Department of Mechanical Engineering

An Optimisation Framework to Reduce the Risk of Hydrate Blockages in Flowlines

Adrian Mark Shuard

This thesis is presented for the Degree of
Doctor of Philosophy
of
Curtin University

October 2017

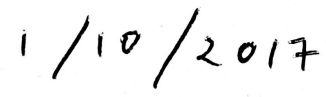
Declaration

To the best of my knowledge and belief this thesis contains no material previously published by any other person except where due acknowledgement has been made.

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university.

Signature:

Date:



Abstract

The formation and subsequent blockage of natural gas hydrates in subsea oil and gas flowlines remains one of the most prevalent flow assurance concerns for a deepwater operator today. The risk for a hydrate blockage to occur is particularly exasperated during any restart operation that follows an extended shut-down period when the fluids contained in the flowline cool to the surrounding sea water temperature and come into contact with any free water that has accumulated at the flowline low-spots. Through the pioneering experimental work of Leporcher et al. [2002] and Volk et al. [2007], it is apparent that a hydrodynamic strategy could potentially be adopted as a means of reducing the risk of hydrate blockage during restart operations, such that the current techniques used to avoid hydrate blockages, including chemical inhibition, thermal control and de-watering exercises, are not required, allowing for production to be brought back online immediately. This dissertation explores the possibility that Computational Fluid Dynamics (CFD) may be used in conjunction with common meta-heuristic optimisation techniques as a tool to assess the risk of hydrate blockage during restart operations in gas-dominated flowlines featuring accumulated water zones, without chemical inhibition and thermal control.

For this purpose, a high-fidelity CFD model of a 3-D half-pipe is first constructed using OpenFOAM® and SALOME in order to investigate the capability of the two-phase interFoam solver to capture turbulent mixtures of air and water and oil and gas at different pipe inclinations. A total of 120 simulations are performed and observations made for stratified, slug, dispersed bubble, annular and froth flow. The values of liquid holdup and pressure drop obtained from the CFD simulations are compared to results obtained from the mechanistic model developed by Petalas and Aziz [2000] at three separate inclinations of +10°, 0° and -10°. The results show that the flow pattern behaviour and numerical values of liquid holdup and pressure drop compare reasonably well to the mechanistic model across all flow regimes considered, giving confidence that the model can be used for optimisation studies, albeit at a computational cost that is too high for practical use.

In order to overcome the excessive computational cost incurred from the high-fidelity CFD model, a modified CFD modelling strategy is proposed using scalable-wall functions for the purpose of minimising the computational expense whilst still providing an acceptable representation of the likely flow patterns involved during a restart operation; stratified and slug flow. This approach is verified using three experimental studies focussing on stratified and slug flow in long pipes and a gas-sweep in a pipe featuring a single low-spot with an accumulated water region. These studies are used to investigate the influence of the longitudinal mesh resolution on the simulation accuracy, where the mesh resolution recommended for use in the framework has been successfully reduced by over 97% when compared to the original model.

Using the modified CFD modelling strategy, a CFD model is constructed to emulate a restart operation of a single low-spot bend using the same geometry presented in the experimental study of Leporcher et al. [2002]. Through this study, a novel strategy for the implementation of a CFD-based optimisation approach to reduce the risk of hydrate blockages in flowlines is proposed. Here, the most appropriate meta-heuristic optimisation techniques available in the DAKOTA toolkit are assessed and a modification is made to the interFoam solver in order to capture the interfacial area between the gas and water phases at each time step, allowing for an estimate of the total amount of hydrate growth to be made using a combination of the kinetic model by Turner et al. [2005] and mass transfer-limited model by Skovborg and Rasmussen [1994]. The results obtained from this study have demonstrated that the surrogate-based optimisation strategy facilitated using DAKOTA has been particularly effective as a means to efficiently project the hydrate growth results obtained from a defined number of sampling points throughout the design space, providing an avenue for the implementation of the optimisation approach in the form of a comprehensive framework.

The optimisation approach used for the single low-spot bend is then further refined through the use of an applied example featuring the Ormen Lange gas field located offshore of Norway. The methodology used in this example is assessed for the purpose of usability and then adapted to form the basis of a comprehensive optimisation framework, applicable to different types of flowline geometries and fluid properties. The final form of the framework is presented as a black box interface, facilitating communication between OpenFOAM®, SALOME and DAKOTA through the use of a text-based user interface to allow for streamlined user input and ensure that the framework may serve as a useful investigative tool for a flowline operator.

Acknowledgements

First and foremost, I would like the opportunity to express my gratitude to my main supervisor, Dr. Hisham Mahmud for his support, guidance and encouragement throughout the course of my doctoral studies. I would also like to thank my co-supervisor, Dr. Andrew King for his invaluable advice regarding the workings of OpenFOAM®. I am also highly indebted to Prof. Marcus Lee for his support, encouragement, and constructive criticism that I believe has strengthened the quality of the final thesis immeasurably. I would also like to acknowledge the support given by Dr. Amandeep Sidhu in his role as chairperson of the thesis committee. Finally, I would like to thank my family, whose support and dedication has allowed me to achieve a lifetime goal. In particular, my wife; Zakiah, children; Latifa, Adam, Alia and Hana, parents; Liz and Max, brother; Justin and sister; Cass, thank you.

Contents

List of Figures

List of Tables

List of Symbols

Greek Symbols

α	Liquid phase fraction/colour function in Equation 4.3
β	Closure coefficient in Equation 4.4
δ_L	Liquid film height
ϵ	Absolute pipe roughness
γ	Bubble distortion coefficient
μ_{int}	Turbulent viscosity in Equation 4.3
ν	Kinematic viscosity
ω	Specific turbulent dissipation rate
ϕ	Interface area density
ψ	Multiplication factor in Equation 5.7
ρ	Density
ρ_h	Hydrate density
σ	Interfacial surface tension
σ_d	Closure coefficient in Equation 4.4
σ_γ	Auxiliary relation in Equation 4.4
σ_ω	Closure coefficient in Equation 4.4
Θ	Pipe inclination
Υ	Mesh resolution parameter

Roman Symbols

Δn	Grid size in the interface region/amount of hydrate growth
$\Delta n/\Delta t$	Hydrate growth rate
Δn_{trans}	Transient hydrate growth
Δt	timestep
ΔT_{sub}	Degree of subcooling
ΔV_{tot}	Volumetric hydrate growth rate
Δx	Longitudinal distance between mesh elements
A	Area

A_i	Interfacial area
B	Tunable parameter in Equation 4.6
C_l	Lift coefficient
C_{int}	Adjustable parameter in Equation 4.3
Co	Courant number
D	Pipe diameter
dP/dx	Pressure gradient
E_L	Liquid volume fraction
f	Friction factor
F_a	Area factor in Equation 5.12
F_d	Normal distance function based upon local parameters
g	Gravitational acceleration
h_L	Liquid height
h_L/D	Ratio of the liquid height to the pipe diameter
H_t	Thickness of the hydrate layer
k	Turbulent kinetic energy
K_1	Kinetic constant in Equation 5.2
K_2	Kinetic constant in Equation 5.2
L	Length
l	Liquid film length
L_w	Wetted length
n_h	Hydration number
P	Pressure/Auxiliary relation in Equation 4.4
p	pressure
r_h	Hydrate growth rate
r_i	Internal radius of pipe aperture with deposited hydrate layer
r_p	Internal pipe radius
Re	Reynolds number
T	System temperature
t	Time
t_{co}	Simulation cut-off time
t_{hb}	Hydrate half-block time
U	Velocity
u	Fitting parameter in Equation 5.2
U_{SG}	Specific gas velocity
U_{SL}	Specific liquid velocity

V	Volume
V_b	Bubble swarm rise velocity
V_{cell}	Mesh cell volume
W_g	Average molecular weight of the hydrate forming components
x_j	Space coordinate
y	Distance to the first cell center normal to the wall
y^+	non-dimensional wall distance

Subscripts

AM	Annular mist
bbl	Bubble
G	Gas
i	Initial
$inlet$	Inlet
int	Interface
L	Liquid
max	Maximum
min	Minimum
ss	Steady state
tot	Total
$trans$	Transient

Chapter 1

Introduction

1.1 Natural Gas Hydrates and Flow Assurance

A natural gas hydrate is an ice-like substance that forms when a small hydrocarbon molecule becomes trapped in a larger water cage to form a solid structure. This can happen when contact occurs between the hydrocarbon molecule and free water at high pressures and low temperatures. The pressure and temperature required for a hydrate to remain stable depends on the properties of the fluid, although a typical hydrate formation temperature would be in the order of 15 - 25°C for a pressure range of 50 - 120 bar. Hydrates were first discovered to occur in nature by Sir Humphry Davy in 1810; and naturally formed hydrates can still be found today in deep oceans and in permafrost. In recent years, hydrates have even been publicised as a future energy source due to the fact that there may be reserves of up to 40×10^{17} standard cubic feet, two orders of magnitude greater than conventional natural gas reserves [Sloan et al., 2009].

The experimental observations made by Hammerschmidt [1934] are recognised as being the first that provide a link between hydrate formation and blockages occurring in subsea natural gas flowlines. Hydrates have since been known to cause blockages in both oil and gas-dominated flowlines, albeit in different ways. Oil-dominated flow exhibits hydrate nucleation at the interface between the oil-phase and the water-phase. These hydrates grow via agglomeration and will eventually become large enough to create a blockage unless the interfacial shear rate is high enough to cause continual breakup of the formed hydrate particles. Conversely, gas-dominated flow exhibits wall deposition and growth in an annular fashion similar to stenosis that eventually leads to flowline blockage, where sloughing of the hydrate may occur depending on the amount of fluid shear [Rao et al., 2011]. Gas-dominated flows are typically con-

sidered to be more critical in terms of the hydrate blockage risk, as it has been shown that some oils exhibit an inherent capability to naturally inhibit the agglomeration of hydrate particles, allowing for a reduced risk of hydrate blockages [Høiland et al., 2011, Sjöblom et al., 2010, Leporcher et al., 1998]. In either case, the hydrate blockage will increase the rate of internal corrosion, structurally weakening the flowline [Obanijesu et al., 2011]. These blockages will also "anneal" if left over an extended period of time, increasing the difficulty of removing them from the flowline.

The removal process of a hydrate blockage usually involves a combination of depressurisation and local heat addition, where both of these techniques raise safety concerns. For hydrate removal involving depressurisation, it is possible that the hydrate blockage can rapidly dissociate from the pipe wall and become a high speed projectile. Sloan et al. [2010] have reported a case where a depressurised hydrate blockage has caused structural failure of an onshore flowline due to a high velocity impact with a sharp bend, leading to a fatality and localised damage to the surrounding area. Additionally, any direct application of heat must be handled with extreme care due to the risk of an explosion. This is because it is known that 1 cubic feet of hydrate material can contain up to 180 standard cubic feet of natural gas, which will readily expand with the addition of heat [Sloan et al., 2009]. Due to the highly critical nature of hydrate formation in subsea flowlines, various techniques have been developed as a means either to avoid the formation of hydrates entirely, or at least manage the rate of hydrate agglomeration to ensure that no blockages occur within the time it takes for the fluids to traverse the length of the flowline. These techniques include;

1. Removal of any free water accumulation via dehydration or pigging.
2. Decreasing the pressure of the system below the hydrate formation pressure.
3. Increasing the temperature of the system beyond the hydrate formation temperature.
4. Injecting an inhibitor to shift the pressure and temperature conditions for hydrate stability, such that higher pressures and lower temperatures are required for the hydrate to remain stable.

Dehydration is used to extract the water from a natural gas flow and is typically carried out on an offshore platform prior to transport via a single phase export

pipeline. No such capability exists to dehydrate the gas in a flowline from the well stream to an offshore platform or onshore process facility. Likewise, the option of decreasing the pressure to avoid the hydrate stability point is only applicable when a hydrate has formed and caused a blockage in the flowline. This methodology is not suitable for hydrate prevention as the well stream pressure must remain high for an acceptable production flow rate [Goodman and Joshi, 2013].

Pigging of the flowline involves propelling a solid/semi-solid object (the pig) through the flowline and is used for the purpose of cleaning deposits such as wax, removing liquids and for flowline inspection. Pigging is a viable option to remove accumulated water and is already widely used on most oil and gas developments situated on land and in shallow waters. For deepwater developments; however, this method is fraught with challenges and technical difficulties that mainly arise due to accessibility issues if subsea launching of the pig is required and control of the pig after launch. Unless properly controlled, the pig can reach unacceptably high velocities due to a high differential pressure occurring in the riser segment, potentially causing damage to topside equipment [Subash et al., 2002]. Other technical difficulties that prevent pigging being used as a viable solution to all developments include; hilly terrain causing the pigs to get stuck, the economic impact of launching multiple pigs on a periodic basis, and the reliability and maintenance of subsea pig launchers for single flowline systems.

The thermal control of flowlines involves either minimising heat loss or adding heat to the system, or both. Insulation is a common approach to minimise heat loss and is typically used from the well head to the processing station. It is not practical for extended flowline lengths as costs become prohibitive, where it has been reported that these costs are in the order of $US1 Million per mile for a subsea flowline [Jassim et al., 2010]. Electrical heating can be applied directly to the flow and is currently in use on flowlines in the waters of the North Sea and West of Africa [Delebecque et al., 2009]. This is practical for short segments of pipe; however, a study conducted by Lenes et al. [2005] have shown that the voltage requirements for pipe sections longer than a few miles are unacceptably high, in the order of several kilovolts (kV). Another strategy that has been used by Murphy Sabah Oil Co. Ltd. in the Kikeh field located offshore of East Malaysia is to elevate the flowlines from above the seabed using what has been termed a Gravity Actuated Pipe [Lemoël et al., 2008]. This has enabled this particular flowline to be subjected to an environmental temperature in the order of 16°C, rather than 4°C at the sea floor. Although this strategy would appear to be

effective for the purposes of hydrate prevention, it is not a practical solution for all deepwater locations due to possible interference from shipping and fishing activities.

Lastly, the injection of chemical inhibitors is a technique used primarily to prevent the formation of hydrates by chemically altering the hydrate formation point such that higher pressures and lower temperatures are required for a hydrate to remain stable. If complete prevention of hydrate formation is not possible, other types of inhibitors can be used to delay the agglomeration of hydrate particles, allowing them to be transported safely downstream. The amount of chemical inhibition required to prevent hydrate formation is determined by a graphical representation of the known range of pressures and temperatures that allow a hydrate to form and remain stable. This representation is commonly known as a pressure-versus-temperature (PVT) curve and is commonly created using commercial phase equilibria software. The expected pressure and temperature conditions of the flowline, depending on the length from the reservoir are then overlaid on the PVT curve. The compiled representation is used by a flowline operator to determine if the flowline will be operating in the hydrate stability zone and will further show the estimated amount of chemical inhibition required to ensure no hydrates are formed. Figure 1.1 shows an example of a PVT curve for a flowline in the Gulf of Mexico presented by Sloan et al. [2010]. Here, it can be seen that a constant injection of approximately 25% methanol (MeOH) is required to keep the flowline completely hydrate free.

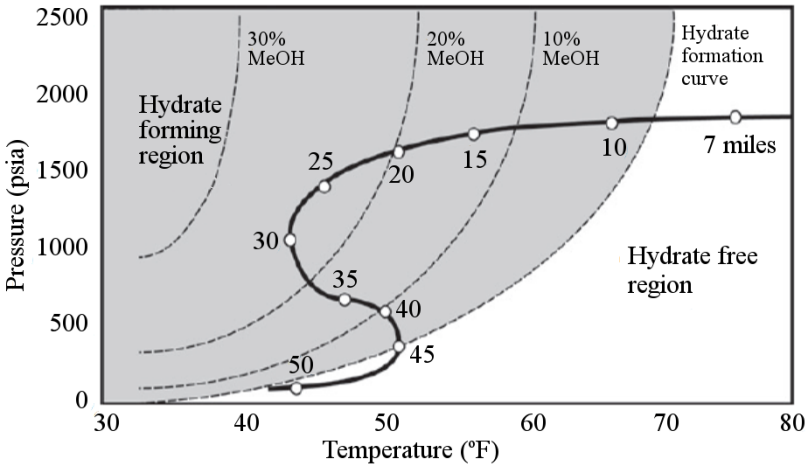


Figure 1.1: A PVT curve of a flowline showing the amount of methanol inhibition (MeOH) required to keep the flowline hydrate free, in this case approximately 25% (adapted from Sloan et al. [2010]).

The main disadvantage with any form of inhibitor is that it must be continually

applied throughout the entire production process. Creek et al. [2011] have provided a monetary cost figure of $US 350,000 per day for the Low-Dosage Hydrate Inhibitor (LDHI) and as much as $US 750,000 per day for methanol. These figures assume a production rate of 50,000 barrels per day at a 40% water cut with a 2% volume dosage for LDHI and a 35% weight dosage for methanol. In addition to this, injection of the inhibitor further reduces the quality of the produced fluid, where severe penalties are incurred if the inhibitor content exceeds the allowable specification set by the refinery [Alapati et al., 2008]. This is due to the additional processes required to remove the excess inhibitor content prior to export.

The complexity involved in preventing, or at the very least, managing the formation of hydrates in subsea flowlines means that hydrates remain one of the most prevalent flow assurance challenges that will be encountered in an offshore oil and gas system; where authors such as Sloan et al. [2009] have stated that hydrate blockages constitute the largest concern by an order of magnitude when compared to asphaltenes, scales and waxes. From a flow assurance perspective; however, one of the most critical tasks for a deep water field operator is ensuring adequate hydrate management during restart operations.

1.2 Restart Operations

A restart operation for a deep water well is particularly challenging in terms of limiting hydrate growth. This is because after an extended shut-down period the fluids contained in the flowline have lost heat to the surrounding sea water and as a result, the system lies well within the hydrate forming region. When these cooled fluids are subject to the high pressure emanating from the well stream, hydrates will form if enough free water is present in the flowline. Free water accumulates during the shut-down period and pools at any of the low-spots that occur in the flowline. The location and severity of the low-spots are dependent on the topography of the sea floor and cannot be avoided without costly preparation, such as the subsea rock installations considered for the Ormen Lange field in Norway, as described by Holden et al. [2005]. In order to ensure hydrate blockages do not occur during a restart operation, a complex start-up procedure is currently required. This procedure must be tailored to each well and adhered to explicitly to ensure no hydrate blockages occur. For example, the following procedure for a cold restart of a deep-water system originally from Bai and Bai [2005] is presented in Figure 1.2.

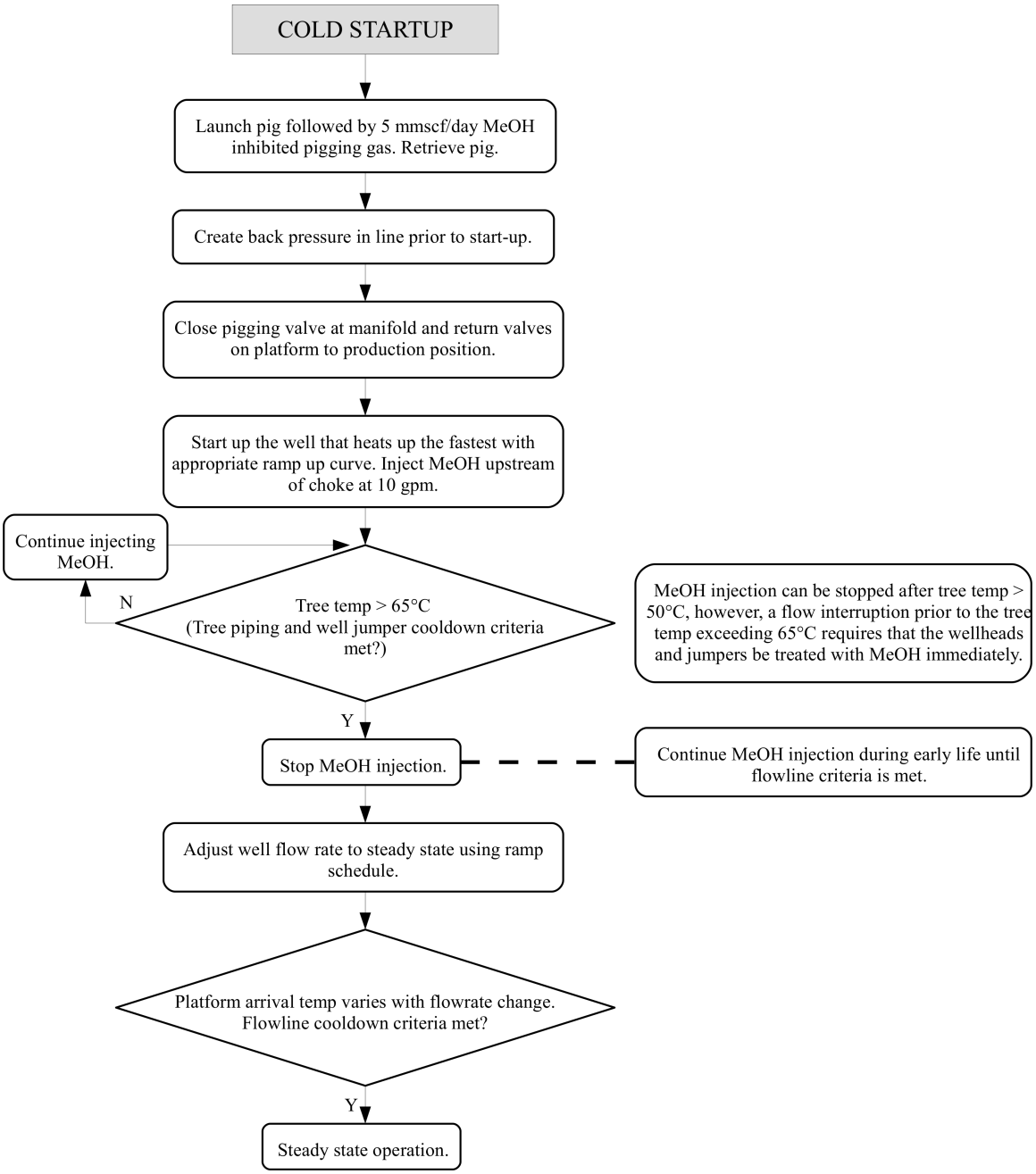


Figure 1.2: An example of the complex procedure required to perform a successful cold restart of a deepwater well, adapted from Bai and Bai [2005] with temperature units changed from Imperial to Metric.

