Numerical Modeling of Horizontal Buoyant Jet Entering Shallow Water with a Crossflow

Ruoyao Ou

Thesis submitted to the University of Ottawa
in partial fulfillment of the requirements for the
Master of Applied Science in Environmental Engineering

Department of Civil Engineering
Faculty of Engineering
University of Ottawa

Abstract

Numerical methods have been widely used for prediction in various complex physical problems such as industrial outfall discharges. The discharge of industrial effluent from the outfall systems can be divided into two categories on the basis of whether salinity or temperature is the inducement to the density difference. The salinity induced buoyant effluent, which is called negatively buoyant jet, has a density higher than the receiving water, thus tending to sink. The temperature induced jet, which is called positively buoyant jet, has a density lower than the receiving water, thus tending to rise. In the present work, the temperature induced horizontal buoyant jet entering shallow water with a uniform crossflow is investigated by numerical simulations using a modified solver in OpenFOAM (twoLiquidMixingFoam). Various turbulence models have been applied in the numerical model to evaluate the effectiveness of these models in predicting the jet behavior. These numerical results are compared with data obtained from a previous experimental study by Johnston et al. (1993). The simulations are run with a number of sets of crossflow-to-jet velocity ratios and densimetric Froude numbers in order to find out their influences on the jet behavior. Additionally, the bed attachment effect, which is one of the major features of the jet, is also investigated. In conclusion, it was found that the k-Equation LES turbulence model outperformed the other two models in terms of the consistency with the experimental data. Both employed RANS models (realizable $k-\varepsilon$ and SST $k-\omega$ model) have weakness in predicting the bed attachment effect. However, they are still capable of predicting the general density distribution in flow field when the bed attachment effect is not present.

Acknowledgements

I would like to express my sincere gratitude to my academic supervisors, Dr. Majid Mohammadian and Dr. Ioan Nistor, for their continued guidance and support throughout all stages of this study.

I also want to thank all the faculty members and my peers at the Department of Civil Engineering at the University of Ottawa for providing valuable advices and suggestions.

Furthermore, I would like to thank my parents for all their help and continuous support since childhood.

Table of Contents

List of Figures

List of Tables

List of Acronyms

CFD: Computational Fluid Dynamic

DNS: Direct Numerical Simulation

LES: Large Eddy Simulation

SGS: Sub-Grid Scale

RANS: Reynolds Averaged Navier–Stokes

SST: Shear Stress Transport

RSM: Reynold Stress Model

DE: Differential Equation

PDE: Partial Differential Equation

FDM: Finite Difference Method

FVM: Finite Volume Method

FEM: Finite Element Method

PISO: Pressure-Implicit with Splitting of Operators

PIMPLE: Semi-Implicit Method for Pressure-Linked Equations

LIF: Laser Induced Fluorescence

JICF: Jet in Crossflow Flow

CVP: Counter-rotating Vortex Pair

MDNF: Momentum Dominated Near Field

MDFF: Momentum Dominated Far Field

BDNF: Buoyancy Dominated Near Field

BDFF: Buoyancy Dominated Far Field

List of Symbols

k: Turbulent kinetic energy [m^2/s^2];

k_{SGS}: Sub-grid scale kinetic energy [m^2/s^2];

ε: Dissipation rate for k in $k-\varepsilon$ models [m^2/s^3];

ω: Dissipation rate for k in $k-\omega$ models [m^2/s^3];

Re: Reynold number;

Re_d: Reynold number of jet nozzle;

Ri: Richardson number;

F_0: Densimetric Froude number;

μ: Dynamic viscosity [kg/m·s];

μ_t: Eddy viscosity [kg/m·s];

v: Kinematic viscosity [m^2/s];

v_{SGS}: Sub-grid scale eddy viscosity;

u_1, u_2, u_3, u_i, u_j: Velocity components [m/s];

u': The fluctuation of the velocity component [m/s];

u_j: Velocity of jet [m/s];

u_a: Velocity of crossflow [m/s];

ρ_j: Density of jet [kg/m^3];

ρ_a: Density of crossflow [m/s];

g: Gravity acceleration [m/s^2];

g': Modified gravity acceleration [m/s^2];

η: Kolmogorov length scale [m];

τ_η: Kolmogorov time scale [s];

τ_{ij}: Reynold tensor stress [$kg/m{\cdot}s^2$];

δ_{ij}: Kronecker delta;

Δ: Cut-off width in LES models [m];

C, b, M, N, n, β: Constants in numerical or mathematical models;

$\sigma_{k,\varepsilon}$: Turbulent Prandtl number for k and ε, respectively;

D_{ij}: Strain rate tensor [s^{-1}];

α_i: Volume fraction of ith fluid;

D_{ab}: Molecular diffusivity between miscible fluids;

d: Diameter of nozzle [m];

θ_o: Discharge angle of jet [°];

Q_0: Mass flux [m^3/s];

M_0: Momentum flux [m^4/s^2];

B_0: Buoyancy flux [m^4/s^3];

Z: Jet elevation [m];

z_{cl}: Vertical elevation of jet trajectory [m];

r: Effective velocity ratio in JICF;

$L_{M,m,B,b}$: Length scales for different regimes in JICF [m];

$G_{M,m,B,b}$: Buoyancy scales for different regimes in JICF[m/s^2];

$\Delta\rho_o$: Density deficit [kg/m^3];

ρ': Relative density;

H: Total depth of water [m];

h: Height of jet nozzle [m];

s: Length of streamwise direction [m];

S_o: Averaged dilution at jet centerline;

T_1: Initial temperature of jet [℃];

T_2: Initial temperature of crossflow [℃];

Chapter 1 Introduction

1.1 Effluent discharge and its environmental impacts

Surface and ground water can be polluted mainly from two sources. One source is from agriculture, where the residue of agricultural fertilizers is the major pollutant. The other is from the discharge of industrial effluents in urban area (Phiri et al., 2005). According to Environment Protection Agency (EPA) of US, the effluent is defined as any form of treated or untreated waste water discharging from a treatment plant or outfall. The discharge of effluents, which is either highly saline or thermal, into ambient water bodies could cause some major environmental impacts since they can deteriorate the water quality, thus threating the well-being of the aquatic creatures. Even worse, industrial effluents, if not properly treated and controlled, could bring toxic chemical into rivers, which tends to accumulate downstream.

In some arid or semi-arid regions, desalination plant is an important facility that helps to alleviate the increasing demand for fresh water due to the population and economic growth. Except for fresh water, desalination plants also generate by-product brine, which normally has a salinity concentration almost twice of the typical seawater (Purnama and Shao, 2015). The open discharge of the concentrated brine into water bodies could degrade the physical, chemical and biological environment of the receiving water bodies (Alameddine and Fadel, 2007). For example, Purnama et al. (2003) found that surface discharge of the concentrated brine has negative impact on the coastal water and promotes seawater intrusion. The brine discharge also effects the biodiversity of littoral fauna and flora according to the data collected by Einav et al. (2003) close to the Dhkelia plant in Cyprus. Therefore, the optimal design of outfall structure is important to minimize these environmental impacts. The general process is summarized in **Figure 1.1** below.

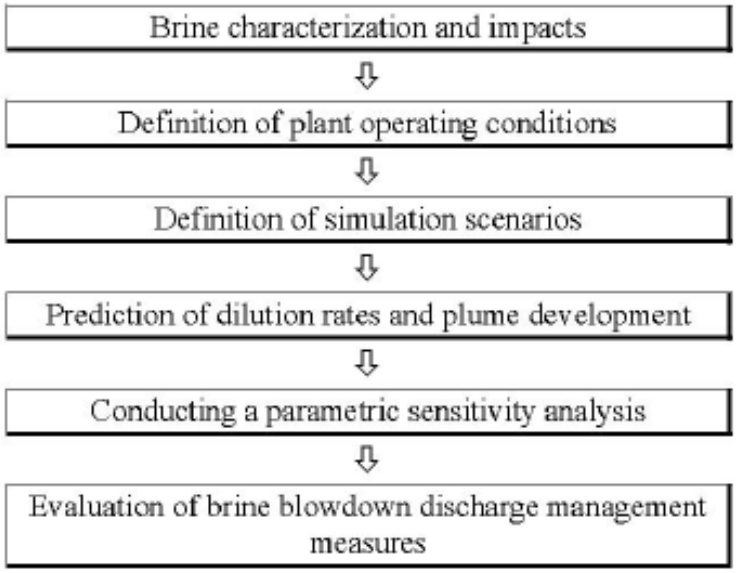


Figure 1.1 The general processes of brine discharge assessment (Alameddine and Fadel, 2007).

The other type of effluent discharge is the thermal discharge. The production of thermal discharge in industry is mainly through thermal power plant. Due to the energy loss during conversion, a large amount of thermal energy is discharged into water bodies, which create high temperature zone in the vicinity of the outfall. This could have great ecological impacts on the aquatic environment, including high mortality and low reproduction rate of some fish species, alteration of local phytoplankton community structure and even loss of biodiversity (Poornima et al., 2005). A typical thermal distribution field of the power plant outfall system can be seen in **Figure 1.2**. Evaluation of the mixing and spread of thermal effluents is essential for the design of thermal discharge outfall in any water environment. Zhao et al. (2015) summarized the important aspects regarding the transport of thermal effluents, which include velocity distribution, gradient of temperature field, volume, buoyancy and momentum fluxes, jet trajectory and cling length, jet bifurcation, entrainment, presence of coherent vortex systems and local geographic settings. With the great development in computational power, numerical modelling becomes one of the most efficient methods to predict the thermal distribution in various conditions.

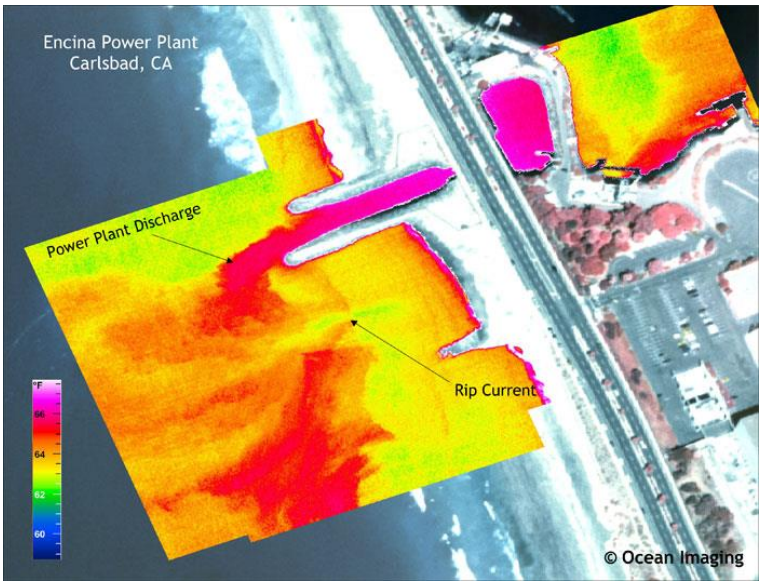


Figure 1.2 Imagine of warm effluent from a power plant entering the ocean using infrared imagining technology (Source: https://www.oceani.com/01img/Services/GeoSpatial/svcs_thermal_encinas.html).

1.2 Objectives of the study

This study focuses on a specific case, that is horizontal buoyant jet entering shallow water with a crossflow. The main objectives of this study are:

- To simulate the horizontal jet entering shallow water with a crossflow in OpenFOAM.
- To compare the numerical data from different turbulence models with data from a previous experimental study by Johnston et al. (1993).
- To estimate the performance of these models on predicting the jet behavior in different environmental parameters.
- To study the influence of bed attachment on the jet behavior by analysing the dataset obtained from the numerical modeling.

1.3 Novelty and Contribution of the study

The novelty of this study include the following:

- This is one of the first studies focusing on the horizontal buoyant jet. Most previous studies have focused on the vertical buoyant jet in a crossflow, while the horizontal

buoyant jet was less the subject of studies. However, the horizontal buoyant jet is more commonly found in the design of power plant outfalls and it is also more complicated as the jet freely develops along three dimensions.

- To the best knowledge of the author, this is the first study, which compares numerical and experimental data, and tests the applicability of the commonly used turbulence models for such horizontal jets.
- This study also focuses on the bed attachment effect and how it influences the mixing and transport of the jet. Additionally, various output parameters are considered in order to investigate the control factors of the bed attachment.
- This study used a unique incompressible multiphase solver, which incorporated the calculation of mass fraction and the Millero & Poisson equation into the the twoLiquidMixingFoam solver in OpenFOAM platform. This generated better predictions compared to the original solver.

The results and findings of this study could help to minimize the impacts of thermal discharges on the water systems when designing outfalls for an industrial power plant. Additionally, this study is the first to test the reliability of several popular turbulence models in predicting the jet behavior in horizontal jet in crossflow, which could also be taken as a reference for future studies.

1.4 Outline of the thesis

The thesis is organized in five chapters. Chapter 1 provides a brief introduction of the industrial effluent discharges. Chapter 2 is the methodology and literature review in which an overview of numerical method is provided in one section. In the following section, the descriptions of turbulent flow, including the essential dimensionaless parameters, turbulent eddy scales and Reynold decomposition method, are discussed first. This is followed by reviews of the methods of turbulent flow modelling and three representative turbulence models. Also, the basic of Computational Fluid Dynamic (CFD) and OpenFOAM software are discussed as well. Finally, this chapter ends up with introductions of the predominated features of several commonly-seen types of jet,

which are positively buoyant jet, negatively buoyant jet and jet in a crossflow.

Chapter 3 is a Technical paper, which is about numerical study of the horizontal jet entering shallow water with a crossflow. This case is modelled by a modified twoLiquidMixingFoam in OpenFOAM. The turbulence models used are realizable $k-\varepsilon$ model, shear stress transport (SST) $k-\omega$ model and large eddy simulation (LES) mode. The simulations cover a range of crossflow-to-jet velocity ratio and Froude number. Also, comparisons are drawn between these numerical data and data obtained from a previous experiment. The bed attachment effect is investigated in this study as well.

Chapter 5 consists of conclusions and recommendations for the future work. It should be noted that the major findings of the thesis are presented at the end of the technical paper, which is Chapter 3, hence the author tried to avoid repetition in this chapter as much as possible and provided a relatively detailed description of how to reach these conclusions.

Chapter 2 Literature review

2.1 Overview of numerical methods

In science and engineering, the phenomenon of interest is usually modelled by differential equations (DEs) or partial differential equations (PDEs). The exact solutions of such equation can not be obtained by the classical mathematical techniques due to the closure problem. Numerical methods apply so-called numerical algorithm, which is a procedure performing a sequence of primitive operations leading to the solution in a finite number of steps (Corless & Fillion, 2013). With the development of computational power and the availability of computers, approximate solution of PDEs can be obtained for most of the situations.

All numerical methods begin with the mathematical description of physical phenomenon, i.e. the set of PDEs discussed above in fluid mechanics, and certain boundary conditions. Choice of boundary conditions is extremely important for any numerical problems as a bad choice of boundary conditions may result in the convergence toward a wrong solution or even the failure of convergence.

In order to solve the fluids problems on a domain of interest, the original PDEs need to be converted to a set of algebraic equations by so-called discretization methods. The most used discretization methods are: Finite Difference Method (FDM), Finite Volume Method (FVM) and Finite Element Method (FEM). Result of imposition of the discretization method is that the continuous exact solutions of PDEs are now replaced by a series of discrete solutions. Each discretized value is placed at specified locations from which the distribution of a variable over the region of interest can be constructed (Moukalled et al., 2016). These locations in which the variables are to be computed are called grid elements or numerical grids. Grids are some small, discrete and non-overlapping elements filling the entire computational domain resulting from domain

discretization process. Common grid shapes include triangle and quadrilateral for two-dimensional geometry, and Tetrahedron, prism, pyramid and hexahedron for three-dimensional geometry. Once this huge system of none-linear algebraic equations has been built, it needs to be solved by a solution method. Solution methods always use an iterative solver, which yields a complete field of solution at different time steps. Normally an iterative solver consists of two levels of iterations: inner iterations, which is responsible for solving of the linear algebraic equations, and iterations, which handle with the nonlinearity and coupling of the equations (Kheirkhah, 2013). Finally, it is important to choose a proper Convergence Criteria, that simply tells the solver when to stop the iterations in order to obtain solutions with sufficient accuracy.

The nature of numerical methods is to capture as closely as possible physical phenomena through numerical procedures. The numerical solution is only an approximation of the exact solution and it must be noted that it may never fit the physical phenomenon completely. Therefore, once applied, the numerical methods must be validated. Validation can be done in many ways, but the general principle is to compare the employed numerical model with reference results (Archambeault & Connor, 2008). Common types of validation methods are (Jauregui and Silva, 2011; Thompson, 1992):

- *Validation using results obtained by other numerical methods.* This method usually involves the application of more than one numerical method or adopts results through previously validated numerical methods.
- *Validation using laboratory results.* This is probably the most widely-used one. This is mainly because the comparison directly shows the consistency of the numerical method with the physical data. However, whenever a measurement is conducted, some external measurement instrument should be introduced and these may interfere with the measurements directly or indirectly. Due to this, it is very important to built the most likely similar experimental and numerical configurations.
- *Validation using analytical solutions.* This method is only limited to some very

simple case in which the known analytical theories are available.

- *Validating through convergence test.* A numerical model is convergent if and only if solutions of a series of models with gradually refined domains tend to reach a specific value. Due to the discrete approximation, numerical model solutions are dependent on the time and distance steps. Multiple numerical solutions with continuously reducing time and distance steps are required to minimize such dependence. Only when further refinement of time and distance steps no longer results in significant changes, the model can be considered as credible.

2.2 Descriptions of turbulent flow

Turbulence is an extremely complex phenomenon characterized by unsteady aperiodic motion, spatially varied fluid properties, and numerous time and space scales. Meanwhile, it plays very important roles in different subjects ranging from fluid mechanic to aircraft design. In this section, some key dimensionless parameters and scales are reviewed. Also, a brief explanation of the Reynold decomposition method will be provided.

2.2.1 Dimensionless parameters and Turbulent scales

The most common method to distinguish between the turbulent flows and the laminar flow is by the dimensionless Reynold number, given as:

$$Re = \frac{\rho u l}{\mu} = \frac{u l}{v} \tag{2.1}$$

where u is a velocity scale, l is a length scale depending on the geometry, μ and v are the dynamic viscosity and the kinematic viscosity respectively. The Reynold number represents the ratio between the inertial forces and the viscous forces within fluid flows. Turbulent regime is marked by high Re value (typically > 2000) which means the inertial forces are dominated over the viscous forces. While, low Re value (typically < 500) tends to form laminar regime. Flow is in a transitional stage (both turbulent and laminar) for intermediate Re value. Another important dimensionless parameter in turbulent studies is the Froude number and it is defined as the ratio of the

inertial forces to the external forces (simply gravitational and buoyant forces in most cases):

$$F_r = \frac{u}{\sqrt{gl}} \tag{2.2}$$

For buoyant turbulent flows, the Densimetric Froude number F_0 is more commonly used and it is defined as:

$$F_0 = \frac{u}{\sqrt{g'l}} \tag{2.3}$$

$$g' = \frac{\Delta\rho}{\rho_0} \tag{2.4}$$

where $\Delta\rho$ is the density difference, ρ_0 is the density of ambient water.

There is another dimensionless coefficient that is sometimes used to account for turbulent generation-the Richardson number Ri, which is a measure of the relative strength of the density gradient in stably stratified shear flows (Galperin et al., 2007), and is given as:

$$Ri = \frac{g}{\rho}\frac{\frac{\partial\rho}{\partial z}}{\left(\frac{\partial u}{\partial z}\right)^2} \tag{2.5}$$

Scales of the turbulent flow vary from large comparable to the dimensions of the physical boundary to small. The concept that usually used to describe the turbulent patterns is “eddy”. Energy cascade is the mechanism resulting in the transfer of energy from large-scale eddies to small-scale eddies. The large-scale eddies is independent of the viscosity and they are controlled by the geometry boundary, while the smallest scale (or so-called “Kolmogorov microscales”) eddies is dominated by the viscosity effects. The length η and time scale τ_η of the smallest eddies can be written as:

$$\eta = \left(\frac{\upsilon^3}{\varepsilon}\right)^{\frac{1}{4}} \tag{2.6}$$

$$\tau_\eta = \left(\frac{\upsilon}{\varepsilon}\right)^{\frac{1}{2}} \tag{2.7}$$

Where υ is the kinematic viscosity, and ε is the dissipation rate. The largest length scale is usually taken as l, which is the characteristic length scale dependent on the

geometry boundary. Accordingly, the largest time scale is l/u.

2.2.2 Reynold decomposition

All turbulent flow variables including velocity, pressure, density etc., exhibit the time-dependent fluctuations around a mean value. The Reynold decomposition indicates that any fluid properties (φ) at the specific point as the sum of a mean value (φ) and a time-dependent fluctuation component, given as:

$$\varphi(t) = \Phi + \varphi'(t) \tag{2.8}$$

The variance $\overline{(\varphi')^2}$ is used to describe the spread of the fluctuation component about the mean value, which is defined as (Versteeg and Malalasekera, 2007):

$$\overline{(\varphi')^2} = \frac{1}{\Delta t}\int_0^{\Delta t} (\varphi')^2 dt \tag{2.9}$$

The turbulent kinetic energy per mass contained at a given location (k) is usually expressed in terms of the velocity variances and is given as:

$$k = \frac{\overline{(u_1')^2} + \overline{(u_2')^2} + \overline{(u_3')^2}}{2} \tag{2.10}$$

Where u_1', u_2' and u_3' are the fluctuation of the velocity components based on the Reynold decomposition.

2.3 Modeling of turbulent flow

Modeling turbulent flow involves solving the partial differential equations numerically. There are three main methods: Reynolds Averaged Navier-Stokes (RANS), Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS). In this thesis, DNS will only be briefly mentioned since it is not considered in this thesis, while RANS and LES modeling will be introduced in more detail.

2.3.1 DNS

DNS is designed to solve the Navier-Stokes equations without any turbulence model. The turbulence of all the spatial scales, ranging from the smallest scales (Kolmogorov microscales) to the largest scales, is resolved. This means that DNS only works on a

sufficiently small grid distance and time step. The choices of the grid distance and the time step are controlled by the ratio's of the largest and smallest scales. The ratio for the length scale is proportional to $Re^{3/4}$, while the corresponding for the time scale is proportional to $Re^{1/2}$. As a result, with a sufficiently high Reynold number, DNS approach is impossible considering the current technology.

