Chapter 5: CFD Theory and Models

5.1 The commercial software ANSYS CFX

ANSYS-CFX is a general purpose computational fluid dynamic software suite that combines an advanced solver with powerful pre- and post-processing capabilities. It includes the following features: An advanced coupled solver that is both reliable and robust, Full integration of problem definition, analysis, and results presentation, An intuitive and interactive setup process, using menus and advanced graphics. ANSYS CFX consists of four software modules: ANSYS CFX-Pre, ANSYS CFX-Solver, ANSYS CFX-Solver Manager and ANSYS CFD-Post.

- *ANSYS CFX-Pre:* is used to define the simulation. The first step is importing the generated meshes in CFX- Pre. Secondly is to specify the physics characters of materials, boundary conditions, initial values and solver parameters. The material databank includes all the commonly used materials, for example, air, water steam, steel and etc. The boundary conditions with a full range of boundary types of inlets, outlets, symmetry and wall specify the boundary details of mass and momentum, heat transfer models and periodicity, pressure level and so on.

- *ANSYS CFX-Solver*: solves all the solution variables for the simulation for the problem specification generated in CFX-Pre. ANSYS CFX-Solver Manager The CFXSolver Manager module is used to control and manage the CFD task. Its major functions includes: specify the input files to the CFX-Solver, start and stop the CFX-Solver, monitor the progress of the solution, set up the CFX-Solver for a parallel calculation.

- *ANSYS CFX-Post:* provides state-of-the-art interactive post-processing graphics tools to analyze and present the CFX simulation results. The important features include quantitative post-processing, state file input, user-defined variables, generation of a variety of graphical objects where visibility, transparency, color, and line/face rendering can be controlled and power Syntax to allow fully programmable session files.

5.2 Mesh generation in ICEM CFD

ANSYS ICEM CFD is used as an integrated mesh generation and post processing tool in engineering applications such as fluid dynamics and structural analysis. It provides advanced geometry acquisition, mesh generation, mesh optimization, and post-processing tools.

5.2.1 Geometry Tools

The geometry tools of ANSYS ICEM CFD can create new and manipulate existing geometry. Furthermore the existing geometries can also be imported in the ANSYS ICEM CFD from their native CAD Interfaces, for example, CATIA, Pro/E and Solid Works. All the geometries used in this thesis work have been created in ANSYS ICEM CFD.

5.2.2 Mesh Tools

ANSYS ICEM CFD's mesh generation tools offer the capability to parametrically create meshes from geometry in numerous formats, for instance, Multiblock structured, Unstructured hexahedral, pyramidal and/or prismatic elements and Quadrilateral and triangular surface meshes etc. In general the meshing modules refer to tetra-, hexa- and prism meshes, hybrid meshes and the shell meshing. A short explanation is shown.

- ANSYS ICEM CFD Hexa Mesh: ICEM CFD Hexa module is a semi-automated meshing module and presents rapid generation of multi-block structured or unstructured hexahedral volume meshes. Blocks can be interactively adjusted to the underlying CAD geometry. The system can generate body fitted internal or external O-Grids automatically. The grid is projected onto the underlying CAD geometry automatically.

- ANSYS ICEM CFD Tetra Mesh: ICEM CFD Tetra is an object oriented unstructured meshing technology. It works directly from the CAD surfaces and fills the volume with tetrahedral elements using the Octree approach. This automatic mesh generation tool is suitable to complex geometries and offers tools for local adaptive mesh refinement and coarsening.

- ANSYS ICEM CFD Hybrid Mesh :
  Hybrid tetrahedral grids consisting of layers of prism elements near the boundary surfaces and tetrahedral elements in the interior can be generated with ICEM CFD Prism. In comparison to pure tetrahedral grids, hybrid tetrahedral grids with near-surface prism layers allow for a more adequate modeling of the close-to-wall physics of the flow field resulting in smaller analysis models, better convergence of the solution, and better analysis results.