To put the impact of the required start-up procedure into context, an example can be seen for the Ormen Lange flowline previously mentioned. Here, two 30" diameter flowlines are used to transport the flow fluid from the reservoir to an onshore processing station located 120 km away at Nyhamna. The initial de-watering is facilitated by launching a pig at the onshore processing station in a loop along the

length of the flowlines to push any fluid originally contained in the flowlines out. The pig is propelled using dehydrated gas taken from the export line after processing and typically more than one pig is used with a maximum speed of less than 0.8 m/s [Burns et al., 2008]. A simple calculation yields that a single pig traversing a round trip length of 240 km would require approximately 3.5 days to complete the journey. After pigging, an mono-ethylene glycol (MEG) type inhibitor is injected with dry gas at the onshore processing station and recirculated throughout the flowline loop to ensure the inhibitor is adequately distributed along both the flowline length. Burns et al. [2008] have provided a time between 32 - 89 hours for MEG inhibitor injected onshore at Nyhmana to complete the loop and return to the slugcatcher. It follows from this that a restart operation after an extended shut-down period would require an absolute minimum of 5 days before production can be brought back online to full production, assuming only one pig is used.

The complex procedure involved in restarting a well appears to contribute to a seemingly unavoidable supply disruption. However, recent experimental research suggests that it may be possible to use a hydrodynamic approach to allow for hydrate-free restart operations without the need for any inhibition techniques and de-watering pigging operations, negating the effect of a restart supply disruption entirely. This research is introduced in the following section.

1.3 Hydrodynamics - An Alternative Strategy for Restart Operations

Experiments conducted by Leporcher et al. [2002] have shown that it may be possible to reduce the risk of a hydrate blockage occurring during restart operations using a hydrodynamic strategy. In these experiments, two separate restart operations using a gas and a gas/oil mixture were simulated in a 0.0508 m diameter pressurised flow loop with a low-spot section filled with variable heights of free water, used to represent an accumulated water phase. Leporcher et al. [2002] observed that no hydrate blockages were formed during any of the high energy restart flows (1 m/s) for either of the two fluid combinations considered. In particular, it was observed that although hydrate nucleation rapidly took place in all high energy flows, a significant amount of the accumulated water was removed from the low-spot section and transported downstream, along with most of the formed hydrate material, contributing to a substantially reduced blockage risk when compared to the low energy flows considered

(0.001 m/s). For all of the low energy flows, hydrate blockages occurred as the restart gas passed at low velocity across the bulk of the static water accumulation at the top of the pipe, causing hydrate material to initial deposit on the upper pipe wall. A typical elapsed time for a subsequent blockage to occur was approximately 4 hours for both fluid combinations. The main conclusion provided by Leporcher et al. [2002] was that high energy restart gas flows that quickly wipe off the initial liquid water columns in the low-spot region lead to a reduced risk of hydrate blockage.

Further evidence of using the hydrodynamic strategy for hydrate free restarts has been presented by Volk et al. [2007], where it was suggested that an optimal restart rate may exist when using an uninhibited flow for low-spot flowline sections with an accumulated water zone, following the same hydrodynamic concept proposed by Leporcher et al. [2002]. This restart rate was assumed by Volk et al. [2007] to be high enough such that hydrate blockages did not occur yet would be limited to avoid well sanding. Volk et al. [2007] conducted a series of flow-loop experiments incorporating a low-spot section; however, an optimal restart rate could not be determined based upon the available data captured. Despite this, the experimental data gathered has provided a direct confirmation of the observations first presented by Leporcher et al. [2002].

Recently, Turner et al. [2015] have argued that there are in fact eight independent mechanisms that contribute to a successful restart operation without hydrate blockages, which include;

1. Nucleation Limitations: where a hydrate crystal has not yet assembled into a critical size for continuous growth.

2. Mass Transfer Limitations: where a small water surface area minimises the potential for hydrate growth to the interface.

3. Heat Transfer Limitations: where heat generated by hydrate formation cannot escape from the forming hydrate phase due to low subcooling and/or a high insulation value.

4. Limiting Reagent-Water: where hydrate growth is limited by small amounts of water present in a system.

5. Limiting Reagent-Guests: where hydrate growth is limited by few hydrate guest components.

6. Stable Dispersion: where hydrate is enabled to flow by suspension in the mobile liquid phase.

7. Partial Thermodynamic Inhibition: where hydrate thermodynamic inhibitors are concentrated and become more effective as a result of water consumption into the growing hydrate phase.

8. Physical Interaction Limitations: where the hydrate has not had the time to assemble into a plugging network of particles or accumulate as a restriction prior to warm-up.

Turner et al. [2015] have described these mechanisms as mutually synergistic when combined, increasing the efficacy when employed together. The experimental work of Leporcher et al. [2002] and Volk et al. [2007] appear to be directly related to items 2, 4 and 8 from the list of mechanisms proposed by Turner et al. [2015]. For example, during a restart operation, the initial hydrate growth is dependent on the interfacial surface area between the hydrate forming components and accumulated water (Item 2). If the flow rate is high enough, an amount of water will be swept out of any low-spots and redistributed downstream, limiting the total free water available for further hydrate growth (Item 4). Using this strategy, a hydrate will most certainly form; however, a blockage will not occur for a certain period of time such that it is possible that a transient production system can remove any hydrate in the system without hindrance to flow, as stated by Turner et al. [2015] (Item 8).

When put into context, the results first presented by Leporcher et al. [2002] are of paramount importance as they suggest that a hydrodynamic strategy may be employed to significantly reduce the risk of hydrate blockages occurring during restart operations, particularity if there is a means to quantify the amount of hydrate risk in a meaningful manner. It is most likely that this quantification of risk will be in the form of an estimated time for a blockage to occur depending on the specific pipe geometry and fluid properties.

1.4 Motivation for This Research

There has currently been no attempt made to identify the optimum restart flow rate based upon an operator-determined level of hydrate blockage risk during a restart operation, or to package the methodology used to determine the optimal flowrate in the form of a framework that is applicable to extended flowline lengths of any geometry and with any fluid properties. A safe operational window may be realisable for a flowline operator if the hydrate blockage risk could be quantified and associated with the restart rate. If this were possible, a hydrate management strategy using hydrodynamics could potentially lead to a major reduction in restart time as the current procedures involved to avoid the formation of hydrates entirely would be made redundant. It is proposed here that the risk of hydrate blockage can be reduced using a hydrodynamic strategy by simulating the transient flow during a restart operation; where the goal is to remove enough free water from the flowline low-spots such that the hydrate blockage risk is minimised. For this purpose, the amount of hydrate risk may be quantified using a combination of the estimated time for a blockage to occur based upon the use of an applicable hydrate growth model coupled with the hydrodynamic characteristics of the flow. Accurate implementation of such a framework will have the following benefits;

1. Enable a reduction in the usage of chemical inhibitors and de-watering pigs during gas-dominated restart operations, leading to cost savings.

2. Reduce the time required for restart operations such that the production flow can be brought online immediately.

3. Provide a framework that may be used for any flowline configuration in any location, which will enable a flowline operator to identify a safe operational window for a reduced risk of hydrate blockages occurring during restart operations.

4. Constitute a substantial contribution to knowledge in the fields of flow assurance and applied optimisation.

1.5 Scope and Objectives

The scope of this research project is to explore the possibility that a simulation-based optimisation approach may be used as an investigative tool to reduce the risk of hydrate blockages in flowlines using the hydrodynamic strategy proposed by Leporcher et al. [2002], with the ultimate goal of producing a comprehensive framework applicable to all types of flowline geometries and fluid properties. In order to achieve this goal, the research objectives have been crafted as follows:

1. To investigate the elements required to construct a simulation-based optimisation framework and identify the applicable techniques that can be used to facilitate the optimisation study, accurately describe the multiphase flow and provide an estimation of the amount of hydrate growth during a transient restart operation in a flowline.

2. To develop and assess a suitable multiphase simulation model in order to provide confidence for its use in the optimisation framework. To further tailor the model for the purposes of describing the flow in long flowline sections.

3. To assess the most appropriate optimisation techniques for use in the optimisation framework by coupling an optimisation toolkit to the simulation model to optimise the single low-spot bend used in the experimental study by Leporcher et al. [2002].

4. To further refine the optimisation framework through the use of a real industrial application, by applying the framework to simulated restart operations of the Ormen Lange natural gas field.

5. To present the most appropriate techniques garnered from Objectives 2, 3 and 4 in the final form of the optimisation framework, applicable to all types of pipe geometries and fluid properties, and including an appropriate user interface.

1.6 Organisation of This Thesis

The remainder of this report is structured as follows; Chapter 2 provides a review of the literature regarding the construction of an optimisation framework, Chapter 3 presents results from the validation of a Computational Fluid Dynamics (CFD) model used to describe the flow, Chapter 4 presents a modified CFD modelling strategy for long flowline sections, Chapter 5 presents the results from the optimisation of a low-spot flowline bend, emulating the gas-dominated restart from the experimental work of Leporcher et al. [2002], Chapter 6 presents the results from an applied industrial example, Chapter 7 presents the finalised version of the optimisation framework and Chapter 8 provides a summary of the work completed in this study and further recommendations for future work that will aid in enhancing the effectiveness of the proposed optimisation framework.

Chapter 2

Building an Optimisation Framework

In this chapter, the manner at which a comprehensive optimisation framework may be constructed is formed based upon a review of the literature. For this purpose, the elements required to build an optimisation framework are first introduced, followed by an investigation of applicable multiphase modelling and validation strategies, available optimisation toolkits and the manner at which the hydrate growth may be estimated. The most appropriate methodology is then identified in order to increase the chances that the constructed optimisation framework will be successful for the purpose of reducing the hydrate blockage risk during restart operations.

2.1 Elements of an Optimisation Framework

For engineering applications, optimisation is defined as the process of finding the best solution of a particular design while satisfying a certain set of requirements. The best solution occurs when the optimum value of either one or more objective functions is identified and the requirements are known as constraints. An optimisation problem for engineering applications is typically written in the form:

$$\begin{aligned} &\text{Minimise/Maximise:} && f(X) \\ &\text{Subject to:} && X \in \Omega \end{aligned}$$

Here, $X = (x_1, x_2, ..., x_n)$ is the vector of design variables, Ω represents the region of feasible solutions throughout the design space and $f(X)$ is an objective function providing numerical assessment for vectors of variables representing alternative solutions [Xiong et al., 2015]. The optimisation methodology applied to solve a specific

problem may be further packaged in the form of a framework as a means to quickly and efficiently accommodate different design configurations.

An optimisation framework constructed for engineering applications will typically involve the simulation of physical systems. The framework will thus consist of an optimisation tool, a simulation tool and a means to provide effective communication between the two. The optimisation tool contains the algorithm used to facilitate the optimisation study and is commonly provided in the form of a toolkit that contains a library of different types of algorithms and methodologies from which the user may choose from. The interface between the optimisation toolkit and simulation tool is considered to be a *black box* interface if no derivatives are available from the simulation response, i.e., if the optimisation algorithm is performed solely on the objective function output garnered from the simulation. Here, the details of the physical processes behind how the simulation reached that particular response are not assessed. The entire framework may be further presented in the form of a graphical user interface (GUI) or text-based user interface (TUI) to increase the ease of use for a practitioner unfamiliar with the specific toolkits used. Figure 2.1 shows an example of a black box interface between an optimisation toolkit and simulation tool.

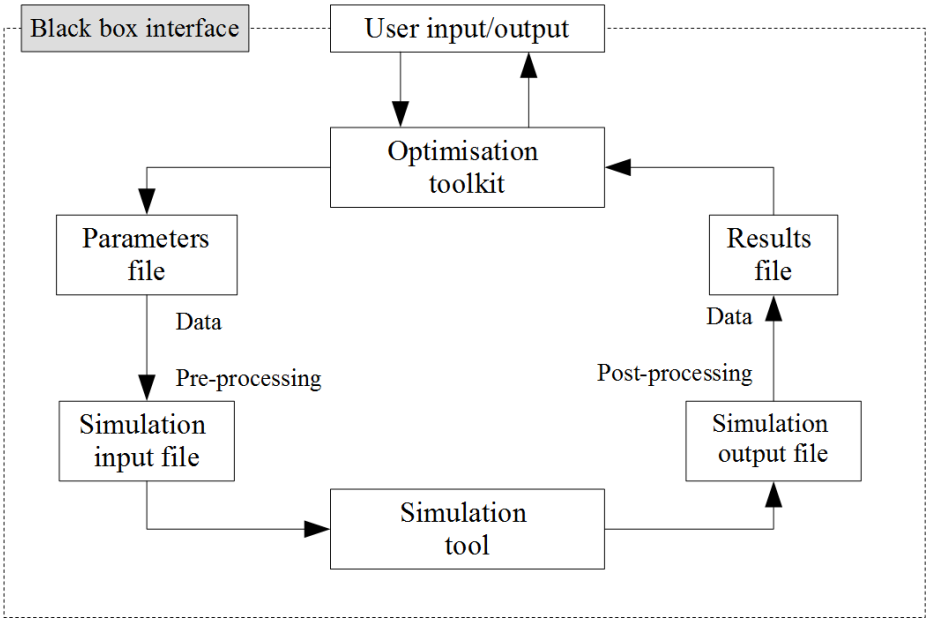


Figure 2.1: Black box interface between optimisation toolkit and simulation tool (adapted from Adams et al. [2014]).

Coupling between the optimisation toolkit and simulation tool is generally conducted by the reading and writing of short script files that can enable the transfer

of data and perform the required pre-processing and post-processing operations that will provide data in the form required by the optimisation toolkit. With reference to Figure 2.1, the parameters file, simulation input file, simulation output file and the results file would all be in the form of a short script file in a format recognised by both the optimisation toolkit and the simulation tool. Using this strategy, an optimisation framework for minimising the risk of hydrate blockages in flowlines during restart operations can be constructed via the coupling of the three following independent categories;

1. An applicable optimisation toolkit to provide a selection of algorithms and facilitate communication with a simulation tool.

2. A simulation tool to describe the transient nature of the flow during restart operations.

3. A user interface to provide an intuitive and automated scheme for easier user input.

A review of the literature encompassing all three categories is presented in the following sections with the goal of defining a suitable methodology for use in this study. Prior to the selection of an appropriate toolkit, it is necessary to review the appropriate stages of the optimisation process. Through this, the necessary requirements of the optimisation toolkit can be defined.

2.2 Process of Simulation-based Optimisation

Nguyen et al. [2014] have presented a comprehensive summary of a three phase simulation-based optimisation process for the design of buildings. Although this is in a different field, the general concepts presented resonate with any simulation based optimisation technique and are reproduced here in Table 2.1, along with a discussion of each phase in separate subsections.

Table 2.1: The phases involved in simulation-based optimisation studies (adapted from Nguyen et al. [2014]).

Phase	Major tasks
Pre-processing	Formulation of the optimisation problem - Computer simulation model - Setting objective functions and constraints - Selecting and setting independent design variables and constraints - Coupling the optimisation algorithm and simulation program - (Optional) Screening out of unimportant variables by using sensitivity analysis - (Optional) Creating a surrogate model to mimic the simulation without being as computationally expensive to do so
Running Optimisation	Monitoring convergence Controlling termination criteria Detecting errors or simulation failures
Post-processing	Interpreting optimisation results (Optional) Response verification and surrogate model verification (Optional) Perform sensitivity analysis on the results Presenting the results

2.2.1 Pre-processing Phase

The pre-processing phase is used to formulate the entire optimisation problem and requires knowledge of the nature of the simulation, the ranges for the design variables and a basic understanding of the interaction between the variables and the objective functions. The variables selected here may be either discreet, continuous or a mixture of both. The most important aspect is to determine a suitable optimisation algorithm that can effectively navigate the entire design space to return the correct optimum, especially for problems where the solution response is unknown. For this reason, a global optimisation algorithm must be used, as its the only way to successfully differentiate between local and global optima. In addition to this, many engineering problems contain multiple-objective functions that are often in conflict with each other and the need for a multi-objective algorithm is immediately apparent. Fortunately, many optimisation algorithms already exist and are classed as either deterministic or stochastic.

Deterministic algorithms are methods belonging to the classical branch of mathematical optimisation and rely heavily on linear algebra to calculate the gradient or Hessian of the response variables. In contrast, stochastic algorithms are methods that generate and use random variables. Through the work of authors such as Cavazzuti [2013], Rao and Savsani [2012] and Arora et al. [1995], it is apparent that deterministic algorithms are not suitable for multi-objective optimisation studies, despite the

advantage of being computationally faster than stochastic methods. This is because deterministic algorithms are intrinsically confined to evaluating only one objective function and also cannot distinguish between local optima and global optima, limiting their usefulness for engineering applications. Recent attempts have been made to combine the speed of the deterministic model with the robustness of the stochastic method in a newly defined hybrid model, as evidenced by the works of Capasso [2015], Collins and Carr [2014] and Friedlander and Schmidt [2012]. Although these efforts have existed for a while, they are still very much in their infancy and have not been implemented in any of the available optimisation toolkits surveyed, making the stochastic algorithm the most appropriate choice for this study.

A particular class of stochastic algorithms, called meta-heuristics, are reportedly the only practical solution to obtain global optimal solutions for non-linear, non-differentiable and non-continuous problems that often exist in the real world [Yang, 2010, Lee and El-Sharkawi, 2008], where other common stochastic algorithms include direct search, stochastic search and stochastic programming. Further evidence advocating the use of meta-heuristic algorithms can be found in the well-known overview and comparison of meta-heuristic approaches by Blum and Roli [2003]. According to Chandrasekar and Ramana [2012], the most popular meta-heuristic algorithms include the Genetic Algorithm (GA) [Goldberg and Holland, 1988], Differential Evolution (DE) [Storn and Price, 1997], Particle Swarm (PS) [Kenndy and Eberhart, 1995] and Simulated Annealing (SA) [Brooks and Morgan, 1995]. The GA, DE and PS are population based techniques inspired by nature and are used to explore the design space via the evolution or replacement of one population of solutions. In contrast, SA is a trajectory based technique which is used to explore the design space via the transition from one feasible solution to another.

In a review of all meta-heuristic techniques for use with engineering optimisation, Xiong et al. [2015] have concluded that population based meta-heuristic methods such as the GA and PS are superior in terms of accuracy compared to trajectory based methods such as SA when considering multi-objective global optimisation involving high numbers of parameters. According to Xiong et al. [2015], trajectory based approaches tended to get stuck in local optima and would not accurately determine the global optimum solution. The use of GAs to solve complex engineering optimisation problems is well established and many examples can be found for applications such as the aerodynamic optimisation of livestock trailers [Gilkeson et al., 2008], jet pumps [Eves et al., 2012], inflatable airfoils [Hauser et al., 2011], wind turbines [Díaz-Casás

et al., 2013], and many others. A review of the literature would suggest that use of the PS technique is less established than the GA, although a comparison of both techniques on the design of space system telescope array configurations by Hassan et al. [2005] has shown comparable results, with the PS technique exhibiting significantly better computational efficiency. Similar results have been presented in a comparison of meta-heuristic techniques by Antosiewicz et al. [2013] to solve the traditional travelling salesman problem (TSP).

Regardless of the exact meta-heuristic optimisation technique employed, it is apparent that comparisons of the performance and efficiency of specific optimisation algorithms are of little importance when used for engineering applications. This is because in the vast majority of optimisation studies involving the use of computational simulations, such as in all of the examples using GAs cited in the previous paragraph, the optimisation algorithm was performed on a surrogate model (meta-model). A surrogate model is used as a means to mimic the response of a high fidelity simulation without being as computationally expensive to do so. Here, the surrogate model is a representation of the actual system response constructed from a certain number of build points located in the design space, where the number and location of the build points is typically determined by a Design and Analysis of Computer Experiments (DACE) method, or a traditional Design of Experiments (DoE) method.

Using surrogate modelling, the optimisation algorithm is now performed on a model of the system and the speed of convergence for each algorithm matters little as the computational expense involved is now entirely weighted in the running of the simulation model at the selected surrogate build points. The work of Johansen [2011] and Ong et al. [2003] provides ample evidence to suggest that surrogate modelling is essential for the simulation based optimisation studies proposed for this study. In particular, for the optimisation of an inflatable airfoil, Johansen [2011] reported that over 533 hours of computational time were saved using a total of 192 processors when using a genetic algorithm based surrogate modelling strategy compared to a standalone genetic algorithm. In addition to the reduction in computational time, Johansen [2011] also observed superior performance and less noise with the use of a surrogate model

The most common types of surrogate models include; polynomial regression models, the Kriging method [Matheron, 1963], Radial Basis Functions (RBF) [Hardy, 1971, Dyn et al., 1986], Moving Least Squares (MLS) [Levin, 1998], Multi Adaptive Regression Splines (MARS) [Friedman, 1991] and Artificial Neural Networks (ANN)

[Smith, 1993, Cheng and Titterington, 1994]. A wealth of literature can be found that discusses various merits of each method and many authors have compared various models to assess the performance and quality of each. In these studies, it has generally been shown that the Kriging method slightly outperforms other techniques [Forrester and Keane, 2009, Glaz et al., 2008, Won and Ray, 2005, Simpson et al., 2004, 2001]. This has not always been the case, however, as Jin et al. [2001] performed a comparative study of the Kriging, MARS, RBF and polynomial surrogate modelling techniques using a suite of 14 mathematical and engineering test problems and concluded that the RBF model was the most dependable model overall in terms of accuracy and robustness, and especially excelled in problems involving small data sets. The complete set of metrics used by Jin et al. [2001] to assess the surrogate quality included; accuracy, efficiency, robustness, transparency and simplicity.

Similar to the optimisation algorithm, construction of the surrogate model typically occurs after all simulations have been run at the selected build points, thus the time taken to implement the actual surrogate model is also likely to be insignificant when compared to the time required for the simulations. In this context, the best approach for a practitioner would appear to be simply to compare all of the available algorithms and surrogate modelling methods contained within a toolkit and select the most effective.

Following the review of the pre-processing optimisation methods, it is evident that the optimisation toolkit must not have a restriction on the number of objective functions, parameters and constraints that can be applied. Furthermore, in order to increase the robustness of the framework constructed, it would seem highly advantageous that the toolkit have a number of multi-objective global optimisation algorithms in order to choose the most appropriate and finally, the capability of the toolkit to provide some form of surrogate modelling capability is essential. Selection of both the surrogate model and optimisation algorithm is to be based upon an analysis of all the available model types provided with the toolkit.

2.2.2 Optimisation Phase

Running the optimisation study includes monitoring the convergence of the solution and detecting any errors. Convergence is defined here as the point at which the final solution is reached. The final solution depends upon the termination criteria inputs and does not necessarily mean that the global optimum has been correctly identified. The termination criteria may be any of the following; maximum number

of generations or iterations, maximum optimisation time, maximum number of equal objective function value evaluations, population convergence or gene convergence. The optimisation process ends if one of these criteria are satisfied. These categories are well established because they are dependant upon the type of optimisation algorithm and are implemented in all optimisation toolkits.