2.3.2 RANS

RANS methods focus on the mean flow field and its properties, which therefore require much lower computational cost. This is probably the most popular approach for flow calculation in industry over the last serval decades. The key factor is to average the Navier-Stokes equations (mostly time-averaged, sometimes ensemble averaged) before solving them numerically. The resulting Reynold averaged N-S equations with index notation are shown as follows:

$$\frac{\partial \overline{u_i}}{\partial x_i} = 0 \tag{2.11}$$

$$\rho \frac{\partial \overline{u_i}}{\partial t} + \rho \overline{u_j} \frac{\partial \overline{u_i}}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial \overline{u_i}}{\partial x_j} + \frac{\partial \overline{u_j}}{\partial x_i}\right) - \rho \overline{u_i'} \overline{u_j'}\right] + \rho g \tag{2.12}$$

The implementation of averaging on the NS equations introduces six unknown terms ($-\rho \overline{u_i'} \overline{u_j'}$), which are also called Reynold stress tensors. The magnitude of Reynold stress terms is generally greater than that of the mean terms in turbulence except for the slip boundary condition where they are considered to be zero. There are two methods in literature for the closure of the system: Eddy viscosity model and Reynold stress equation model.

Eddy viscosity model

This model is based on the Boussinesq assumption (1877), which states that Reynold stresses are proportional to the mean rates of deformation, which is given as follows:

$$-\rho \overline{u_i'} \overline{u_j'} = \tau_{ij} = \mu_t \left(\frac{\partial \overline{u}_i}{\partial x_j} + \frac{\partial \overline{u}_j}{\partial x_i}\right) - \frac{2}{3}\rho k \delta_{ij} \tag{2.13}$$

Where μ_t is the eddy viscosity, and $k = \frac{1}{2}(\overline{u'^2} + \overline{v'^2} + \overline{w'^2})$ is the turbulent kinetic

energy per mass. The first term on the right hand side is analogous to the viscous stress in Newton's law of viscosity. The second term contains the Kronecker delta, which is given as:

$$\delta_{ij} = \begin{cases} 1, i = j \\ 0, i \neq j \end{cases} \tag{2.14}$$

This makes sure that the normal Reynold stresses are specified correctly. Some of the turbulence models use this viscosity model include $k-\varepsilon$ model, Realizable $k-\varepsilon$ model, RNG $k-\varepsilon$ model and SST $k-\omega$ model etc.

Reynold stress model (RSM)

Unlike the Eddy viscosity model which assumes the relation between the Reynold stress tensor and the deformation, RSM uses the Reynold stress transport equations to represent the Reynold stress tensor in the Reynolds-averaged NS equations. As a result, the directional effects of the Reynold stress field can be considered (Versteeg and Malalasekera, 2007). The exact transport equation for the Reynold stress takes the form of:

$$\frac{\partial R_{ij}}{\partial t} + u_j \frac{\partial R_{ij}}{\partial x_j} = P_{ij} + D_{ij} - \varepsilon_{ij} + \Pi_{ij} + \Omega_{ij} \tag{2.15}$$

Where P_{ij} is the production term of R_{ij}, D_{ij} is the diffusion transport term, ε_{ij} is the dissipation term of R_{ij}, Π_{ij} is the transport term by turbulent pressure-strain interactions, and Ω_{ij} is the transport term by rotation. More importantly, only the production term P_{ij} are exact, which can be written as:

$$P_{ij} = \rho\left(R_{ik}\frac{\partial u_j}{\partial x_k} + R_{jk}\frac{\partial u_i}{\partial x_k}\right) \tag{2.16}$$

Each of the remaining terms have to be modeled before solving the equation numerically. Therefore, it can be expected that RSM model is much more expensive computationally comparing to the eddy viscosity model.

2.3.3 LES

LES is one of the most promising approach for computing turbulent flow and it is an intermediate form of computation approach that could track the behavior of the large

eddies. It can predict unsteady state problems with much lower computational cost compared to DNS. According to Versteeg and Malalasekera (2007), the small eddies are almost isotropic and have uniform behavior, while the large eddies, which extract energy from the mean flow field, are more anisotropic with their behaviors influenced by boundary condition, external forces and domain geometry. The essence of the LES approach that it computes the large eddies and models the effect of the small with a sub-grid scale model. **Figure 2.1** is a graphical representation showing the principle of separating large and small eddies, and how to handle them respectively. The separation is done by applying a spatial (or low-passing) filtering operation to the full-scale flow field. The detailed design of the filter includes a filter function and a cut-off width. All turbulent eddies at larger scales than the cut-off width is resolved, while the eddies smaller than the cut-off width is filtered out and is later modeled with a Sub-Grid Scale (SGS) model.

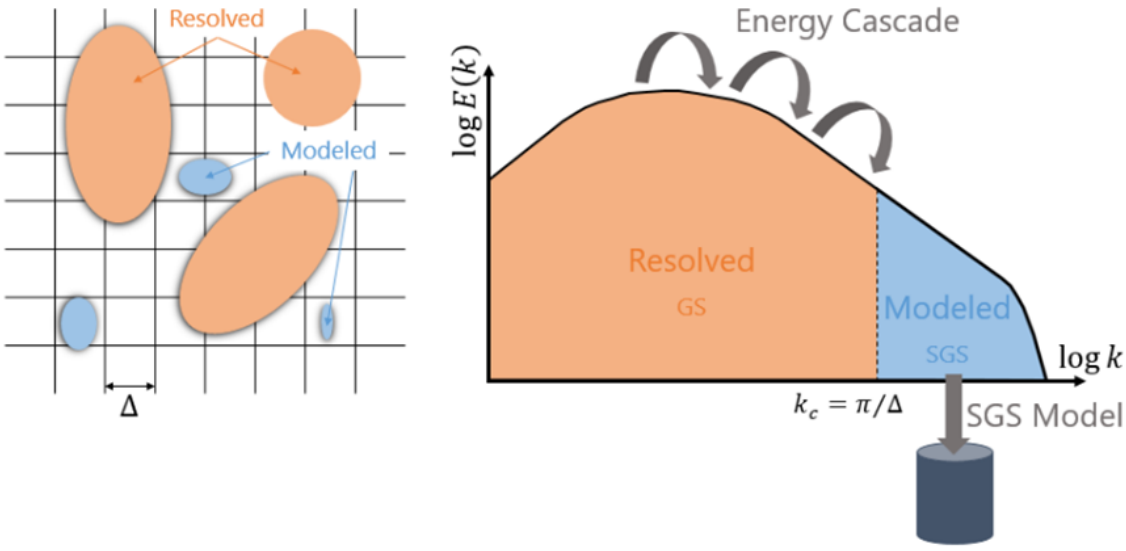


Figure 2.1 Graphical representation of the principles of LES approach (Fumiya, 2016).

A filter operation in LES method is usually defined as:

$$\bar{\phi}(x,t) = \iiint_{-\infty}^{\infty} G(x,x',\Delta)\phi(x',t)dx_1'dx_2'dx_3' \qquad (2.17)$$

Where $G(x,x',\Delta)$ is the filter function and Δ is the cut-off width, $\bar{\phi}(x,t)$ is the filtered function and $\phi(x',t)$ is the original (unaltered) function.

The cut-off width is normally selected to be the cubic root of the gird cell volume, given

as:

$$\Delta = \sqrt[3]{\Delta x \Delta y \Delta z} \tag{2.18}$$

The filtered Navier-Stokes equations (mass and momentum conservation) are given as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho \overline{u_i})}{\partial x_i} = 0 \tag{2.19}$$

$$\frac{\partial (\rho \overline{u_i})}{\partial t} + \frac{\partial \left(\rho \overline{u_i} \overline{u_j}\right)}{\partial x_j} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{\partial}{\partial x_j}\left(\mu \frac{\partial \overline{u_i}}{\partial x_j}\right) - \frac{\partial \tau_{ij}}{\partial x_j} \tag{2.20}$$

Where the overbar denotes filtered velocity, and τ_{ij} represents the sub-grid scale stresses. A comparison between the filtered velocity and the full-scale velocity is shown in **Figure 2.2.** The SGS stress is the equivalence of the Reynold stress in RANS and it can considered as the contribution of the momentum transport within the SGS eddies, which is given as:

$$\tau_{ij} = \rho \overline{u_i u_j} - \rho \overline{u_i} \overline{u_j} \tag{2.21}$$

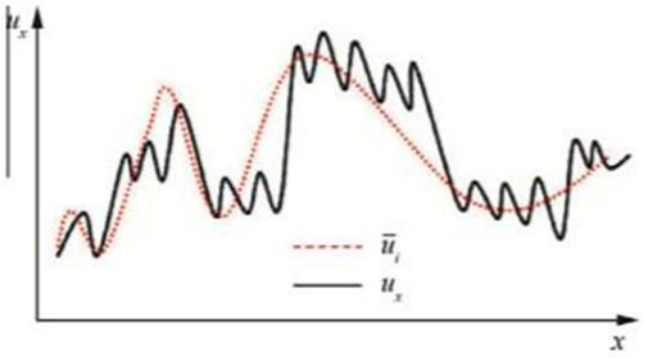


Figure 2.2 Plot of the the filtered velocity and the full-scale velocity (Yang, 2015).

2.4 Turbulence models

There are a wide range of turbulence models for commercial use, which are established based on the above modeling methods. This section will introduce three of the most common.

2.4.1 Realizable $k-\varepsilon$ model

This model is one of the most used RANS turbulence model, and it is based on the standard $k-\varepsilon$ model, which is a two-equation model, one for the turbulent kinetic

energy k and one for the dissipation rate ε. The realizable $k-\varepsilon$ model is considered more accurate compared to the standard $k-\varepsilon$ model in predicting the distribution of the dissipation rate of planar and round jets, and flow fields with strong curvature and recirculation because of the using of an improved method to calculate the turbulent viscosity (Shaheed et al., 2019). The two transport equations in this model are given as follows:

$$\frac{\partial k}{\partial t}+\frac{\partial k u_i}{\partial x_i}=\frac{\partial}{\partial x_i}\left(\left(\upsilon+\frac{v_t}{\sigma_k}\right)\frac{\partial k}{\partial x_i}\right)+G_k-\varepsilon \tag{2.22}$$

$$\frac{\partial \varepsilon}{\partial t}+\frac{\partial \varepsilon u_i}{\partial x_i}=\frac{\partial}{\partial x_i}\left(\left(v+\frac{v_t}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon}{\partial x_i}\right)+\sqrt{2}C_{1\varepsilon}S_{ij}\varepsilon-C_{2\varepsilon}\frac{\varepsilon^2}{k+\sqrt{\upsilon\varepsilon}} \tag{2.23}$$

where G_k represents the generation of turbulent kinetic energy from the mean velocity field, σ_k and σ_ε are the turbulent Prandtl number for k and ε. In addition, $\left(\upsilon+\frac{v_t}{\sigma_k}\right)$ and $\left(v+\frac{v_t}{\sigma_\varepsilon}\right)$ represent the effective diffusivity for k and ε respectively. The equation for the turbulent kinematic viscosity at each location:

$$v_t=C_\mu\frac{k^2}{\varepsilon} \tag{2.24}$$

Furthermore, the model constants C_μ, $C_{1\varepsilon}$ and $C_{2\varepsilon}$ are treated differently. To be more specific:

C_μ is calculated by:

$$C_\mu=\frac{1}{A_0+A_s\frac{kU^*}{\varepsilon}} \tag{2.25}$$

$$U^*=\sqrt{S_{ij}S_{ij}+\widetilde{\Omega}_{ij}\widetilde{\Omega}_{ij}} \tag{2.26}$$

$$\widetilde{\Omega}_{ij}=\overline{\Omega}_{ij}-\varepsilon_{ijk}\omega_k-2\varepsilon_{ijk}\omega_k \tag{2.27}$$

where $\overline{\Omega}_{ij}$ is the mean rate of rotation tensor, and ω_k is the angular velocity. The constants A_0 and A_s are determined as follows:

$$A_0=4 \tag{2.28}$$

$$A_s=\sqrt{6}cos\varphi \tag{2.29}$$

$$\varphi = \frac{1}{3} Arc\cos\left(\min\left(\max\left(\sqrt{6}W, -1\right), 1\right)\right) \quad (2.30)$$

$$W = \frac{S_{ij}S_{jk}S_{ki}}{\tilde{S}^2} \quad (2.31)$$

$C_{1\varepsilon}$ is determined by:

$$C_{1\varepsilon} = \max\left(\frac{\eta}{5+\eta}, 0.43\right) \quad (2.32)$$

$$\eta = S\left(\frac{k}{\varepsilon}\right) \quad (2.33)$$

$C_{2\varepsilon}$ is usually specified as 1.9 according to Shih (1994). Eq. (2.24) and Eq. (2.25), which is a uniquely defined turbulent viscosity, plus the transport of the mean squared vorticity fluctuation are used to composed the transport equation of the dissipation rate ε (Kheirkhah, 2013).

2.4.2 Shear Stress Transport (SST) $k-\omega$ model

This model is first proposed by Menter (1994) and it is considered a more advanced model which combines $k-\omega$ and $k-\varepsilon$ models. The $k-\omega$ model is designed to predict adverse pressure gradient and boundary layers, but has poorer performance at free flows (Killingstad, 2018). In this model, the specific turbulence dissipation rate ω, which refers to the rate at which turbulence kinetic energy is converted in to thermal internal energy per unit volume and time, substitute ε. The relation between them is:

$$\omega = \frac{\varepsilon}{k\beta^*} \quad (2.34)$$

where the constant β^* often set to be 0.09. The governing equations of this model are:

$$\frac{\partial k}{\partial t} + \frac{\partial}{\partial x_i}(u_i k) = \frac{\partial}{\partial x_i}\left(\left(\upsilon + \frac{v_t}{\sigma_k}\right)\frac{\partial k}{\partial x_i}\right) + P - \beta^*\omega k \quad (2.35)$$

$$\frac{\partial \omega}{\partial t} + \frac{\partial}{\partial x_i}(u_i \omega) = \frac{\partial}{\partial x_i}\left(\left(\upsilon + \frac{v_t}{\sigma_{\omega 1}}\right)\frac{\partial \omega}{\partial x_i}\right) + \frac{\gamma}{v_t}P - \beta\omega^2 + 2(1-F_1)\frac{\sigma_{\omega 2}}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i} \quad (2.36)$$

with P computed by:

$$P = \tau_{ij}\frac{\partial u_i}{\partial x_j} \quad (2.37)$$

where τ_{ij} is the Boussinesq assumption of eddy viscosity. The model constants are specified as: $\sigma_k = 0.85$, $\beta = 0.083$, $\sigma_{\omega 1} = 0.5$, $\gamma = 0.56$ and $\sigma_{\omega 2} = 0.856$.

The additional function F_1 is given as below:

$$F_1 = \tanh(arg_1^4) \tag{2.38}$$

and arg_1 is determined as:

$$arg_1 = min\left[max\left(\frac{\sqrt{k}}{\beta^*\omega d}, \frac{500\nu}{d^2\omega}\right), \frac{4\rho\sigma_{\omega 2}k}{CD_{k\omega}d^2}\right] \tag{2.39}$$

with $CD_{k\omega}$ determined by:

$$CD_{k\omega} = \max\left(2\rho\sigma_{\omega 2}\frac{1}{\omega}\frac{\partial k}{\partial x_i}\frac{\partial \omega}{\partial x_i}, 10^{-20}\right) \tag{2.40}$$

where d is the distance from the point to the neatest wall.

2.4.3 k-Equation LES model

The diversity of LES method is that we can select different SGS models to model the small eddies that is filtered out. In OpenFOAM, LES with the implicit filtering is applied, in which only the filter width is specified and the filter is not. kEqn LES model starts with the eddy viscosity assumption, which defines the SGS stress tensor as follows:

$$\tau_{ij} = -\upsilon_{SGS}\left(\nabla\overline{u}_\iota + \nabla\overline{u}_\iota^{\,T}\right) + \frac{2}{3}k_{SGS}\delta_{ij} \tag{2.41}$$

Where υ_{SGS} is the sub-grid scale eddy viscosity, and k_{SGS} is the sub-grid scale kinetic energy. The sub-grid kinetic energy can be correlated to the normal stress component and is written as:

$$k_{SGS} = \frac{1}{2}\tau_{kk} = \frac{1}{2}(\overline{u_k u_k} - \overline{u_k}\cdot\overline{u_k}) \tag{2.42}$$

The sub-grid scale eddy viscosity is computed with k_{SGS}:

$$\upsilon_{SGS} = C_k\Delta\sqrt{k_{SGS}} \tag{2.43}$$

Where C_k is a model constant whose default value is 0.094, and Δ is the cut-off width. The approach to calculate Δ is specified in the *turbulenceProperties* file in OpenFOAM. The cubeRootVol method is used in this paper, which specified Δ as:

$$\Delta = \sqrt[3]{\Delta x\Delta y\Delta z} \tag{2.44}$$

The transport equation of the sub-grid scale kinetic energy k_{SGS} is:

$$\frac{\partial k_{SGS}}{\partial t}+\frac{\partial \bar{u}_j k_{SGS}}{\partial x_j}=\frac{\partial}{\partial x_j}\left(\left(\upsilon+\frac{\upsilon_{SGS}}{\sigma_{k_{SGS}}}\right)\frac{\partial k_{SGS}}{\partial x_j}\right)-\rho\tau_{ij}:\bar{D}_{ij}-C_\epsilon\frac{\rho k_{sgs}^{\frac{3}{2}}}{\Delta} \quad (2.45)$$

where the operator: is the double inner product, and $\bar{D}_{ij}$ is the resolved-scale strain rate tensor defined as:

$$\bar{D}_{ij}=\frac{1}{2}\left(\frac{\partial \bar{u}_i}{\partial x_j}+\frac{\partial \bar{u}_j}{\partial x_i}\right) \quad (2.46)$$

In this model, a new transport equation of the sub-grid kinetic energy Eq. (2.45) is used to account for the effects of the production, dissipation and diffusion.

2.5 Computational Fluid Dynamic (CFD) and OpenFOAM platform

2.5.1 Introduction to CFD

The N-S equations are a set of combined partial differential equations, which is too hard to be solved analytically. However, the analytical solution can be found for some simple geometries and boundary conditions. In most case, the equations need to be approximated by numerical method. Computational Fluid Dynamic (CFD) is the subject that applied numerical techniques to solve problems associated with complex fluid flow. There are many existing turbulence models in CFD, which represent and process the N-S equation differently. Some of them will be briefly introduced later in this chapter.

Computational Fluid Dynamic (CFD) is essentially a methodology for solving a series of partial differential equations describing a fluid system. These encompass equations representing basic conservation principles (mass, momentum and energy), as well as additional transport equations for specific field properties such as the turbulent kinetic energy k and the dissipation rate ε. Typical work flow used in CFD study is demonstrated in **Figure 2.3** below.

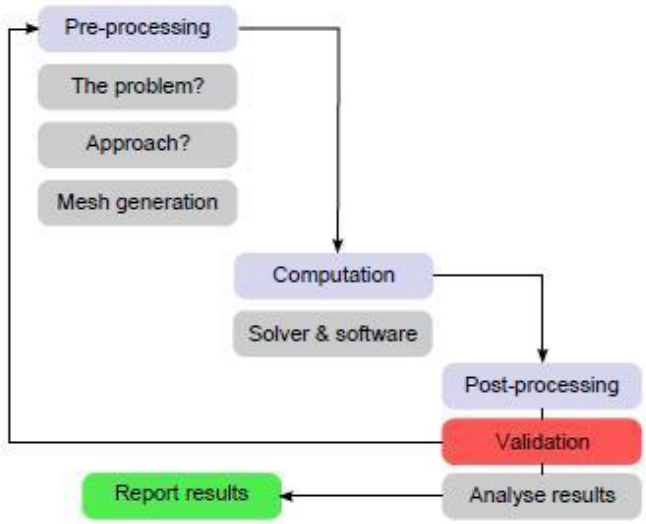


Figure 2.3 Typical work flow in CFD (Medina et al., 2015).

Pre-processing stage can be the most challenge step for a CFD study, since it involves evaluation of the nature and the dominant features of the problem, selecting efficient simulation methods (i.e. DNS, LES or RANS), mesh generation and assigning appropriate boundary conditions. The second stage in the work flow is to carry out computation. This is achieved by a CFD solver or software. Users can select either open source (no cost and with access to the source code, but no technical support) or commercial (typical without access to the source code) CFD software for their own considerations. Finally, the post-processing stage includes validation of the results and representation & analysis of the satisfied results (Medina et al., 2015). The proposed work flow is strictly followed in this thesis in order to obtain accurate and convincing results.

2.5.2 OpenFOAM

All numerical simulations in this thesis are conducted using the OpenFOAM model (Open Field Operation and Manipulation), which is a free object-oriented software written in C++ language containing lots of source codes for solvers and utilities. It was first created by Henry Weller in 1989 and was released open source as “OpenFOAM” through OpenCFD, which is the company founded by Henry and his colleagues (“About OpenFOAM,” n.d.). OpenFOAM is structured in form of libraries with functions that can be called by users to solve CFD problems. Also, it allows users to freely view and

modified the source code which has benefited its own development. Nowadays, it is one of the leading free, open source software for computational fluid dynamic. OpenFOAM is assembled from following components (Jasak, n.d.) :

- *Fundamental libraries:* containing discretization, mesh handling etc. Functions in this category are widely shared across many applications.
- *Physical modelling libraries:* containing liquid & gasses models, viscosity models and turbulence models etc.
- *Utilities:* containing mesh import and manipulation, parallel processing, domain decomposition, post processor hook-up (allowing user to add their own post-manipulation or visualization tools such as Paraview, Fieldview, OpenDX and Ensight) etc.
- *Executables:* including different fluid solvers (offering an algorithm to find the solution of PDEs, custom executable for specific physics segment.

OpenFOAM is directly programmed in terms of objects (i.e.U, phi, rho) defined as a field of values over mesh elements regardless of their dimension, rank or size (Hossein, 2013). To be more precise, the entire field of interests with the calculated values for all possible directions is represented by a single object class defined in *GeometricField.* Unique differential operators are used within OpenFOAM in order to represent partial differential equations in their natural language. They are defined as follows:

- $fvm :: ddt = {\partial}/{\partial t}$ (first order time derivative)
- $fvm :: d2dt2 = {\partial^2}/{\partial t^2}$ (second order time derivative)
- $fvc \ :: grad = \nabla$ (gradient term)
- $fvm :: div = \sum_i {\partial}/{\partial x_i}$ (divergence term)
- $fvm :: laplacian = \sum_i {\partial^2}/{\partial x_i^2}$ (laplacian term)

where the fvm operator is defined for implicit equations which must be solved iteratively, the fvc operator is defined for explicit equation which can be solved immediately.