5.2.3 Mesh Quality

A good mesh quality is essential for performing a good CFD analysis. Therefore, assessment of the mesh quality before performing large and complex CFD analysis is very important. ICEM/CFD offers the possibility to check the mesh parameters, such as grid angles, aspect ratios etc. General recommendations of the Best Practice Guidelines for the use of CFD in Nuclear Reactor Safety Applications [Best Guidelines(2007)] for generating high quality grids are:

- Avoid grid angles below 20° and above 160°
- Avoid jumps in grid density: Growth factors between adjacent volumes should be smaller than 2.
- Avoid non-scalable grid topologies: Non-scalable topologies can occur in block structured grids and are characterized by a deterioration of grid quality under grid refinement.
- Avoid grid lines which are not aligned with the flow direction (e.g. tetrahedral meshes, in thin wall boundary layers). Computational cells which are not aligned with the flow direction can lead to significantly larger discretization errors.
- Avoid high grid aspect ratios: This criterion depends on the flow solver. For standard iterative solvers, aspect ratios should not be larger than 10 to 50 to obtain convergent solutions. Solvers with multi-grid acceleration can absorb higher aspect ratios.
- Use a finer and more regular grid in critical regions, e.g. regions with high gradients or large changes such as free surfaces.
- Avoid the presence of non-matching grid interfaces in critical regions. An arbitrary grid interface occurs when there is no one-to-one correspondence between the cell faces on both sides of a common geometry face.
- In areas where local details are needed, the local grid refinement can be used to capture fine geometrical details. If grid refinement is used, the additional grid points should lie on the original boundary geometry, and not simply be a linear interpolation of more grid points on the original coarse grid.

These guidelines have been taken into account for our work, as it can be seen in Chapter 7.

5.3 Fundamental equations of fluid dynamics

The fluid is considered as a continuous medium and its behavior is described by the three conservation laws of physics for macroscopic media that were already introduced in chapter 2.

- The conservation of mass as in Eq.2.13

- The conservation of linear momentum as in Eq.2.14

- The conservation of energy: Having the same form of Eq.2.15. But it may be interesting to comment that an alternative form of this energy equation, which is suitable for low-speed flows is also available in ANSYS CFX, the Thermal Energy Equation. By taking the dot product of $\vec{v}$ with the momentum equation and substracting it from the total energy equation it is obtained

$$\frac{\partial(\rho \cdot h)}{\partial t} - \frac{\partial p}{\partial t} + \nabla.(\rho \cdot h \cdot \vec{v}) = \nabla \cdot (\lambda \nabla T) + \vec{v} \nabla \mathrm{p} + \overline{\overline{\tau}} \colon \nabla \vec{v} + S_E \qquad \text{Eq.5. 1}$$

The thermal energy equation follows by interpreting the enthalpy h as the internal energy e, thus eliminating the term P/ρ, and neglecting the term $-p\nabla\vec{v}$. It has the form:

$$\frac{\partial(\rho \cdot h)}{\partial t} + \nabla.(\rho \cdot h \cdot \vec{v}) = \nabla \cdot (\lambda \nabla T) + +\overline{\overline{\tau}} \colon \nabla \vec{v} + S_E \qquad \text{Eq.5. 2}$$

This approach is applicable for low-speed flows and fluids where variable density effects are negligible. But in our case the general total energy equation will be used.

5.4 Turbulence models

As it has been remarked in the previous chapters, a turbulence model is a computational procedure to close the system of mean flow equations, this is to be able to calculate the Reynolds stresses and the scalar transport terms. As said, the employed models belong to the eddy viscosity hypothesis approach, for the continuous liquid phase, 3 models will be seen, and for the gaseous disperse phase only the one used, a zero equation model

5.4.1 Continuous or liquid phase

Although there is no need to say it, it is to be reminded to the reader, that all the variables below, although, without subscripts, correspond to the liquid phase.