Errors may arise during the optimisation process due to insolvable solution spaces, infeasible variables or output reading errors [Nguyen et al., 2014]. Many optimisation toolkits also have an in-built error checking and mitigation methodology. An example of this is the Design Analysis Kit for Optimisation and Terrestrial Applications (DAKOTA) software [Adams et al., 2014], which includes a simulation failure capturing approach that allows the user to abort, retry, recover or continue the simulations upon the detection of an error emanating from the simulation. It would appear that error capturing capability is not strictly essential provided the simulation response is robust and well tested, although would be highly advantageous as a time-saving measure when testing any new simulation tool.

Finally, for any high-fidelity simulations involving large numbers of mesh elements, the ability for the optimisation toolkit to provide a parallel processing capability is essential and is primarily used to reduce the simulation runtime during the DACE study by taking advantage of all of the processing power available. Advocates for the use of parallel computing in optimisation studies can even be found as far back as Schnabel [1984]. Fortunately, the ideas from previous decades have come to fruition and the use of parallel computing has now become the norm, thus it is unlikely to be a problem in this study.

2.2.3 Post-processing Phase

The post-processing phase is where the results are interpreted in the form of charts, diagrams or tables in order to derive useful data to determine the optimal solution. The final solution should be verified to confirm the reliability and robustness of the solution. Nguyen et al. [2014] have found there is no standard rule to perform verification in the literature, although a number of approaches have been identified as shown in Table 2.2.

Table 2.2: Optimisation verification methods as in Nguyen et al. [2014].

Author	Verification Method
Hasan et al. [2008]	Brute-force search
Eisenhower et al. [2012]	Comparison of a surrogate model to the original model
Tuhus-Dubrow and Krarti [2010]	Sensitivity analysis

The brute-force approach used by Hasan et al. [2008] is an exhaustive search method where the intention is to evaluate all possible areas of the design space. For any form of GA algorithm similar to that used by Hasan et al. [2008], this can be accomplished by increasing the population size and number of generations. The downside of this approach is immediately apparent due to the excessive number of simulations required in order to check the entire design space, rendering this technique not suitable for the current study.

Alternatively, Eisenhower et al. [2012] compared the results obtained from a constructed surrogate model to the "original" simulation model, in this case EnergyPlus. In this manner, the surrogate model is checked by a number of new simulation runs at the defined optimal point and errors can be expressed in an applicable metric such as those listed in Table 2.3, where n is the number of build points, $o(x_i)$ is the value obtained from the simulation, $p(x_i)$ is the value predicted by the surrogate model and $\bar{o}$ is the mean of the true response variable.

Table 2.3: Common error metrics used to evaluate the accuracy of surrogate models.

Error metric	Formula
Sum Squared	$\sum_{i=1}^{n}[o(x_i) - p(x_i)]^2$
Mean Squared	$\frac{1}{n}\sum_{i=1}^{n}[o(x_i) - p(x_i)]^2$
Root Mean Squared	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}[o(x_i) - p(x_i)]^2}$
Sum Absolute	$\sum_{i=1}^{n}\|o(x_i) - p(x_i)\|$
Mean Absolute	$\frac{1}{n}\sum_{i=1}^{n}\|o(x_i) - p(x_i)\|$
Maximum Absolute	$\max\|o(x_i) - p(x_i)\|$
R^2	$\frac{\sum_{i=1}^{n}[p_i - \bar{o}]^2}{\sum_{i=1}^{n}[o_i - \bar{o}]^2}$

All of the error metrics listed in Table 2.3 are likely to generate similar trends provided they are used consistently. Generally, lower errors are better but care must be used for interpolatory models such as Kriging, where errors will almost always be zero [Adams et al., 2014]. The R^2 method is an exception, as the value falls on an interval in the range [0,1], with 1 representing a data match. The Root Mean Squared and both of the absolute error metrics are potentially advantageous as they are on the same scale as the constructed surrogate models and build point data. The advantage of this method is that any of the additional simulations used to assess the surrogate robustness can be added as build points and used to construct an even more accurate surrogate model. In practise, this technique has been used by authors such as Khatir et al. [2013], Eves et al. [2012] and Gilkeson et al. [2008], where the additional simulation points used have been classified as validation points.

A variation of this technique is called cross-validation and is used to predict how well a surrogate model will respond to unseen data by iteratively removing subsets of the original build point data. According to Adams et al. [2014], the build points are randomly divided into k partitions, where k separate surrogate models are constructed, each excluding the k^{th} partition of the data. Each model is then evaluated at the points that were excluded in its generation using the metrics introduced earlier. The advantage of this technique is that no additional build points are required. If the number of partitions is equal to the number of build points, the resulting method is known as leave-one-out cross validation or prediction error sum of squares (PRESS).

In the last method considered by Nguyen et al. [2014], Tuhus-Dubrow and Krarti [2010] performed a sensitivity analysis of the solution by varying the optimisation inputs to check how much the output would vary. Similar to the brute-force approach, a sensitivity analysis used in such a manner will also require a large amount additional simulations in order to see a proper trend emerge between the input and output response and is thus unlikely to have any foreseeable advantages for the expensive computational simulations used in this study.

A review of the optimisation verification techniques has shown that both the brute-force and sensitivity analysis methods would involve too many additional simulations, especially considering the simulations are likely to be computationally expensive. As a result, the comparative method used by Eisenhower et al. [2012] is the most plausible for use in this study, whereby it is suggested that a cross-validation technique be used to evaluate the quality of the surrogate models without the need for any additional build points.

2.3 Review of Available Optimisation Toolkits

The information obtained from the review of the process involved for simulation-based optimisation can now be used in order to survey potential optimisation toolkits suitable for this study. The article presented by Nguyen et al. [2014] consists of a list of current optimisation toolkits applicable for use in the building industry, and fortunately, a number of the toolkits provided are general in nature and can be applied to any form of engineering optimisation problems. The list presented by Nguyen et al. [2014] has been reproduced here to show these optimisation toolkits, together with three additional programs presented in a review of surrogate modelling techniques for engineering based optimisation by Wang and Shan [2007]. This combined list is presented in Table 2.4, where the categories to showcase the capabilities of each toolkit are in the same form as Nguyen et al. [2014].

Table 2.4: A summary of the optimisation toolkits reviewed (adapted from Nguyen et al. [2014] with additional input from Wang and Shan [2007]).

Software	Open source?	Multi-objective algorithm	Parallel processing?	Handling discreet + continuous variables	Parametric study?	Sensitivity analysis	Generic for simulation programs?	Multiple algorithms?	User interface?	Objective function flexibility	Parameter flexibility	Surrogate model?	Operating system	Reference
Altair HyperStudy®	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	Win, Linux, Mac	[1]
DAKOTA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Win, Linux, Mac	[2]
iSIGHT	✗	✓	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	Win, Linux	[3]
LS-OPT	✗	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	Win, Linux, Mac	[4]
MATLAB toolkit	✗	✓	✓	✓	✓	✓	✓	✓	✓/✗	✓	✓	✓	Win, Linux, Mac	[5]
modeFrontier®	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓	Win, Linux Mac	[6]
ModelCenter®	✗	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✗	Win	[7]
Optimus®	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Win, Linux	[8]
VisualDOC	✗	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	Win, Linux, Mac	[9]

[1] Ratzel and Kocer [2007]
[2] Adams et al. [2014]
[3] Van der Velden and Koch [2010]
[4] Stander et al. [2007]
[5] MATLAB [2010]
[6] Nardin et al. [2009]
[7] Malone and Papay [1999]
[8] OPTIMUS [2004]
[9] Balabanov et al. [2015]

It is immediately apparent that many effective tools are available to implement optimisation frameworks and a wealth of literature can be found that showcase the merits of each (for example; HyperStudy® - Eves et al. [2012], ModelCenter® - Gur et al. [2011], DAKOTA - Axdahl [2015], and many more). All of the toolkits listed in Table 2.4 provide the capability to integrate simulation tools from third-party vendors, create and automate simulation workflows, set simulation parameters and run multidisciplinary simulation processes. Of these, DAKOTA is the only comprehensive non-commercial optimisation toolkit readily available in the public domain and is shown to be the only software that fulfils each category presented in Table 2.4, and as such, it is immediately intended for use in this study.

DAKOTA is a public domain licensed optimisation toolkit developed by Sandia National Laboratories that includes methods for optimisation, non-linear uncertainty qualification, design of experiments, statistical sampling, parametric studies, sensitivity analysis and surrogate modelling [Giunta, 2002]. DAKOTA employs a generic interface capability based on UNIX commands that permits linking to a variety of commercial and custom simulation codes, making it ideal for engineering applications that involve CFD or Finite Element Analysis (FEA). DAKOTA has been developed using the C++ programming language and enables the usage of a wide variety of internal and external optimisation algorithms by executing them via a system call interface that utilises standardised file types for the input parameters and the solution output.

It is worth mentioning that recent authors such as Perez et al. [2012] and Martins and Marriage [2007] have advocated the use of object-oriented optimisation frameworks built using higher level programming languages such as Python and Matlab as a means to interface with a larger variety of simulation software. This is because an optimisation software compiled using the C/C++ programming language is typically limited to interfacing with simulation software written in the same language. A number of frameworks have been created for this purpose such as YALMIP [Löfberg, 2004] and TOMLAB [Holmström et al., 2010] which are open-source but are interfaced via MATLAB. Additionally, OpenOpt [Kroshko, 2007] and pyOpt [Perez et al., 2012] are two alternative Python based frameworks that also offer a number of different optimisation algorithms via a system call interface. Despite the extraordinary efforts to bring these projects to light, none appear to have the usability, support level and capability of DAKOTA, which has thus been recommended for use in this project, provided a compatible software can be found to model the multiphase flow. As such,

a review of the appropriate methods to model the multiphase flow is presented in the following section.

2.4 Modelling of Transient Multiphase Flow

The transportation of untreated hydrocarbons in subsea flowlines is characterised by a multiphase flow mixture typically comprising of liquid crude, natural gas, water and sand. The multiphase flow will exhibit different characteristics depending on the pipe shape, size, orientation, material properties, flow properties and fluid component phase fractions. Typically, exact measurements of the flow characteristics are only available at the wellhead choke and at the topside location, thus the true characteristics of the flow along the length of the flowline are unknown. This level of uncertainty is the main factor that contributes to the state of over-conservatism regarding inhibition techniques seen in the industry. Transient multiphase flow models were created out of the need that the characteristics of the flow be accurately modelled throughout the entire system in order to provide more comprehensive flow assurance solutions.

A transient multiphase flow model for long flowline sections generally comprises of a 1-D drift-flux model that reduces a fully 3-D flow to 1-D using averaged equations and a coarse grid (pipe segments) of several tens to hundreds of meters of length [Gharaibah et al., 2015]. These 1-D models contain industry-standard algebraic closure relations for slip and friction depending upon the flow pattern. The closure relations are based upon experimental data and allow for the "tuning" of the models to attain a high level of accuracy. The further usage of a 1-D coarse mesh enables a highly efficient and fast simulation time, whereby transient simulations can be run for entire flowline lengths, making the use of this tool extremely attractive for flowline operators. Table 2.5 lists the most common commercial software used in the oil and gas industry together with multiphase simulation capability. Here, "3P" represents three-phase and "2P" represents two-phase respectively.

Table 2.5: Summary of transient multiphase flow simulation tools for long flowlines.

Software	Developer	License	Capability
FlowManager Dynamic™	FMC Technologies	Commercial	3P
LedaFlow®	Kongsberg Oil and Gas Tech.	Commercial	3P
OLGA	SPT Group	Commercial	3P
TACITE	Schneider Electric	Commercial	2P

Of the software listed, OLGA commands a dominating market share of approximately 90% [Bendiksen, 2015]. The dominance of OLGA is in part due to the wealth of experimental data used to formulate the closure laws. In particular, the performance of the OLGA code has benefited from the OLGA Verification and Improvement Project (OVIP), an industrial project with the participation of eleven oil and gas companies which lead directly to the improvement of closure laws, made possible by the large collection of field data made accessible by the OVIP members [Irfansyah et al., 2005]. Despite the market dominance of OLGA, comparisons with other codes can show similar results. For instance, a comparison of the OLGA and LedaFlow® codes was performed by Belt et al. [2011] with reference to data from the Boussens flow loop located in France. Here, it was shown that both codes performed equally well in terms of predicting the correct pressure gradient and liquid holdup, especially in cases where the flow was nearly horizontal. Belt et al. [2011] did report however that two weak points currently exist for both codes; an over prediction of the liquid holdup in stratified flow and a relative pressure gradient error of up to 70% in vertical slug flow.

Despite the obvious advantage of using an industry standard tool, all of the multiphase flow software presented in Table 2.5 are only available commercially. A single OLGA license for transient flow costs in excess of $US 250k, which limits the applicability of its use in academic studies. As an alternative to an expensive commercial package, the multiphase flow may be modelled using CFD software instead [Lo and Fiala, 2010]. CFD models have already found widespread use in the oil and gas industry although this has been restricted to detailed modelling of specific regions of interest due to computational constraints. An example of this is the LedaFlow® Quasi-3D (Q3D) package that is used to resolve pipe segments deemed critical by the user by using slice-averaged equations of mass, momentum and energy to resolve the flow [Ellul, 2010]. It is only recently that the advent of increased computational power and resources has enabled CFD to be applied to more extensive regions of interest, such as the use of CFD models to accurately predict a phenomena known as *terrain-induced* slugging, as shown in the studies by Chica et al. [2012], Jia [2012] and Ragab et al. [2008].

Gharaibah et al. [2015] have provided an overview of current CFD tools and methodologies for the simulation of multiphase flow with particular interest on multiphase pipe flow, sand particle transport, flow induced vibration and the simulation of processes such as flow pumps, wet-compression, gas-lift and separation. As the

multiphase flow in long flowlines is predominately either stratified or slug, Gharaibah et al. [2015] have reported that the flow may be modelled using a Eulerian-Eulerian approach, which can actually be applied to all types of multiphase flow, provided proper phase interaction models such as the Volume of Fluid (VOF) model by Hirt and Nichols [1981] are defined. The VOF method has been used by many authors to resolve multiphase flows, such as evidenced by the works of Stenmark [2013], Mazumder [2012] and Lo and Fiala [2010]. Comparisons between CFD and commercial multiphase solvers can also be found. José et al. [2014], using the VOF model implemented in Fluent, have produced comparable results to LedaFlow® Q3D for two-phase air/water and CO_2/oil flows in a closed system flow wheel. Here, both models exhibited close agreement to laboratory data; however, LedaFlow® Q3D exhibited a faster computational time due to less elements used. Despite this, the comparison given by José et al. [2014] gives credence that CFD is able to accurately model the multiphase flow. The higher number of elements used in 3-D CFD analyses seems to be unavoidable, although this effect could possibly be reduced by selecting appropriate wall functions or eliminating symmetric geometry to reduce the model size.

It is evident that the vast majority of CFD studies regarding multiphase flow are confined to straight pipes. These studies are typically limited to 2-D analysis [Desamala et al., 2013], or are limited in scope to a single flow regime [Rzehak and Kriebitzsch, 2015, de Sampaio et al., 2008]. Examples for 3-D analysis of multiple flow regimes have used different computational solvers depending on the specific flow regime under investigation, such as Vasquez et al. [2012]. This is due to the complexity for a single solver to accurately model the behaviour of changing flow regimes. Bestion [2014] has summarised the challenges when attempting to use a system wide CFD solver for multiple flow regimes when considering thermal-hydraulics of nuclear reactors. Here, it was reported that system wide code was typically limited to generating a macroscopic description of the flow, whereby calibration of the code is usually required if small scale effects need to be resolved accurately. In the context of this study, it is of the utmost importance to characterise the limitations of the solver selected for the framework through the use of an appropriate validation study to yield insight into potential limitations on the specific flow regimes that can be modelled.

Interestingly, most of the CFD studies presented in the literature regarding multiphase flow in flowlines are limited to two-phase flow and any instances three-phase analysis of multiphase pipe flow could not be found. This is likely to become a

problem as Irfansyah et al. [2005] have reported that modelling multiphase system as two-phase is only applicable as long as the liquid hydrocarbon and water phases are well mixed. In such a case, it can be assumed that limited slippage exists between these two phases and the flow can be considered to be a flow of gas and liquid. Problems will arise in the two-phase modelling of a three-phase gas-dominated flow (gas-condensate-water) depending on the amount of condensate present, as slippage between the condensate and water phase will occur which will not be captured by the model. Despite this, recent CFD studies of gas-condensate flowlines are still using a two-phase model; such as the modelling of a multiphase flow splitting into two separate vertical risers by Worthen and Henkes [2015] for the Shell floating LNG vessel. The results presented by Worthen and Henkes [2015] showed that slug flow in the vertical riser was not adequately reproduced, where it was concluded that this was due to the lack of slippage between the gas and liquid phases, which is a direct confirmation of the problem highlighted by Irfansyah et al. [2005].

The two-phase limitation could possibly be due the fact that accurate modelling of three-phase flows is exceedingly challenging and attempts to resolve this are still in there infancy. Another possibility could be due to the lack of any applicable validation methods to apply to the model in order to ensure sufficient accuracy. As a means of investigating the second possibility and to determine a suitable validation technique for use in this study, a review of state-of-the-art multiphase validation strategies is presented in the next section.

2.5 Validation Techniques for Multiphase Flow

Steady state hydrodynamic modelling is a technique that is widely used to help explain the behaviour and characteristics of multiphase flows [Shippen and Bailey, 2012]. This technique has been used in the past to numerically verify the accuracy of CFD models; as seen in the work of Vasquez et al. [2012], de Sampaio et al. [2008] and Munkejord et al. [2005]. These models are distinct from the transient models described in the previous section and are limited to predicting the flow regime in straight pipes, from which the values of the liquid holdup and pressure gradient can be determined depending upon input variables such as pipe diameter, pipe inclination, pipe roughness and the fluid properties. The main advantage of the steady state hydrodynamic models is that the underlying mechanics are readily accessible, meaning that the governing equations and closure laws can easily be implemented in a third party tool such as Scilab, MATLAB or EXCEL in order to solve the flow equations.

A graphical history of steady state multiphase flow models has recently been presented by Shippen and Bailey [2012] and includes the current state-of-the-art models that are still applicable at the time of this study. This representation is reproduced here in Figure 2.2.

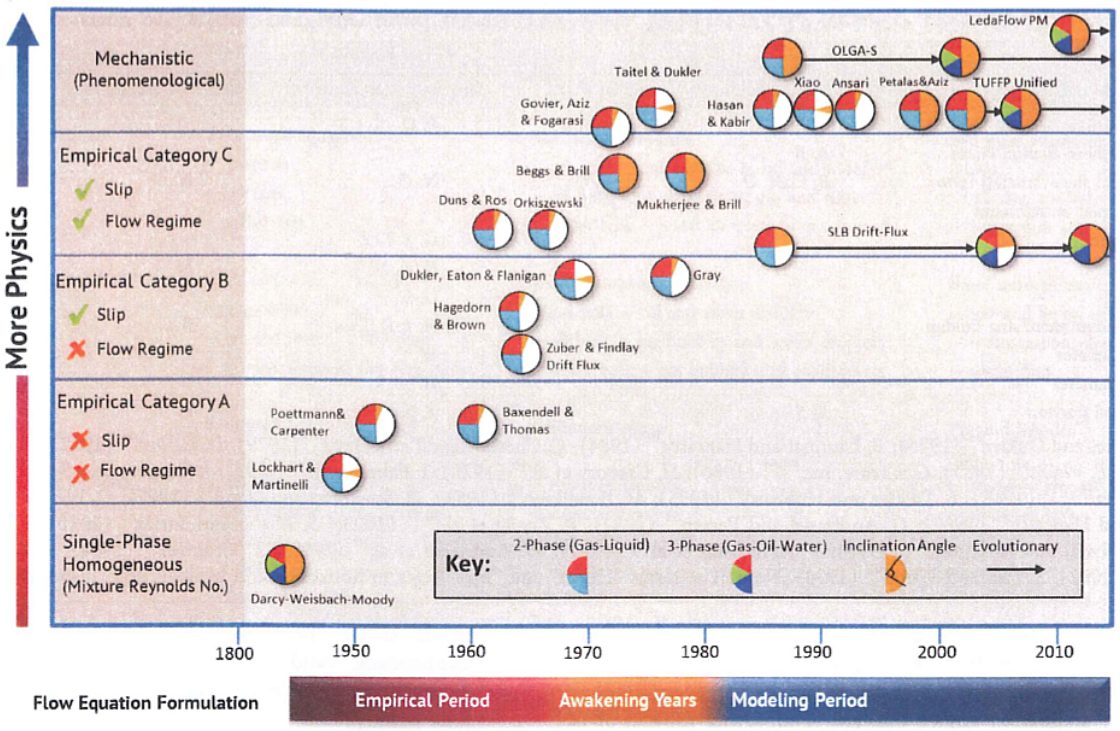


Figure 2.2: History of steady state multiphase flow models [Shippen and Bailey, 2012].

The incremental progress made throughout the years can be clearly seen in Figure 2.2 and it would seem that models that describe three-phase flows are on the cusp of the next phase of development. Although it would appear that some good agreement with experimental data has been presented for three-phase models such as by Lawrence et al. [2013], these three-phase models are currently limited to describing the behaviour of stratified and slug flows and are far from a comprehensive solution that exits for the description of two-phase flow for all flow patterns. Additionally, similar to the transient flow models described in the previous section, current three-phase models are only commercially available, such as LedaFlow® PM, OLGA-S and the SLB-Drift Flux model. The Tulsa University Fluid Flow Project (TUFFP) model [Sarica and Zhang, 2006] appears to be an alternative; however, recent project reports suggest this project is still very much under development and will also only be commercially available, based upon information provided in the 76^{th} and 81^{st} Semi-Annual

Board Meeting Reports available online.

Due to the lack of an accessible and comprehensive verification method for three-phase flow, it is proposed that the current CFD simulation used in this study is limited to a two-phase description of the flow. For this reason, the framework will only be applicable to gas-dominated flows with very low Condensate-Gas Ratios (CGR), where it is assumed that the gas and condensate may be modelled as a single phase. No information could be found regarding a minimal CGR for the successful modelling using two-phase flow. In light of this, although it is proposed that a CFD package is incorporated into the current optimisation framework constructed in this study, it would be prudent to ensure the framework is as modular as possible, whereby a more representative modelling tool may be swapped in place if the results provided by the CFD simulation prove to be inadequate. The ideal case being if the framework and methodology presented in this study are compatible with OLGA.

Despite this, an additional benefit arises as Leporcher et al. [2002] have concluded that a gas-dominated restart operation constitutes the highest risk for a hydrate blockage occurring in a flowline. In particular, Leporcher et al. [2002] observed that in the presence of an oil phase, most of the hydrates that form during a restart operation will agglomerate while remaining in the bulk phase, allowing the hydrates to be transported downstream and leading to a reduced blockage risk. As such, an optimisation framework presented for gas-dominated flows may be more applicable to the current industry needs. Another salient point is distinctly evident from the top row of Figure 2.2, which is that with the exception of the SLB-Drift Flux model, recent progress of steady state multiphase flow modelling has been limited to the development of mechanistic models.