Consider the standard differential equation of momentum conservation:

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) = -\nabla p + \mu\nabla^2\boldsymbol{u} \tag{2.47}$$

The mentioned operators convert the original equations into the following form:

```
 solve
(
    fvm::ddt(rho, u)
   +fvm::div(phi, u)
    +fvm::Laplacian(nu, u)
     ==
   -fvc::grad(p)
);
```

The correspondence between these two forms is very clear, therefore it is accessible for users who are not familiar with C++ programming language.

The typical structure of case directory in OpenFOAM is presented in **Figure 2.4** below.

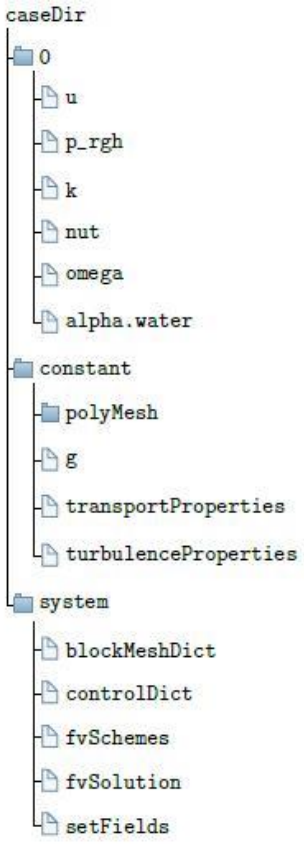


Figure 2.4 Structure of a typical OpenFOAM folder.

The *0* directory consists of a bunch of files that define boundary conditions and initial

value of the variables of interests. For example, the boundary conditions and initial fields of velocity *U* and pressure *P* are determined from the files *0/U* and *0/P* in 0 directory respectively.

The *constant* directory contains files that control relevant physical properties including file *g* for gravitation g, file *transportProperties* for density ρ and viscosity ν and file *turbulenceProperties* for specification of turbulence.

The *system* directory contains at least three files with information about the simulation setup. The file *ControlDict* is mainly concerned with time parameters related to the simulation case such as start time, end times and time step, and data output control. While the rest two files *fvSchemes* and *fvSolution* are used to set discretization schemes and solution algorithms respectively.

- *Discretization schemes:* For each term within the equations that are solved in solver, a numerical discretization scheme has to be assigned. These schemes are given in different sub-dictionaries in the *fvSchemes* file. Time derivative terms $(\frac{\partial}{\partial t})$ are normally discretized with Euler scheme which is a transient, first order implicit and bounded schemes (Greenshields, 2018a). They are all put under the *ddtSchems* sub-dictionaries. The time steps during the simulation are given in terms of the Courant number C:

  $$C = \frac{u\Delta t}{\Delta x} \tag{2.48}$$

  Where u is the velocity. The maximum courant number is set in the *ControlDict* file and the time step is adapted using this relation during the simulation according to the courant number.

  Gradient terms (∇) are set in the *gradSchemes* sub-dictionaries. The default discretization scheme for these terms is *Gauss Linear*. *Gauss* represents the Gaussian integration, while *Linear* denotes the linear interpolation or central differencing schemes which is required for discretization.

  Divergence terms $(\nabla \cdot)$ are set in the *divSchemes* sub-dictionaries. There is no default scheme for divergence terms as they are naturally different. Some

commonly used schemes are Gauss linear, Gauss limitedLinear, Gauss vanleer and Gauss upwind etc. The advection terms are usually in the form of $div\ (phi, \dots)$, where phi represent the volumetric flux of velocity on the cell faces.

Other main types of terms and their corresponding sub-dictionaries include Laplacian terms (*laplacianSchemes*), surface normal gradient terms (*snGradSchemes*) and terms of interpolation values (*interpolationSchemes*).

- *Solution algorithms:* Three most used solution algorithm in OpenFOAM for the flow problems are the semi-implicit method for pressure-linked equations (SIMPLE) algorithms, the pressure-implicit split-operator (PISO) and the pressure-implicit method for pressure-linked equations (PIMLE) algorithm. The idea of these algorithms is coupling equations for mass and momentum conservation. PISO and PIMPLE are used for transient condition and PIMPLE for steady state (Greenshields, 2018b). The following **Figure 2.5** shows the procedures of PIMPLE algorithm adopted in OpenFOAM. Distinguished from the PISO algorithm, the PIMPLE has a unique outer correction loops (i.e. cycling over a given time step for a number of iterations) and equation under-relaxation between outer correctors (Robertson et al., 2015).

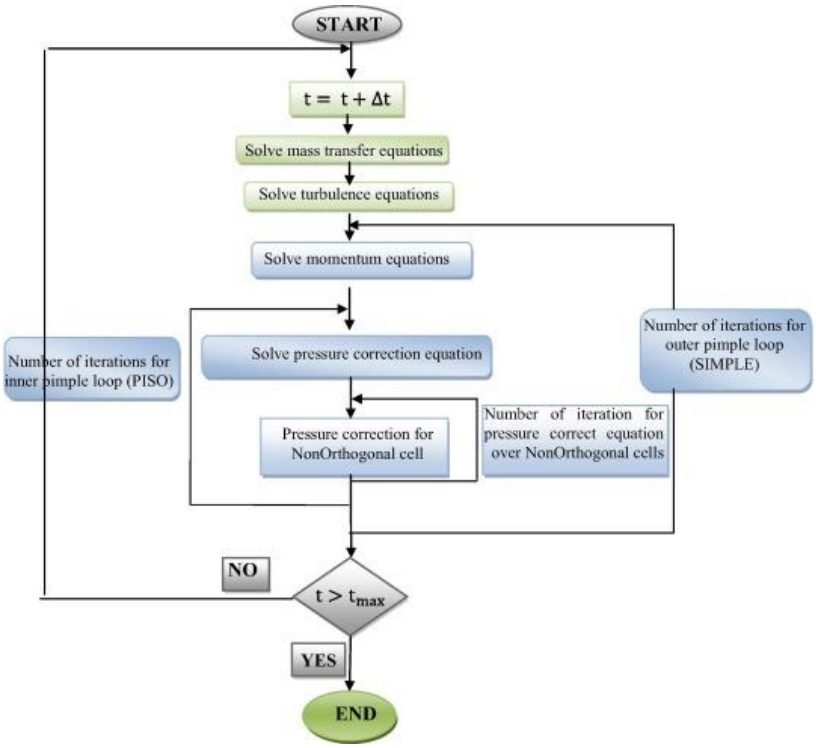


Figure 2.5 Flowchart of PIMPLE solution procedures used in OpenFOAM (Roohi et al., 2016).

Additionally, the *polyMesh* folder within the *constant* directory are where the mesh or grid system is described in terms of points, faces, owner, neighbour and boundary.

OpenFOAM contains a wide range of solvers that covers different physical conditions including compressible, incompressible, miscible, immiscible and multiphase etc. This thesis is using a customized incompressible multiphase solver modified from the original twoLiquidMixingFoam. Details about the modifications are discussed in the technical paper chapter later.

For any multiphase solvers like twoLiquidMixingFoam in OpenFOAM, the method of fluid volume is used for tracking the interaction of multiple fluid phases. The volume fraction $\alpha(x,t)$, which is both spatial and time dependent, are defined as:

$$\alpha(x,t) = \begin{cases} 1, & x \in \Omega_f \\ 0, & else \end{cases} \tag{2.49}$$

Where Ω_f represents the section of the domain occupied by one fluid f. The discrete volume fraction of ith fluid (α_i) in the system is computed as:

$$\alpha_i = \frac{\iiint \alpha(x,t)dxdydz}{V} \tag{2.50}$$

The advection-diffusion equation for the volume fractions of miscible fluids is given as :

$$\frac{\partial \alpha_i}{\partial t} + \nabla \cdot (\boldsymbol{u}\alpha_i) = \nabla \cdot \left(\left(D_{ab} + \frac{v_t}{S_c} \right) \nabla \alpha_i \right) \tag{2.51}$$

Where

α_i is the volume fraction

D_{ab} is the molecular diffusivity between miscible fluids

v_t is the turbulent eddy viscosity

S_c is the turbulent Schmidt number, which is a coefficient represents the ratio between the rates of turbulent transport of momentum and the turbulent transport of any scalar parameters. Due to the law of mass conservation, the following constraint needs to be satisfied:

$$\sum_{i=1}^{n} \alpha_i = 1 \tag{2.52}$$

Additionally, the bulk density ρ and viscosity μ at each location in every time steps are defined as follows:

$$\rho = \sum_i \rho_i \alpha_i \tag{2.53}$$

$$\mu = \sum_i \mu_i \alpha_i \tag{2.54}$$

2.5.3 Mesh generation in OpenFOAM

OpenFOAM software is also equipped with two approaches to generate mesh for numerical simulation.

blockMesh

The *blockMesh* utility is the most used method of mesh generation and it creates parametric meshes with grading and curved edges. The entire computational domain is divided into a set of three-dimensional hexahedral blocks, each of which is defined by 8 vertices. The definition of each block contains entries for vertex numbers, number of cells in each direction and cell expansion ratios, which is written as follows:

```
//
Blocks
(
    hex (0 1 2 3 4 5 6 7)
    (10 10 10)
    simpleGrading (1 2 3)
);
//
```

All relevant information is written into a dictionary file named *blockMeshDict* located in the *constant/polyMesh* directory. The disadvantages of the *blockMesh* utility are that it is time consuming and not user friendly, and can not handle with complex geometry.

snappyHexMesh

The snappyHexMesh is a very powerful utility that can solve the above problems. The

snappyHexMesh also begins with the blockMesh utility by which the background mesh is first generated. This background mesh needs to encompass the all geometry surfaces which consists of a group of STL files. All STL files used in this study are created in *Salome* (an open-source software that can generate geometry for CFD studies). With the background mesh and these STL files, the snappyHexMesh carves out the desired geometry according to the STL files within the background mesh. The summary of the procedures is shown in **Figure 2.6**. In addition to the ability to handle complex geometries, the snappyHexMesh has the advantage of defining new region where a further refinement can be assigned.

In order to generate mesh using snappyHexMesh utility, following files need to be included in the system directory:

- *blockMeshDict:* This file is used to create the background mesh surrounding the geometry surfaces prior to snappyHexMesh.
- *meshQualityDict.* This file contains parameters for checking the mesh quality.
- *surfaceFeatureExtractDict:* This file controls the surfaceFeatureExtract utility which is used to extract the edges with sharp angles based on the angle specified in *includeAngle* sub-dictionary.
- *snappyHexMeshDict:* This File contains all the settings for running snappyHexMesh. The Geometry sub-dictionary lists all the surface files used by the *snappyHexMesh*, except the background mesh. A name is also assigned to them for later references. In addition, this sub-dictionary allows users to specify regions of the domain that they want to further refine. The default shapes include box, sphere, cylinder, plane, cone, disk etc. The following content are one example that shows how to define extra regions for further refinement in this sub-dictionary:

```
//
geometry
{
box2
{
```

```
    type searchableBox;
    min (3 0 0);
    max (4.6 0.8 0.200675);
 }
box3
{
    type searchableBox;
    min (3 0.8 0);
    max (5.2 1.3 0.200675);
}
}
//
```

Three steps involved in running snappyHexMesh are castellating, snapping and layering. The *castellatedMeshContrls* sub-dictionary is used for user-defined edge refinement in the surfaces and the regions specified previously. The *snapControls* contains some important parameters including *nSolverIter* (the number of mesh displacement iterations) and *nFeatureSnapIter* (the umber of feature edge snapping iterations). The additional layering procedures uses the settings in *addLayerControls* sub-dictionaries. In the head of the *snappyHexMeshDict* file, users can set the three components either true or false dependent on their requirements.

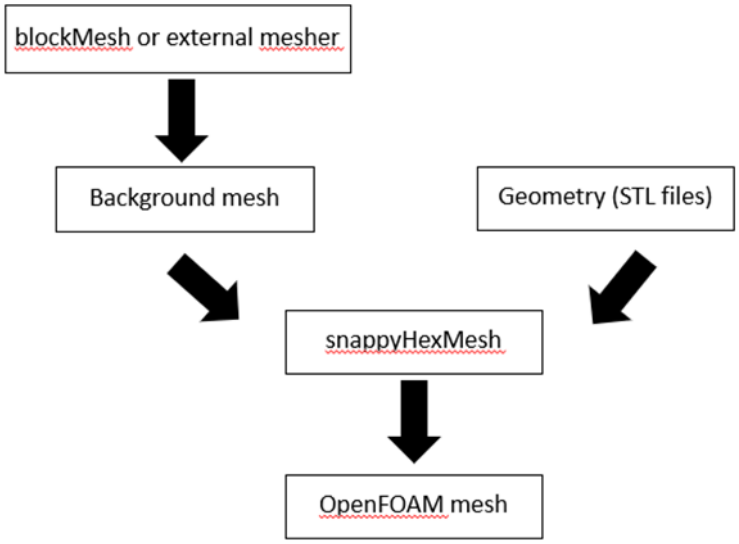


Figure 2.6 The procedures of mesh generation in the snappyHexMesh.

2.6 Some typical types of jet

2.6.1 Positively Buoyant jet

A positively buoyant jet occurs when a light fluid is discharged into a stagnant heavier one. This type of jet can usually be found in multistage flash desalination plant, which generates high temperature waste water from the cooling system of the plant (Kheirkhah et al., 2014). The positively buoyant jet tends to curve all the way up until the surface regime due to the buoyancy effect if the source is remote from any confined boundary. **Figure 2.7** is a schematic of a typical positively inclined buoyant jet entering in a unconfine ambient water.

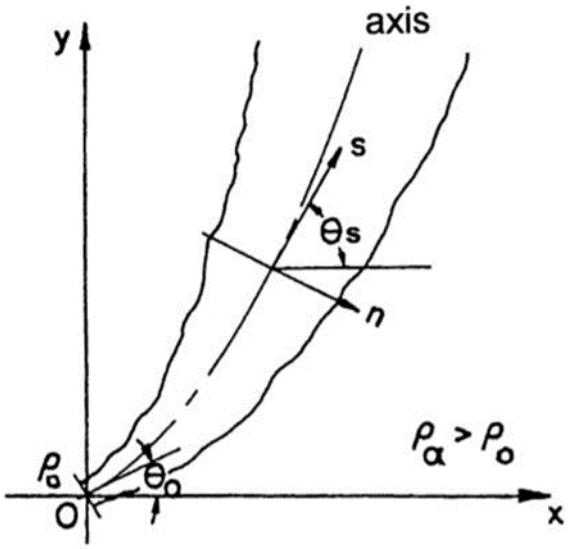


Figure 2.7 Schematic of a positively inclined buoyant jet (Angelidis, 2002).

The jet trajectory is determined by connecting the points of maximum time-averaged concentration in each cross section. θ (θ_o or θ_s) is the inclination angle, which denotes the angle of direction of the instantaneous velocity in the central axis and the horizontal direction. Theoretically, θ reaches to 90^o at end of the trajectory and the jet becomes vertically upward.

Positively buoyant jet with a wall attachment (or so-called wall jet) is another topic that have been extensively studied. This type of jet normally occurs in regions where jet is discharged in touch with a solid horizontal boundary (shown in **Figure 2.8** below).

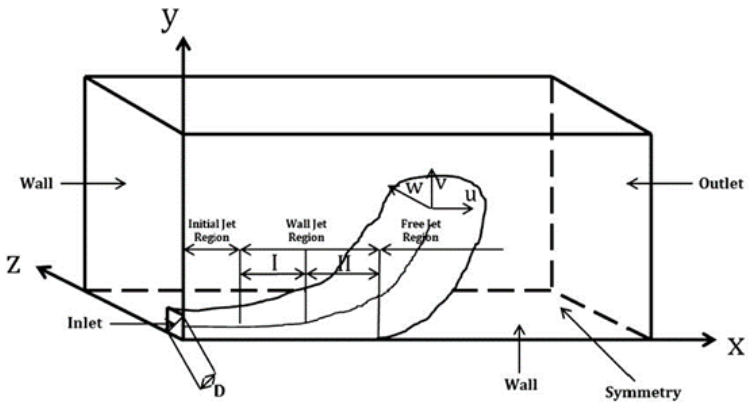


Figure 2.8 Schematic view of the wall jet with coordinate system (Kheirkhah, 2013).

There are three different regions for a wall jet-initial jet region, wall jet region and free jet region. In the initial jet region, jet flows directly along wall boundary due to the

supply of initial momentum. Then, in the wall jet region, it keeps clinging to the wall after the initial region for some distances due to the wall attachment effect. Finally, the buoyant effect becomes dominant and drags the jet into free region. According to Sharp (1975), the dilution of such wall jets is almost twice that of an equivalent free jet. Therefore, the wall jet is widely used for design of the submerged outfall discharging high-temperature effluent. Rajaratnam and Pani (1974) conducted an experimental study about a three-dimensional wall jet discharging from square, circular, elliptic, rectangular and equilateral triangular nozzles. They found that there is a similarity between the distribution of the axial velocity in the central and transverse planes in the fully developed flow. Meanwhile, a few researchers have recently studied the wall jet numerically. For example, Huai et al. (2010) simulated the positively buoyant wall jet using the realizable $k-\varepsilon$ model. Based on the numerical results, they determined the cling length, centerline trajectory and temperature dilutions at some specific cross sections. Additionally, they also compared their numerical data with the experimental data obtained by Sharp (1975), and demonstrated that their model could be effective in predicting the velocity distribution and temperature dilutions. The distribution of velocity and temperature dilutions had a similarity along the axial direction at centerline in the near field.

2.6.2 Negatively Buoyant jet

The negatively buoyant jet normally occurs in desalination or geothermal plants, which discharge brine, as well as industrial waste (Papakonstantis et al., 2011). If the jet is discharged horizontally, it directly curves down due to the buoyant effect. If the jet is discharged inclinedly, it rises to a maximum height (or named initial terminal height Z_f) before starting to sink down toward the bed. **Figure 2.9** shows some essential features of the positively buoyant inclined jet.

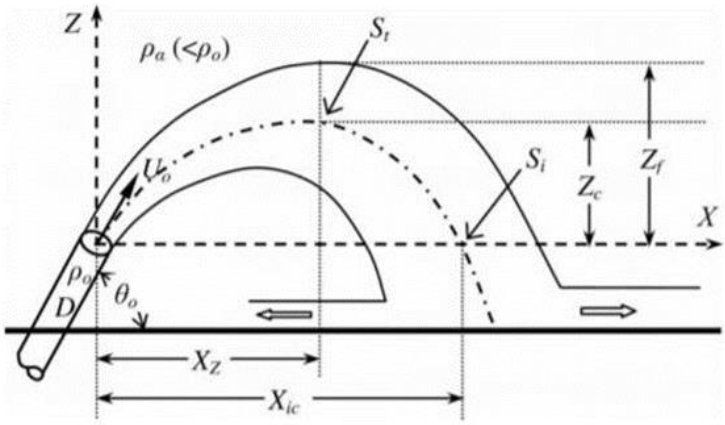


Figure 2.9 Schematic view of a negative inclined buoyant jet (Papakonstantis et al., 2011a).

the jet trajectory is also determined by connecting the points of maximum time-averaged concentration in each cross section and is plotted in a dotted line. θ_0 is the angle of jet initial velocity, which is believed to be an important controlled factor of the jet behavior. The jet reaches its terminal height Z_f at a horizontal distance of X_z from the center of jet nozzle. Similarly, Z_c is the terminal height of the jet trajectory, which has the same horizontal distance with Z_f. X_{ic} represents horizontal distance from the coordinate origin to where jet returns to its initial elevation. S_t and S_i are the minimum dilution (corresponding to maximum concentration) at terminal point and return point respectively. According to Papakonstantis et al. (2011b), all the above parameters can be expressed in forms of θ_0 and several initial jet parameters. These initial jet parameters are mass fluxes (Q_0), momentum (M_0) and buoyancy (B_0), which are defined as follows:

$$Q_0 = \frac{\pi d^2}{4} U_0, M_0 = U_0 Q_0, B_0 = g' Q_0 \tag{2.55}$$

Additionally, two characteristic length scales $L_M = M_o^{3/4}/B_o^{1/2}$ and $L_Q = Q_o/M_o^{1/2}$ are also defined. The importance of these two length scales is that the ratio of them can be related to the jet densimetric Froude number F_o (Fischer et al., 2013):

$$\frac{F_o}{\left(\frac{\pi}{4}\right)^{\frac{1}{4}}} = \frac{L_M}{L_Q} \tag{2.56}$$

For $l_M/l_Q \gg 1$ or $F_o \gg 1$, the jet is dominated by initial momentum, for $l_M/l_Q \ll 1$

or $F_o \ll 1$, the initial momentum becomes insignificant and the jet is mainly driven by buoyant effect.

With these assumptions, the jet elevation Z and time-averaged dilution at random location (X, Z) within the jet center plane can be expressed with the characteristic length scales and discharge angle θ_0 as (Roberts et al., 1997):

$$Z = DF_o C(\theta_o) \tag{2.57}$$

$$S = F_o f\left(\frac{x}{DF_o}, \frac{z}{DF_o}\right) \tag{2.58}$$

F_o can be obtained from Eq. (2.56) above. For a inclined buoyant jet with a certain discharge angle θ_o at any location (x, z), $C(\theta_o)$ and $f(\frac{x}{DF_o}, \frac{z}{DF_o})$ are constants, which can be determined by experimental studies. By applying Eq (2.57) and (2.58), the above parameters, namely Z_c, Z_f, S_t and S_i, which are important features of the jet, can be calculated. Kikkert et al. (2007) studied the effectiveness of using analytical solutions to predict the behavior of negative inclined jet by comparing with data from experiments. Results showed that the ranges of discharge angle and initial Froude number for decent predictions by the analytical solutions are $25°$-$90°$ and 14-99 respectively.

Zeitoun et al. (1970) applied both experimental and numerical method to investigate the directly upward and inclined dense jets with some densimetric Froude numbers. Based on their results, a jet discharged at a $60°$ angle leads to the longest trajectory, and thus the maximum dilution, compared to any other angles. They also suggested that this finding can be applied to the design of outfall systems for desalination plants. Roberts et al. (1997) further studied the negatively buoyant jet at an angle of $60°$ into stationary receiving water experimentally using advanced equipment including laser-induced fluorescence (LIF) and a micro conductivity probe. The time-averaged LIF images showed a complex additional mixing region downstream of the impact point, which results in higher dilution level than that of the impact point. They also reported that the fluctuations of concentration reduce with distance from the jet, which could be explained by collapse of turbulence under the effect of density stratification. Although,

negatively inclined jet with a $60°$ discharge angle is generally recommended for design of outfalls to achieve a maximum dilution efficiency, $60°$ discharge angle leads to a relatively high terminal height, and thus this angle may not be appropriate for some shallow water conditions. Therefore, Shao and Law (2010) investigated the behavior of $30°$ and $45°$ negatively inclined jet, which are usually considered for many coastal shallow water environments.