The $k - \varepsilon$ turbulence model

The model $k - \varepsilon$ is by far the most popular for the calculation of turbulent flows. It has the advantage that no geometry-related parameters are included in the modeling. A disadvantage of this model for numerical calculations is that the coupling of the conservation equations with the overall system of equations is very stiff and shows a slow convergence. The $k - \varepsilon$ model introduces two variables in the system of equations, the turbulent kinetic energy k and the turbulence dissipation rate ε. The effective viscosity that operates with the velocity gradients is defined as

$$\mu_{eff} = \mu + \mu_t \qquad \text{Eq.5. 3}$$

Where μ_t is the turbulence viscosity. The $k - \varepsilon$ assumes the following relation between the turbulence viscosity, k and ε.

$$\mu_t = C_\mu \rho \cdot \frac{k^2}{\varepsilon} \qquad \text{Eq.5. 4}$$

Where C_μ is a constant. The two transport equations for k (kinetic energy) and ε (its dissipation rate) are:

$$\frac{\partial \rho k}{\partial t} + \nabla(\rho \vec{u} k) = \nabla\left[\left(\mu + \frac{\mu_t}{\sigma_k}\right)\nabla k\right] + P_k + P_{kb} - \rho\varepsilon \qquad \text{Eq.5. 5}$$

$$\frac{\partial \rho \varepsilon}{\partial t} + \nabla(\rho \vec{u} \varepsilon) = \nabla\left[\left(\mu + \frac{\mu_t}{\sigma_\varepsilon}\right)\nabla \varepsilon\right] + \frac{\varepsilon}{k}\left[C_{\varepsilon 1}(P_k + P_{\varepsilon b}) - C_{\varepsilon 2}\rho\varepsilon\right] \qquad \text{Eq.5. 6}$$

Where $C_{\varepsilon 1}$, $C_{\varepsilon 2}$, σ_{ε} and σ_k are constants. P_k is the turbulent generation and is determined by:

$$P_k = \mu_t \nabla \vec{u} \cdot (\nabla \vec{u} + \nabla \vec{u}^T) - \frac{2}{3} \nabla \cdot \vec{u} (3\mu_t \nabla \cdot \vec{u} + \rho k) \qquad \text{Eq.5. 7}$$

P_{kb} and $P_{\varepsilon b}$ account for the influence of buoyancy forces. When the full buoyancy model is used, P_{kb} takes the form:

$$P_{kb} = -\frac{\mu_t}{\rho \sigma_\rho} g \cdot \nabla \rho \qquad \text{Eq.5. 8}$$

And for the Boussinesq buoyancy model:

$$P_{kb} = -\frac{\mu_t}{\rho \sigma_\rho} \beta g \cdot \nabla \rho \qquad \text{Eq.5. 9}$$

$P_{\varepsilon b}$ is taken proportional to P_{kb} and must be positive

$$P_{\varepsilon b} = C_3 \max{(0, P_{kb})} \qquad \text{Eq.5. 10}$$

<u>The $k - \omega$ turbulence model</u>

Another widely used two-equation turbulence model is the $k - \omega$ model by Wilcox [Wilcox(1998)]. In this model, two transport equations are solved for kinetic energy k and for the characteristic turbulent frequency ω. The $k - \omega$ assumes the following relationship between k, ω and μ_t:

$$\mu_t = \rho \frac{k}{\omega} \qquad \text{Eq.5. 11}$$

The two transport equations for k and ω are the following

$$\frac{\partial \rho k}{\partial t} + \nabla(\rho \vec{u} k) = \nabla \left[\left(\mu + \frac{\mu_t}{\sigma_k} \right) \nabla k \right] + P_k + P_{kb} - \beta' \rho k \omega \qquad \text{Eq.5. 12}$$

$$\frac{\partial \rho\omega}{\partial t} + \nabla(\rho\vec{u}\omega) = \nabla\left[\left(\mu + \frac{\mu_t}{\sigma_\omega}\right)\nabla\omega\right] + \alpha\frac{\omega}{k}P_k + P_{\omega b} - \beta\rho k\omega \qquad \text{Eq.5. 13}$$

Where β', β, α, σ_k and σ_ω are constants. $P_{\omega b}$ is calculated by:

$$P_{\omega b} = \frac{\omega}{k}[(\alpha+1)C_3 \cdot \max(o, P_{kb}) - P_{kb}] \qquad \text{Eq.5. 14}$$

The Shear Stress Transport (SST) turbulence model

A major boundary layer effect is the separation from a surface under adverse pressure gradient conditions. Separation has a strong effect on the near-wall turbulence and therefore on the turbulent heat transfer. The SST model, has demonstrated the capability of accurate separation predictions in numerous cases. The idea behind the SST model is to combine the best elements of the $k-\varepsilon$ and the $k-\omega$ model with the help of a blending function *F1*. Blending means that both models are taken into account in the transition region by a proportion that depends on the gradients. *F1* is one near the surface and zero in the outer part and for free shear flows. It activates the Wilcox model in the near-wall region and the k-e model for the rest of the flow. By this approach, the attractive near-wall performance of the Wilcox model can be utilized without the potential errors resulting from the free stream sensitivity of that model.

