_{21} \) [%] |
|----------------|----------------|-----|-----|------------|------------|
| 8.000          | 6.500          | 0.8125 | 0.477 | 0.833      | 0.602      |
| 0.033          | 0.208          | 0.7230 | 0.748 | 0.522      | 0.427      |

*Index 1, 2 and 3 represent case A, B and C, respectively

The convergence conditions for draining time, \( t_{\text{drain}} \) and flow rate, \( Q \) are monotonic as the value of convergence ratio, \( R \) is more than zero and less than one for both cases. Hence these two parameters are applicable in the Grid Convergence Index (GCI) study. The draining time, \( t_{\text{drain}} \) can be expressed theoretically with the following Equation 38.

\[
t_{\text{drain}} = \sqrt{\frac{h_0 - h}{g}} \left( \frac{d_e}{d_n} \right)^2
\]

(38)

Here, \( d_e \) is the tank diameter, \( d_n \) is the nozzle diameter, \( t \) is the draining time, \( h_0 \) is the initial water level, \( h \) is the water level at a \( t \) time and \( g \) is gravitational acceleration. This equation is derived with the assumption that the flow is irrotational and inviscid. On another hand, the equation for the flow rate, \( Q \) can be estimated following in Equation 39, where \( v \) is flow velocity and \( A_n \) is cross-sectional area inside the outlet nozzle.

\[
Q = \int_{A_n} v \cdot dA_n
\]

(39)

Roache [22] has done a valuable contribution to systematic and most common methodology for the grid refinement studies, which is called Grid Convergence Index (GCI) method. GCI is based on grid refinement error estimator derived from the generalized Richardson Extrapolation. The percentage of differences between the computed and asymptotic values is calculated by using GCI. It demonstrates how far is the error of the computed value with the asymptotic value. How much the solution of the computed value would change with further refinement also can be illustrated with GCI. A small value of GCI percentage shows the computed value is approaching asymptotic range. The GCI for fine grid can be interpreted as shown in Equation 40.

\[
GCI_{i+1,i} = F_s \frac{|\varepsilon_{i+1,i}|}{f_i^{(r+p-1)}}
\]

(40)

where \( F_s \) is safety factor. As three different grids are used in this study, the safety factor should be \( F_s = 1.25 \) [24]. Otherwise, for the refinement with two grids, the safety factor is \( F_s = 3.0 \). Table 6 shows that GCI for parameter draining time, \( t_{\text{drain}} \) and flow rate, \( Q \) from three different meshes is good with the decrement value from GCI\(_{32} \) to GCI\(_{21} \) (GCI\(_{21} \) < GCI\(_{32} \)). The results demonstrate that the dependency of numerical method on the mesh size has been decreased when the GCI for the finer grid (GCI\(_{21} \)) is lower than the coarser grid(GCI\(_{32} \)). Therefore, the results of simulation will not be of much difference as further refinement of the grid is changed.

Table 6. GCI and order of accuracy for parameter: draining time, \( t_{\text{drain}} \) and flow rate, \( Q \)

| \( \varepsilon_{32} \) | \( \varepsilon_{21} \) | \( R \) | \( p \) | GCI \(_{32} \) [%] | GCI \(_{21} \) [%] |
|----------------|----------------|-----|-----|------------|------------|
| 8              | 6.5            | 0.8125 | 0.477 | 0.833      | 0.602      |
| 0.033          | 0.208          | 0.7230 | 0.748 | 0.522      | 0.427      |

*Index 1, 2 and 3 represent case A, B and C, respectively

Based on the results in Table 7, Figure 5 is plotted and it displays the monotonic convergence. It shows that the values obtained for three different meshes (fine, medium and coarse grids) for draining time, \( t_{\text{drain}} \) are just slightly lower than the Richardson extrapolated value. On the other hand, based
on the results in Table 8, Figure 6 is plotted and it shows a different line pattern from Figure 5. This demonstrates that the Richardson extrapolated value is slightly lower than the values obtained for three different meshes for flow rate, $Q$. However, both of these solutions are still in the range of finer GCI as shown in Figure 5 and Figure 6.

Table 7. Calculations of global results normalized by the Richardson extrapolated value for parameter draining time, $t_{\text{drain}}$

| Case       | $A$ (Fine), $f_1$ | $B$ (Medium), $f_2$ | $C$ (Coarse), $f_3$ | $f_{\text{RE}}$ |
|------------|-------------------|---------------------|---------------------|-----------------|
| $t_{\text{drain}}$ | 58.5              | 52                  | 44                  | 70.45           |

| $f_{\text{RE}}$ | $f_1$ | $f_2$ | $f_3$ | $f_{\text{RE}}$ |
|-----------------|-------|-------|-------|-----------------|
| 0.8303          | 0.738 | 0.525 | 1     |

Table 8. Calculations of global results normalized by the Richardson extrapolated value for parameter flow rate, $Q$

| Case       | $A$ (Fine), $f_1$ | $B$ (Medium), $f_2$ | $C$ (Coarse), $f_3$ | $f_{\text{RE}}$ |
|------------|-------------------|---------------------|---------------------|-----------------|
| $Q$        | 0.1843            | 0.2085              | 0.242               | 0.141           |

| $f_{\text{RE}}$ | $f_1$ | $f_2$ | $f_3$ | $f_{\text{RE}}$ |
|-----------------|-------|-------|-------|-----------------|
| 1.3             | 1.489 | 1.728 | 1     |