Unlike experimental studies, numerical studies were rarely conducted until recent years. Vafeiadou et al. (2005) first used a three-dimensional numerical, called CFX-5, to simulate the negative inclined jet at various angles. Their numerical results were generally consistent with the experimental data except the rise height and the distance to the impact point tend to be slightly underestimated comparing to experimental results observed by Robert et al. (1997). Elhaggag et al. (2011) conducted their numerical studies of the negative inclined jet for areas of limited extent using Fluent CFD software. The numerical results were used to identified the vertical jet penetration

2.6.3 Jet in Crossflow

Jet in crossflow (JICF) is an important flow field, which find practical applications in many engineering problems, such as cooling in turbine blades, pollutant dispersal, outfall designs etc, for its ability to promote mixing with crossflow and to introduce a controlled force into flow field (Karagozian 2003; Muppidi and Mahesh 2005). The effective velocity ratio r, which is believed to be the primary influential factor, can be given as:

$$r = \left(\frac{\rho_j u_j^2}{\rho_a u_a^2}\right)^{1/2} \tag{2.59}$$

where u_j is the jet velocity, u_a is the velocity of the crossflow, ρ_j is the jet fluid density and ρ_a id the density of the crossflow fluid. This equation can be simplified to $r = u_j/u_a$ for flows with constant density or small density difference.

Vertical jet in crossflow is the most studied type of jet in crossflow. It is symmetrical and characterized by complex flow structures and vortex systems. There are several

comprehensive reviews of the physical behavior of round vertical jet in crossflow conducted by previous researchers (Fric and Roshko, 1994; Karagozian, 2014; Chakraborty and Bhattacharjee, 2017 etc).

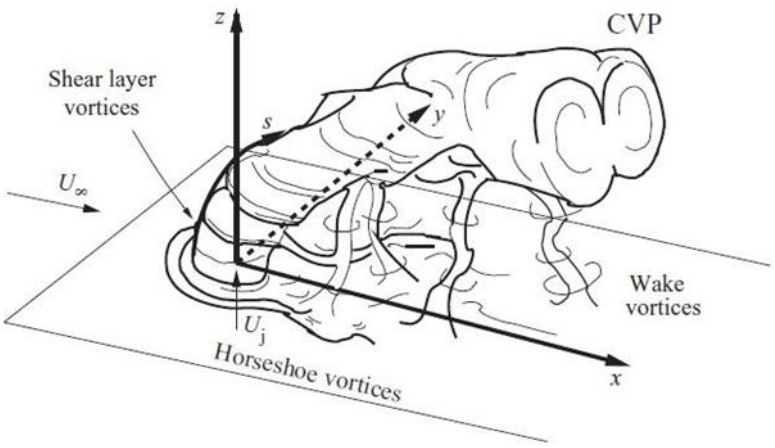


Figure 2.10 Schematic a vertical jet in crossflow with potential vortex systems and orientation of coordinate axes (Karagozian, 2014).

Four potential coherent vortex structures can be identified in the near field of the jet due to the interaction of the jet and crossflow (shown in **Figure 2.10**), they are: (1) the counter rotating vortex pair (CVP), (2) the shear layer vortices, (3) the wake vortices and (4) the horseshoe vortices.

The counter rotating vortex pair (**Figure 2.11 (a)**) is the dominant structure in the jet cross-section. Its generation is considered to be associated with the evolution of vorticity in the jet's shear layer and the pressure difference between the upstream and the downstream regions (Karagozian, 2014). The CVP is essentially a representation of the mean flow field caused by the momentum of the initial jet. Some previous researches (Kamotani and Greber 1972; Fearn and Weston 1974) have performed detailed measurements of the velocity and vorticity field in the cross section of a CVP.

The shear layer vortices (**Figure 2.11 (b)**) dominate the windward side of the initial section of the jet. It is believed to be a result of the Kelvin-Helmholtz instability of the jet's shear layer.

The wake vortices (**Figure 2.11 (c)**) form on the leeside downstream of the jet. Fric &

Roshko (1994) mentioned the interesting similarity between the wake vortices of the jet and the solid cylinders with the only difference of the jet or the solid cylinder as obstacles.

The horseshoe vortices system (**Figure 2.11 (d)**) forms when the crossflow enters into the adverse pressure region created by jet and it is characterized by two separate streams wrapping around the jet base issuing from the wall into the crossflow. Krothapalli et al. (1990) study a case in which crossflow encounters a rectangular jet flow and concluded that the formation of the horseshoe vortex can be periodic.

The CVP and horseshoe vortices are the properties of the mean flow field, and they are easy to identified in the simulation using RANS models. On the contrary, shear layer and wake vortices are instantaneous and temporal properties, which can be removed after time averaging.

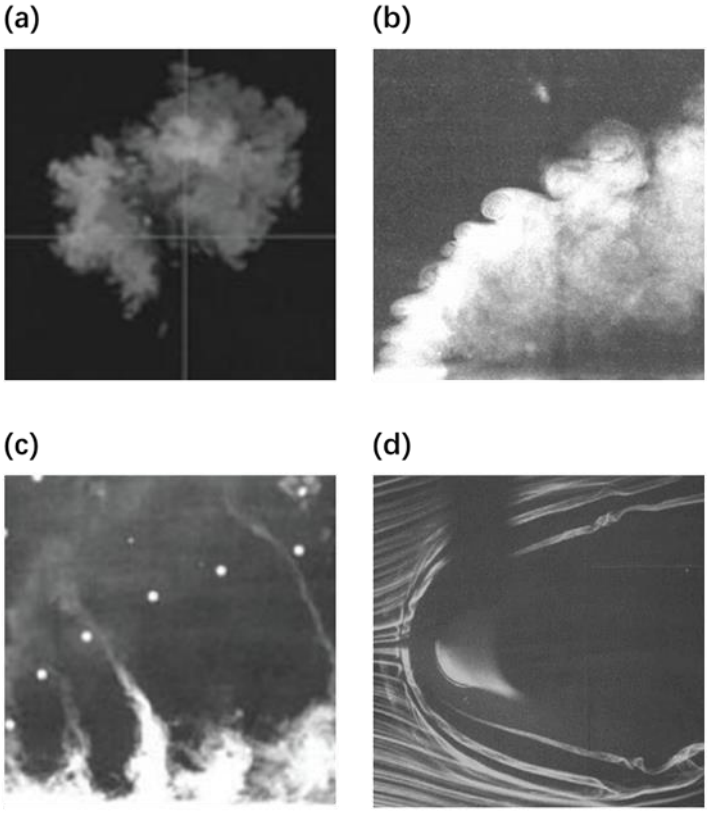


Figure 2.11 Examples of various vortex structures. (a) the counter rotating vortex pair (b) the shear layer vortices (c) the wake vortices (c) the horseshoe vortices (Fric and Roshko, 1994).

In addition to the various vortex structures, the recirculating zone and Coanda

attachment effect are two other phenomena that might be recognized for jets entering a crossflow. The recirculating region (**Figure 2.12**) generally occurs in the wake downstream of the outlet when the jet interacts with the ambient crossflow. The jet movement is restricted by the presence of the solid surface on the near bank, thus resulting in a recirculating region with low pressure.

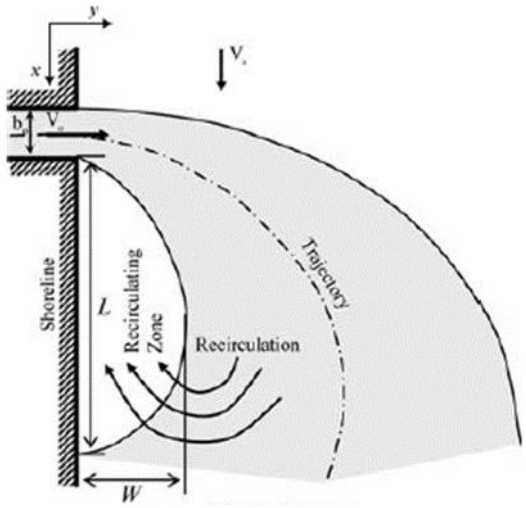


Figure 2.12 Plan view of the recirculating zone in jet in the crossflow (Kim and Cho, 2006).

The Coanda attachment effect refers to the phenomenon of the attachment of flow to solid walls. In hydrodynamic, when a jet discharge is very close to a solid boundary in the near field, a strong attachment effect occurs. This process appears as the jet flows along this solid boundary first, before it is finally free from the attachment. Regions where such strong near field interactions happen are characterized by the minimum dilution, thus highest pollutant concentrations. Therefore, this condition needs to be avoided when designing an outfall system.

Chapter 3 Technical Paper – Numerical Modeling of Horizontal Buoyant Jet entering Shallow Water with a Crossflow

Abstract – Jet in crossflow (JICF) is a type of flow field that has many engineering applications particularly in coastal waters. This issue is also relevant for some major environmental problems such as the discharge of thermal industrial effluents in other water bodies. The major objective of this study is to test the effectiveness of various turbulence models, including the realizable $k-\varepsilon$, Shear Stress Transport (SST) $k-\omega$ and the k-Equation Large Eddy simulation (LES) to predict horizontal buoyant jets entering shallow water flowing as a crossflow. Additionally, the effect of the jet attachment to the bed, which is one of the major jet characteristics, is also investigated. These investigations are performed by running simulations covering a range of crossflow-to-jet velocity ratios and densimetric Froude numbers using a modified twoLiquidMixingFoam solver in OpenFOAM. This unique solver considers the temperature dilutions, instead of the concentration dilutions. Comparisons are drawn between these numerical results and data obtained from a previous experimental study by Johnston et al. (1993). In this work, it is found that the k-Equation LES model outperformed the other two Reynolds-averaged Navier–Stokes (RANS) turbulence models in terms of consistency with experimental data. It was found that the jet bed attachment is significantly influenced by the crossflow-to-jet velocity ratio and density difference.

Index Terms – OpenFOAM, CFD, twoLiquidMixingFoam, RANS, LES

3.1 Introduction

For jets in a crossflow, such as the thermal discharges in costal waters, the mixing and transport behavior in the near as well as the far field are different and it is therefore necessary to distinguish between them. Wright (1977) introduced an entire set of length and buoyancy scale relationships [seen Appendix **A**] with the consideration of crossflow momentum, which is constructed in terms of kinematic volume flux Q_0, kinematic momentum flux M_0, kinematic buoyancy flux B_0 and crossflow velocity U_a. The classification depends on the dominance of a particular source in the flow field (either momentum or buoyancy) and whether the crossflow momentum is strong enough compared to the jet momentum. For example, in the Momentum Dominated Near Field (MDNF), where the momentum of the jet is dominant over the effect of crossflow, length scale L_M and buoyancy scale G_M depend both on M_0 and B_0:

$$L_M = \frac{M_0^{\frac{3}{4}}}{B_0^{\frac{1}{2}}} = \left(\frac{\pi}{4}\right)^{\frac{1}{4}} dF_0 \tag{3.1}$$

$$G_M = \frac{B_0^{\frac{3}{2}}}{M_0^{\frac{5}{4}}} = \left(\frac{\pi}{4}\right)^{\frac{1}{4}} (g_0')F_0^{-1} \tag{3.2}$$

Similarly, in the Momentum Dominated Far Field (MDFF), where the effect of the crossflow dominates jet momentum, the corresponding length scale L_m and the buoyancy scale G_m only include U_a and M_0. This principle of constructing the length & buoyancy scales also applies to the Buoyancy Dominated Near Field (BDNF) and the Buoyancy Dominated Far Field (BDFF), which are defined as regimes where buoyancy is dominant over the effect of crossflow and the effect of crossflow dominates buoyancy, respectively.

The shallow water condition, compared with the deep water condition, translates into extra confinements on the development of jet from either bed or free surface. Meanwhile, the presence of crossflow further complicates the situation. Johnston et al. (1993) suggested three distinct jet types for the shallow water condition: bed jet (jet

source close to the bed boundary with $\frac{Z_0}{L_M} \approx 0.1, \frac{(H-Z_0)}{L_M} \approx 0.9$, where H is the depth of water and z_0 is the height of jet nozzle), intermediate jet ($\frac{Z_0}{L_M} \approx 0.5, \frac{(H-Z_0)}{L_M} \approx 0.5$) and surface jet (jet source close to the free surface with $\frac{Z_0}{L_M} \approx 0.9, \frac{(H-Z_0)}{L_M} \approx 0.1$). The bed jet is characterized by a strong Coanda attachment effect, and the intermediate jet only experiences Coanda effect, while the surface jet is generally considered free from Coanda influence. The authors also found that the bed jet dips temporarily towards the bed due to the instability induced by the current.

Unlike vertical jets in crossflows, which have been of long-term interest in studies, horizontal jets in crossflows are limited in the literature, with only a few experimental studies conducted for certain conditions. Lee and Neville (1987) conducted an experimental study about the horizontal outfalls with Froude numbers ranging from 1 to 10, and found relationships for both the jet centerline dilution and the location of the surface boil for the BDNF and BDFF. Lee (1989) studied the horizontal jets with greater momentum (Froude numbers between 15-90) entering crossflow conditions. He suggested that the dilution level of horizontal jets in crossflow is less than that of corresponding vertical jets. Johnston et al. (1993) conducted a series of experiments studying thermal jets entering crossflow conditions with a focus on the mean flow patterns and their integral quantities. They heated the jet before discharging it at right angle to the current. Temperature measurements in the water tank were measured by thermistor probes located in stainless-steel tubes. They indicated that the location of these probes could be set to an accuracy of within 2 mm. Three sets of data were repeated under stationary condition four times to assess the repeatability of results for identical initial conditions. They finally suggested that the entire jet can be divided into three sections (**Figure 3.1**) due to the constant exposure of crossflow: a symmetrical section aligned with the direction of the jet source, an unsymmetrical section with a compression zone outward and an expansion zone inward, and another symmetrical section aligned with the current direction.

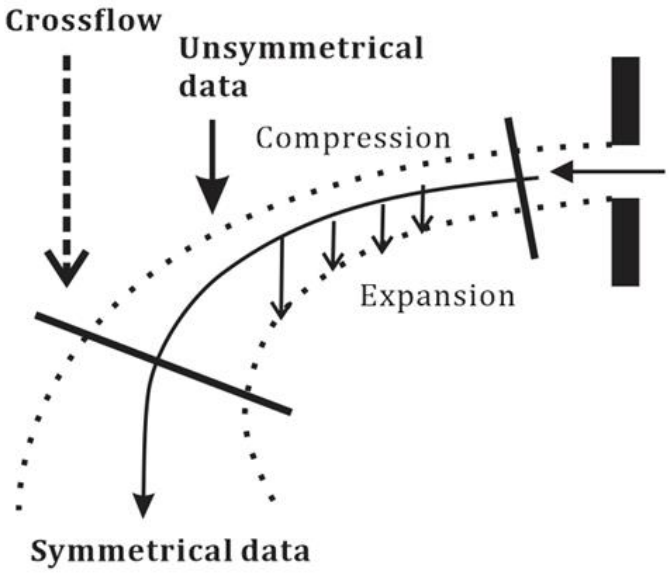


Figure 3.1 Plan of jet mainstream (Based on Johnston et al., 1993).

As for the numerical investigations, there are several studies about the mixing behavior of surface horizontal jets in a crossflow. One of the first numerical studies associated with horizontal jets in a crossflow was conducted by McGuirk and Rodi (1979), who analyzed a side outfall into an open channel numerically using a three-dimensional mathematical model. More commonly, the three-dimensional continuity and Reynolds averaged equations with the Boussineq's and hydrostatic assumptions, and scalar advection-diffusion equations, were used. For example, Kim and Seo (2000) adopted a concept of the numerical slot diffuser and modeled the thermal mixing behavior using a numerical model. Their numerical results were generally in good agreement with the measured values of the actual submerged multiport diffuser. Kim and Cho (2006) conducted numerical studies of the thermal water discharged from surface and submerged side outfalls in shallow and deep ambient water. They found that, in deep water, where no recirculating zone is recognized, the surface discharge has a higher minimum dilution than the submerged discharge.

Several researchers also studied horizontal surface jet discharges into the crossflow. Jirka (2007) simulated both shallow and deep-water jets using an integral surface jet model. He mainly focused on turbulent entrainment of surface jets, and classified it into: horizontal entrainment, vertical entrainment, frontal entrainment, interfacial

entrainment and advected puff entrainment. Anwar (1987) conducted an experimental study to map the thermal distribution of a warm (73℃) surface jet entering a cold (15℃) crossflow by measuring the horizontal and vertical isotherms. He found that the jet axis was strongly dependent on water depth. Additionally, he also identified a slow-moving and recirculated zone on the lee side of the surface jet in the proximity of the inner wall of the flume. Gharavi et al. (2020) studied the flow characteristics of turbulent mixing regimes of surface buoyant jets entering a crossflow with Stereo-PIV measurements. By analyzing the mean velocity streamlines, they observed a downstream-propagating vortex at the transverse planes within the flow structure, which could occur up to a distance of 35W (where W is the jet exit width in their study) downstream of the discharge point. Additionally, a circulation zone was also identified at the longitudinal planes by analyzing the mean excess velocity streamlines.

In the present study, the authors adopted both the Reynolds Averaged Navier-Stokes (RANS) and the Large Eddy simulation (LES) approaches to simulate the horizontal jet entering shallow water with a uniform crossflow. The major objectives are to compare the simulation results with the existing experimental data and evaluate the performance of two approaches on the prediction of transport and mixing behavior of the jet in the near field. This paper is organized as follows. First, the authors introduce the governing equations and the numerical methodology. Next, the results of convergence testing are presented. Then, the performance of the different turbulence models for each case are evaluated in terms of three aspects including the horizontal jet trajectory, the vertical jet trajectory, and the average dilution on the trajectory, by comparing numerical results with the existing experimental data. In addition, the influence of the crossflow-to-jet velocity ratio on the jet behavior is also investigated using these simulations. Next, more numerical scenarios are designed in order to investigate how the variations in density difference could influence the vertical elevation and the dilution rate of jets. This section also involves a discussion about the bed attachment effect, which is one of the predominant features in this study. Finally, conclusions and recommendations are provided for the closure of this study.

3.2 Governing equations

The fundamental system of equations describing viscous flow in flow mechanics is known as the Navier-Stokes system of equations. With the assumption of incompressible and Newtonian fluid, the equations can be written as:

$$\nabla \cdot \boldsymbol{u} = 0 \tag{3.3}$$

$$\rho\left(\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{u} \cdot \nabla \boldsymbol{u}\right) = -\nabla p + \mu \nabla^2 \boldsymbol{u} + \rho g \tag{3.4}$$

where $\boldsymbol{u}$ is the velocity vector, p is the pressure, ρ is the density, g is the local gravitational acceleration, the dynamic viscosity is given by μ. Eq. (3.3) represents the conservation of mass for incompressible flow. Eq. (3.4) follows Newton's second law of motion. In the present study, only incompressible fluids are considered. This means that the density ρ is independent of pressure. However, the fluid density is still not constant as it varies with temperature T and salinity S. The variation of the density ρ with the temperature T and salinity S can be described by the equation proposed by Millero & Poisson (1981):

$$\rho = \rho_T + AS + BS^{\frac{3}{2}} + CS \tag{3.5}$$

where

$$A = 8.24493 \times 10^{-1} - 4.0899 \times 10^{-3}T + 7.6438 \times 10^{-5}T^2 - 8.2467 \times 10^{-7}T^3 + 5.3875 \times 10^{-9}T^4$$

$$B = -5.72466 \times 10^{-3} + 1.0227 \times 10^{-4}T - 1.6546 \times 10^{-6}T^2$$

$$C = 4.8314 \times 10^{-4}$$

ρ_T is the water density as a function of temperature:

$$\rho_T = 999.842594 + 6.793952 \times 10^{-2}T - 9.095290 \times 10^{-3}T^2 + 1.001685 \times 10^{-4}T^3 - 1.120083 \times 10^{-6}T^4 + 6.536336 \times 10^{-9}T^5$$

In addition to the Navier-Stokes equations, the authors are also interested in concentration fields of other scalar variables including the salinity S, temperature T and volume fraction α. The conservation equation for the scalar variables in Einsteinian notation is given by:

$$\frac{\partial(\phi)}{\partial t} + \frac{\partial(u_i\phi)}{\partial x_i} = D\frac{\partial^2\phi}{\partial x_i^2} + Q_s \tag{3.6}$$

where ϕ can be any scalar parameters, D is the isotropic diffusion coefficient for this scalar parameter and Q_s is the source term. The remaining three terms from left to right are: the transient term, the advective term and the diffusion term, respectively.

For each fluid phase, a volume fraction $\alpha(x,t)$ is introduced. The volume fraction is both time and space dependent, and is defined as:

$$\alpha(x,t) = \begin{cases} 1, & x \in \Omega_f \\ 0, & else \end{cases} \tag{3.7}$$

where Ω_f represents the section of the domain occupied by one fluid f. For each grid within the computational domain, the discrete volume fraction of ith fluid (α_i) in the system is computed as:

$$\alpha_i = \frac{\iiint \alpha(x,t)dxdydz}{V} \tag{3.8}$$

The advection-diffusion equation for the volume fractions of fluids is given as follows:

$$\frac{\partial\alpha_i}{\partial t} + \nabla\cdot(\boldsymbol{u}\alpha_i) = \nabla\cdot\left(\left(D_{ab} + \frac{v_t}{S_c}\right)\nabla\alpha_i\right) \tag{3.9}$$

where D_{ab} is the molecular diffusivity between fluids, v_t is the turbulent eddy viscosity, and S_c is the turbulent Schmidt number, which is a coefficient that represents the ratio between the rates of turbulent transport of momentum and the turbulent transport of any scalar parameter.

In the present study, the relevant equations are discretized using the finite volume method and the numerical simulations are conducted with OpenFOAM, which is an open source software.

3.3 Methodology and Numerical setups

Whether a numerical modeling is convincing relies largely on the pre-processing stage. Once the flow problem and features of interests are identified, a computational domain needs to be defined first. Many factors have to be taken into account in order to have good numerical results. Boundary conditions and mesh quality are the two most important factors.