Mechanistic models are based on fundamental laws and observations and can typically offer more accurate modelling than an empirical model. This is because an empirical model is based exclusively upon experimental data and accurate results are generally limited by the range of data on which they are based [Ellul et al., 2004]. The latest significant edition for two-phase mechanistic models is the Petalas and Aziz mechanistic model [Petalas and Aziz, 2000], which has been upgraded based on the substantial pioneering work of Taitel and Dukler [1976], Barnea et al. [1982], Xiao et al. [1990] and Gomez et al. [1999] and now can be applied to all pipe geometries and any fluid properties. Included in the model are various amendments to parameters such as the interfacial friction factor and the effects of wall roughness and liquid entrainment have been added. The Petalas and Aziz model has compared well to

data from the Stanford Multiphase Flow Database [Petalas and Aziz, 1995], which consists of over 1800 experiments using actual well data and over 20,000 laboratory measurements. A positive endorsement for the model can be found in the experimental work of Oddie et al. [2003], where the model was able to predict the observed flow patterns for water/gas and oil/water in inclined pipes with a high degree of accuracy and liquid holdup with a reasonable degree of accuracy.

The high level of validation makes the Petalas and Aziz mechanistic model ideal for comparison to CFD models and it is proposed that this mechanistic model can be used for verification of the two-phase CFD model presented in this study. If a good comparison is shown, the CFD model can then be used in confidence as part of the optimisation framework. Validation will be based upon comparing the liquid holdup, pressure gradient and flow pattern identification for two-phase flow in straight pipes at different inclinations to the data provided by Petalas and Aziz [2000], which consists of two separate flows of air/water and oil/gas at three inclinations of 0°, +10° and -10°.

2.6 CFD Modelling Packages

It is becoming increasingly apparent that open-source CFD packages have the ability to solve for complex fluid dynamic problems with capability at least at par with commercial software tools such as ANSYS CFX, ANSYS Fluent, StarCCM+® and COMSOL Multiphysics®. Examples of code comparisons can be found in the literature that point to recommendations for the use of open-source software in one way or another [Asim et al., 2015, Mackenzie et al., 2015, Lindmeier et al., 2010]. This is because all CFD codes are implemented using the same fundamental concepts that describe the flow by solving the Navier-Stokes equations using a spacial discretisation technique, implemented using the Finite Volume Method. Differences are considered to arise only in the presentation of the software and level of support given to a practitioner [Shengwei, 2013]. There is typically no support for open-source software apart from various internet forums and user mailing lists; however, open-source software is still widely used due to the absence of licensing costs, automatic parallelisation capability and access to the underlying source code, something a commercial software would not offer. For this reason the use of open-source CFD packages have been considered for the implementation in this framework.

Haghgoo [2013] has complied a list of current open-source software available for the purpose of identifying a suitable package for the modelling of coal gasification. To determine the most applicable CFD software, a test case involving flow around a cylinder was used at three separate Reynolds numbers to observe separation of the boundary layer with comparison to experimental results. Comparisons were made using SU2, OVERFLOW, OpenFOAM®, Mfix and Nek5000. Here, OpenFOAM® was slightly preferred over the other options available as it is ready-made for a wide range of applications and contains options such as the ability to select different discretisation schemes that the other open source CFD software did not contain. Other authors such as Chen et al. [2014] have favoured the use of OpenFOAM® for reasons such as:

- The contained codes are extensible for many customised applications.
- The generality of the various OpenFOAM® libraries and solvers empowers the user to solve nearly all CFD problems comprehensively.
- The pre and post-processing interfaces are well designed, powerful and user friendly.
- Dynamic meshes (moving grids) can be used and manipulated for dynamically changing geometry.
- The object-oriented C++ code development strategy makes it convenient for users to incorporate their own submodels.

It is worth noting that Chen et al. [2014] have also commented that the learning curve is steep, whereby the use of OpenFOAM® is learnt solely from sources such as online user groups and CFD communities. This is no doubt a challenge; however, it is not considered to be a limiting factor in this study. Thus, it is proposed OpenFOAM® is used here, particularly due to the script interface which will readily enable coupling with DAKOTA.

OpenFOAM® is a C++ library consisting of solvers and utilities with the primary purpose for use in CFD applications. Since the programming language is C++, it is compatible with DAKOTA and evidence exists in the literature that shows the successful coupling of OpenFOAM® and DAKOTA for CFD based optimisation studies, such as in Bailardi et al. [2015], Ger and Johansen [2011]. Two-phase flow is handled

in OpenFOAM® using interFoam, a transient solver for two incompressible, isothermal and immiscible fluids using an interface capturing technique based a modified VOF approach, which incorporates an additional interfacial compression flux term to reduce the effects of numerical smearing of the interface [Emad, 2014].

Validation of the interFoam solver has so far been confined to the specific interest of certain users and its applicability to a wide range of multiphase flow applications is yet to be established. Deshpande et al. [2012] has presented a well-known summary of the completed interFoam validation efforts and added further validation for pure advection, high Weber number flows and surface tension dominated flows. Any further review of the literature will suggest that use of interFoam to model two-phase pipe flow is scarce and has also been confined only to visual inspection of flow patterns based upon a specified flow map. Labeaga and Omagogeascoa [2013] simulated two-phase flows of air and water in open and closed channels at different inclinations with comparison to the Taitel and Dukler flow map. The results reported by Labeaga and Omagogeascoa [2013], however, were inaccurate due to a low mesh density and lack of a three-dimensional model. Thaker and Banerjee [2013] performed a three-dimensional numerical simulation of air and water flow in a 10 mm diameter horizontal pipe with flow pattern results compared by visual inspection to the experimental work of Vaze and Banerjee [2011]. Here, the numerical results for the stratified, slug, plug and annual flow patterns compared well visually to the experimental results however no attempt was made to obtain and compare values of the liquid holdup and pressure gradient.

The absence of a comprehensive validation study for two-phase pipe flow using interFoam presents an opportunity to compare results obtained from the interFoam solver to the Petalas and Aziz mechanistic model described in the previous subsection. This will not only provide confidence that the model created may be used in this optimisation framework, but provide a foundation to increase the accuracy and effectiveness of the interFoam solver for future studies. As a result, the applicable optimisation toolkit, solver and validation technique have all been selected and one final aspect remains; to consider how the hydrate growth may be modelled during the restart operation. As it has been proposed in this study that an estimation in the amount of hydrate growth can be used to determine a qualitative means of defining the risk of hydrate blockage during a restart operation, the applicable means to model the hydrate growth has been investigated and is presented in the following subsection.

2.7 Modelling Hydrate Growth

A number of hydrate growth models have been developed to estimate the rate at which a hydrate forming gas is consumed to become hydrate material, using either kinetic, mass transfer or heat transfer limited models. A list of the main hydrate growth models has been compiled by Sloan and Koh [2007] and is reproduced here in Table 2.6, with the inclusion of an additional kinetic model by Turner et al. [2005]. Here, the term $f - f_{eq}$ is the driving force based on fugacity, $x_{int}^{j} - x_{b}^{j}$ is the driving force based on the bulk liquid mole fraction and $T_{eq} - T_{sys}$ is the driving force based on the degree of sub-cooling.

Table 2.6: Hydrate growth models identified by Sloan and Koh [2007], with an additional model by Turner et al. [2005].

Growth Model	Driving Force/Model Features	Author(s)
Growth kinetics	$(f - f_{eq})$	Englezos et al. [1987]
Growth kinetics	$(f - f_{eq})$ Minor modification to Englezos' model	Malegaonkar et al. [1997]
Growth kinetics	$(T_{eq} - T_{sys})$	Turner et al. [2005]
Mass transfer	$(x_{int}^{j} - x_{b}^{j})$ Simplification/modification to Englezos' model	Skovborg and Rasmussen [1994]
Mass transfer	Based on field phase theory	Svandal et al. [2006]
Mass transfer	Based on Monte Carlo cellular automata	Buanes et al. [2006]
Heat transfer	Curved film front growth on water-hydrate former interface	Uchida et al. [1999]
Heat transfer	Curved film front growth on water-hydrate former fluid interface	Mori [2001]
Heat transfer	Straight film front growth on water side of water-hydrate former interface	Freer et al. [2001], Mochizuki and Mori [2006]

The heat transfer limited models shown in Table 2.6 are restricted to estimating the lateral film front growth and are not applicable for use in this study, as it can be reasonably expected that the ambient temperature for a subsea flowline will remain essentially constant throughout the length of the flowline until the riser base, such that the heat transfer mechanism will only play a minor role in the overall amount of hydrate growth. In contrast, for both types of kinetic and mass transfer limited hydrate growth models, the hydrate growth rate is a function of the interfacial area between the hydrate forming components and the free water phase, coupled with the degree of subcooling between the hydrate equilibrium temperature at the system pressure and the system temperature.

The hydrate growth model developed by Englezos et al. [1987] is possibly the most well known kinetic model and is considered by Sloan and Koh [2007] to be the pio-

neering foray into hydrate modelling, mainly providing a foundation for future work albeit with many restrictions that govern its usage. In particular, the experimental data used for this model was only captured for the hydrate growth of the methane, ethane and carbon dioxide hydrate forming compounds that typically comprise the "sI" type of hydrate. As such, the model cannot be used in confidence for other types of hydrates such as "sII" and "sH". Furthermore, Sloan and Koh [2007] have even suggested that the entire role of hydrate growth kinetics is much smaller than either mass or heat transfer when considering actual hydrate growth in turbulent pipes, possibly rendering any form of estimation based solely upon growth kinetics irrelevant. The subsequent model developed by Malegaonkar et al. [1997] only contains minor modifications to the model by Englezos et al. [1987] to account for slight inconsistencies and as such, is subject to the same restrictions highlighted by Sloan and Koh [2007].

In order to address the restrictions to the models by Englezos et al. [1987], Skovborg and Rasmussen [1994] developed a simplified model by modelling the entire hydrate process based on the mass transfer of the gas through the liquid film at the gas-liquid interface. Unfortunately, this model is also far from a perfect solution, as it appears to be only a data-fit without any further theoretical significance, with additional systematic driving force errors [Sloan and Koh, 2007]. Two of the more recent mass transfer models developed are based upon phase theory [Svandal et al., 2006] and a hybrid of Monte Carlo and cellular automata [Buanes et al., 2006]. A study by Buanes et al. [2009] has shown that both of these models produce comparable results, however these models are limited to estimating the hydrate growth in carbon dioxide and methane and are not considered for use in this study.

It is thus largely evident that implementation of any type of hydrate growth model is far from a robust and comprehensive solution. Furthermore, all of the current hydrate growth models are only relevant to the empirical data on which they were based, where most of these hydrate models appear to be obtained from high-pressure reactor experiments, meaning that any relevance to actual flowlines is largely unknown. Despite this, both Di Lorenzo et al. [2014] and Aman et al. [2016] have provided the most relevant examples for gas-dominated flow, where both studies feature a comparison of the kinetic model by Turner et al. [2005] and the mass transfer limited model by Skovborg and Rasmussen [1994] to experimental flowloop data for annular flow. Unfortunately, the results from both studies have proven to be somewhat inconclusive. Using the same experimental methodology, it would appear

that at lower velocities (4.6 m/s) the mass transfer model was more accurate, although at higher velocities (8.7 m/s) the kinetic model was more representative. In both cases the kinetic model predicted higher growth rates than the mass transfer model.

In light of this, the best approach would appear to be to compare both of the Skovborg and Rasmussen [1994] or Turner et al. [2005] models, or to combine the results of both models as a means to produce a conservative estimate, particularly as a restart operation involving agitation of the gas/water interface would conceivably involve both kinetic and mass transfer effects. Incidentally, both of the studies by Di Lorenzo et al. [2014] and Aman et al. [2016] have used a constant interfacial area, calculated for annular flow based upon a combination of semi-empirical correlations from authors such as Pan and Hanratty [2002] and Beggs and Brill [1973]. A constant area approach; however, is not applicable for any kind of transient operation. Instead, the use of CFD to model the transient response of the system will enable a more accurate prediction of the hydrate growth, in a manner that is not currently possible with the traditional 1-D drift-flux models and has not yet been observed in the literature. Furthermore, as the estimations of the hydrate growth are built from the response of the simulation, any hydrate model can be used, interchanged or refined if necessary, without any additional computational expense. This is particularly advantageous considering the uncertainty involved with the implementation of the hydrate growth models.

2.8 Summary

A review of the processes involved in building an optimisation framework has been conducted. For this purpose, the framework can be split into three separate parts; an optimisation toolkit, a simulation tool and an interface to connect the two. It has been found that many optimisation toolkits are available that already have the capability to provide an interface to the simulation software, such that the only interface required to build a comprehensive framework is in the form of a GUI or TUI, which can be used to present the framework in a more intuitive manner. The available optimisation toolkits have been assessed following a review of the optimisation process and DAKOTA has been proposed for use in this study due to it being the most capable program as shown in Table 2.4.

The review of the multiphase flow modelling techniques presented in Section 2.4 has shown that most flowline operators use commercial software based upon the 1-D

drift flux modelling approach. These commercial models are expensive which put them out of reach for typical studies in academia. Instead, the use of CFD has been proposed for use in this study, although it has been found that current CFD studies of multiphase flow are limited to describing two-phase flows due to a likely combination of modelling complexity and lack of verification techniques available for three-phase flow. Hence, the scope of the restart operations presented in this study has been limited solely to two-phase gas-dominated flows, which will minimise the erroneous modelling of condensate and water as a single phase. The Petalas and Aziz mechanistic model has been selected to verify the CFD model due to the ease of its implementation and validation with experimental data.

A final review of CFD modelling packages has lead to OpenFOAM® being selected to model the two-phase flow using the interFoam solver. As both OpenFOAM® and DAKOTA are based on the C++ programming language, linkage between the two is expected to be straightforward and robust. Finally, a review of hydrate growth models has shown that current developments are not yet able to provide comprehensive solutions and many restrictions govern the usage of such models. In light of this, it has been proposed to use the kinetic model developed by Turner et al. [2005] in conjunction to the mass transfer model by Skovborg and Rasmussen [1994] in order to account for mass transfer effects due to agitation of the gas/water interface during the restart operation.

Chapter 3

Validation of a CFD Model for Two-Phase Pipe Flow

In this chapter, a 3-D CFD model is built using OpenFOAM® for the purpose of validating the solver, mesh and turbulence model for subsequent implementation in the optimisation framework. Validation of the constructed CFD model is accomplished by comparing the simulated values of the pressure gradient and liquid holdup obtained from the CFD model to the mechanistic model by Petalas and Aziz [2000]. A total of 120 CFD simulations are conducted using the interFoam solver for two separate two-phase flow mixtures of air and water and oil and gas at 0°, +10° and -10° inclinations. The fluid properties and specific inclinations used for these simulations directly coincide with the data presented by Petalas and Aziz [2000]. The CFD simulation points are selected such that comparisons can be made across multiple flow regimes, including; stratified wavy, slug, dispersed bubble, froth and annular flow. The work presented here constitutes the culmination of work required to achieve the first part of Objective 2 from this study.

3.1 Implementation of the Mechanistic Model

The methodology for the implementation of the mechanistic model is presented completely in Petalas and Aziz [2000] and is not discussed in detail here. Instead, an overview of the procedure used in this study is presented in Figure 3.1, which has been adapted directly from Petalas and Aziz [2000].

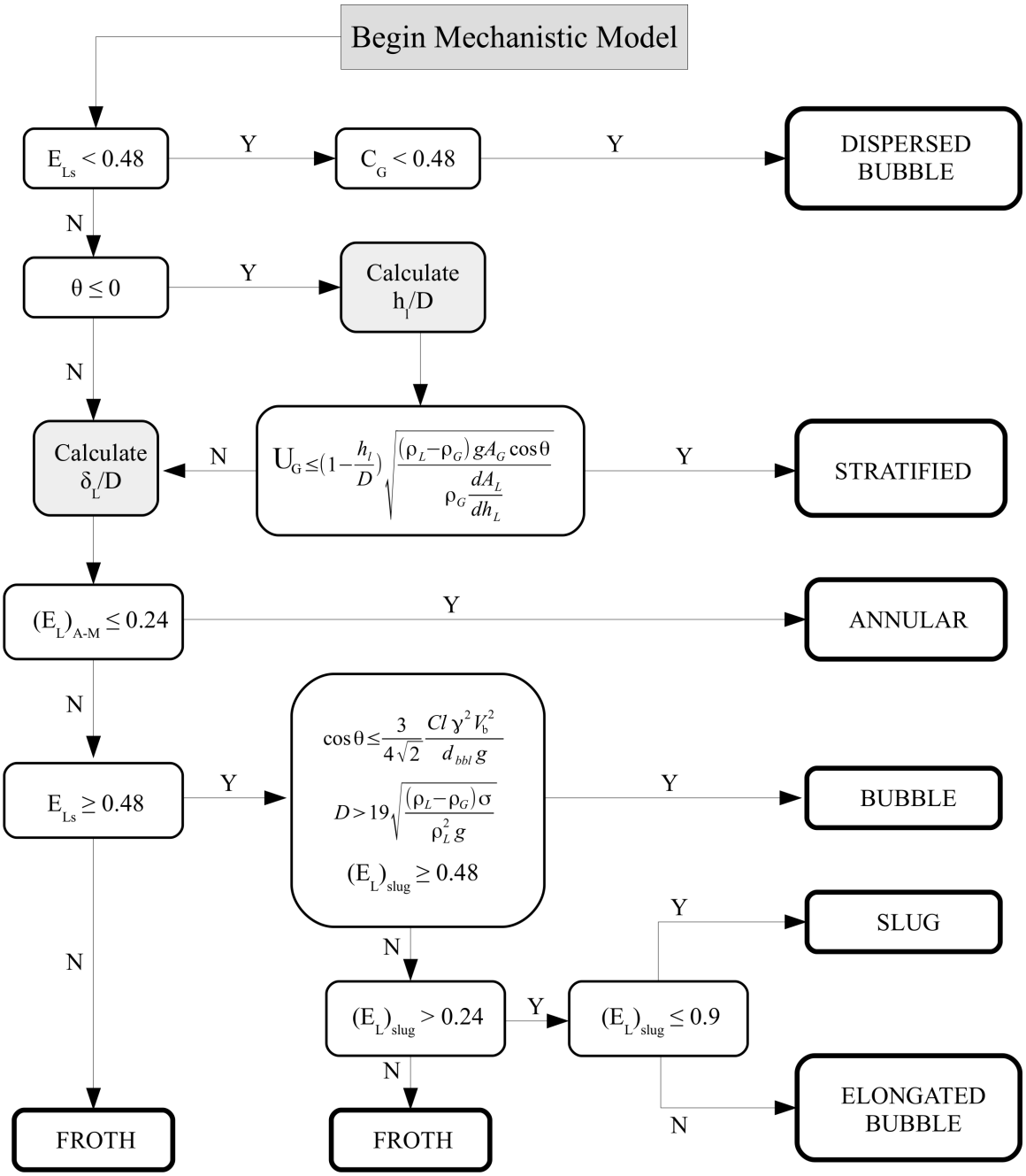


Figure 3.1: The procedure used to implement the Petalas and Aziz mechanistic model, adapted from Petalas and Aziz [2000].

Using the procedure shown in Figure 3.1, the mechanistic model has been implemented in this study using Scilab 5.5.2. It is important to note that some simplifications have been made as the model is only for use within $\pm 10°$ inclinations. These simplifications include:

- No provision has been made to distinguish between stratified and stratified-wavy flow. This is because the liquid holdup and pressure gradient calculations are the same.

- The liquid phase velocity criterion present in the mechanistic model; $U_L \leq \sqrt{\frac{gD(1-h_l/D)cos\theta}{f_L}}$ is not used as it is only applicable to steeply declined pipes.

- Calculation of the minimum film thickness, δ_{min}, is not performed as it is only applicable to steeply inclined pipes.

In addition, the turbulent Darcy friction factor, f, has been calculated in this study using the method proposed by Swamee and Jain [1976], i.e.

$$f = \frac{0.25}{\left[log_{10}\left(\frac{\epsilon}{3.7\times D} + \frac{5.74}{Re^{0.9}}\right)\right]^2} \tag{3.1}$$

An alternative method is to used Haaland's formula [Haaland, 1983], however it was found that results were comparable. Finally, the transition between turbulent and laminar flow is set as a hard boundary at a Reynolds number, Re, of 2100; below this, the laminar friction factor is calculated as:

$$f = \frac{64}{Re} \tag{3.2}$$

3.2 Numerical Method

OpenFOAM® 2.1.x was used to simulate the flow using the two-phase interFoam solver. For increased accuracy, it is intended to resolve the boundary layer rather than use a wall function approach. For this purpose, the Steady State Transport (SST) $k-\omega$ turbulence model has been selected for all simulations. This model uses a blending function to transition between the standard $k-\omega$ model for near wall regions to a high Reynolds number type $k-\epsilon$ model in the outer portion of the boundary layer. Menter et al. [2003] have suggested that the $k-\omega$ model is substantially more accurate for near wall layers and the SST $k-\omega$ model has successfully been used to model pipe flow by authors such as Knotek et al. [2016] for stratified/slug flow, Rzehak and Kriebitzsch [2015] for bubbly flow and Fontes et al. [2008] for annular flow. It is recognised that authors such as Yeoh and Tu [2009] have suggested that a single turbulence model cannot reasonably predict all turbulent states involved in multiphase flow. This is

no doubt an issue and may result in a reduced level of accuracy; however, due to computational constraints, this issue has not been investigated in further detail in the current study.

Two 3-D half-pipe CFD models were constructed to simulate both of the air/water and oil/gas flows, where a vertical symmetric plane was employed as a means to decrease the number of mesh elements by half. In order to ensure an accurate comparison is made, the same fluid properties and geometries are used as presented in the article by Petalas and Aziz [2000]. As a result, the CFD model of air/water system has a diameter of 0.052 m and a length of 4 m and the oil/gas system has been scaled up to a diameter of 0.1569 m and a length of 12 m. The geometry and mesh for both models were constructed using SALOME 7.4.0 and exported to OpenFOAM® via the ideasUnvToFoam utility, where both meshes comprised of 352,628 hexahedral elements and were selected such that the boundary layer was resolved in all simulations using 8 graded layers. The main advantage of resolving the boundary layer is that the mesh can remain unchanged throughout all simulations provided that the y^+ value for each phase is below a calculated value of 6 [Guerrero, 2014]. Here, y^+ is a parameter that represents a non-dimensional wall distance for a wall-bounded flow and is calculated using the yPlusRAS utility in OpenFOAM®. A more detailed description of how y^+ has been calculated is presented in Appendix E.2. Figure 3.2 shows the topology of the mesh at the inlet plane, where an equal division for the inlet air/gas and water/oil flows can be seen.