Although, unlike single phase fluid flow problem, no standard turbulence model is tailored for gas-liquid flow, numerical investigation has revealed that standard $k-\varepsilon$ model tends to predict unrealistically high gas void fraction peak close to the wall [Frank et al.(2004)].The shear stress transport turbulence model was developed by [Menter (1994)], applying the two-equation $k-\omega$ model near the wall and the two-equation $k-\varepsilon$ model in the bulk flow, as it has already been said above. This has been shown to provide more realistic prediction of gas volume fraction close to the wall of the flow domain. The advantage of the SST over the $k-\varepsilon$ model can be seen for adverse pressure gradient conditions.

Besides combining these two models, the SST model complements them with a special equation for the eddy viscosity, offering higher reliability and accuracy while maintaining stability and computing time. An additional feature of the SST model is the introduction of an upper limit for the turbulent shear stress in boundary layers in order to avoid excessive shear-stress levels typically predicted with Boussinesq eddy-viscosity models.

5.4.2 Disperse or Gas phase

Dispersed Phase Zero Equation Model

For the gas, in bubbly flow, a dispersed phase zero equation model is generally used, modeling the turbulent viscosity of the gas phase as a function of the turbulent viscosity of the continuous phase, as follows:

$$\nu_{td} = \frac{\nu_{tc}}{\sigma} \rightarrow \mu_{td} = \frac{\rho_d}{\rho_c}\frac{\mu_{tc}}{\sigma} \qquad \text{Eq.5. 15}$$

The parameter σ is a turbulent Prandtl number relating the dispersed phase kinematic eddy viscosity ν_{td} to the continuous phase kinematic eddy viscosity ν_{tc}.

5.5 Near Wall Treatment

An important issue in the accurate prediction of industrial turbulent flows is the formulation and the numerical treatment of the equations in regions close to solid walls. The near-wall formulation determines the accuracy of the wall shear stress and the wall heat transfer predictions and has an important influence on the development of boundary layers, including the onset of separation. Typically the two following approaches are used to model the flow in the near-wall region:

- The wall-function method: In the scalable wall-function approach, the viscosity affected sublayer region is bridged by employing empirical formulas to provide near-wall boundary conditions for the mean flow and turbulence transport equations. These formulas connect the wall conditions (e.g. the wall shear stress) to the dependent variables at the near-wall grid node, which is presumed to lie in the fully-turbulent region of the boundary layer. The major advantage of the wall function approach is that it conserves valuable computer resources and it avoids the need to account for viscous effects in the turbulence model. One of the major drawbacks of the wall-function approach is that the predictions depend on the location of the point nearest to the wall and are sensitive to the near-wall meshing; refining the mesh does not necessarily give a solution of increasing accuracy. The problem of inconsistencies in the wall-function in case of fine grids can be overcome with the use of the scalable wall-function formulation developed by CFX. This approach is available for all high Re models.

- The low-Reynolds-number method: An alternative approach to the use of wall-functions is to use a fine-grid analysis extending through the viscosity-affected sublayer close to the wall. The low-Re approach requires a very fine grid in the near-wall zone. Computer storage and runtime requirements are usually larger than those of the wall-function approach and care must be taken to ensure good numerical resolution in the near-wall region to capture the rapid variation of the variables. This imposes strong restrictions on the grid resolution in the near-wall region.