3.3.1 Geometry and Boundary conditions

In this study, the two most dominant features are crossflow and shallow water environment, while, the fluctuation of free surface has little influence on the development of jet. The sketch of the simulation geometry is shown in **Figure 3.2**.

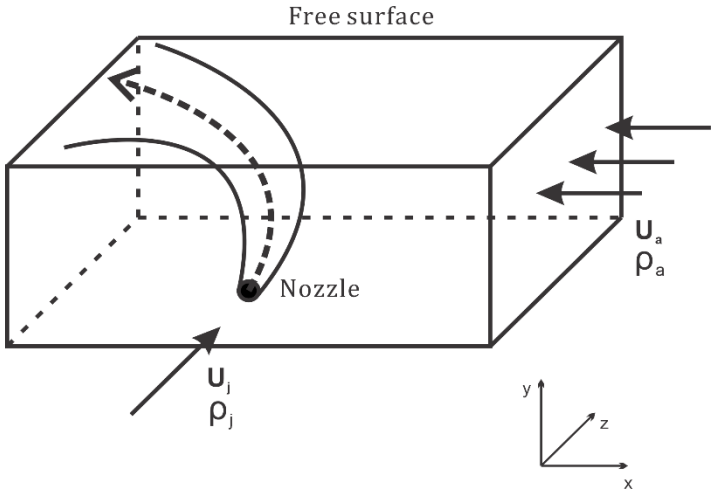


Figure 3.2 Domain of simulation and initial conditions with coordinates.

The dimensions of the computational domain were set to be as similar as possible to those of the experimental study of Johnston et al. (1993). Johnston's experimental study was conducted in a $10m \times 3.5m \times 0.3m$ water tank with a receiving water depth of approximately $0.2m$. This numerical study does not model the free surface, which means that free surface fluctuations are ignored. Therefore, the resulting dimensions of the numerical study are $10m$ length, $3.5m$ width and $0.2m$ deep. The round nozzle is placed at one of the side walls with its center location around $x = 5m, z = 0.02m$ and diameter $0.027m$ are based on the experiment.

The boundaries in OpenFOAM are represented as patches composed of cell faces. In order to solve the PDEs, all patches and internal fields need to be assigned appropriate boundary and initial condition. **Table 3.1** summarizes all boundary conditions employed for the simulation using the Realizable $k - \varepsilon$ turbulence model. In the **Table 3.1**, **U, P_rhg, alpha.Swater, k, ε** and **nut** represent boundary conditions for the velocity, pressure minus hydrostatic pressure, volume fraction of jet fluid, turbulent kinetic energy, turbulent kinetic energy dissipation and eddy viscosity, respectively.

Table 3.1 Boundary conditions for simulation using the Realizable $k - \varepsilon$ turbulence model.

Boundary conditions	U	P_rgh	alpha.Swater	k
Inlet	fixedValue	fixedFlux Pressure	zeroGradient	fixedValue
Outlet	zeroGradient	totalPressure	zeroGradient	zeroGradient
Atmosphere	Slip	fixedFlux Pressure	inletOutlet inletValue 0 Value 0	zeroGradient
Front wall	noSlip	fixedFlux Pressure	zeroGradient	kqRWall Function
Back wall and Bottom	noSlip	fixedFlux Pressure	zeroGradient	kqRWall Function
Nozzle	fixedValue	fixedFlux Pressure	fixedValue	fixedValue

ε	nut
fixedValue	zeroGradient
zeroGradient	zeroGradient
zeroGradient	zeroGradient
epsilonWall Function	nutWall Function
epsilonWall Function	nutWall Function
fixedValue	zeroGradient

Initial values related to the nozzle boundary condition are given by:

$$u_1 = U \tag{3.10}$$

$$u_2 = u_3 = 0 \tag{3.11}$$

$$k = 0.06u_1^2 \tag{3.12}$$

$$\varepsilon = 0.06\frac{u_1^3}{d} \tag{3.13}$$

where u_1, u_2, and u_3 are velocity components, U is the jet velocity for the nozzle boundary and d is the nozzle diameter. For the inflow, a similar approach is used with *d* being the water depth. Eq. (3.12) and Eq. (3.13) are based on Huai et al. (2010). The *noSlip* condition, which sets the velocity value to zero at walls, is used for the velocity boundary conditions of solid wall patches. The *zeroGradient* boundary condition simply adopts the quantity from the nearest cells to the patch. The *fixedValue* condition specifies a fixed quantity for the patch. The *fixedFluxPressure* and the *TotalPressure* are boundary conditions for pressure. The former could adjust the pressure gradient

according to the flux value specified by the velocity boundary condition.

The *TotalPressure* condition sets the static pressure at the patch based on a specified total pressure, which is uniformly zero in this study. The *inletOutlet,* in most cases, is the same as the *zeroGradient* condition, but switches to *fixedValue* if the flow enters the domain. The value of the *fixedValue* is specified under a subdirectory called *inletValue*.

For variables of the model transport equations, including k, ε and nut (eddy viscosity), solid wall patches are handled using the wall functions (Balen et al., 2010). According to the turbulent boundary layer theory, the near wall regions can be divided into three sections: the viscous sub-layer, the buffer layer and the logarithmic layer. The wall functions refer to the empirical equations for the wall region. The use of these equations can ensure better simulation within the turbulent boundary layer.

Another required boundary condition is the free surface. Ideally, one should ensure the best similarity between the computational setup and the real flow setting. However, due to the limitations of computational resources, some compromises may be needed. The simulations herein are only concerned with the turbulent mixing and transport below the free water surface. In addition, the volumetric flux of the jet is extremely small compared to the amount of water in the tank. Therefore, the authors reasonably assumed here that the free surface is flat with very small fluctuations. The approach applied here is to use a rigid lid with *slip* boundary condition (shown in **Table 3.1**), which is a frictionless wall boundary condition. This would allow flow in the horizontal directions, but set the flow to zero in the vertical direction, thus preventing both inflow and outflow. By considering the nature of the simulations, this is considered a reasonable approximation which eliminates the presence of the free surface, without excluding its influence on jet development below the water.

3.3.2 Solution procedure and Turbulence models

The solver used in this study is an adjusted twoLiquidMixingFoam. Details with respect

to this solver are discussed later in this section. The twoLiquidMixingFoam is a transient multiphase solver for two incompressible miscible liquids using the Pressure Implicit Method for Pressure-Linked Equations (PIMPLE) algorithm, and it has been used and validated in many studies. The procedures of PIMPLE algorithm are shown in **Figure 3.3** below.

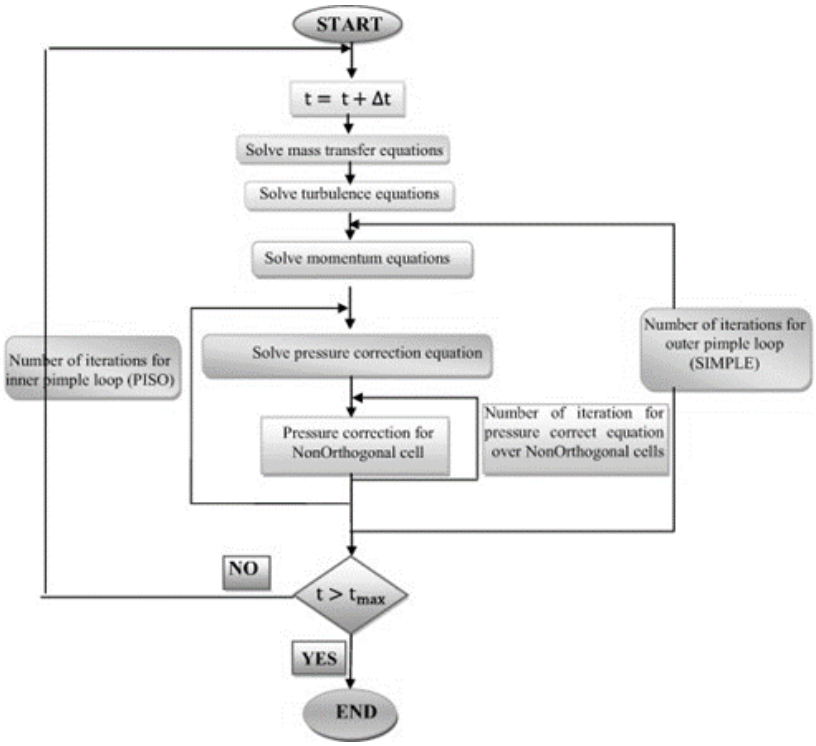


Figure 3.3 Flowchart of the PIMPLE solution procedures used in OpenFOAM (Roohi et al., 2016).

For any multiphase solver such as the twoLiquidMixingFoam in OpenFOAM, the parameter of volume fraction α_i, which represents the proportion of the volume of a single fluid in each cell of the computational domain, is used for tracking the interaction of multiple fluid phases. Due to the law of mass conservation, the following constraint needs to be satisfied:

$$\sum_{i=1}^{n} \alpha_i = 1 \tag{3.14}$$

Additionally, the bulk density ρ at each location every time step is defined as follows:

$$\rho = \sum_i \rho_i \alpha_i \tag{3.15}$$

However, Eq. (3.15) may not be directly applicable to the mixing process caused by the

temperature-induced differences. This is because the thermal energy transfer is mainly based on mass (not volume). Therefore, the volume fraction was converted into the equivalent mass fraction before calculating the temperature dilutions. This solver is revised to compute field density in two steps instead of one:

1- Convert volume fraction into corresponding mass fraction based on the volume-mass relation, and calculate mixing temperature T as:

$$T = \alpha_{1,m} T_1 + \left(1 - \alpha_{1,m}\right) T_2 \tag{3.16}$$

where $\alpha_{1,m}$ is mass fraction conversion from volume fraction α, T_1 and T_2 are temperature of the jet and the receiving water respectively. Therefore, the resulting field temperature $T(\boldsymbol{x}, t)$, which is dependent on both simulation time and location in the domain, can be expressed as follows:

$$T(\boldsymbol{x}, t) = T_1 \frac{\alpha_1 \, \rho_j}{\alpha_1 \, \rho_j + (1 - \alpha_1) \, \rho_a} + T_2 \frac{(1 - \alpha_1) \, \rho_a}{\alpha_1 \, \rho_j + (1 - \alpha_1) \, \rho_a} \tag{3.17}$$

Where ρ_j and ρ_a are density of the jet and the receiving water.

2- Calculate the density field using the Millero & Poisson equation (Eq.3.5), which describes the non-linear relation between temperature (ºC) and water density.

Further, Eq. (3.15) is replaced by Eq. (3.16) and Eq. (3.17). This change is mainly reflected in the *CreateFields.H* file in the OpenFOAM library. The entire *CreateFields.H* is presented in **Appendix B**. The application of this modified twoLiquidMixingFoam solver can better represent the influences of temperature mixing on density variations. It also generated more consistent results compared to the experimental data in the present study.

In the present paper, three different turbulence models, including (1) the Realizable $k - \varepsilon$ model, (2) the SST $k - \omega$ model and (3) the k-equation LES model are applied to jets discharging in the crossflow. These three models are all typically used in various fluid dynamic problems. The Realizable $k - \varepsilon$ and SST $k - \omega$ models are improved versions of the standard $k - \varepsilon$ and $k - \omega$ models, respectively. Both of them use the RANS method and two-equation closures. In this study, the authors also used an LES model developed in OpenFOAM platform called k-Eqn LES, which is a typical LES

model with one equation eddy-viscosity model, and is expected to perform better in predicting the turbulent mixing behavior compared to the RANS models as it applies a spatial (or low-pass) filtering operation. This allows it to compute the large eddies and model the effect of the small eddies with an extra sub-grid scale (SGS) model. The results of these three models are compared with the those from the experimental results of Johnston et al. (1993) and subsequently discussed.

3.3.3 Mesh generation

The performance and accuracy of the numerical study largely depends on the computational mesh quality. The latter has to be fine enough in order to capture the essential features of the real phenomenon.

The mesh generator used in this article is the SnappyHexMesh in OpenFOAM, which can generate 3D meshes containing hexahedra (hex) and split-hexahedra (split-hex) automatically from triangulated surface geometries. This utility is widely-used in various complex geometries because of its ability to conform to the surfaces by iteratively refining the mesh. **Figure 3.4** provides a summary of the major procedures of mesh generation using SnappyHexMesh utility.

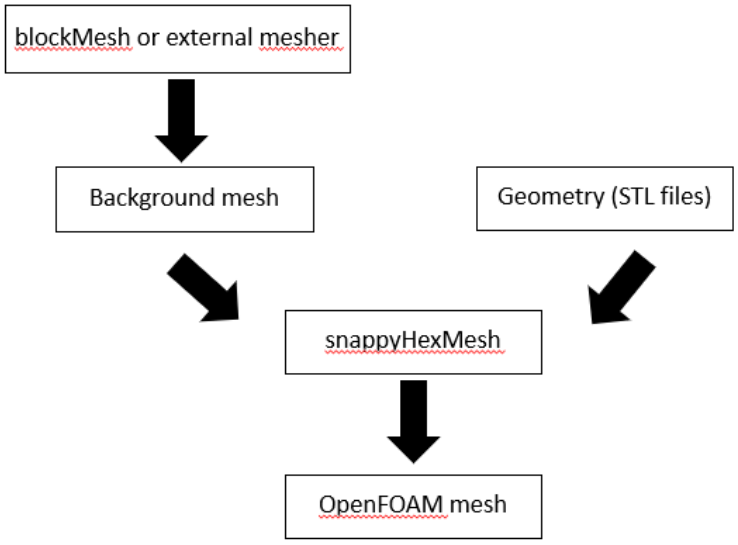


Figure 3.4 The procedures of mesh generation in snappyHexMesh.

In this study, the jet nozzle is given a higher refinement level than the rest of the patches (shown in **Figure 3.5**) due to its small size. In addition, further refinement was also

performed as shown in **Figure 3.6.**

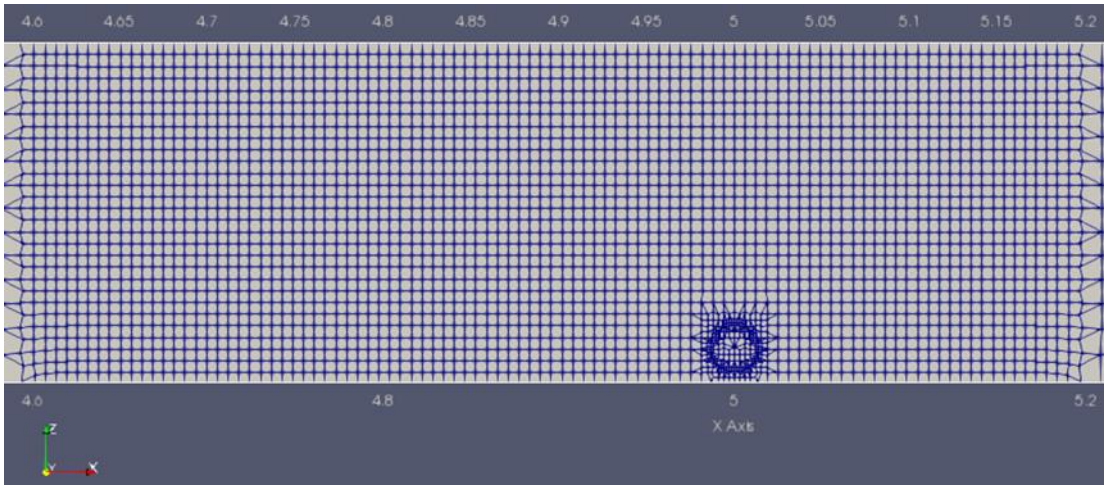


Figure 3.5 Side view of the mesh after using snappyHexMesh.

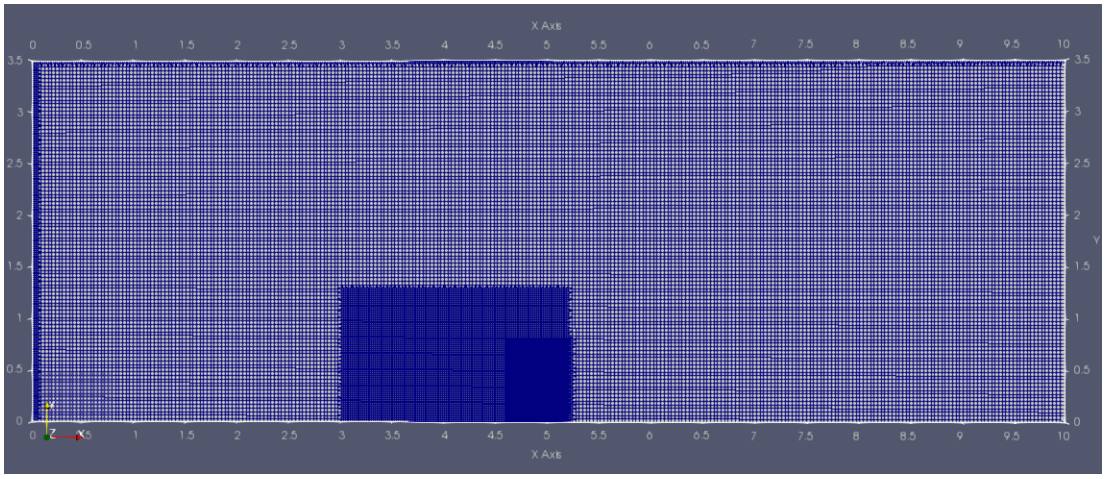


Figure 3.6 Plan view of the mesh after using snappyHexMesh.

3.4 Convergence tests

Model convergence is tested with systematically refined grids. The resulting coarse, medium, fine and extra fine meshes are used in the convergence test. **Table 3.2** is a summary of some the model properties and of the mesh quality parameters of these grids.

Table 3.2 Number of cells and mesh quality parameters of the grids.

Grid	Cells	Max volume	Max aspect ratio	Max skewness
Coarse	1.12×10^{5}	2.33×10^{-7}	8.72	1.28
Medium	5.09×10^{5}	4.84×10^{-8}	4.83	1.74
Fine	1.18×10^{6}	1.52×10^{-8}	4.42	1.20
Extra fine	1.39×10^{6}	1.17×10^{-8}	5.82	1.94

Two velocity profiles right in front of the center of the jet nozzle are used to check for convergence. For each velocity profile, the velocity magnitudes $|u| = \sqrt{u_1^2 + u_2^2 + u_3^2}$ are nondimensionalized by the crossflow velocity U_a with the results plotted against the nondimensionalized distance x/d.

Figure 3.7 (a) and (b) show test runs with profiles located at $y = 1d$ and $y = 5d$ right in front of the jet nozzle, respectively. The Realizable $k-\varepsilon$ turbulence model is used in these two tests. The grid convergence tests were run with three different mesh grids: coarse, medium and fine. The results showed that the medium and fine grids converged towards the same velocity profile, while the coarse grid underestimated the velocity in the proximity of the jet nozzle.

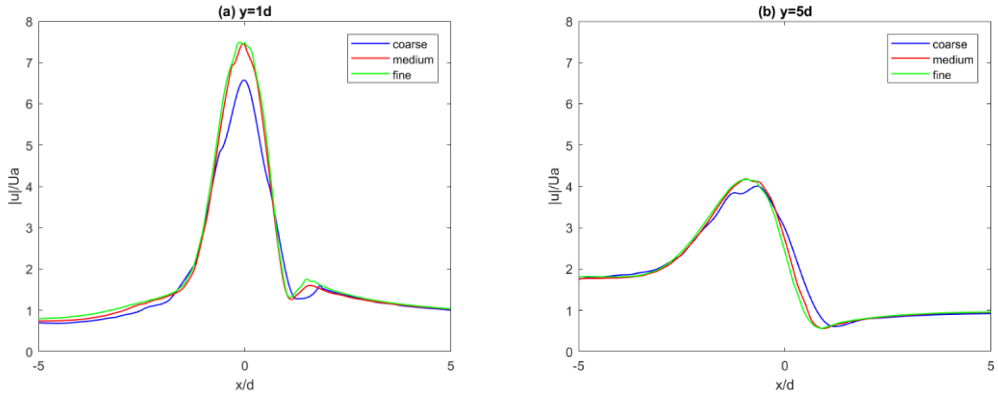


Figure 3.7 Results of convergence test for Realizable $k-\varepsilon$ turbulence model: (a) velocity profile located at $y = 1d$. (b) velocity profile located at $y = 5d$.

Similar tests were also run with medium, fine and extra fine grids for the SST $k-\omega$ model. Corresponding results (**Figure 3.8** (a) and (b)) showed that the medium and the fine grids converged towards the same velocity profile, while the coarser grid underestimated the velocity in the proximity of the jet nozzle.

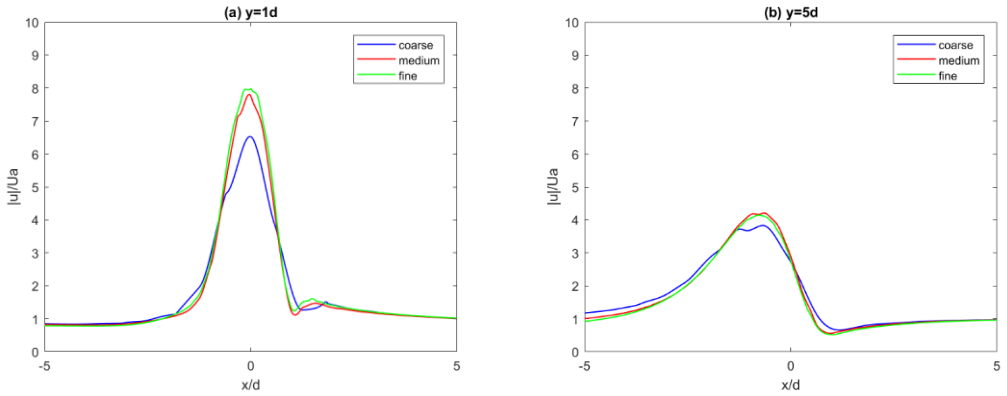


Figure 3.8 Results of convergence test for SST $k-\omega$ model: (a) velocity profile located at $y = 1d$. (b) velocity profile located at $y = 5d$.

For the k-Equation LES model, a finer mesh, compared to that used with the RANS models, was developed in order to trace the behavior of large eddies. Therefore, the grid convergence tests were run with medium, fine and extra fine grids (**Figure 3.9** (a) and (b)). The results showed that the fine and the extra fine grids converged towards the same mean velocity profile, while the medium grid underestimated the velocity in the proximity of the jet nozzle.