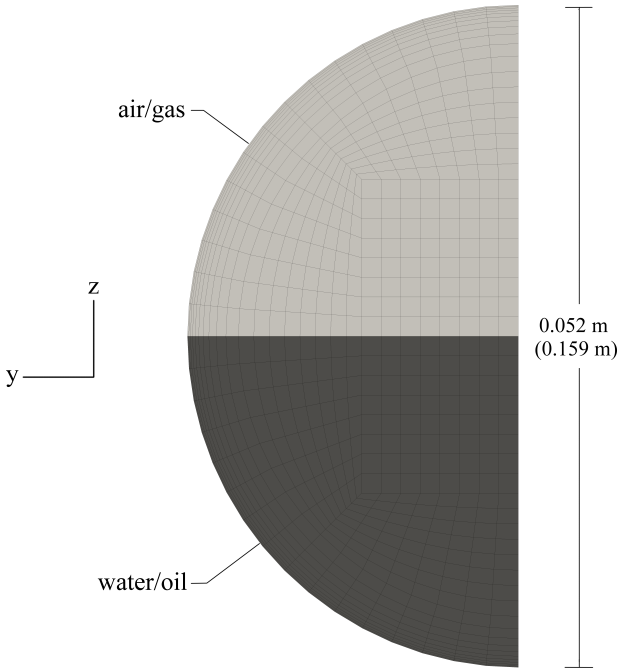


Figure 3.2: The mesh cross section used for the 0.052 m diameter half-pipe (air/water flow) and the 0.159 m half-pipe (oil/gas flow).

For all simulations, the pipe was initially filled completely with the liquid phase and as shown previously in Figure 3.2, an equal fraction of gas and liquid was introduced at the inlet upon commencement of the simulation (t = 0 s). The simulations were then run until the flow from both phases introduced at the inlet had time to pass through the entire length of the pipe and exit the computational domain. The values of the liquid holdup and pressure gradient used for the mechanistic model comparison were distance-averaged from a point defined by the furthest turbulent entrance length calculated for each phase (whichever phase was greater) to a point 0.5 m from the pipe outlet for the air/water simulations and 2 m for the oil/gas simulations. This was done in order to reduce any effect that the inlet and outlet pressures may have on the solution. The residuals were checked using the final residual data located in the log file. Gravitational acceleration was applied in the vertical (Z) direction and the effect of the different pipe inclinations was realised by altering the appropriate vector components of the gravitational acceleration in the *g* file contained in the *constant* file directory.

In order to ensure correct solution control, an adjustable timestep was used with an initial value of 1e-07s in conjunction with a MaxCo (mean Courant number) limit of 0.3 and a MaxAlphaCo (interface Courant number) limit set to 0.1. The standard

numerical schemes for interFoam are used with the exception of *limitedLinearV 1* for the velocity component advection term that was used to improve solution stability. The calculations for the pressure-velocity coupling are performed using the PIMPLE (merged PISO-SIMPLE) algorithm with five corrector loops and interface compression was used for the determination of the phase fraction to yield a sharp interface. The other main system parameters used in the analysis are shown in Table 3.1.

Table 3.1: System parameters.

Parameter	Air/Water System	Oil/Gas System
Pipe diameter	0.052 [m]	0.1569 [m]
Pipe length	4 [m]	12 [m]
Gas density	1.28 [kg/m^3]	130.35 [kg/m^3]
Liquid density	1000 [kg/m^3]	841.45 [kg/m^3]
Gas kinematic viscosity	7.8e-06 [m^2/s]	1.38e-07 [m^2/s]
Liquid kinematic viscosity	1e-06 [m^2/s]	3.27e-06 [m^2/s]
Interfacial tension	0.74 [N/m]	0.2 [N/m]
Pipe roughness (absolute)	4.57e-04 [m]	3.05e-03 [m]

All results obtained from this analysis were viewed and analysed using ParaView 5.0.0, an open-source, multi-platform data analysis and visualisation application.

3.3 Boundary Conditions

In all simulations the velocity was specified as $fixedValue$ at the inlet and *zeroGradient* at the outlet. Conversely, the pressure was specified as *zeroGradient* at the inlet and $fixedValue$ at the outlet. A no-slip boundary condition is applied to the pipe wall by setting the velocity as zero and a symmetry boundary condition was imposed at the longitudinal section of the half-pipe in the X-Z plane shown previously in Figure 3.2. The boundary conditions used for all simulations are listed in Table 3.2 and a more detailed description of the terms used to define the boundary conditions is presented in Appendix E.1.

Table 3.2: Boundary conditions.

Parameter	Inlet	Outlet	Wall
U	fixedValue	zeroGradient	fixedValue uniform 0
p_rgh	zeroGradient	fixedValue uniform 0	zeroGradient
k	fixedValue	zeroGradient	fixedValue
nut	Calculated uniform 0	Calculated uniform 0	nutkRoughWallFunction
omega	fixedValue	zeroGradient	omegaWallFunction

3.4 Mesh Independency and Symmetry Analysis

Prior to commencement of the comparative study, a mesh independency analysis was performed for the air/water system by comparing the simulated values of the liquid holdup and pressure gradient to the total number of cells in the mesh for all flow patterns considered. Using the same cross section as shown in Figure 3.2, the total cell count was increased from an initial number of 16,000 until a point where the results obtained from the CFD simulations were less than 5% of the previous case. Figure 3.3 shows the evolution of the pressure gradient depending on the total cell count of the mesh for all flow patterns with comparison to the values obtained from the mechanistic model.

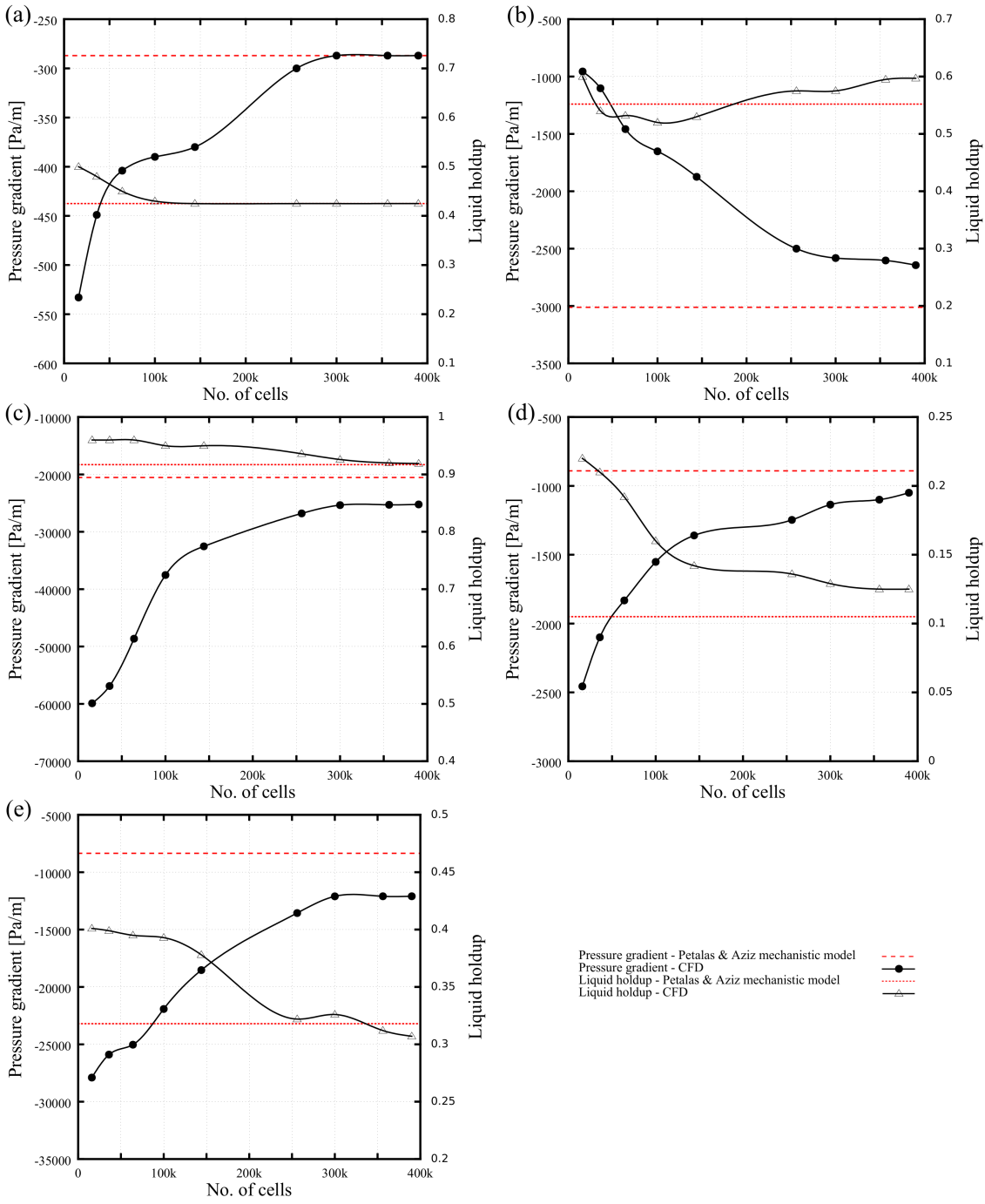


Figure 3.3: The evolution of the simulated pressure gradient and liquid holdup depending on the total mesh cell count with a comparison to the theoretical values obtained from the mechanistic model over all flow regimes considered where; a) stratified wavy flow (U_{SG} = 2m/s, U_{SL} = 0.5 m/s, inclination = -6.25°) , b) slug flow (U_{SG} = 3 m/s, U_{SL} = 3 m/s), c) dispersed bubble flow (U_{SG} = 1.5 m/s, U_{SL} = 12 m/s), d) annular flow (U_{SG} = 18 m/s, U_{SL} = 0.3 m/s) and e) froth flow (U_{SG} = 12 m/s, U_{SL} = 3 m/s).

By observation from Figure 3.3, the point at which the simulated CFD results for both the pressure gradient and liquid holdup lie within a ±5% error for all flow patterns occurs at a total cell count above 350,000 cells. With the exception of slug flow, all flow patterns exhibit the same behaviour as the cell count is increased, whereby the values of the liquid holdup and pressure gradient reduce. For slug flow, the opposite can be seen, where the pressure gradient increases with each increment. This is because the true formation of slugs can only be observed for cell counts greater than 250,000 cells. Below this number, no slugs would form and the flow was observed to be stratified wavy, leading to a lower simulated value for the pressure gradient.

Unfortunately, the high number of cells is a limiting factor towards the intended application for extended flowline lengths, especially considering the magnitude of the errors that are apparent for all flow patterns shown in Figure 3.3 when a comparison is made to the mechanistic model for mesh sizes less than 200,000 cells. As a result, to adequately model a two-phase oil/gas flowline of 1 km length for all flow patterns would require an extremely unrealistic cell count in excess of 29 million. This suggests that another approach is necessary, such as to use wall functions instead of modelling the boundary layer. Nevertheless, analysis of the mesh presented in this chapter is still useful to characterise the interFoam solver for two-phase flow and discover any limitations apparent with the boundary layer modelling strategy. The model can also be used to provide benchmark results for an additional model specifically constructed for longer flowline lengths. Development of this additional model is shown in the following chapter and the results and discussion from the comparison of the mechanistic model to the original model is shown in the remainder of this chapter.

Following the mesh indepedency analysis, five simulations of the air/water system were run on a full 3-D horizontal pipe to determine the suitability of using a symmetric boundary condition for all flow patterns. The mesh had the same cross sectional resolution as shown in Figure 3.2 and comprised of 705,256 elements. Figure 3.4 shows the longitudinal (Ux) and lateral (Uy) velocities per radial distance (sampled in the Z direction) at the geometric midpoint of the pipe for each of the separate flow patterns.

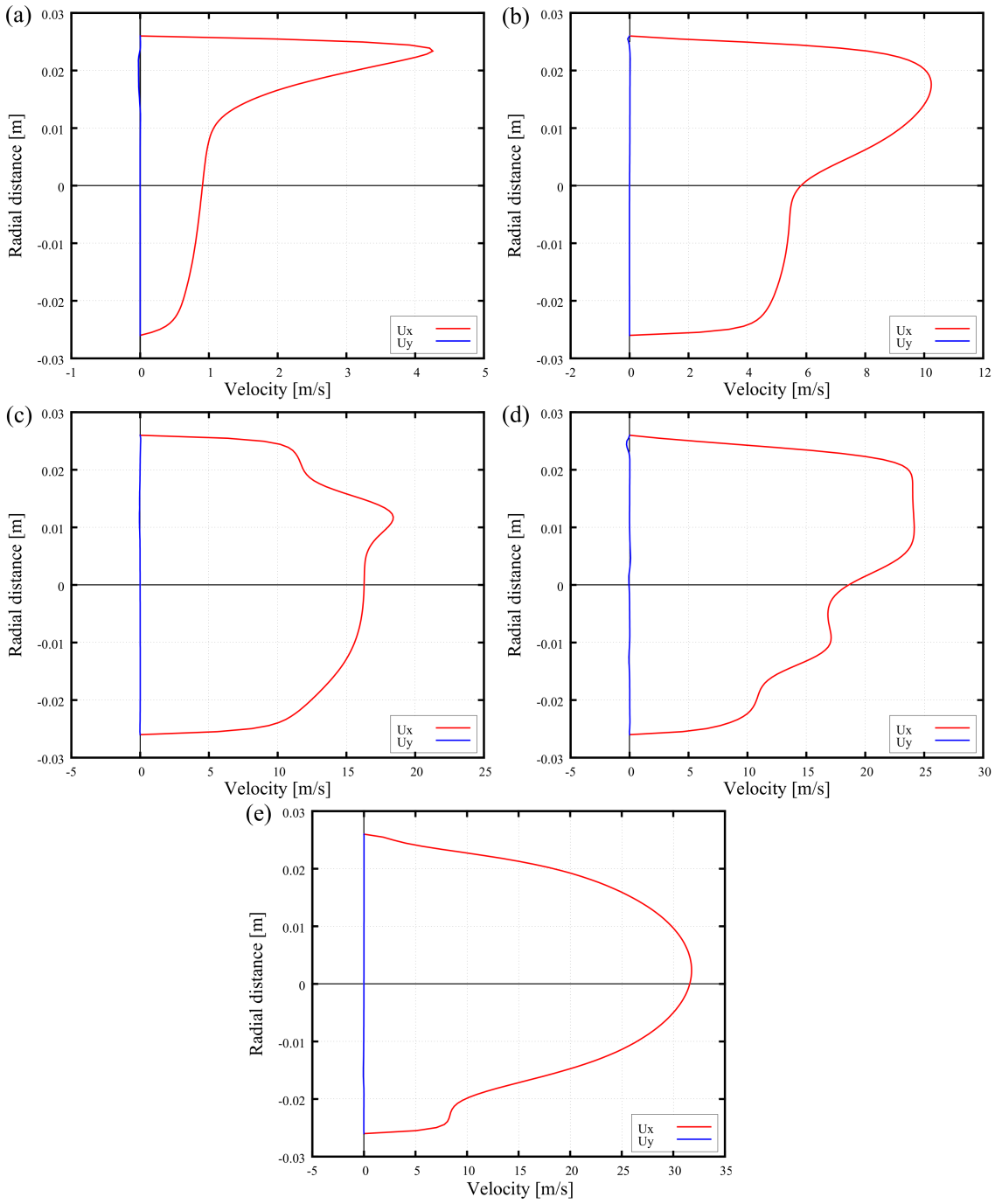


Figure 3.4: The simulated longitudinal (Ux) and lateral (Uy) velocities recorded at the pipe geometric midpoint, sampled in the vertical (Z) direction for; a) stratified wavy flow ($U_{SG} = 2$ m/s, $U_{SL} = 0.5$ m/s), b) slug flow ($U_{SG} = 3$ m/s, $U_{SL} = 3$ m/s), c) dispersed bubble flow ($U_{SG} = 1.5$ m/s, $U_{SL} = 12$ m/s), d) annular flow ($U_{SG} = 18$ m/s, $U_{SL} = 0.3$ m/s) and e) froth flow ($U_{SG} = 12$ m/s, $U_{SL} = 3$ m/s).

The behaviour of the longitudinal velocity for two-phase stratified wavy, slug and annular flow shown in Figure 3.4 is consistent with results reported by Thaker and Banerjee [2013], while the behaviour of the dispersed bubble flow regime is consistent with results presented by Ekambara et al. [2012]. A suitable comparison could not be found in the literature for froth flow in horizontal pipes; however, the observed results are consistent with the definition that froth flow is a transition between slug and mist flow regimes. Nevertheless, the results for all flow patterns show that the lateral velocities are minimal across the geometric mid-plane. It can thus be concluded that the symmetric boundary condition is applicable for use in this study as an effective means to reduce the required cell count by half.

3.5 Simulation Points

20 simulations have been run for each of the three separate pipe inclinations considered with both sets of fluid properties, bringing the total to 120 simulations conducted. The different superficial inlet velocities for the liquid and gas phases are shown superimposed on the approximate flow regime map in Figure 3.5 for the air/water flow and Figure 3.6 for oil/gas flow, whereby; a) is for the horizontal pipe, b) the +10° inclined pipe and c) the -10° declined pipe, respectively. The flow regime map for each inclination has been determined by the mechanistic model.

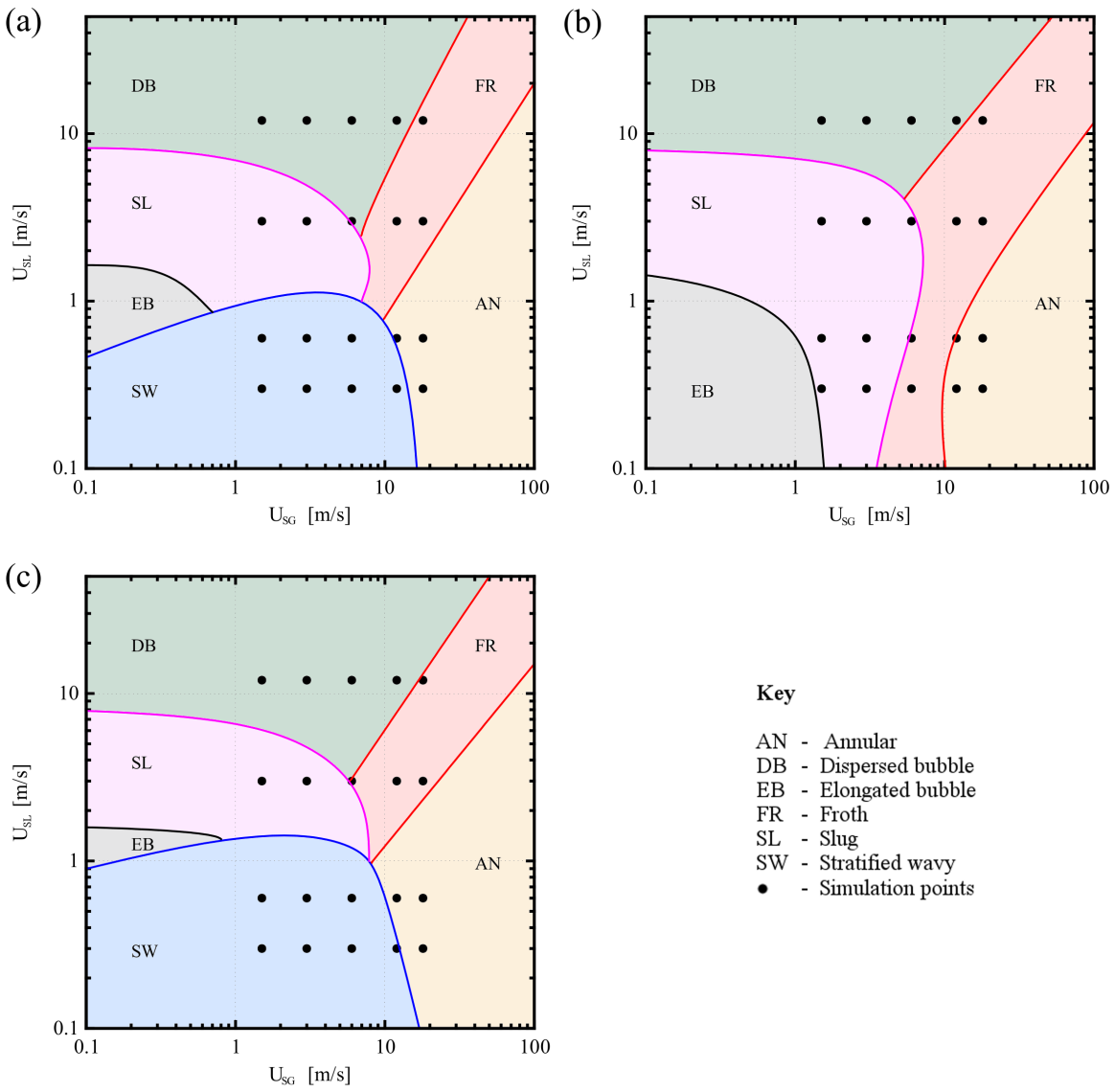


Figure 3.5: The CFD simulation points for the air/water fluid combination superimposed on the flow regime map generated by the mechanistic model; a) the horizontal pipe, b) +10° inclined pipe and c) the -10° declined pipe.