As it has been explained, wall functions are not always desirable, as they neglect the influence of the viscous sublayer. Especially for flows at low device Reynolds numbers, the omission of the sublayer can have a significant effect on the solution. For example, the mass flow rate in a pipe or channel flow at low Reynolds numbers can be in error by 10 % and more, due to the relatively large influence of the viscous portion of the boundary layer. For external flows, the effect is usually smaller, as the displacement of the outer flow is then a second order effect. It was therefore desirable to offer the user a formulation, which will automatically switch from wall-functions to a low-Re formulation, as the grid is refined, and so was done in Ansys CFX with the automatic near wall treatment. [Ansys2(2009)]

The idea behind *the automatic near-wall treatment* is that the model shifts gradually between a viscous sublayer formulation and wall functions, based on the grid density. As already pointed out, low-Re k-e based models do generally not satisfy the requirements imposed by complex industrial flow simulations. The current method is therefore based on the k-w near wall formulation. The ω-equation is well suited for this task, as it provides analytical solutions, both for the sublayer and the logarithmic region. A blending function depending on $y+$ is therefore, designed, to see it in detail please go to the cited reference [Ansys2(2009)].
The automatic near wall treatment was then, employed in this work.

5.6. Closure models for bubble coalescence and breakup Γ, Ω

In CFX the models of [Luo and Svendsen (1996)] and [Prince and Blanch (1990)] are the default closure model for the MUSIG approach. The closure model is one weak point and limits the application of MUSIG approach. So other proposals have been made for example in [Liao et al.(2010)].

5.6.1 The breakup kernel function

Of Luo and Svendsen (1996), considers the collision effect of turbulence eddy. According to the model, breakage of a bubble occurs only when the kinetic energy of the bombarding eddy is large enough and exceeds the increase in surface energy required for the breakage, which is called energy constraint [Wang et al.(2003)].Break up kernels are often expressed as a function of the break up function $f_{BV} = \frac{m_j}{m_i}$

$$\Omega(m_i, f_{BV}m_i)\ [\tfrac{1}{s}] = F_B \cdot \frac{0.923}{d_i^{\frac{2}{3}}} \varepsilon^{\frac{1}{3}}\left(1-\alpha_g\right) \int_{\xi_{min}}^{1} \frac{(1+\xi)^2}{\xi^{\frac{11}{3}}} \cdot \exp\left(-\chi\right) d\xi \qquad \text{Eq.5. 16}$$

$$\chi = \frac{12(f_{BV}^{\frac{2}{3}} + (1-f_{BV})^{\frac{2}{3}} - 1)\sigma}{\beta \rho_l \varepsilon^{\frac{2}{3}} d_i^{\frac{5}{3}} \xi^{\frac{11}{3}}} \qquad \text{Eq.5. 17}$$

5.6.2 The coalescence kernel function

Of [Prince and Blanch (1990)], assumes that the coalescence of two bubbles occurs in three steps. First, the bubbles collide trapping a small amount of liquid between them. This liquid film then drains until the liquid film separating the bubbles reaches a critical thickness. The film them ruptures and the bubbles join together. The coalescence kernel is therefore modeled by a collision rate of two bubbles and a collision efficiency relating to the time required for coalescence:

$$\Gamma(m_i, m_j)\ [\tfrac{1}{m^3 s}] = \left(\theta_{ij}^T + \theta_{ij}^B + \theta_{ij}^S\right)\eta_{ij} \qquad \text{Eq.5. 18}$$

Being the three terms turbulence, buoyancy and shear contribution (this last one is neglected in our case).The collision efficiency is modeled by comparing the time required for coalescence t_{ij} with the actual contact time during the collision τ_{ij}, Where the efficiency has the form of Eq.4. 22**,** and the drainage time and the contact time of Eq.4. 19 and Eq.4. 21 respectively (although in these equations the subscripts have not been considered).

r_{ij} , the equivalent radius, is defined as:

$$r_{ij} = \left(\frac{1}{2}\left(\frac{1}{r_i} + \frac{1}{r_j}\right)\right)^{-1} \qquad \text{Eq.5. 19}$$

The turbulent contribution to collision frequency is:

$$\theta_{ij}^T = F_{Cturbulence} S_{ij}\left(u_{ti} + u_{tj}\right)^{1/2} \qquad \text{Eq.5. 20}$$

Where the cross-sectional area of the colliding particles and the turbulent velocity are:

$$S_{ij} = \frac{\pi}{4}\left(r_i + r_j\right)^2 \qquad \text{Eq.5. 21}$$

$$u_{ti} = \sqrt{2}\varepsilon^{1/3} d_i^{1/3} \qquad \text{Eq.5. 22}$$

Where again the cross-sectional area is the same considered by Yao and Morel, being $F_{Cturbulence}$ a calibration factor.