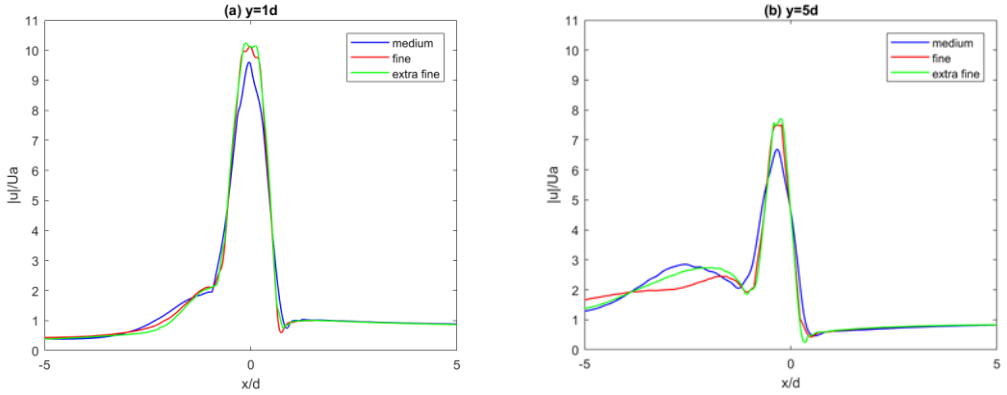


Figure 3.9 Results of convergence test for the k-Equation LES model: (a) velocity profile located at $y = 1d$. (b) velocity profile located at $y = 5d$.

From these convergence tests, the further refined meshes with no significant changes in the velocity profiles by were considered to be fine enough and selected based on considerations related to the computational cost. As such, the medium mesh was used for the Realizable $k-\varepsilon$ and SST $k-\omega$ models, while the fine mesh was used for the kEqn LES model.

3.5 Result and Discussion

3.5.1 Comparison between experimental and numerical data

In order to compare and verify the models, 4 cases from the referential experiment were selected as the modeling conditions. **Table 3.3** and **Table 3.4** present summaries of these conditions and of some of the most relevant parameters. u_a and u_j, which are the velocities of the jet and crossflow, respectively, jointly determined the velocity ratio r, which is an important indicator of the crossflow intensity. Densimetric Froude number F_0 is the ratio of the flow inertia to buoyancy forces. Additionally, L_M and G_M are length and buoyancy scales respectively (Eq.3.1 and Eq.3.2), and they are commonly used for non-dimensional analysis.

It is worth mentioning that Plumes #1-8 in the original experiment by Johnston, which share the same location parameter with Plumes #9-12, were not selected.

Table 3.3 Conditions of the numerical studies based on the corresponding experiments.

#Plume	u_a(m/s)	u_j(m/s)	r (u_a/u_j)
9	0.02	0.4	0.05
10	0.04	0.4	0.10
11	0.06	0.429	0.14
12	0.14	0.424	0.33

[1] **Note:** $(h - z_o)/z_o \approx 9.06$

Table 3.4 Parameters of interests of the numerical studies.

#Plume	F_o	L_M	G_M	ρ_j(kg/m³)
9	14.43	0.374	0.00182	995.366
10	14.49	0.376	0.00180	995.393
11	14.55	0.378	0.00203	995.007
12	14.56	0.378	0.00199	995.074

Since the mixing of the buoyant jet is a three-dimensional process, both the horizontal and vertical jet trajectories are important aspects of the jet's physical behavior. The three-dimensional jet trajectory is defined as the time-averaged streamline across the center of the round jet exit (Kim and Cho, 2006; Zhang and Yang, 2017). Additionally, the average dilution along the streamline is also another aspect that was considered.

Horizontal jet trajectory

Horizontal jet trajectory is obtained by projecting the three-dimensional time-averaged center streamline to the $x-y$ plane. **Figure 3.10** (a)-(d) compare the resulting horizontal jet trajectories of different turbulence models with the experimental results. The length scale used for nondimensionalization is L_M as the jet is momentum-dominated and only the mixing behavior in the near field is considered.

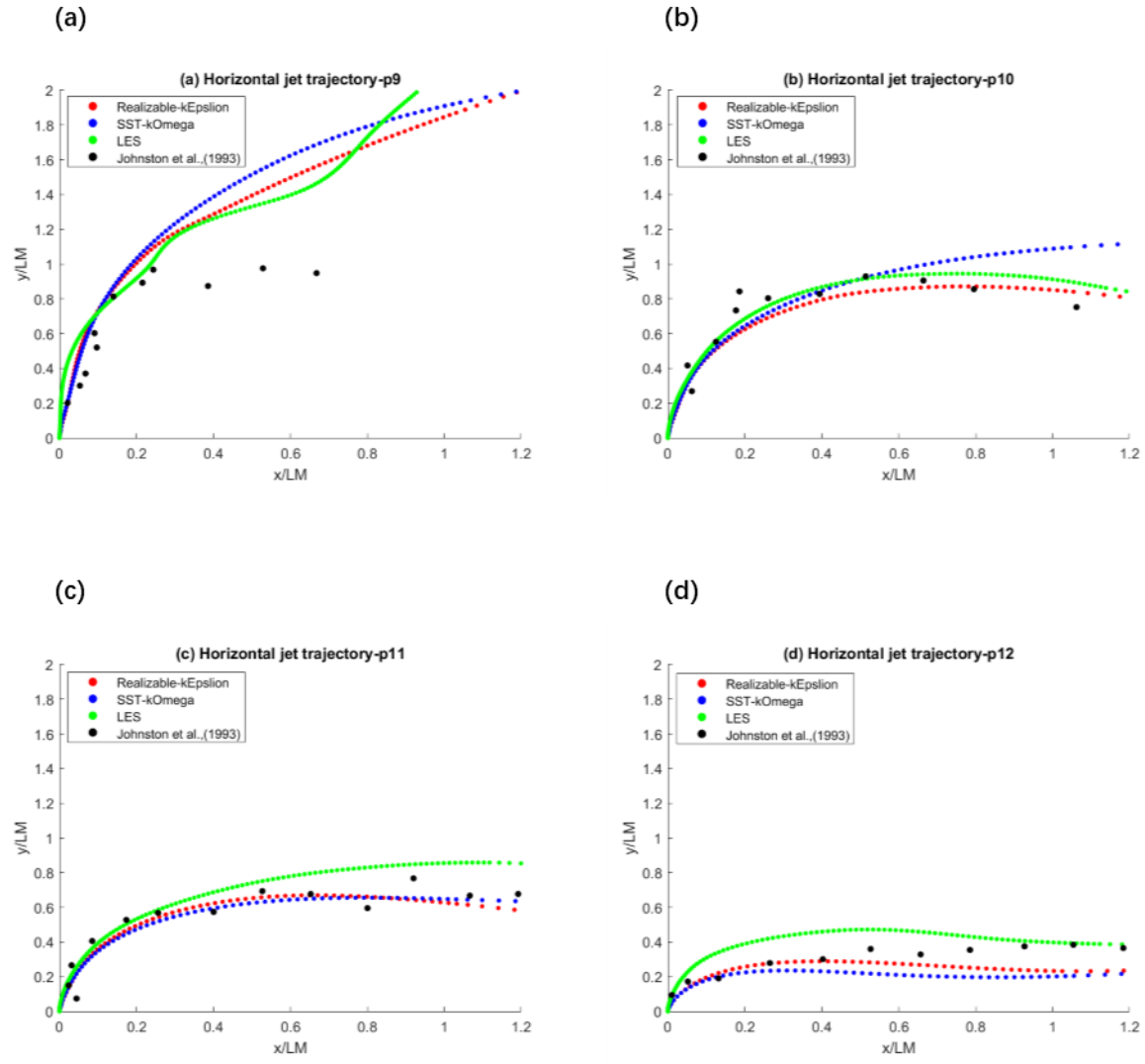


Figure 3.10 Horizontal jet trajectory of the current-induced buoyant jet: (a) Plume 9, (b) Plume 10, (c) Plume 11, (d) Plume 12.

In general, the numerical results of the three turbulence models are in good agreement

with the experimental data, except for Plume #9. Both the numerical and the experimental data show that the penetration of the jet into the ambient water declines as the velocity ratio r increases. This is because the strong momentum of crossflow impedes and bends the jet toward the inner wall. As for the inconsistency observed for the case of plume #9, further experimental and numerical studies are required to determine the reason.

Vertical jet trajectory

Vertical jet trajectory is plotted (**Figure 3.11** (a)-(d)) in terms of the streamwise direction s, which is the incorporation of both $x-$ and $y-$ directions. This allows a more comprehensive representation of how the jet rises along the horizontal plane. Apart from the horizontal penetration, the presence of the crossflow also strongly influences the vertical development of the jet. To be more specific, the vertical rise of the jet is initially restricted in a small crossflow of 0.02 ms^{-1} (plume #9), indicating the attachment of the jet to the bottom. The restriction is released in a larger crossflow of 0.04 ms^{-1} (plume #10), which could be explained by the reducing the bed attachment effect. According to Johnston et al. (1993), the vertical trajectory of plume #10 is highly consistent with the trajectory obtained from a stagnant jet entering unconfined conditions. In an even larger crossflow (0.06 and 0.14 ms^{-1}), the rise of the jet is again restricted, but, this time, it is believed to be caused by the strong momentum of the crossflow itself.

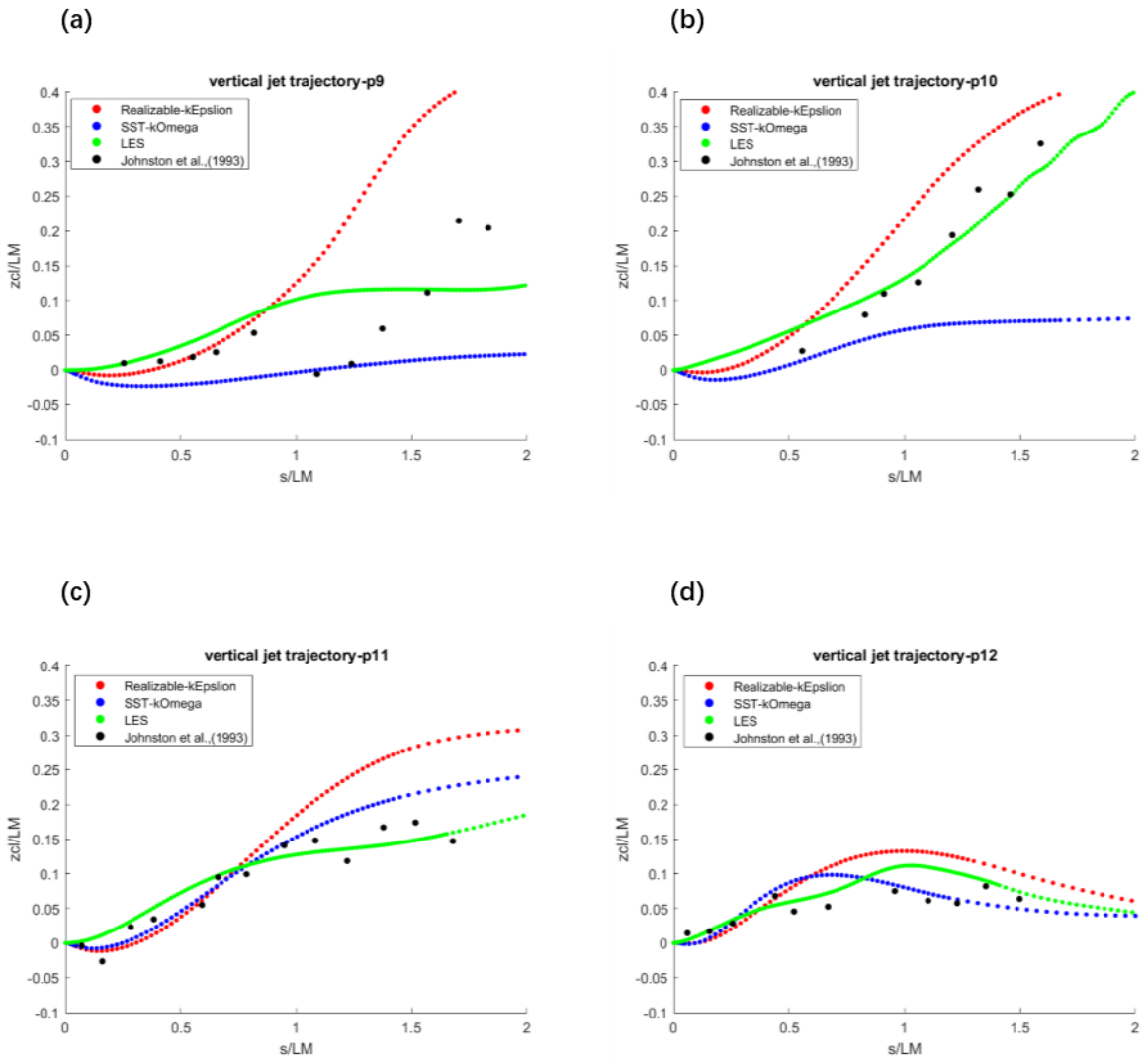


Figure 3.11 Vertical jet trajectory of the current-induced buoyant jet: (a) Plume 9, (b) Plume 10, (c) Plume 11, (d) Plume 12.

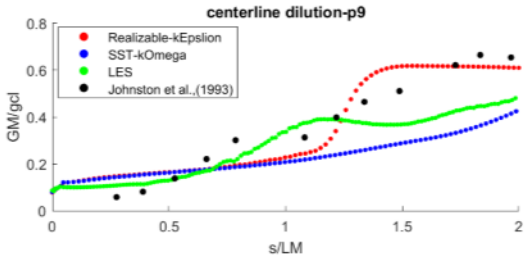
The Realizable $k-\varepsilon$ model tends to overestimate the vertical trajectory elevation, which might show its inaccuracy in predicting the attachment effect. In contrast, the SST $k-\omega$ model seems to exaggerate the effect of attachment to the bed, thus underestimating the vertical rise, especially in relatively small crossflow (plume #9 & plume #10). The LES model generally performs well in predicting the vertical elevation of the jet except for plume #9.

Averaged dilution on the jet trajectory

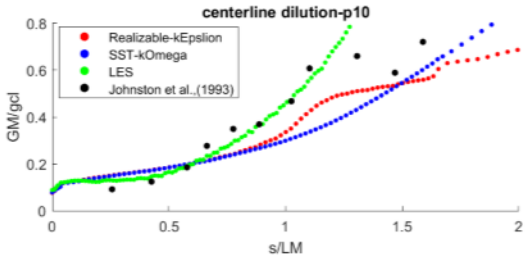
In Johnston's experimental analysis, the nondimensional averaged dilution on the jet trajectory is calculated as $g'_{cl} = \Delta\rho_{cl} g / \rho_w$. The streamwise direction s, instead of the separate $x-$ or $y-$ directions, is used again to represent the averaged dilution on the jet trajectory.

The averaged dilution can be promoted by the crossflow to some extent (**Figure 3.12** (a)-(d)). As indicated in the vertical trajectory, the LES outperforms the two RANS models in general, and both the Realizable $k-\varepsilon$ and SST $k-\omega$ models fail to provide satisfactory predictions of the averaged centerline dilution, predominantly in small crossflow conditions, while, their predictions are much better for cases with large crossflows. The LES model shows good overall performance, regardless of the magnitude of the crossflow velocity.

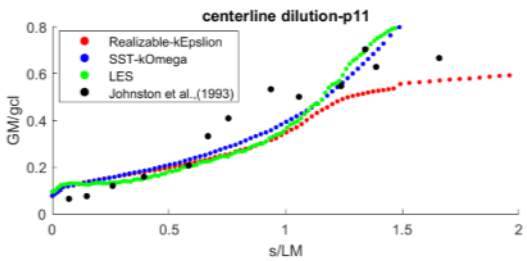
(a)



(b)



(c)



(d)

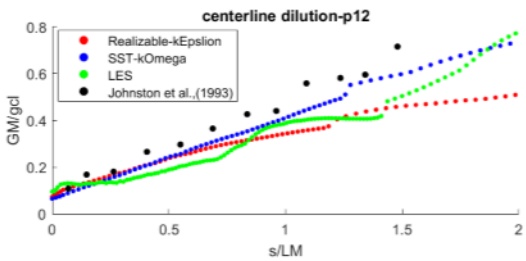


Figure 3.12 Average trajectory dilution of the current-induced buoyant jet: (a) Plume 9, (b) Plume 10, (c) Plume 11, (d) Plume 12.

Based on the results above, the k-Equation LES model is able to predict various aspects of mean jet behavior in most cases. There are only a few exceptions including the averaged centerline dilution for plume #9 and #11 shown in **Figure 3.12** (a) and (c), while the Realizable $k-\varepsilon$ and SST $k-\omega$ models provide better predictions respectively. Given that the LES model has significant advantage of predicting the vertical jet trajectory (**Figure 3.11**), this study postulates it to be the most accurate among those selected to simulate the horizontal buoyant jet entering shallow water with a crossflow. The Realizable $k-\varepsilon$ model and the SST $k-\omega$ model show some inaccuracies in simulating some of the important conditions such as the shallow water and bed restriction. However, the predictions of both of these models in strong crossflows seems to be better than in weak crossflows. This could be due to the fact that in strong crossflow, wall attachment is less significant, thus leading to smaller difference between the general results of LES and RANS models.

3.5.2 Bed attachment effect

Figure 3.13-Figure 3.15 show the simulation results of these three turbulence models for plume #9. All figures include two contour maps of the relative density within two streamwise sections and five streamlines. The relative density is defined as $\rho' = (\rho_a - \rho)/(\rho_a - \rho_j)$. The selected streamlines are shown in **Figure 3.16**, which also share the identical coordinate system with **Figure 3.2**. For the sake of convenience, the authors refer to these five streamlines as the middle, left, upper, right and lower. The middle streamline at the center is shown by red color in the subsequent figures.

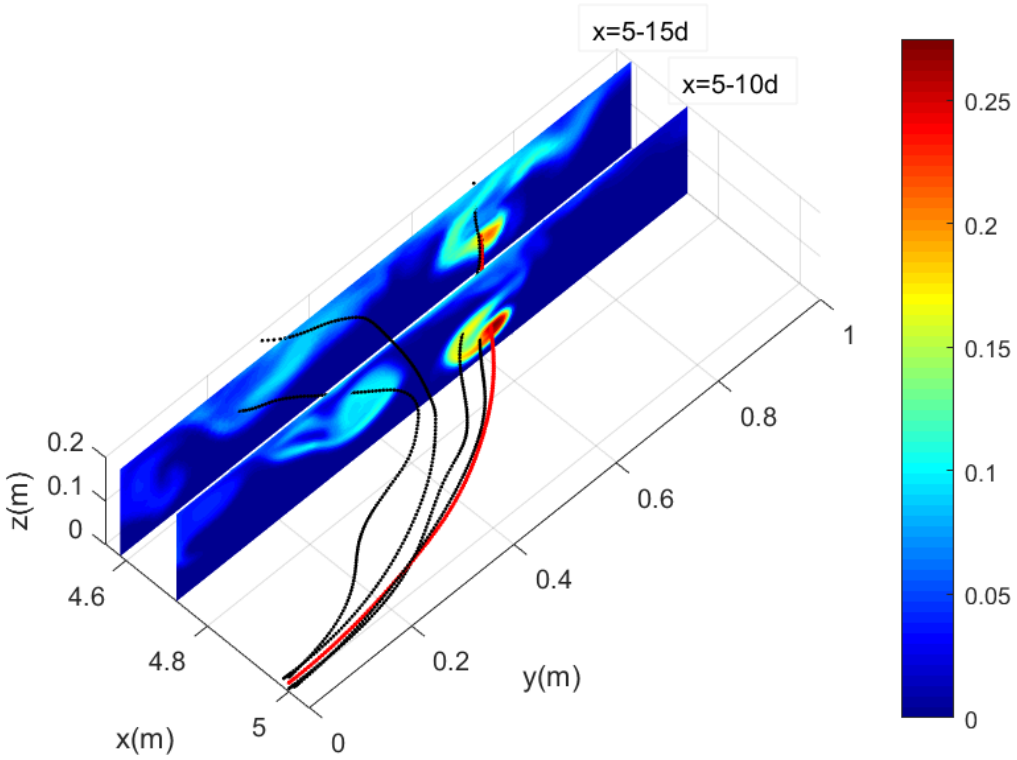


Figure 3.13 Contour maps of the relative density in x=5-10d and x=5-15d and representative streamlines in plume #9, predicted by LES model.

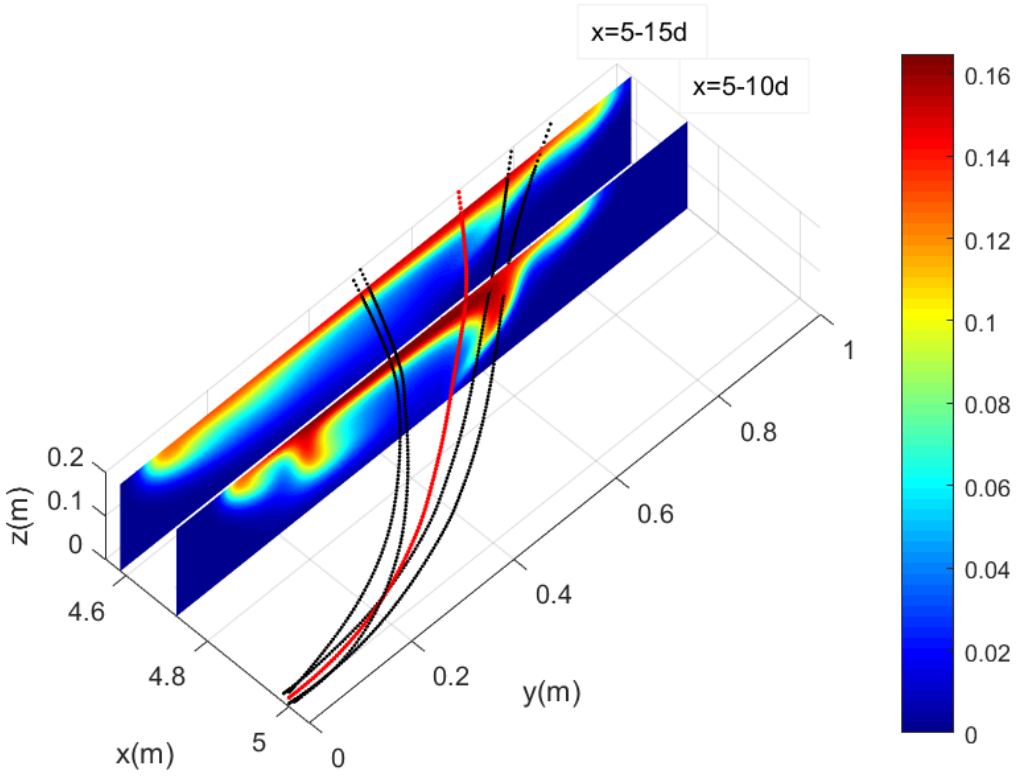


Figure 3.14 Contour maps of the relative density in x=5-10d and x=5-15d and representative streamlines in plume #9, predicted by Realizable $k-\varepsilon$ model.