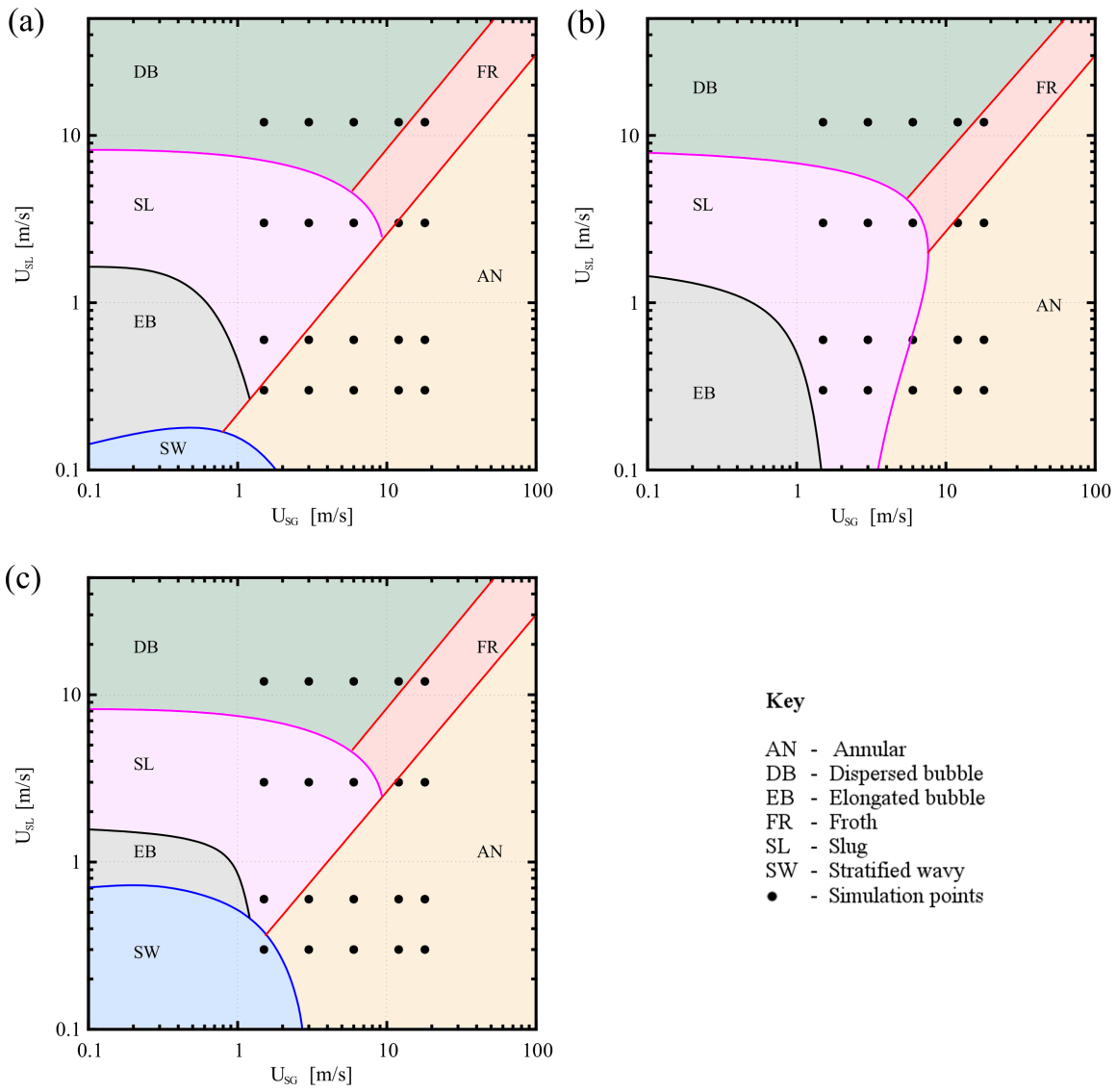


Figure 3.6: The CFD simulation points for the oil/gas fluid combination superimposed on the flow regime map generated by the mechanistic model; a) the horizontal pipe, b) +10° inclined pipe and c) the -10° declined pipe.

3.6 Results

3.6.1 Air and Water Flow

3.6.1.1 Flow Pattern Visualisation

The general types of visualisation generated for each flow pattern considered in this analysis; stratified wavy, slug, dispersed bubble, froth and annular mist flow are presented graphically in terms of volume fraction of the water phase (alpha.water) in Figure 3.7. Here, the black colour represents water (alpha.water = 1) and the grey colour represents air (alpha.water = 0). The interface between the two phases is represented by a very faint light grey colour (alpha.water = 0.5).

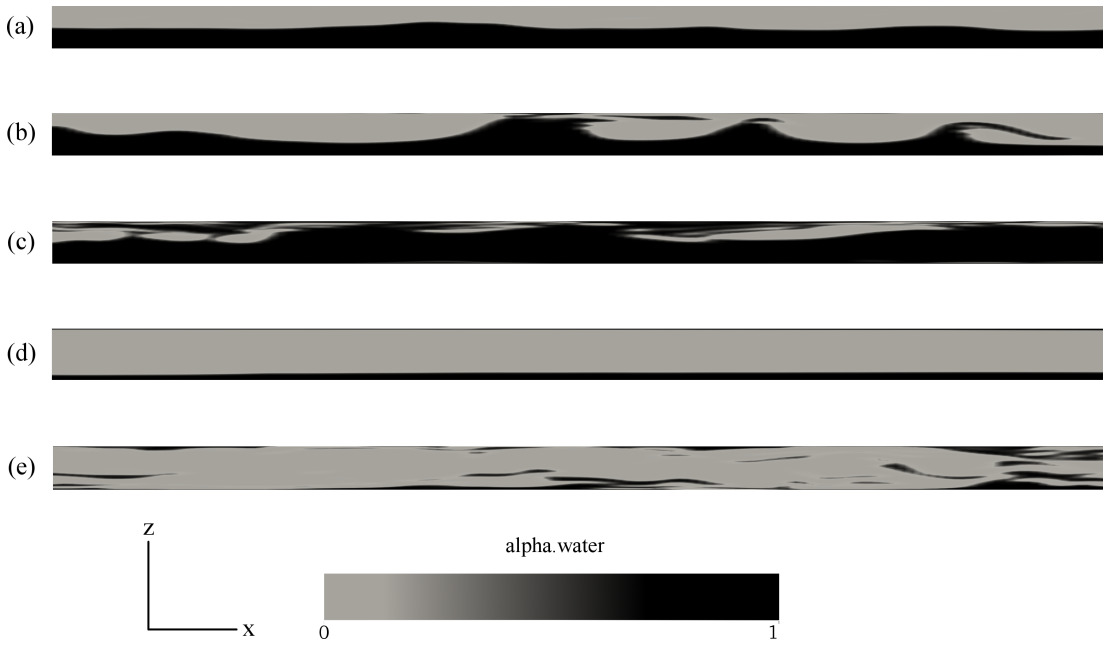


Figure 3.7: Flow pattern visualisation for horizontal flow; (a) stratified wavy (U_{SG} = 1.5 m/s, U_{SL} = 0.6 m/s), (b) slug (U_{SG} = 3 m/s, U_{SL} = 3 m/s), (c) dispersed bubble (U_{SG} = 1.5 m/s, U_{SL} = 12 m/s), (d) annular mist (U_{SG} = 18 m/s, U_{SL} = 0.3 m/s) and (e) froth flow (U_{SG} = 18 m/s, U_{SL} = 3 m/s).

Stratified Wavy: Results show a good visual correlation in all cases between the CFD results and the flow pattern identified by the mechanistic model. Waves form periodically at the interface and are carried downstream. The wave height was shown to be dependent on the velocity difference of the two phases as postulated in the Kelvin-Helmholtz wave instability theory [Yih, 1969], where an increase in the liquid inlet velocity correctly yielded an increase in liquid holdup while an increase in gas velocity yielded a decrease in liquid holdup. The -10° declined pipe correctly

predicted a reduction in liquid holdup for all cases and stratified flow was correctly shown not to exist in the +10° inclined pipe.

Slug: The existence of slug flow was predicted correctly for all simulations with visualisation results closely matching the numerical and experimental study reported in the literature by Vallée et al. [2008]. Slug flow was initialised by a wave forming at the interface of the two phases. As the liquid velocity was greater than observed in all cases of stratified wavy flow, the liquid holdup had increased to a point such that waves that were formed at the interface were high enough to impact the top pipe surface causing periodic slugging to occur at regular intervals. The effect of gravity was evident with slugging occurring at lower liquid velocities in the +10° inclined pipe, and higher liquid velocities for the -10° declined pipe.

Dispersed Bubble: Upon commencement of the simulations in the dispersed bubble flow regime, the liquid height quickly reached the top surface of the pipe and small groups of air bubbles were observed to be entrained in the main fluid body and carried downstream. The longitudinal mesh density was not fine enough to capture all the detailing on the bubbles which appear to coalesce and elongate in this study, as opposed to being purely spherical as defined in the literature [Andreussi et al., 1999]. Any changes in pipe inclination had only a negligible effect on the visualisation of the flow pattern, which is consistent with the flow pattern map shown previously in Figure 3.5.

Annular Mist: All cases of the annular flow patterns were consistent with the mechanistic model predictions. The visualisation correctly consisted of a layer of fluid at the radial extremity of the pipe with an annular core of high velocity gas; however, no evidence of mist existed in the core owing to the limited mesh size. The lower section of the layer was observed to be noticeably thicker (typically by a factor of 2 or more) due to the influence of gravity, which is consistent to results presented in the experimental study conducted by Kumar et al. [2013].

Froth: Visualisation of froth flow deviated significantly from the experimental results reported in the literature by Gajbhiye and Kam [2011]. Instead of a foamy type flow, it was observed that all simulated cases of froth flow comprised of wispy areas of highly elongated water segments contained within the air phase. The visual results however, were significantly different from all other flow pattern visualisations and thus can be said to match the mechanistic model predictions. Froth flow was initiated by an increased gas phase velocity which caused an initial liquid wave to

quickly reach the top surface of the pipe, where it was subsequently broken up and carried downstream in elongated liquid streaks.

3.6.1.2 Liquid Holdup and Pressure Gradient Comparison

The values obtained for the liquid holdup and pressure gradient depending on the specific flow regime are presented for both the CFD and mechanistic models in Figures 3.8 and 3.9, respectively, where; a) stratified wavy, b) slug c) elongated bubble, d) annular and e) froth flow. In this representation, the specific velocity input value for each of the gas and liquid phases can be deduced from the flow maps presented in Figure 3.5, whereby the simulation numbers shown in Figures 3.8 and 3.9 represent the simulation point from Figure 3.5 when read from left to right, starting from the lower left most point in the flow pattern zone. In addition, dotted lines have been used in Figures 3.8 and 3.9 for visualisation purposes to indicate consecutive simulations that are at the same inlet liquid superficial velocity as referenced by the flow map in Figure 3.5. All results presented in both figures have additionally been reproduced in tabular form in Appendix A.1.

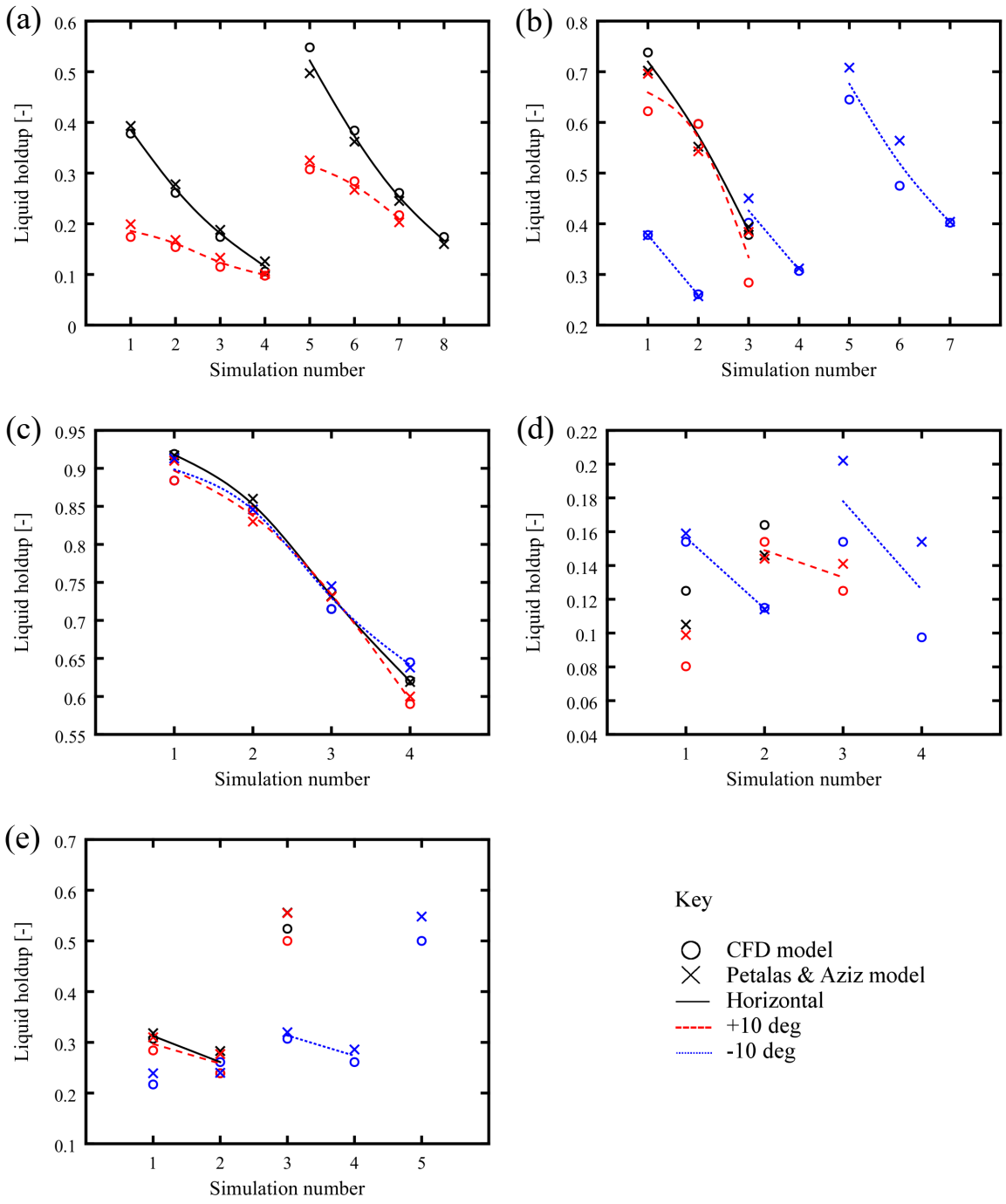


Figure 3.8: The air/water liquid holdup results for each flow regime; (a) stratified wavy, (b) slug, (c) dispersed bubble, (d) annular and (e) froth. The additional lines show the average value between the CFD model and the mechanistic model for simulation points at the same specific liquid velocity.

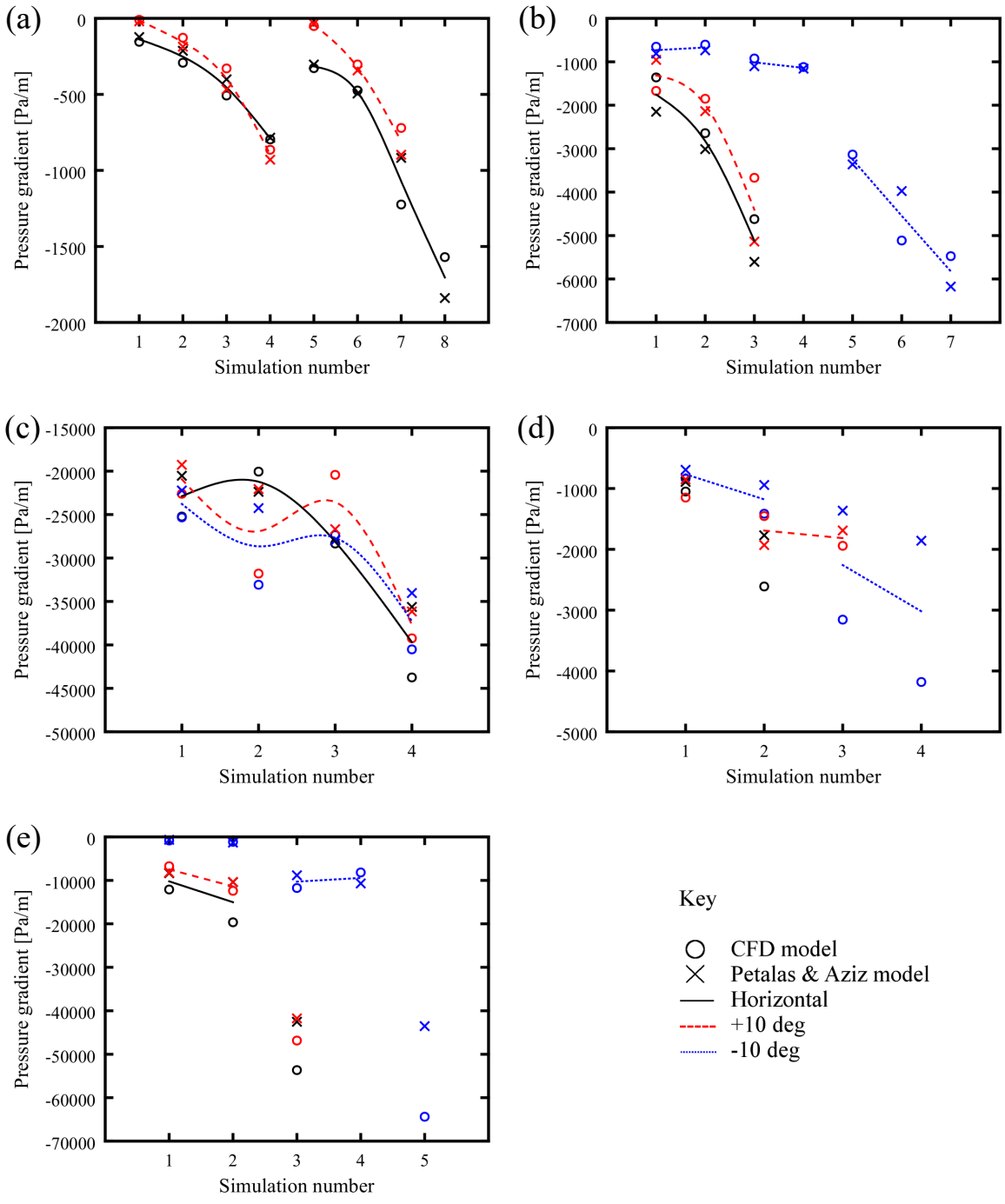


Figure 3.9: The air/water pressure gradient results for each flow regime; (a) stratified wavy, (b) slug, (c) dispersed bubble, (d) annular and (e) froth. The additional lines show the average value between the CFD model and the mechanistic model for simulation points at the same specific liquid velocity.

The error magnitude for the complete dataset for air/water flow is presented in Figure 3.10 and values of the average error magnitude for the liquid holdup and pressure gradient depending on the pipe inclination and flow pattern type are presented in Figure 3.11.

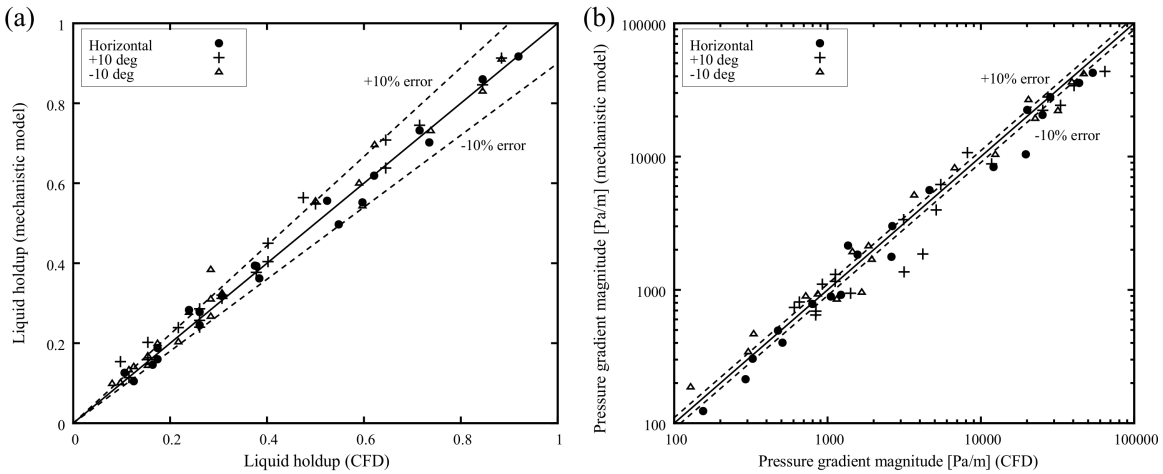


Figure 3.10: The error magnitudes for the complete dataset for air/water flow; a) liquid holdup and b) pressure gradient.

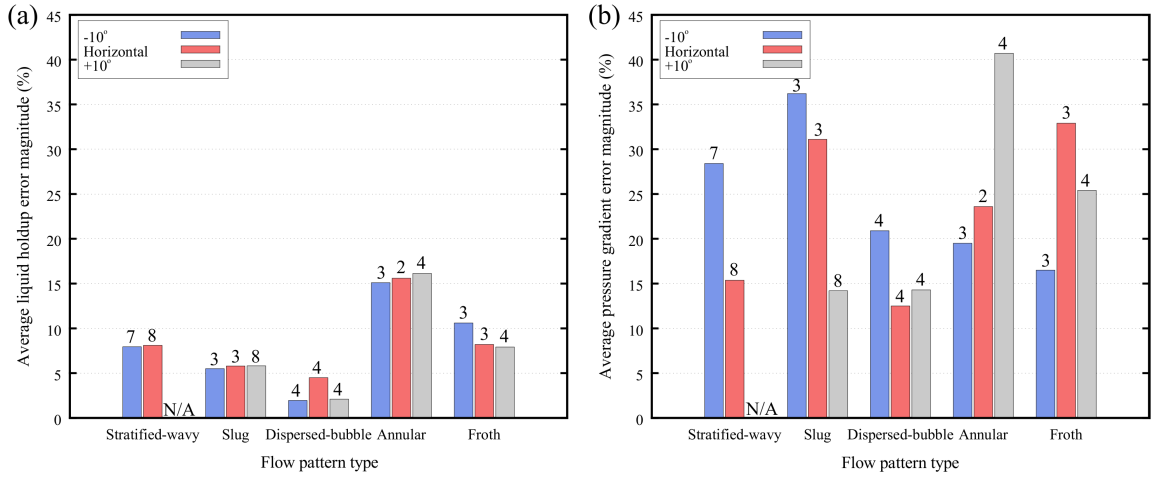


Figure 3.11: The average error magnitude for all flow regimes and pipe inclinations for air/water flow; a) liquid holdup and b) pressure gradient. The values located above each column represent the number of simulation points used to calculate the average error.

3.6.2 Oil and Gas Flow

3.6.2.1 Flow Pattern Visualisation

The flow pattern visualisation for the oil/gas flow was observed to be largely identical to that observed in the air/water flow, and in all cases the observed flow patterns were in agreement with the regimes predicted by the mechanistic model. Stratified wavy flow was only modelled for one case, in the -10° declined pipe and the effect of the higher density in the oil phase was observed for the slug flow cases, where any waves that form on the interface are not as numerous as those observed in any of the air/water cases. Likewise, the interval for slugs to form was noticeably larger when compared to the air/water flows.

3.6.2.2 Liquid Holdup and Pressure Gradient Comparison

With the exception of stratified wavy flow, values of the liquid holdup and pressure gradient for all other flow regimes are presented in Figure 3.12 and Figure 3.13. The presentation of the results shown here follows the same manner as the air/water flow and the complete dataset is presented in tabular format in Appendix A.1.