The buoyancy contribution is:

$$\theta_{ij}^{B} = F_{Cbouyancy} S_{ij} \left| U_{rj} - U_{ri} \right| \qquad \text{Eq.5. 23}$$

Where $F_{Cbouyancy}$ is another calibration factor and

$$U_{ri} = \sqrt{\frac{2.14\sigma}{\rho_l d_i} + 0.505 g \cdot d_i} \qquad \text{Eq.5. 24}$$

5.7 Implementing one group Interfacial area transport equation in Ansys CFX

Ansys CFX allows defining a transport equation for any scalar quantity by a field velocity. The general form of this transport equation, where we have already substituted the interfacial area concentration is:

$$\frac{\partial(\alpha a_i)}{\partial t} + \nabla.(\vec{v}\alpha a_i) - \nabla.\left[\alpha\left(D_g^{a_i} + \frac{\mu_{td}}{Sc_{t,d}}\right)\right] = S_{a_i} + T_{a_i} \qquad \text{Eq.5. 25}$$

Where a_i is the interfacial area concentration per unit volume, α is the void fraction, $D_g^{a_i}$ is the kinematic diffusivity for the scalar to be transported and is normally set by the user, $Sc_{t,d}$ is the turbulent Schmidt number of the disperse phase, S_{a_i} is the external volumetric source term, and T_{a_i} is the total source due to inter-phase transfer across interfaces with other phases, and is not considered in the adiabatic case.

This equation differs from Eq.4. 4 in the presence of the diffusive term (third term) and of the void fraction inside the derivative terms.

Since the form of the transport equation cannot be modified by the user, a possible strategy to overcome the constraints of the software has be to found.

5.7.1 Eliminating the diffusive term

Three different attempts have been realized to reduce or eliminate the influence of the diffusive term:

- Define the Turbulent Schmidt number like it follows

$$Sc_{t,d} = -\frac{\mu_{td}}{D_g^{a_i}} \qquad \text{Eq.5. 26}$$

in order to obtain a constant value of zero inside the divergence of the diffusive term.

- Implement the complete term with opposite sign on the right hand of the equation Eq.4. 4

- Reduce the influence of the diffusive term without eliminating it, as suggested by Customer Support of Ansys, by setting a low value for the kinematic diffusivity D (1e-15) and a sufficient high value for the Schmidt Number (1000 or above), in our case, 10000 was set.

The first two attempts led the solver to become very instable and it has not been possible to reach convergence for any of the cases under exam. Overflows have also been experienced.

The strategy at point three resulted in the only viable way to reduce the influence of the diffusion term.

The derivatives in the source term were calculated by implementing “UserCel” functions and routines in the code.

5.7.2 The Transformed Source Term

The presence of the void fraction multiplying the transported variable in Eq.5. 25 leads to the necessary implementation of a supplementary term on the right hand of the equation. This term takes in account the presence of supplementary mixed derivative terms in the equation with respect to Eq.4. 4.

Therefore, the final form of the source term to be implemented, and shown in, is:

$$S_{a_i} = \alpha \left[\sum_j \phi_j + \left(\frac{2a_i}{3\alpha}\right)\left(\frac{\partial \alpha}{\partial t} + \nabla . \vec{v_i}\alpha\right)\right] + a_i\left(\frac{\partial \alpha}{\partial t} + \vec{v_i}\nabla\alpha\right) \qquad \text{Eq.5. 27}$$

The first term is implemented as “Source” it means that it is applied to the bulk of the fluid and then multiplied for the relative volume fraction automatically by the code. The second term is defined as “Fluid Source” and is applied directly to the related fluid.

Like the derivatives in the Source term, the sink/source terms for interfacial area concentration, and some other variables were implemented either with “UserCel” functions, or with subroutines prepared in Fortran.