Figure 5. Comparison of parameter draining time, $t_{\text{drain}}$, between three different meshes and Richardson Extrapolation estimation

Figure 6. Comparison of parameter flow rate, $Q$, between three different meshes and Richardson Extrapolation estimation

Besides, error of these two parameters (draining time, $t_{\text{drain}}$, and flow rate, $Q$) can be defined from this Richardson extrapolated value and discrepancy between simulation value. The error is determined by Equation 41.

$$E_t = \left| \frac{f_t - f_{\text{RE}}}{f_{\text{RE}}} \right|$$  \hspace{1cm} (41)

Based on the calculation of RMS error, Figure 7 shows relative error between Richardson extrapolated value with the finer grid is only 0.169% for the parameter draining time, $t_{\text{drain}}$ and 0.216% for the parameter of flow rate, $Q$. These illustrate that the finer grid has approached asymptotic value, where the error due to the spatial discretization has been reduced significantly.
Figure 7. Percent of RMS error for parameter: (a) draining time, $t_{\text{drain}}$ and (b) flow rate, $Q$ with three different meshes and Richardson Extrapolation estimation.

3.2. Comparison of numerical data from Case A (fine grid) with previous data

Table 9 shows the comparison of draining time result for liquid level obtained in this study with that from studies by Park and Sohn [2]. It highlights that an excellent agreement is achieved between the results, especially with the experimental result from Park and Sohn.

| Case                           | Draining Time, $t_{\text{drain}}$ [s] |
|-------------------------------|-------------------------------------|
| Park & Sohn (Experimental)    | 58.16                               |
| Park & Sohn (Theoretical)     | 69.08                               |
| Park & Sohn (Numerical)       | 63.18                               |
| Current study (Case A)        | 58.50                               |

4. Conclusion

This grid refinement study was successfully achieved by using OpenFOAM framework. OpenFOAM displays a good result and simultaneously saves cost since no license is required. The flow behavior was simulated via PISO algorithm with first order accuracy of discretization scheme. The multiphase solver – interFoam and Volume of Fluid (VOF) were adopted to track the multiphase problem. The value of Grid Convergence Index (GCI) from coarser to finer grid was found to be decreasing as the grid was refined. The finer grid (Case A) was relevant to be applied in the future studies as the value of GCI is less than 1%.

Acknowledgements

This research was financially supported by Malaysia Ministry of Higher Education (MOHE) under Research University Grant (RUG) project of Universiti Teknologi Malaysia FRGS PY/2015/05383 and TWAS 13-272, and the use of High-Performance Computer (HPC) Universiti Teknologi Malaysia facilities was also appreciated.

References

[1] Zubair M, Abdullah M Z and Ahmad K A 2013 Computational and Mathematical Methods in Medicine 2013
[2] Park I S and Sohn C H 2011 International Communications in Heat and Mass Transfer 38 1044-9
[3] Versteeg H K and Malalasekera W 2008 An Introduction to Computational Dynamics - The Finite Volume Method (Essex: Longman Scientific & Technical)
[4] Jasak H 1996 Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows PhD Thesis University of Zagreb
[5] Rechiman L M, Centero M I and Dari E A 2014 de Mecanica Computacional XXXIII 2927-38
[6] Yu B, Tao W Q, Zhang D S and Wang Q W 2001 *Numerical Heat Transfer, Part B: Fundamentals* 40 343-65
[7] Al-Rabeh A 1993 *Computer Methods in Applied Mechanics and Engineering* 109 131-41
[8] Ali M S M 2011 *Flow and Noise Associated with The Interaction of A square Cylinder with A Downstream Flat Plate* PhD Thesis University of Adelaide
[9] Shyy W 1985 *Journal of Computational Physics* 57 415-38
[10] Papadakis G and Bergeles G 1995 *International Journal of Numerical Methods for Heat & Fluid Flow* 5 49-62
[11] Ferziger J H and Peric M 1996 *Computational Methods for Fluid Dynamics* (Berlin: Springer)
[12] Robertson E, Choudhury V and Walters D 2015 *Computer and Fluids* 123 122-45
[13] Shulze L and Thorenz C 2014 *11th International Conference on Hydroscience & Engineering*
[14] Hirsch C 1991 *Numerical Computation of Internal and External Flow* (Oxford: Butterworth-Heinemann)
[15] Courant R, Friedrichs K and Lewy H 1967 *IBM J. Res. Dev.* 11 215-34
[16] Jasak H, Weller H and Gosman A 1999 *International Journal for Numerical Methods in Fluids* 31 431-49
[17] Rusche H 2002 *Computational Fluid Dynamics of Dispersed Two-phase Flows at High Phase Fractions* PhD Thesis Imperial College London
[18] Rechiman L M, Centero M I and Dari E A 2014 *de Mecanica Computacional* XXXIII 2927-38
[19] Isa R I 1986 *Journal of Computational Physics* 62 40-65
[20] Patankar S 1980 *Numerical Heat Transfer and Fluid Flow* (Boca Raton: CRC Press)
[21] Damian S M 2013 *An Extended Mixture Model for the Simultaneous Treatment of Short and Long Scale Interfaces* PhD Thesis Universidad Nacional Del Litoral
[22] Roache P J 1994 *Journal of Fluids Engineering* 116 405-41
[23] Stern F, Wilson R V, Coleman H W and Paterson E G 2001 *Journal of Fluids Engineering* 123 793-802
[24] Wilcox D C 2006 *Turbulence Modelling for CFD* (California: D C W Industries)