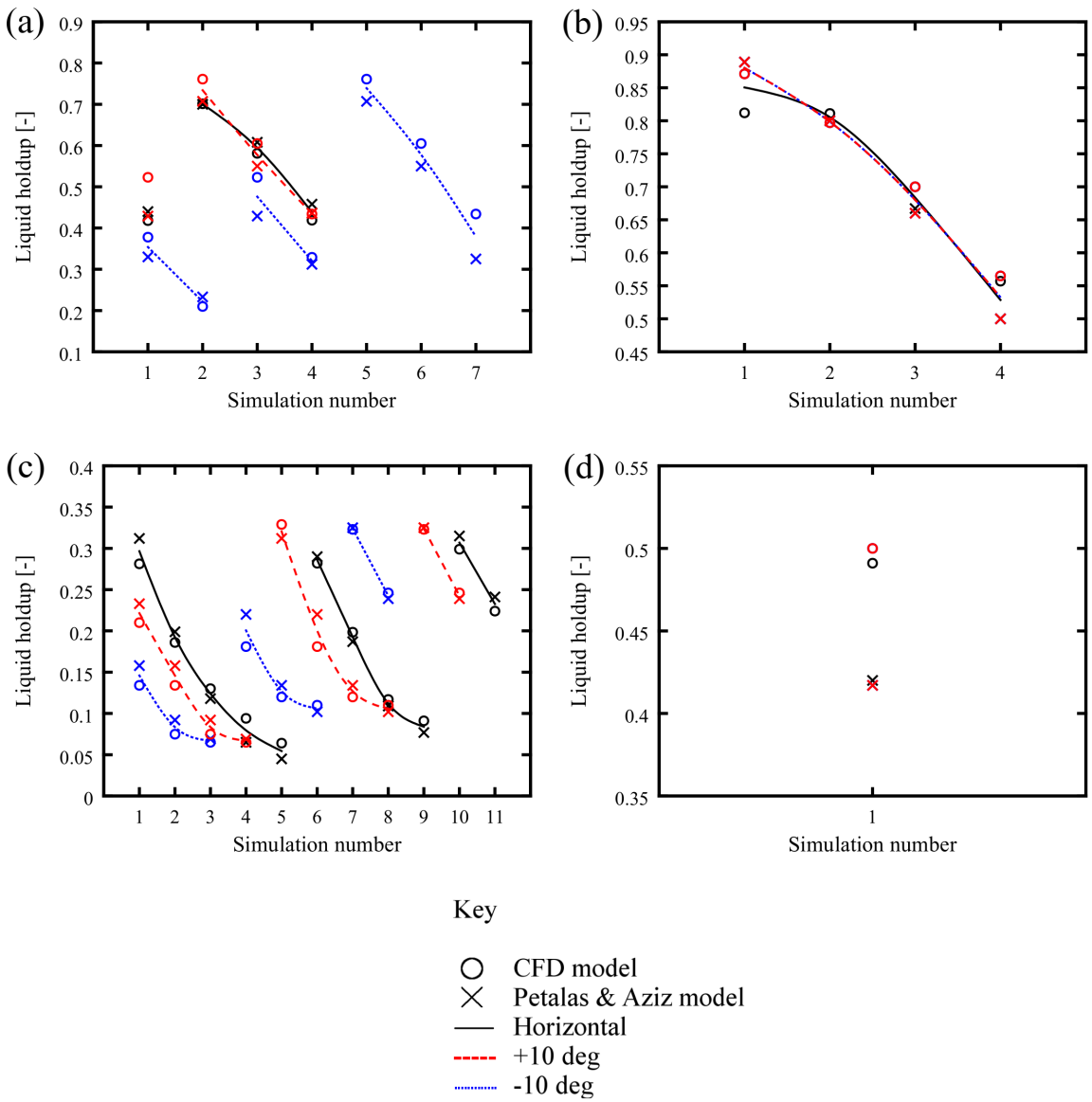


Figure 3.12: The oil/gas liquid holdup results for each flow regime; (a) slug, (b) dispersed bubble, (c) annular mist and (d) froth. The additional lines show the average value between the CFD model and the mechanistic model for simulation points at the same specific liquid velocity.

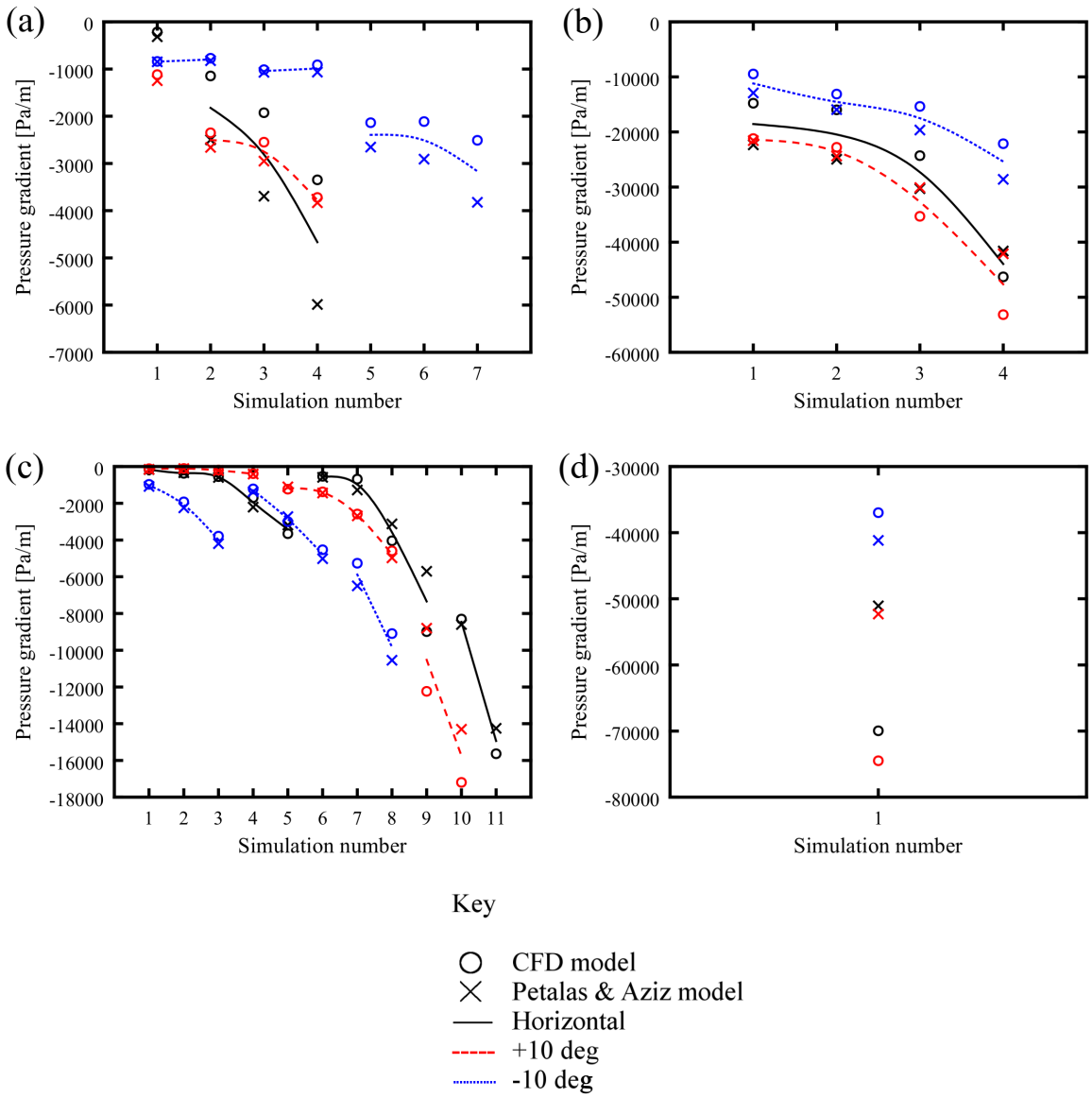


Figure 3.13: The oil/gas pressure gradient results for each flow regime; (a) slug, (b) dispersed bubble, (c) annular mist and (d) froth. The additional lines show the average value between the CFD model and the mechanistic model for simulation points at the same specific liquid velocity.

The error magnitude for the complete dataset for oil/gas flow is presented in Figure 3.14 and the values of the average error magnitude for the liquid holdup and pressure gradient depending on the pipe inclination and flow pattern type are presented in Figure 3.15.

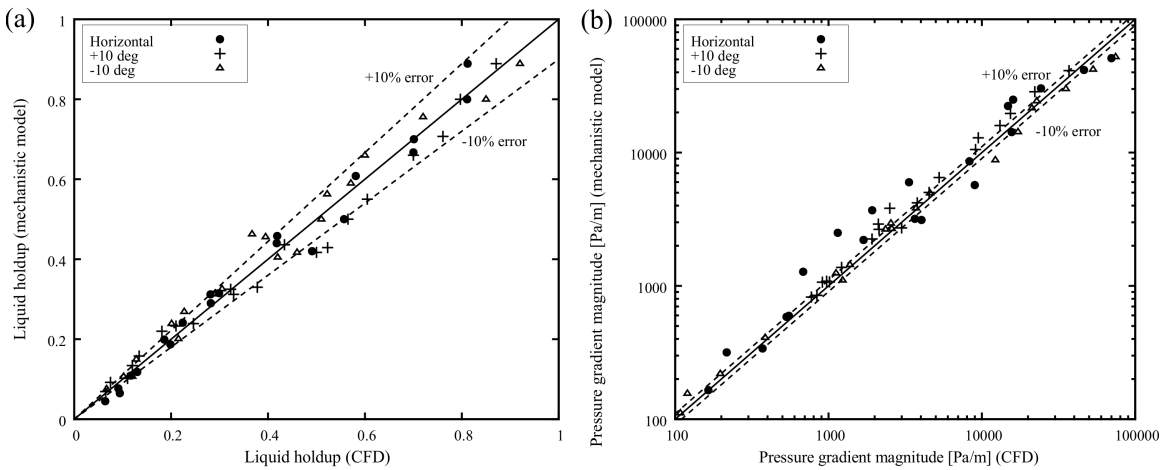


Figure 3.14: The error magnitudes for the complete dataset for oil/gas flow; a) liquid holdup and b) pressure gradient.

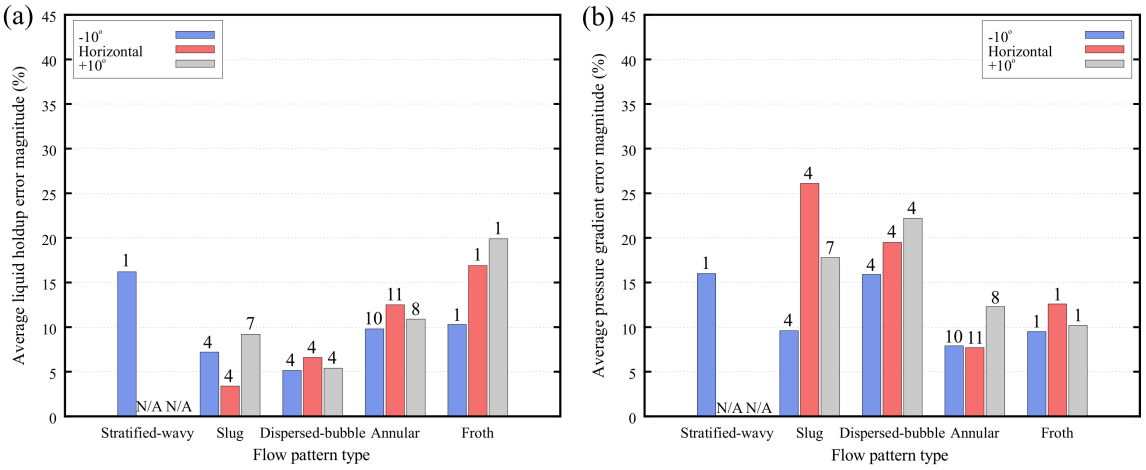


Figure 3.15: The average error magnitude for all flow regimes and pipe inclinations for oil/gas flow; a) liquid holdup and b) pressure gradient. The values located above each column represent the number of simulation points used to calculate the average error.

3.7 Discussion

3.7.1 Air and Water Flow

From the CFD response, it was correctly observed that the effect of changing pipe inclination acted to shift the flow regime boundaries on the flow pattern map. The boundaries of flow patterns such as stratified wavy, elongated bubble, slug and froth flow are particularly influenced by changes in inclination, whereas dispersed bubble and annular mist exhibit little change. The OpenFOAM® simulations correctly depicted the changes in flow regimes when compared to the mechanistic model.

Stratified wavy flow was simulated for 8 cases using the horizontal pipe and 7 cases using the -10° declined pipe. A very good correlation is observed in all cases between the CFD and mechanistic model for the liquid holdup with a mean error of less than 8%, with error margins very similar for both horizontal and declined flow. The pressure gradient comparison shown in Figure 3.9 part (a) has displayed good accuracy for all cases considered with the exception of simulation points conducted near the transition to annular flow. The effect of increasing the inlet gas velocity correctly translated to a reduction of liquid holdup and an increase in pressure gradient.

Slug flow was simulated for a total of 14 cases across all three inclinations. Excellent accuracy of the liquid holdup is observed with an average error less than 6% compared to the mechanistic model with error margins highly consistent across all inclinations. High pressure gradient errors of 30% are observed in the horizontal and declined pipes whereby the increased accuracy evident in the inclined pipe is attributed to the three additional simulation points at a much lower liquid velocity which showed good correlation to the mechanistic model.

Dispersed bubble flow exhibited the most accurate results for the liquid holdup and pressure gradient, with a mean errors less than 5% and 20%, respectively. It is possible that this is due to the high liquid phase fraction of the flow aiding more accurate determination of the pressure gradient due to a more viscous and dense combined flow. In addition, throughout all changes of pipe inclination the dispersed bubble regime is the least affected in terms of pressure gradient variations. It can be seen that the flow pattern map remains roughly the same size for each respective inclination shown in Figure 3.5. As this phase exhibits the least change it is the least affected by gravitational effects, which could explain the lowest mean error recorded.

Annular mist flow was simulated for a total of 9 cases. The results for the liquid holdup presented in Figure 5.11 exhibit the greatest error margin (<17%) compared to all other flow patterns considered. Mean errors in pressure gradient were within 20-40%, depending on the pipe inclination when compared to the mechanistic model. A good consistency is evident for liquid holdup across all inclinations with a variation of less than 1%. The highest pressure gradient errors occurred in the inclined pipe and are attributed to the two additional simulation points near the flow boundary with the froth flow regime.

Froth flow was simulated for 3 cases in the horizontal pipe, 4 cases for the inclined pipe and 3 cases in the declined pipe. The liquid holdup and pressure gradient comparisons were generally good with mean errors less than 11% and 35% observed, respectively. The magnitude of the errors reported may be attributed to the deviation in the flow pattern visualisation observed, whereby it would appear that the mesh density was not fine enough in the longitudinal direction to provide an accurate depiction of the flow.

Overall, it can be seen that liquid holdup results are better for flows involving equal or higher flow rates for the liquid phase when compared to the gas phase, with dispersed bubble flow showing the highest accuracy and annular mist flow being the least accurate. Regardless of flow regime, a good consistency for the liquid holdup error margin is achieved between each inclination. This consistency is not immediately apparent when comparing errors in the pressure gradient, where large fluctuations are observed between each flow regime and inclination, especially for simulation points located near flow regime boundaries. Despite this, it is seen that dispersed bubble flow exhibits the least mean error for the pressure gradient comparison, as in the case for the liquid holdup.

The superior results observed in the dispersed bubble flow regime may possibly be explained by the use of weighted average values for the calculation of density and viscosity in the interFoam solver when solving the Navier-Stokes equations. In particular, the density ratio of the two phases is known to be a contributing factor in introducing erroneous disturbances at the interface, called spurious currents; as suggested in the works of Deshpande et al. [2012], Galusinski and Vigneaux [2008] and Harvie et al. [2006]. It is possible that these spurious currents are more apparent for flows involving a high velocity gas phase, such as annular or froth flow where the subsequent liquid holdup is low. Deshpande et al. [2012] have presented a criterion

that enables control over the growth of the spurious currents by introducing a timestep constraint, although its use has not been explored in this study.

Despite the magnitude of the pressure gradient error margins recorded, the overall results give confidence that interFoam, and its implementation of the modified VOF method using the SST $k-\omega$ turbulence model can still be used to give reasonable solutions to two-phase flow across multiple flow regimes, particularity stratified wavy, slug and dispersed bubble.

3.7.2 Oil and Gas Flow

The visualisation results for the two-phase mixture of oil/gas were observed to be largely in line with that of the air/water flows. The same set of superficial liquid and gas inlet velocities were used as the air/water flow which resulted in an increased amount of the number of annular and slug flow simulations. The average error margins for values of liquid holdup were largely similar to the air/water flow although a much more accurate set of results was observed for the pressure gradient for all flow patterns considered, where the maximum average error margin was approximately 26% for the oil/gas compared to approximately 40% for the air/water flow.

In a manner consistent with the results for the two-phase air/water mixture, the elongated bubble regime exhibited the lowest mean error for the liquid holdup. However, contrary to the air/water results, the mean error for the pressure gradient for both annular and froth flow was lower than dispersed bubble flow. For annular flow, this may be attributed to the greatly increased number of simulations points conducted for the oil/gas mixture compared to air/water mixture (29 vs 9, respectively). Likewise, froth flow was only modelled 3 times for the oil/gas mixture compared to 12 for the air/water and is likely to have been a contributing factor to the difference in the average error magnitudes observed.

From a computational perspective, it was observed that the oil/gas flow exhibited significantly faster computational times, typically in the order of two times or greater. This is directly related to the variable timestep used, whereby the actual timestep is limited by the Courant number (Co), where $\mathrm{Co} = U(\Delta t/\Delta x)$. Here, U is the velocity magnitude, Δt is the timestep and Δx is the longitudinal distance between the mesh elements. As alluded to in the discussion of the air/water results in the previous subsection, if spurious currents were created at the interface of the air/water flow due to the high density ratio, the timestep will reduce to compensate (as Co, Δx are

fixed) and lead to the smaller timesteps observed (oil/gas: $\Delta t \approx 2 \times 10^{-4}$ s, air/water: $\Delta t \approx 1 \times 10^{-5}$ s or lower). Fortunately, for a gas-dominated restart, the density ratio exhibited is expected to be closer to that of the oil/gas system, as such it should help to foster faster computational times for the framework with an increased level of accuracy.

3.7.3 InterDyMFoam

The visual results presented previously in Figure 3.7 show that the mesh density in the lateral direction (X) must be greatly increased in order for the interFoam solver to capture the dispersed bubble and froth flow regimes more accurately. As the intention is to use the model for long flowline applications, the total cell count must be kept to a minimum and further increasing the mesh density in the lateral direction is not currently plausible. Krumrick et al. [2016]; however, have had more success in producing the correct flow patterns for the dispersed bubble and froth flow regimes when simulating multiphase pipe flow using the interDyFoam solver included in OpenFOAM®. Like the interFoam solver, interDyMFoam can solve for two incompressible, isothermal and immiscible fluids using the VOF approach. The difference here lies in the option to use additional mesh motion and mesh topology functions, including adaptive re-meshing, which has been successfully used by Krumrick et al. [2016] to refine the mesh at the interface between the two-phases. Figure 3.16 shows a depiction of an air/water froth flow presented by Krumrick et al. [2016] using adaptive re-meshing, where differences are immediately apparent from the results presented in Figure 3.7 of this study.

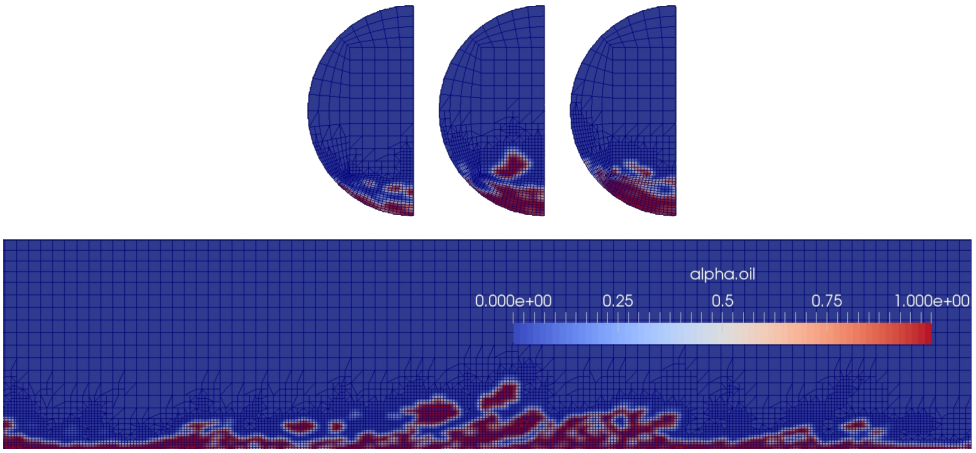


Figure 3.16: The interDyMFoam solver has been used by Krumrick et al. [2016] to generate more accurate visualisations of dispersed bubble and froth flows (adapted from Krumrick et al. [2016]).

Although it is evident that the use of interDyMFoam can produce superior visualisation results when compared to interFoam for dispersed bubble and froth flow, Krumrick et al. [2016] have shown that prediction of the pressure gradient is not necessarily more accurate than the results presented in this study when compared to the results generated by the Petalas and Aziz mechanistic model. Furthermore, the use of adaptive meshing for near wall regions must be handled with care due to fluctuations in the y^+ values that may produce erroneous results depending on the boundary conditions at the wall patch. For this reason, it appears more appropriate to first consider the use of wall functions with the interFoam solver as a means of reducing the total cell count for longer flowline applications. A more detailed discussion of the pressure gradient results from the interDyMFoam solver can be found in Krumrick et al. [2016] and the development a modified modelling strategy for long flowline sections is presented in the following chapter.

3.8 Summary and Conclusions

A total of 120 CFD simulations have been completed for two-phase mixtures of air/water and oil/gas in straight pipe sections at three separate inclinations using the interFoam solver. Comparisons have been made from the simulated values of the liquid holdup and pressure gradient to the Petalas and Aziz mechanistic model with additional flow pattern observations recorded. All of the flow patterns observed in the CFD analysis were consistent with the mechanistic model predictions; however, visualisations of the dispersed bubble and froth flow regimes deviated from observations reported in the literature due to the limited mesh density in the longitudinal direction. For both flow mixtures, a good correlation exists for the values of liquid holdup, with the highest average error occurring in the annular-mist flow regime, while dispersed bubble flow exhibited the best result correlation. The average error in the pressure gradient was observed to be highest in the annular flow regime and least in the dispersed bubble flow regime for the air/water mixture. In contrast to this, the average error for the pressure gradient in the oil/gas system was observed to be the least in the annular flow regime, and the highest for slug flow, potentially attributed to by the increased number of simulation points. Following the results of this study, the following conclusions have been made;

- interFoam as a system wide solver is capable of capturing changes in the various flow patterns with a reasonable degree of accuracy for liquid holdup and

pressure gradient, suggesting the interFoam solver is suitable for the purposes of describing the flow in the optimisation framework.