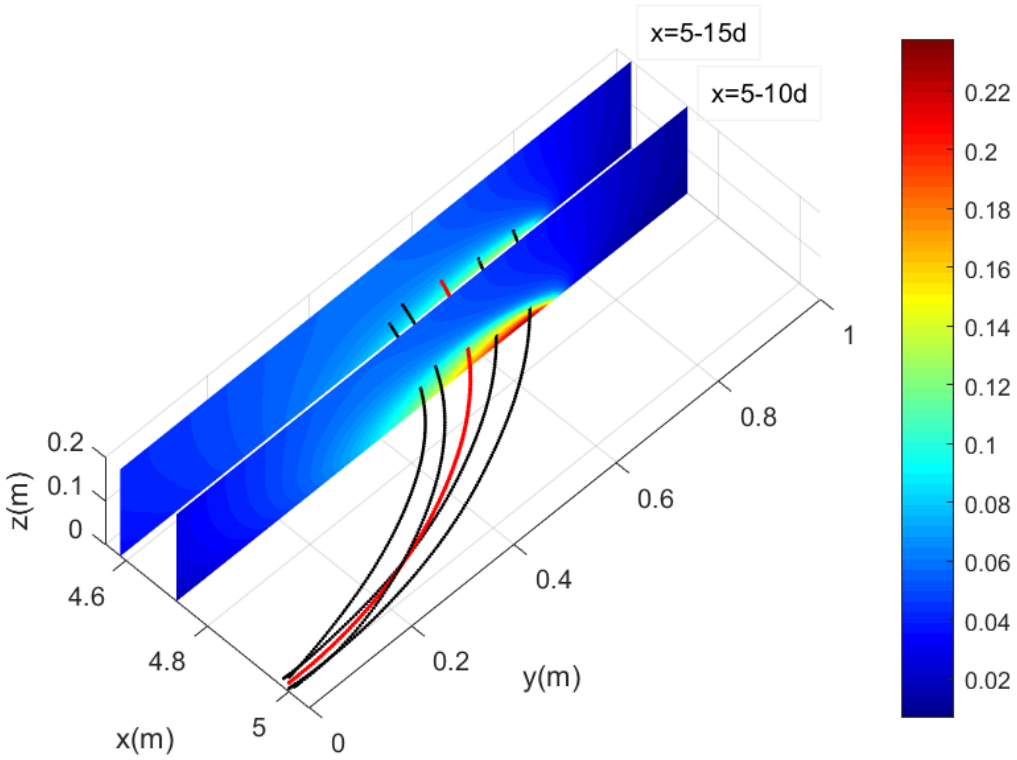


Figure 3.15 contour maps of the relative density in x=5-10d and x=5-15d and representative streamlines in plume #9, predicted by SST $k-\omega$ model.

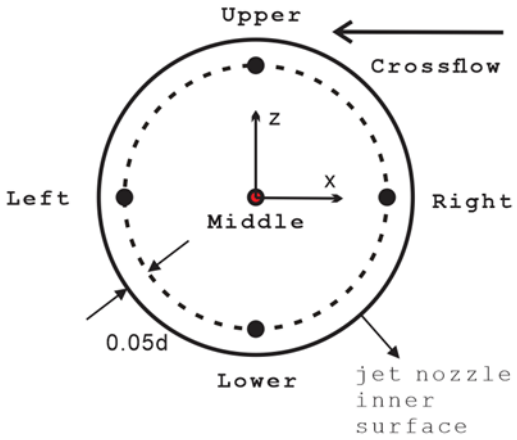


Figure 3.16 Selected three-dimensional streamlines of the flow field.

The LES model clearly shows the separation of the jet into two parts, a phenomenon caused by the effect of bed attachment. The trajectory that is subjected to the attachment effect is bent less by the crossflow, thus penetrating deeper into the ambient water than the one that is not attached to the bed. The middle streamline (in red), which has been

selected to represent the jet trajectory previously, is strongly attached to the bed. This could explain the discrepancies of predictions in jet horizontal trajectory between numerical and experimental data shown in **Figure 3.10 (a)**.

Meanwhile, the Realizable $k-\varepsilon$ model fails to predict the attachment effect. The entire jet body rises to the free surface without any delay. In contrast, the SST $k-\omega$ model greatly exaggerates the attachment effect and results in very little vertical rise of the jet. Predictions of the general spatial density distributions by these turbulence models become more uniform when the effect of bed attachment is restrained or eliminated by a stronger crossflow. Plume #10, where the crossflow-to-jet velocity ratio is 0.10, is believed to be fully free from the bed attachment effect according to both Johnston et al. (1993) and **Figure 3.17**. When the velocity ratio continues to increase, such as Plume #11 with R$= 0.14$, the crossflow not only removes the attachment effect, it also puts extra restriction on the elevation of the jet (**Figure 3.18**). This explains why predictions of the jet trajectories for Plume #10, #11 and #12 by the three turbulence models are more similar compared to the predictions for Plume #9. This also indicates the fact that even the RANS models are generally capable of capturing some mean field features of jets if bed attachment is not present.

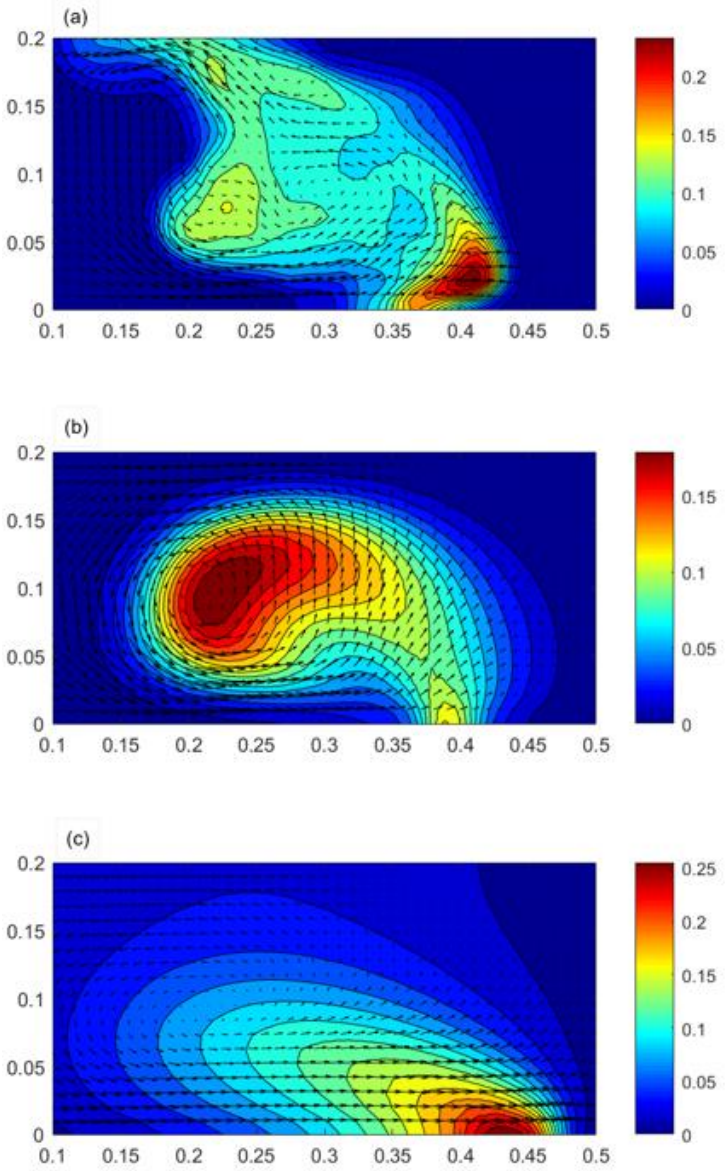


Figure 3.17 Contour maps of the relative density of Plume #10 in x=5-10d with velocity vector (a) LES model, (b) Realizable $k-\varepsilon$ model, (c) SST $k-\omega$ model.

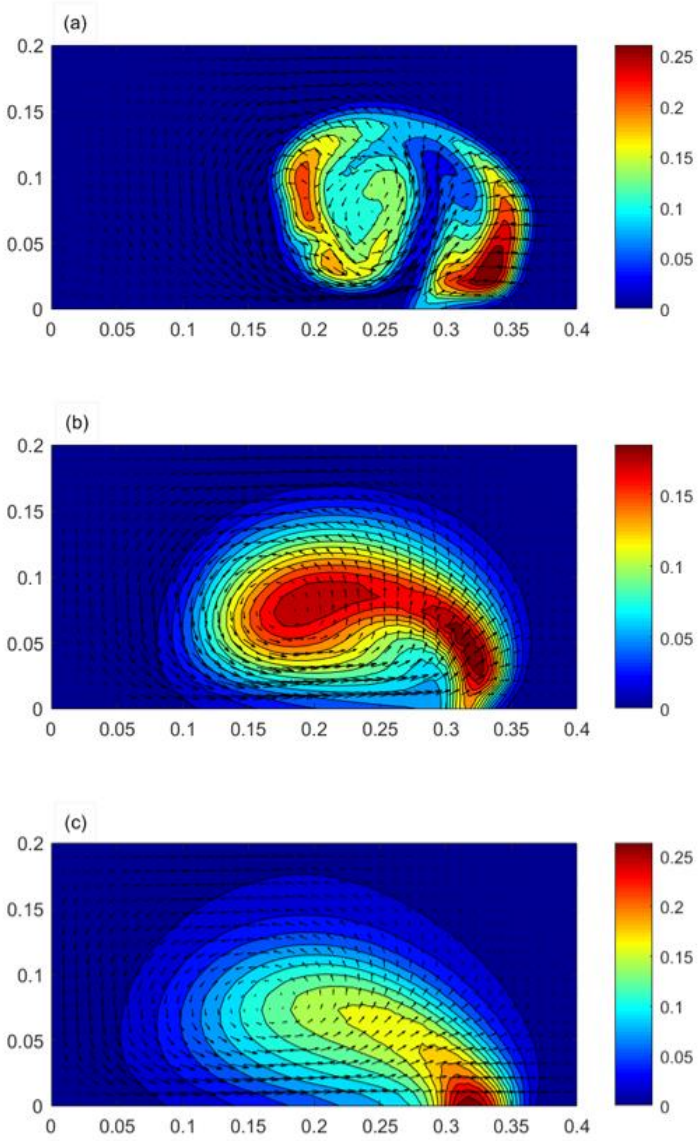


Figure 3.18 Contour maps of the relative density of Plume #11 in x=5-10d with velocity vector (a) LES model, (b) Realizable $k-\varepsilon$ model, (c) SST $k-\omega$ model.

However, one may still argue that the LES model can predict the relatively small and instantaneous turbulent structures better than the other two models and can reproduce more details of the mixing process.

3.5.3 Variations in density difference and its influences on the bed attachment

According to the discussion in the previous section, the dilution level of the attached section can be low compared to other regions surrounding it, which should be of importance for outfall design. Based on the parameters of plume #9, the other four numerical cases are used to investigate the influence of the density difference on jet behavior (**Table 4.3**).

Table 3.5 Some important parameters of relevant numerical setups.

Setups	F_o	d(m)	ρ_j(kg/m^3)	T(°C)	$(\rho_a - \rho_j)/\rho_j$
P9-0.8	11.54	0.027	993.77	35.75	0.0044
P9-0.9	12.99	0.027	994.70	33.01	0.0035
P9	14.43	0.027	995.37	30.93	0.0028
P9-1.1	15.87	0.027	995.86	29.27	0.0024
P9-1.2	17.32	0.027	996.24	28.00	0.0020

These cases are designed according to 80%, 90%, 110% and 120% of the Froude number of the original plume #9, respectively, and the corresponding jet density and temperature are computed from different Froude numbers. Apart from the density difference, any other conditions were kept exactly the same as plume #9. **Figure 3.19** demonstrates how the attached sections of jet rise along with the horizontal distance to nozzle center.

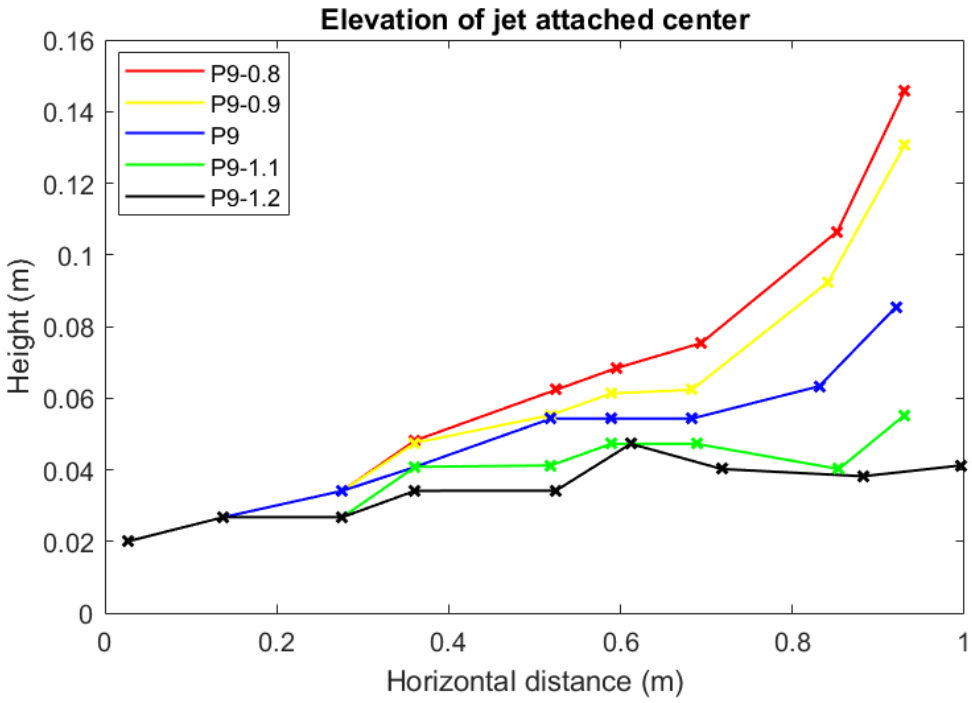
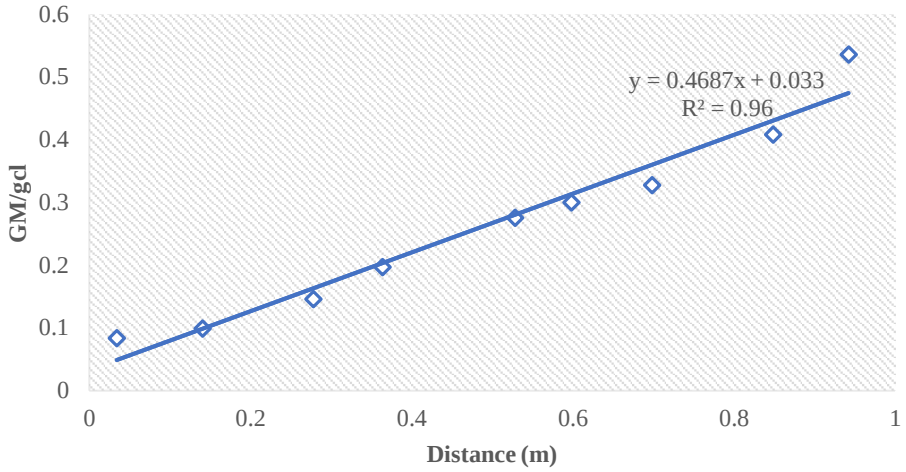


Figure 3.19 Comparison of the elevation of jet attached center along with distance.

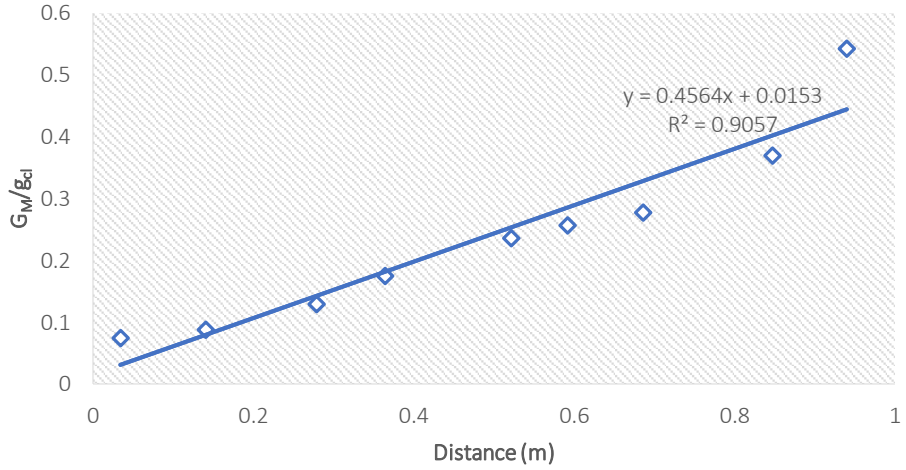
Each point is determined by the identifying the maximum relative density or the minimum dilution in that particular cross section. A smaller Froude number, which corresponds to a greater density difference between the jet and ambient water, enables the jet to detach from the bed more quickly than the larger Froude number. However, the influence of various density differences does not occur until around 0.2m distance from the jet nozzle. Before that, the trends for all cases is relatively uniform. When the density difference is sufficiently small, such as for cases P9-1.1 and P9-1.2, the bed attachment effect can endure over a very long distance. This could lead to a less diluted jet.

The nondimensional dilution parameters G_M/g_{cl} of the points with minimum dilution in each slice are plotted versus the three-dimensional distance to the center of nozzle $\sqrt{(x-5)^2+y^2+z^2}$, and shown in **Figure 3.20** below.

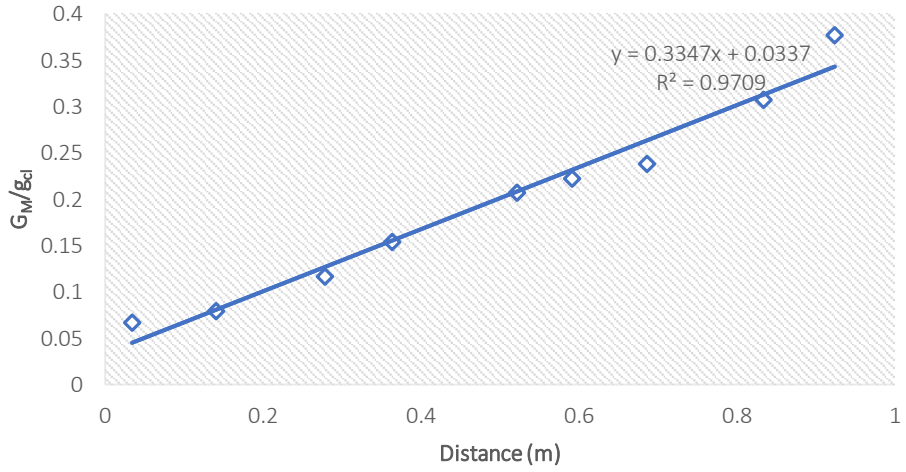
(a) P9-0.8



(b) P9-0.9



(c) P9



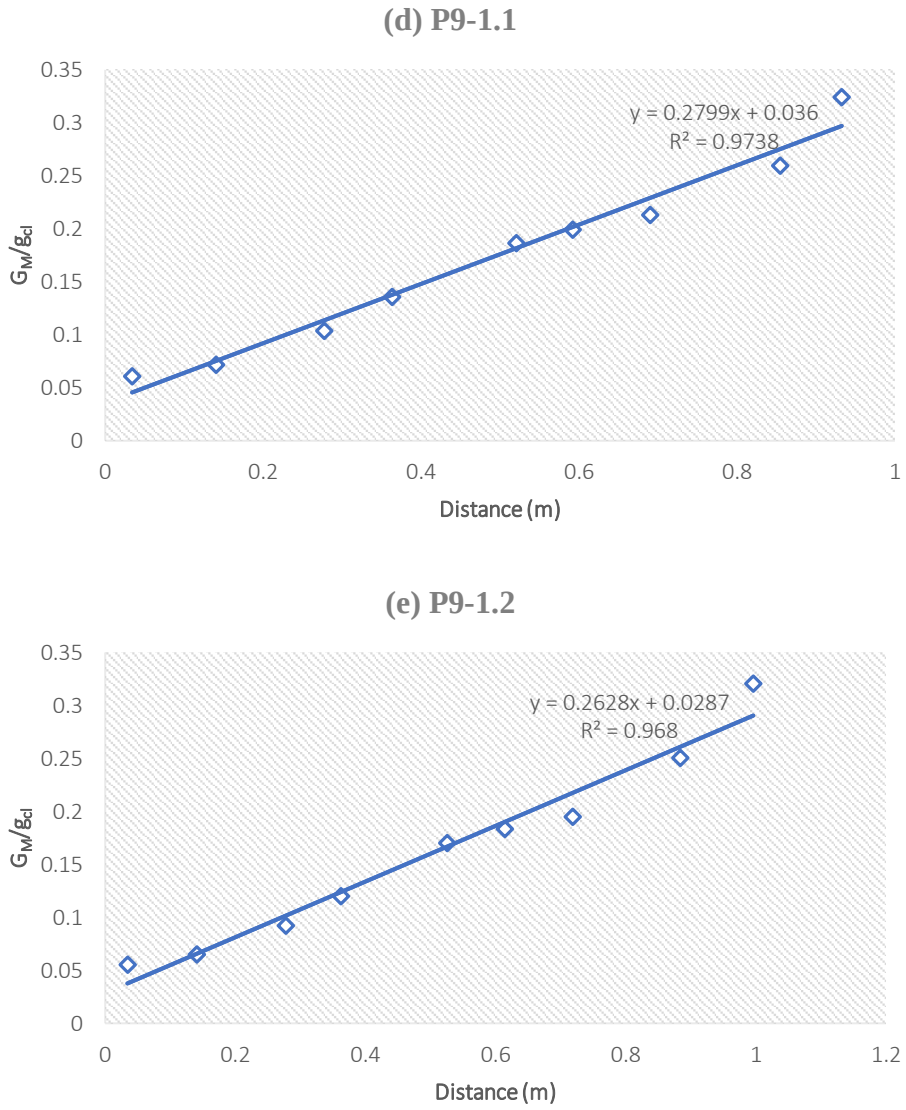


Figure 3.20 Plots of the nondimensional dilution versus the distance for (a) P9-0.8, (b) P9-0.9, (b) P9, (d) P9-1.1, (e) P9-1.2.

The distribution of the points for all cases is generally linear. A linear regression analysis is performed for each case. The calculated R^2 values are all larger than 0.9, which means that the linear model is a good representation of the experimental data.

3.6 Conclusion

A horizontal jet entering shallow water with a crossflow was investigated in this study. In the numerical setup, the jet is positively-buoyant as its density is lower than that of the ambient water and is discharged from a side nozzle in the proximity of the bed

boundary. The major conclusions of this paper are:

- The LES model generally outperformeds the other two models in modelling the horizontal jet entering shallow water with a crossflow.
- Both employed RANS models (realizable k-ε and SST k-ω model) exhibited weaknesses in predicting the bed attachment effect. However, they were still capable of predicting the general density distribution in the flow field when the attachment effect was not present.
- There is a strong linear correlation between the center dilution level of the clinging section and its distance to the nozzle. Additionally, the dilution rate grows with the increasing density difference within a certain range.

Chapter 4 Conclusions and Recommendations

4.1 Conclusions

Discharge of the thermal water generated by power plants back into different water bodies using a side outfall is a common method that is widely used in engineering practices. Therefore, studying the mixing and transport behavior of this type of discharge in the near field has become a research priority to reduce the potential impacts of thermal pollution on the aquatic systems. This thesis considered three turbulence models, which are realizable $k-\varepsilon$ model, SST $k-\omega$ model and k-equation LES model. The experimental data, which was used to validate the numerical models, derived from Johnston et al. (1993). The numerical simulations were conducted in OpenFOAM using a modified twoLiquidMixingFoam solver, which used the mass fraction instead of the volume fraction to compute mixing temperature, and considered the non-linear relation between the water temperature and its density. The numerical setups covered both a wide range of crossflow-jet velocity ratio and Froude number (or density difference) in order to investigate the influences of both factors on the jet development. The computational grids used in simulations were selected based on a set of convergence tests.

In the section of model validation, the horizontal, vertical jet trajectory and the average trajectory dilution were checked in order to validate the numerical models, since the combination of these three parameters could provide an overall description of the mean field behavior of jet. Correspondingly, even though there is a strong consistency between the numerical and experimental results in terms of these parameters, it does not necessarily mean that the numerical models are able to capture every detail of the mixing and transport processes due to the presence of instantaneous vortex structures and secondary flows. The results showed that LES model with the newly modified

solver generally performs well in this condition compared to other two turbulence models. While, predictions by realizable $k-\varepsilon$ and SST $k-\omega$ models are only consistent with the experiment in relatively large crossflow-jet velocity ratio (R= 0.14 for Plume #11 and R= 0.33 for Plume #12).

Plume #9 demonstrates a strong bed attachment effect, in which a portion of the jet is confined to the bed and penetrates deeper into the ambient water, and meanwhile, the rest of the jet is not significantly influenced by the bed and it rises much quicker to the free surface due to the strong buoyancy effect. Both realizable $k-\varepsilon$ and SST $k-\omega$ models have their weakness in predicting the bed attachment effect. The former one is unable to reproduce the bed attachment, while the latter one tends to overestimate the attachment effect as it predicted that the entire jet was confined to the bed.

The bed attachment effect can be restrained or eliminated as the increasing of crossflow-to-jet velocity ratio. As suggested by Johnston et al. (1993), a velocity ratio of R= 0.10 in Plume #10 is enough to remove the bed attachment and can obtain the vertical trajectory that is almost identical to that of the jet entering unconfine environment. This is also consistent with what have been shown in **Figure 3.18**. greater velocity ratio, such as R= 0.14 & 0.33 in Plume #11 and #12, not only removes the bed attachment, it also reduces the vertical elevation of jet in a different manner than the attachment effect (shown in **Figure 3.19**).

The magnitude of Froude number (or density difference) also plays an important role in controlling the attachment effect, thus influencing the jet elevation. In general, the greater jet-to-ambient water density difference is, the easier the jet is to overcome the bed attachment. However, in the early stage of jet development, such influence of variations in density difference is not significant. Additionally, there is a very strong linear relationship between the dilution level at the clinging section and the distance to the nozzle center. Larger density difference also leads to greater dilution rate, which can be described by the slope in linear regression analysis. The slopes are 0.4678, 0.4564, 0.3347, 0.2799 and 0.2628 for P9-0.8, P9-0.9, P9, P9-1.1 and P9-1.2 respectively.

4.2 Limitations and recommendations for Future work

Limitations of this study mainly lie in:

- The duration of all simulations in this study was 70 second.
- This study did not compare the twoLiquidMxingFoam with other multiphase solvers, including pisoFoam, interFoam, interMixingFoam, etc.
- The assumption used to explain the inconsistencies and discontinuity in plume #9 is not sufficiently solid due to the lack of relevant experimental data.
- The linear relationship and its effective range proposed in Section 3.5.3 need further studies.
- This numerical study only covered very limited physical conditions; therefore the applicability of major results needs to be validated in a wider range of physical conditions.

Correspondingly, the following recommendations are proposed for the potential future work related to this topic:

- Simulations can be run for longer duration to study the long-term impacts of the thermal jet on the ambient water.
- Several other multiphase solvers in OpenFOAM, such as pisoFoam, interFoam, interMixingFoam, could also be used to simulate horizontal jet entering shallow water with a crossflow. It could be expected that using of different solvers may result in significant differences in results, even if the turbulence model used would be kept identical, since the mechanics and assumptions behind them vary significantly.
- More experimental data, especially for Plume #9, is required in order to bridge the gap between numerical and existing experimental results.
- More systematic numerical studies can be conducted to further identify the effective range of density difference, which is proposed in section 3.5.3.
- Location of the nozzle can also be considered as a factor to study its influence on the jet development.

References:

Abou-Elhaggag, M.E., El-Gamal, M.H., Farouk, M.I., 2011. Experimental and Numerical Investigation of Desalination Plant Outfalls in Limited Disposal Areas. Journal of Environmental Protection 02, 828.

About OpenFOAM [WWW Document]. CFD Direct. https://cfd.direct/openfoam/about/.

Alameddine, I., El-Fadel, M., 2007. Brine discharge from desalination plants: a modeling approach to an optimized outfall design. Desalination 214, 241–260.

Angelidis, P.B., 2002. A numerical model for the mixing of an inclined submerged heated plane water jet in calm fluid. International Journal of Heat and Mass Transfer 45, 2567–2575.

Archambeault, B., Connor, S., 2008. Proper model validation is important for all EMI/EMC applications, in: 2008 IEEE International Symposium on Electromagnetic Compatibility. Presented at the 2008 IEEE International Symposium on Electromagnetic Compatibility, pp. 1–8.

Awrejcewicz, J., 2011. Numerical Analysis: Theory and Application. BoD – Books on Demand.

Balen, W. van, Blanckaert, K., Uijttewaal, W.S.J., 2010. Analysis of the role of turbulence in curved open-channel flow at different water depths by means of experiments, LES and RANS. Journal of Turbulence 11, N12.

Chakraborty, S., Bhattacharjee, U., n.d. Review of Jets in a Cross Flow-Experimental and Numerical Approach 7, 15.

Corless, R.M., Fillion, N., 2013. A Graduate Introduction to Numerical Methods. Springer New York, New York, NY.

Davies, P.A., Mofor, L.A., Fernando, H.J.S., 1995. Laboratory studies of mixed buoyant jets in shallow cross-flows. technical note. Proceedings of the Institution of Civil Engineers - Water, Maritime and Energy 112, 75–78.

Einav, R., Harussi, K., Perry, D., 2003. The footprint of the desalination processe

s on the environment. Desalination 152, 141–154.

Fearn, R., Weston, R.P., 1974. Vorticity Associated with a Jet in a Cross Flow. AIAA Journal 12, 1666–1671.

Fischer, H.B., List, J.E., Koh, C.R., Imberger, J., Brooks, N.H., 2013. Mixing in I nland and Coastal Waters. Elsevier.

Fric, T.F., Roshko, A., 1994. Vortical structure in the wake of a transverse jet. J. Fluid Mech. 279, 1–47.

Galperin, B., Sukoriansky, S., Anderson, P.S., 2007. On the critical Richardson n umber in stably stratified turbulence. Atmospheric Science Letters 8, 65–6 9.

Greenshields, C., 2018a. OpenFOAM v6 User Guide: 4.4 Numerical schemes [W WW Document]. CFD Direct. https://cfd.direct/openfoam/user-guide/v6-fvsc hemes/

Greenshields, C., 2018b. OpenFOAM v6 User Guide: 4.5 Solution and algorithm control [WWW Document]. CFD Direct. https://cfd.direct/openfoam/user-gu ide/v6-fvsolution/

Greenshields, C., 2018c. OpenFOAM v6 User Guide: 5.4 Meshing with snappyHe xMesh [WWW Document]. CFD Direct. https://cfd.direct/openfoam/user-gu ide/v6-snappyhexmesh/

Holz, M., Heil, S.R., Sacco, A., 2000. Temperature-dependent self-diffusion coeffi cients of water and six selected molecular liquids for calibration in accurat e 1H NMR PFG measurements. Phys. Chem. Chem. Phys. 2, 4740–4742.

Huai, W., Li, Z., Qian, Z., Zeng, Y., Han, J., Peng, W., 2010. Numerical Simulat ion of Horizontal Buoyant Wall Jet. J Hydrodyn 22, 58–65.

Jasak, H., n.d. OpenFOAM: Introduction and Overview 18.

Jauregui, R., Silv, F., 2011. Numerical Validation Methods, in: Awrejcewicz, J. (Ed.), Numerical Analysis - Theory and Application. InTech.

Johnston A. J., Nguyen N., Volker R. E., 1993. Round Buoyant Jet Entering Shal low Water in Motion. Journal of Hydraulic Engineering 119, 1364–1382.

Kamotani, Y., Greber, I., 1972. Experiments on a Turbulent Jet in a Cross Flow. AIAA Journal 10, 1425–1429.

Karagozian, A.R., 2014. The jet in crossflow. Physics of Fluids 26, 101303.

Karagozian, A.R., 2003. Background on and Applications of Jets in Crossflow, in: Karagozian, A.R., Cortelezzi, L., Soldati, A. (Eds.), Manipulation and Co ntrol of Jets in Crossflow, International Centre for Mechanical Sciences. S pringer, Vienna, pp. 3–13.

Keffer, J.F., Baines, W.D., 1963. The round turbulent jet in a cross-wind. Journal of Fluid Mechanics 15, 481–496.

Kheirkhah Gildeh, H., 2013. Numerical Modeling of Thermal/Saline Discharges in Coastal Waters (Thesis). Université d'Ottawa / University of Ottawa. http: //dx.doi.org/10.20381/ruor-3030

Kheirkhah Gildeh H., Mohammadian A., Nistor I., Qiblawey H., 2014. Numerical Modeling of Turbulent Buoyant Wall Jets in Stationary Ambient Water. J ournal of Hydraulic Engineering 140, 04014012.

Kikkert G. A., Davidson M. J., Nokes R. I., 2007. Inclined Negatively Buoyant Discharges. Journal of Hydraulic Engineering 133, 545–554.

Killingstad, P.E., 2018. A study of dead water resistance Reynolds Averaged Nav ier Stokes simulations of a barge moving in stratified waters.

Kim, D.G., Cho, H.Y., 2006. Modeling the buoyant flow of heated water dischar ged from surface and submerged side outfalls in shallow and deep water with a cross flow. Environ Fluid Mech 6, 501–518.

Large Eddy Simulation (LES), CFD WITH A MISSION [WWW Document], n. h ttps://caefn.com/openfoam/les

Lattemann, S., Höpner, T., 2008. Environmental impact and impact assessment of seawater desalination. Desalination 1–3, 1–15.

Lee Joseph H. W., Neville-Jones Peter, 1987. Initial Dilution of Horizontal Jet in Crossflow. Journal of Hydraulic Engineering 113, 615–629.

McGuirk, J.J., Rodi, W., 1979. Mathematical modelling of three-dimensional heat

ed surface jets. Journal of Fluid Mechanics 95, 609–633.

Medina, H.J., Beechook, A., Saul, J., Porter, S., Aleksandrova, S., Benjamin, S., 2015. Open source Computational Fluid Dynamics using OpenFOAM. Pres ented at the Light Aircraft Design: Methods and Tools.

Menter, F.R., 1994. Two-equation eddy-viscosity turbulence models for engineerin g applications. AIAA Journal 32, 1598–1605.

Modeling the mixing of heated water discharged from a submerged multiport diff user. Journal of hydraulic Research, 2000, 38(4): 259-270.

Moukalled, F., Mangani, L., Darwish, M., 2016. The Finite Volume Method in C omputational Fluid Dynamics: An Advanced Introduction with OpenFOAM and Matlab, Fluid Mechanics and Its Applications. Springer International P ublishing, Cham.

Muppidi, S., Mahesh, K., 2005. Study of trajectories of jets in crossflow using di rect numerical simulations. Journal of Fluid Mechanics 530, 81–100.

Panitz, T., Wasan, D.T., 1972. Flow attachment to solid surfaces: The Coanda eff ect. AIChE Journal 18, 51–57.

Papakonstantis, Ilias G., Christodoulou, G.C., Papanicolaou, P.N., 2011. Inclined n egatively buoyant jets 1: geometrical characteristics. Journal of Hydraulic Research 49, 3–12.

Papakonstantis, Ilias G., Christodoulou, G.C., Papanicolaou, P.N., 2011. Inclined negatively buoyant jets 2: concentration measurements. Journal of Hydrau lic Research 49, 13–22.

Phiri, O., Mumba, P., Moyo, B.H.Z., Kadewa, W., 2005. Assessment of the impa ct of industrial effluents on water quality of receiving rivers in urban areas of Malawi. Int. J. Environ. Sci. Technol. 2, 237–244.

Poornima, E.H., Rajadurai, M., Rao, T.S., Anupkumar, B., Rajamohan, R., Narasi mhan, S.V., Rao, V.N.R., Venugopalan, V.P., 2005. Impact of thermal dis charge from a tropical coastal power plant on phytoplankton. Journal of T hermal Biology 30, 307–316.

Purnalna, A., Al-Barwani, H.H., Al-Lawatia, M., 2003. Modeling dispersion of bri ne waste discharges from a coastal desalination plant. Desalination 155, 41 –47.

Rajaratnam, N., Pani, B.S., 1974. Three-Dimensional Turbulent Wall Jets. Journal of the Hydraulics Division 100, 69–83.

Reynolds Stress Model - an overview | ScienceDirect Topics [WWW Document], n.d. https://www.sciencedirect.com/topics/engineering/reynolds-stress-model

Roberts Philip J. W., Ferrier Adrian, Daviero Greg, 1997. Mixing in Inclined Den se Jets. Journal of Hydraulic Engineering 123, 693–699.

Robertson, E., Choudhury, V., Bhushan, S., Walters, D.K., 2015. Validation of O penFOAM numerical methods and turbulence models for incompressible bl uff body flows. Computers & Fluids 123, 122–145.

Roohi, E., Pendar, M.-R., Rahimi, A., 2016. Simulation of three-dimensional cavit ation behind a disk using various turbulence and mass transfer models. Ap plied Mathematical Modelling 40, 542–564.

Ruiz, A.M., Lacaze, G., Oefelein, J.C., 2015. Flow topologies and turbulence scal es in a jet-in-cross-flow. Physics of Fluids 27, 045101.

Shaheed, R., Mohammadian, A., Kheirkhah Gildeh, H., 2019. A comparison of st andard k–ε and realizable k–ε turbulence models in curved and confluent c hannels. Environ Fluid Mech 19, 543–568.

Shao, D., Law, A.W.-K., 2010. Mixing and boundary interactions of 30° and 45° inclined dense jets. Environ Fluid Mech 10, 521–553.

Sharp, J., 1975. Technical note. the use of a buoyant wall jet to improve the dillu tion of a submerged outfall. Proceedings of the Institution of Civil Engine ers 59, 527–534.

Shih, T.-H., 1994. A New Reynolds Stress Algebraic Equation Model.

The structure of a jet in cross flow at low velocity ratios: Physics of Fluids: Vol 16, No 6 [WWW Document], n.d. https://aip.scitation.org/doi/10.1063/1.16 97397

Thompson, D.B., 1992. Numerical Methods 101—Convergence of Numerical Mod els. Presented at the Hydraulic Engineering: Saving a Threatened Resource —In Search of Solutions, ASCE, pp. 398–403.

Vafeiadou, P., Papakonstantis, I., Christodoulou, G., 2005. Numerical Simulation of Inclined Negatively Buoyant Jets. The 9th international conference on e nvironmental science and technology, September. 2005: 1-3.

Versteeg, H.K., Malalasekera, W., 2007. An introduction to computational fluid d ynamics: the finite volume method, 2. ed., [Nachdr.]. ed. Pearson/Prentice Hall, Harlow.

Wright, S.J., 1977. Mean Behavior of Buoyant Jets in a Crossflow. Journal of the Hydraulics Division 103.

Yuan, L.L., Street, R.L., 1998. Trajectory and entrainment of a round jet in cross flow. Physics of Fluids 10, 2323–2335.

Zeitoun, M.A., McIlhenny, W.F., 1971. Conceptual Designs Of Outfall Systems F or Desalination Plants. Presented at the Offshore Technology Conference, Offshore Technology Conference.

Zhang, L., Yang, V., 2017. Flow Dynamics and Mixing of a Transverse Jet in Cr ossflow—Part I: Steady Crossflow. J. Eng. Gas Turbines Power 139.

Zhao, Y.J., Zeng, L., Zhang, A.L., Wu, Y.H., 2015. Response of current, tempera ture, and algae growth to thermal discharge in tidal environment. Ecologic al Modelling, Ecological management for human-dominated urban and regi onal ecosystems 318, 283–292.

Zhiyin, Y., 2015. Large-eddy simulation: Past, present and the future. Chinese Jo urnal of Aeronautics 28, 11–24.

Appendix A

Length scales and Buoyancy scales (from Johnston et al., 1993)

LENGTH SCALES:

$$L_B = Q_o^{3/5}/B_o^{1/5} = (\pi/4)^{2/5} dF^{2/5}$$

$$L_b = B_o/u_a^3 = (\pi/4) d^{5/2} F (g_o')^{3/2} u_a^{-3}$$

$$L_M = M_o^{3/4}/B_o^{1/2} = (\pi/4)^{1/4} dF$$

$$L_B = M_o^{1/2}/u_a = (\pi/4)^{1/2} (g_o')^{1/2} F d^{3/2} u_a^{-1}$$

BUOYANCY SCALES:

$$G_B = B_o/Q_o = g_o'$$

$$G_b = u_a^5/B_o = (\pi/4)^{-1} (g_o')^{-3/2} F^{-1} d^{-5/2} u_a^5$$

$$G_M = B_o^{3/2}/M_o^{5/4} = (\pi/4)^{\frac{1}{4}} (g_o') F^{-1}$$

$$G_m = u_a^3/M_o^{1/2} = (\pi/4)^{-1/2} (g_o')^{-1/2} F^{-1} d^{-3/2} u_a^3$$

Appendix B

CreateFields.H file in the modified twoLiquidMixingFoam

```
Info<< "Reading field p_rgh\n" << endl;
volScalarField p_rgh
(
    IOobject
    (
        "p_rgh",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);

Info<< "Reading field U\n" << endl;
volVectorField U
(
    IOobject
    (
        "U",
        runTime.timeName(),
        mesh,
        IOobject::MUST_READ,
        IOobject::AUTO_WRITE
    ),
    mesh
);

#include "createPhi.H"

Info<< "Reading transportProperties\n" << endl;
incompressibleTwoPhaseMixture mixture(U, phi);

volScalarField& alpha1(mixture.alpha1());
volScalarField& alpha2(mixture.alpha2());

const dimensionedScalar& rho1 = mixture.rho1();
const dimensionedScalar& rho2 = mixture.rho2();
//const dimensionedScalar& Temp1 = mixture.Temp1();
//const dimensionedScalar& Temp2 = mixture.Temp2();
```

```
dimensionedScalar Dab("Dab", dimViscosity, mixture);
dimensionedScalar Temp1("Temp1", dimless, mixture);
dimensionedScalar Temp2("Temp2", dimless, mixture);

// Read the reciprocal of the turbulent Schmidt number
dimensionedScalar alphatab("alphatab", dimless, mixture);

// Need to store rho for ddt(rho, U)
volScalarField T("T",
Temp1*((alpha1*rho1)/(alpha1*rho1+alpha2*rho2))+Temp2*((alpha2*rho2)/(alpha1*rho1+alpha
2*rho2)));
volScalarField rho("rho", rho1/998.2063*(999.842594+6.793952e-2*T-(9.095290e-
3*pow(T,2))+(1.001685e-4*pow(T,3))-(1.120083e-6*pow(T,4))+(6.536336e-9*pow(T,5))));
rho.oldTime();


// Mass flux
// Initialisation does not matter because rhoPhi is reset after the
// alpha1 solution before it is used in the U equation.
surfaceScalarField rhoPhi
(
    IOobject
    (
        "rhoPhi",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::NO_WRITE
    ),
    rho1*phi
);

// Construct incompressible turbulence model
autoPtr<incompressible::turbulenceModel> turbulence
(
    incompressible::turbulenceModel::New(U, phi, mixture)
);


#include "readGravitationalAcceleration.H"
#include "readhRef.H"
#include "gh.H"
```

```
volScalarField p
(
    IOobject
    (
        "p",
        runTime.timeName(),
        mesh,
        IOobject::NO_READ,
        IOobject::AUTO_WRITE
    ),
    p_rgh + rho*gh
);

label pRefCell = 0;
scalar pRefValue = 0.0;
setRefCell
(
    p,
    p_rgh,
    pimple.dict(),
    pRefCell,
    pRefValue
);

if (p_rgh.needReference())
{
    p += dimensionedScalar
    (
        "p",
        p.dimensions(),
        pRefValue - getRefCellValue(p, pRefCell)
    );
    p_rgh = p - rho*gh;
}

mesh.setFluxRequired(p_rgh.name());
mesh.setFluxRequired(alpha1.name());

#include "createMRF.H"
```