- The 3-D half-pipe model implemented using a vertical symmetric plane has been successful and will improve the computational efficiency of the constructed framework for longer flowline sections provided that any lateral deviation is minimised.

- Changes in pipe inclination were accurately reflected in the CFD simulations for both fluid combinations considered.

- The mesh density in the longitudinal direction must be finer to provide a more accurate visualisation of dispersed bubble and froth flow.

- Usage of the SST $k - \omega$ turbulence model with a resolved boundary layer is appropriate to describe the flow with the the current set of boundary conditions used.

- Although it has been shown that the interFoam solver is capable of describing the two-phase flow, the CFD modelling strategy must be modified to further reduce the number of mesh elements used in order for the framework to be suitable for longer flowline applications.

Chapter 4

A CFD Modelling Strategy for Long Flowlines

In this chapter, the CFD approach using OpenFOAM® is modified to enable the optimisation framework to be more suitable for longer flowline applications. The excessive cell count limitations of the previous model are addressed by using a scalable wall function approach to drastically reduce the number of cells required. To complement this, the scope of the modified CFD model is also reduced to concentrate on the likely flow patterns observed during a gas-dominated restart operation; stratified and slug flow. In order to validate the modified CFD modelling approach, two separate verification studies are conducted that are derived from the experimental work of Espedal [1998] for stratified flow and Ujang [2003] for slug flow. These experimental studies also feature in a comparative CFD study by Lo and Fiala [2010] using STAR-CD. Subsequently, the results presented by Lo and Fiala [2010] are also compared to using the modified modelling strategy. Following this, an additional case study is then been presented to determine whether the modified model can accurately predict a free water section being extricated from a low-spot pipe section during a gas-sweep, based upon the experimental work of Birvalski [2015]. Descriptions of the rationale behind the modified model, experimental case studies, numerical method used and the comparative results are presented in the following sections. The work presented in this chapter constitutes the culmination of work required to achieve Objective 2 from this study.

4.1 Rationale Behind the Modelling Strategy

While it was shown in Chapter 3 that the interFoam solver is capable of resolving two-phase flows of air/water and oil/gas across multiple flow regimes with reasonable

predictions for the liquid holdup and pressure gradient, the excessive cell count used, equivalent to above 29,000 cells per meter length for the oil/gas system imposes severe restrictions on the usefulness of this modelling technique. In order to further reduce the total cell count and achieve a higher level of practicality, it is proposed to use a wall function approach as the basis of a modified modelling strategy suitable for long flowline sections during restart operations.

Wall functions are used as a means to resolve near-wall flows by mathematically representing the viscous sublayer and buffer layer regions of the boundary layer. Authors such as Craft et al. [2002] have suggested that only one-hundredth of the computational effort is required to implement the wall function approach when compared to the resolved boundary layer approach and as such, the wall function approach is highly preferred for industrial calculations. However, most importantly, Craft et al. [2002] have also mentioned that the track record for the implementation of the wall function schemes is poor, partly due to inappropriate implementation and partly due to inherent limitations within the schemes themselves. A visual comparison of both the resolved boundary layer approach used in Chapter 3 and the wall function approach can be seen in Figure 4.1 parts a) and b), respectively, which has been adapted from the work of Guerrero [2014].

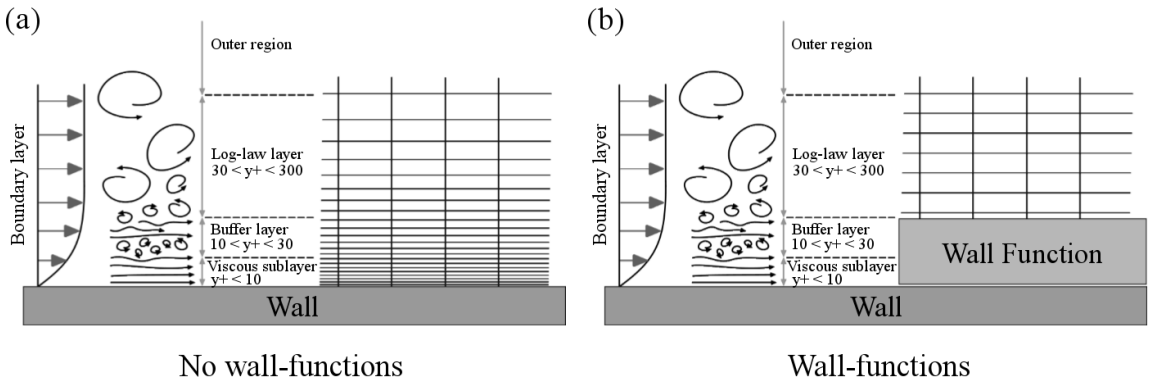


Figure 4.1: Two separate CFD modelling approaches; a) the boundary layer modelling approach and b) the wall function modelling approach (adapted from Guerrero [2014]).

As noticeable in Figure 4.1, the wall function approach will constitute a substantial reduction in the amount of elements required as the fine mesh region closest to the wall does not need to be resolved. However, in contrast to the resolved boundary layer approach, the wall function approach presents some difficulties in implementation when considering flow across the multiple flow regimes that may be expected to occur

in long flowlines. In particular, in order to correctly implement the wall function approach, the height of the first cell normal to the wall must now lie such that the equivalent y^+ value is located within the log-law layer at all times, a region with a y^+ value of between 30 to 300, as shown previously in Figure 4.1. As each different flow pattern exhibits different velocity profiles that can vary by orders of magnitude, it is highly probable that the height of the first cell normal to the wall will need to be changed for each and every simulation in order to compensate. Since the calculation of the required first cell normal to the wall is an iterative process, this method has the potential to become highly problematic when considering its intended use with optimisation studies over entire flowline lengths.

Fortunately, there are two options that can be considered on order to alleviate this restriction. The first option involves the use of a scalable wall function to allow the height of the first cell normal to the wall to be located anywhere such that the equivalent y^+ value is within an expanded range of between 1 to 300. Using OpenFOAM®, scalable wall functions are applicable to the $k-\omega$ type turbulence models and can be implemented by using the kqRWallFunction and the omegaWallFunction as the wall boundary conditions for the turbulent kinetic energy (k) and the specific turbulent rate of dissipation (ω), respectively. The second option is based upon the observation that the most likely flow regimes experienced during a gas-dominated restart will be a combination of stratified, slug and dispersed bubble flow. It may be possible to eliminate the necessity to model froth and annular flow outright as the high liquid and gas velocities required to experience froth and annular flow make it unlikely to occur in a gas-dominated restart scenario. While it is recognised that annular flow occurs at lower velocities in larger diameter pipes, as seen in Hanratty [2013], the amount of accumulated water held in the low-spots prior to the restart operation will be a limiting factor in restricting the existence of a sustained annular flow pattern.

In contrast, stratified flow will be present in all restart operations due to the presence of a stagnant water phase that accumulates at the low-spots of the flowline during shut-down, and accurate modelling of this phase is essential. Likewise, the presence of a free-water phase in a flowline will undoubtedly result in some form of slugging during a restart operation. It is likely that the slugging experienced during a restart operation will be a combination of hydrodynamic, terrain-induced and operationally induced, as described by Bai and Bai [2005], and accurate modelling of the flow pattern is also of paramount importance. Lastly, dispersed bubble flow may exist if a slow moving gas phase passes through a static water accumulation,

which will most likely occur in upward segments of undulating flowlines, or in other steep sections of the flowline such as risers.

A problem arises here, however, as despite the high accuracy of the pressure gradient and liquid holdup results for the dispersed bubble phase when compared to the mechanistic model in Chapter 3, the inability to accurately capture the behaviour of the dispersed bubbles will impose a substantial limitation on the accuracy of the model to estimate the amount of hydrate growth. This limitation arises due to the use of the interfacial area between the gas and liquid phases to calculate the hydrate growth rate, which is used in both hydrate models by Skovborg and Rasmussen [1994] and Turner et al. [2005], as mentioned previously in Section 2.7. Fortunately, dispersed bubble flow will only be prevalent if the gas velocity is low enough such that the two-phase gas/water interface is not pushed down to become stratified or form into a liquid slug, as can be seen in the experimental work of Leporcher et al. [2002]. As mentioned previously in Chapter 1, Leporcher et al. [2002] have observed that a high velocity gas phase leads to a reduced risk of hydrate blockages if a certain amount of accumulated water is swept from the low-spot and redistributed downstream. Due to this observation, the dispersed bubble flow regime has been omitted from any analysis with the modified model. Despite the inability of the modified model to capture hydrate growth for droplets, it is intended that the modular design of the framework will still allow the optimisation strategy to be applicable for any future studies if a more accurate script-based modelling tool is available for the inclusion of dispersed bubble flow.

In this study, it is thus proposed to use the scalable wall function approach with an emphasis on stratified and slug flow. Without the explicit need to produce a CFD model to capture all flow patterns, the chances for the model to produce accurate and usable results have increased at a greatly reduced computational expense. In order to validate the modified modelling strategy, three case studies have been selected in addition to the Petalas and Aziz mechanistic model used in the previous chapter. These three case studies are based upon the experimental work of Espedal [1998], Ujang [2003] and Birvalski [2015]. The first two of these studies encompass stratified and slug flow, respectively, while the last case study involves a gas-sweep from a pipe featuring a single low-spot. A description for each of these studies is presented in the following section.

4.2 Description of the Case Studies

All three of the experimental studies by Espedal [1998], Ujang [2003] and Birvalski [2015] involve a two-phase flow combination of air and water flowing through perspex pipes. Due to slight variations in the diameter and length used in each of the experimental studies, a separate CFD model has been constructed for each analysis. The length of each CFD model constructed is considerably longer than the two original models presented in Chapter 3 and is a useful means to assess the performance of the modified models for longer pipe sections. The geometry used for each case study is presented in Table 4.1, along with a comparison to the original CFD models used in the air/water and oil/gas studies presented in Chapter 3.

Table 4.1: A summary of the geometry used in each of the additional CFD studies compared to the original studies presented in Chapter 3.

CFD study	Diameter [mm]	Length [m]	L/D
Original air/water model from Chapter 3	50.8	4	76.9
Original oil/gas model from Chapter 3	156.9	12	76.5
Stratified flow comparison with Espedal [1998]	60	18	300
Slug flow comparison with Ujang [2003]	77.9	37	560
Gas-sweep comparison with Birvalski [2015]	50.8	40	785

4.2.1 Stratified Flow Comparison with Espedal [1998]

Espedal [1998] performed an experimental study using a two-phase mixture of air/water in a 60 mm diameter horizontal pipe of 18 meter length for stratified flow and produced a comprehensive dataset showing values of the pressure gradient (dP/dx) and the liquid height ratio (h_L/D) for a wide range of superficial velocities at a number of near-horizontal pipe inclinations. For the purposes of a comparative study, the pressure gradient and liquid height results from four separate tests that were carried out at an inclination of -0.5° from the horizontal have been selected for comparison. These four particular experiments also feature in a comparative study by Lo and Fiala [2010] using STAR-CD, the results of which have also been compared to in this study, along with the Petalas and Aziz mechanistic model predictions. The associated test numbers, specific velocities of air (U_{SG}) and water (U_{SL}) and the pressure gradient and liquid height ratio results for the four tests conducted by Espedal [1998] are presented in Table 4.2, and the system parameters used in this analysis are presented in Table 4.3. It should be noted that the liquid height ratio used by Espedal [1998]

refers to the ratio of the height of the liquid phase within the pipe taken at the pipe mid section to the pipe diameter and is not the liquid holdup used in the previous chapter.

Table 4.2: A small selection of the original dataset presented by Espedal [1998] used to validate the modified modelling strategy for stratified flow. All four tests feature a two-phase mixture of air/water at an inclination of -0.5° from the horizontal.

Test number	U_{SG} [m/s]	U_{SL} [m/s]	dP/dx [Pa/m]	h_L/D [-]
FN0139	2.984	0.1190	4.94	0.2261
FN0140	2.982	0.1776	6.33	0.3003
FN0141	2.976	0.2640	9.21	0.3921
FN0503	2.975	0.3159	11.56	0.4473

Table 4.3: The system parameters used in the comparison with Espedal [1998].

Parameter	Value
Gas density	1.19 [kg/m^3]
Liquid density	999 [kg/m^3]
Gas viscosity	1.83e-05 [Pa·s]
Liquid viscosity	9.7e-04 [Pa·s]
Interfacial tension	0.0728 [N/m]

4.2.2 Slug Flow Comparison with Ujang [2003]

Ujang [2003] has experimentally studied the initiation and the subsequent evolution of hydrodynamic slugs using a two-phase flow of air and water in a 77.9 mm diameter horizontal pipe section of 37 meter length. From the results obtained, Lo and Fiala [2010] have compared the slug frequency measured by the liquid height at multiple locations along the length of the pipe and the average slug length along the pipe at a particular flow (U_{SG} = 4.64 m/s, U_{SL} = 0.611 m/s, 1 atm). In this study, the results of both Ujang [2003] and Lo and Fiala [2010] at this particular flow have been compared to in order to assess the capability of the modified model to capture slug flow. As no data could be found for the pressure gradient and liquid holdup for this experiment, the results have been compared to the estimations generated from

the Petalas and Aziz mechanistic model. The same system parameters used in the comparison with Espedal [1998] are used for this comparison.

4.2.3 Gas-sweep Comparison with Birvalski [2015]

Birvalski [2015] has performed an experimental study on a low-spot pipe bend to observe the response of an initial accumulated water zone to seven different inlet velocities of air, where the amount of water in the low-spot varied in four increments from a minimum of 50 ml to a maximum of 300 ml. Upon initiation of the air flow, the length of the water layer in the downward and upward sections of the low-spot was measured at steady state conditions. A depiction of the experimental setup used by Birvalski [2015] is shown in Figure 4.2 part a), along with an illustration of the initial liquid pool and the subsequent response of the liquid layer due to an incoming gas flow in part b).

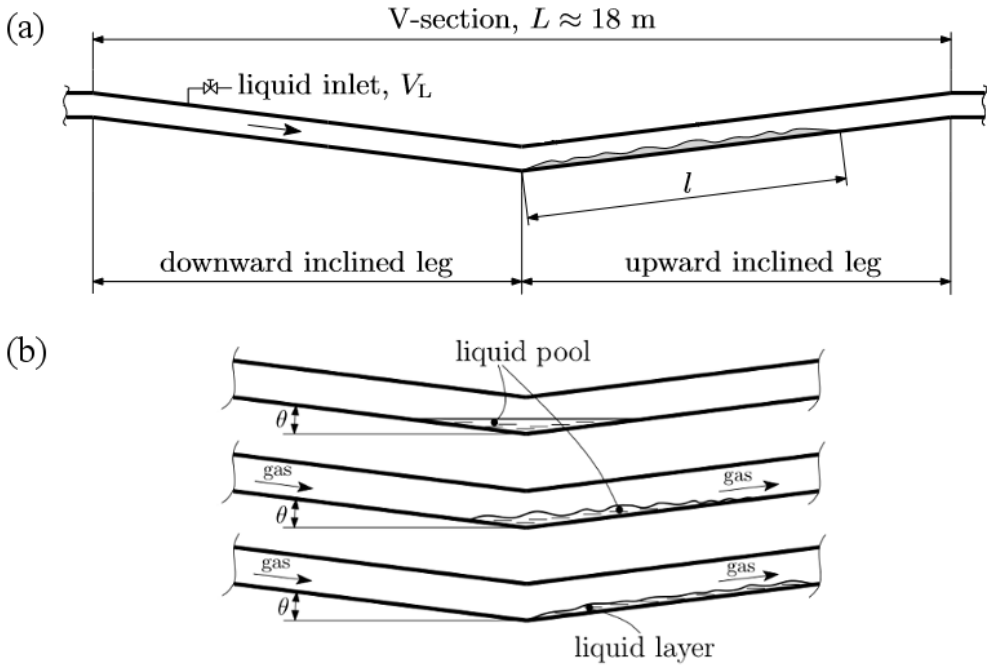


Figure 4.2: a) The experimental setup used by Birvalski [2015] for the gas-sweep of an accumulated water section in a 50.8 mm diameter pipe with a single low-spot bend, b) the response of the liquid layer due to the incoming air flow (both images adapted from Birvalski [2015]).

The horizontal length of the low-spot section was approximately 18 meters, which was followed by a 30 meter section of horizontal pipe connected to an open vessel. The CFD model used in this study features the same low-spot geometry, although includes

an additional 10 m long entrance length to ensure the gas flow is fully developed prior to the first downward bend and a 10 m long exit length, which has been shortened considerably as the elongated water film in the upward pipe segment is not intended to pass through the exit. As shown in Figure 4.2 part b), the downward length of the water layer is compressed upon commencement of the gas flow and the upward length becomes elongated. In this study, both the downward and upward lengths of the water layer from an initial volume of 300 ml have been compared at the seven separate inlet air velocities present in the experimental study of Birvalski [2015]. Two additional air flow velocities (1 m/s, 1.5 m/s) were added to assess the response of the modified model at the onset of contraction and elongation of the liquid film layer in the downward and upward pipe segments, respectively. The full system parameters used in this study are shown in Table 4.4 and have been chosen in order to be consistent with the values presented by Birvalski [2015].

Table 4.4: The system parameters used for the comparison with Birvalski [2015].

Parameter	Value
Pipe inclination	1.7°
Gas density	1.19 [kg/m^3]
Liquid density	997 [kg/m^3]
Gas viscosity	1.83e-05 [Pa·s]
Liquid viscosity	1e-03 [Pa·s]
Interfacial tension	0.0728 [N/m]

4.3 Numerical Method

The updated version of OpenFOAM®, version 2.3.x was used to simulate all two-phase flows present in this chapter using the interFoam solver. The implementation of the interFoam solver and the associated numerical schemes used have remained unchanged from the description presented previously in Chapter 3. Similarly, the SST $k-\omega$ turbulence model has been used for all simulations. This was because, complimentary to the results presented by Lo and Fiala [2010] for the air/water stratified flow comparison with Espedal [1998], Lo and Tomasello [2010] have shown that the standard SST $k-\omega$ produced more a accurate set of pressure gradient results and comparable liquid height results when compared to both the standard $k-\omega$ and standard $k-\epsilon$ turbulence models. The cross-section used by Lo and Tomasello [2010]

to evaluate the turbulence models is similar to that of the modified model used in this study, which is shown in Figure 4.3.

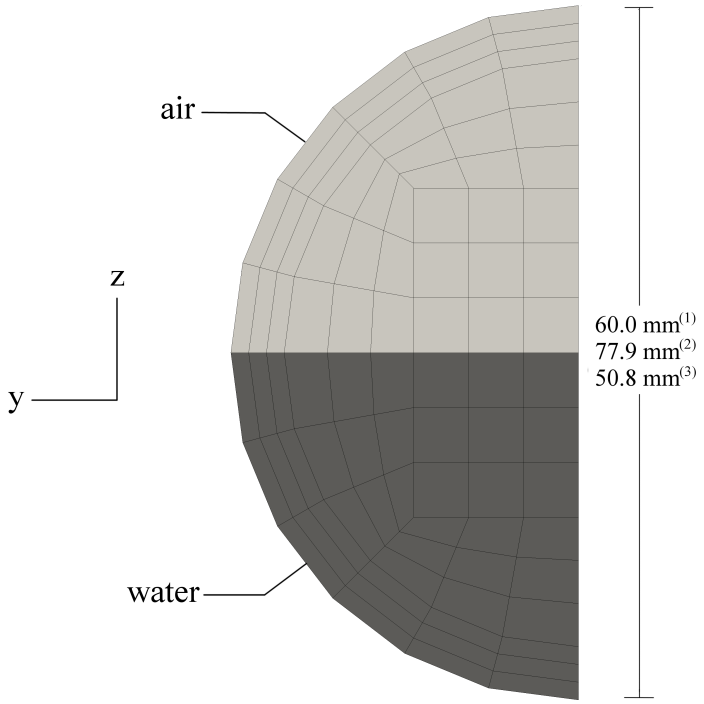


Figure 4.3: The modified mesh cross-section and the respective diameters used for the comparison of to the experimental datasets of (1) Espedal [1998], (2) Ujang [2003] and (3) Birvalski [2015]. An equal fraction for air and water is shown at the inlet plane which is used for the stratified and slug flow comparisons.

SALOME 7.4.0 has been used to generate the meshes used in all simulations. The generation of the mesh used for the stratified and slug flow comparisons is similar to the original 3-D half-pipe model presented in the previous chapter, where the only difference arises due to the reduced number of cross section elements used. For the gas-sweep comparison with Birvalski [2015], the strategy to ensure an adequate hexahedral mesh over the entire model is slightly more complex and is presented in greater detail in Appendix E.3, together with a procedure to create the accumulated water region using the setFields utility in OpenFOAM® prior to commencement of the simulation. Additionally, the SALOME code used for the gas-sweep comparison is shown in Appendix B.2. For the comparisons of stratified and slug flow, initiation of the simulations follows the procedure outlined in Chapter 3, whereby the pipe was initially filled with water and equal fractions of air and water were introduced at the inlet, as shown previously in Figure 4.3.

In order to adapt the boundary conditions to enable a scalable wall function approach using OpenFOAM®, the boundary condition used for the turbulent kinetic energy (k) at the wall patch is simply changed from a fixed value to the kqRWallFunction and given an initial value of zero. For implementation of the omegaWallFunction boundary condition, the initial value for the specific turbulent dissipation rate (ω) was calculated from the standard formula;

$$\omega = 10\frac{6\nu}{\beta y^2} \tag{4.1}$$

Where the constant, β has a value of 0.075, ν is the kinematic viscosity of the fluid adjacent to the wall and y is the distance from the first cell center normal to the wall. As all simulations are based on experimental data using pipe materials made from perspex, the nutkWallFunction is used as opposed to the nutkRoughWallFunction used in Chapter 3. Table 4.5 shows the complete list of updated boundary conditions used in the modified model.

Table 4.5: The updated OpenFOAM® boundary conditions used for each of the modified models.

Parameter	Inlet	Outlet	Wall
U	fixedValue	zeroGradient	fixedValue uniform 0
p_rgh	zeroGradient	fixedValue uniform 0	zeroGradient
k	fixedValue	zeroGradient	kqRWallFunction
nut	Calculated uniform 0	Calculated uniform 0	nutkWallFunction
omega	fixedValue	zeroGradient	omegaWallFunction

4.4 Results

For each of the stratified, slug and gas-sweep comparative studies, five separate mesh resolutions have been used. These mesh resolutions are all based upon the modified mesh cross-section shown previously in Figure 4.3 and are defined in terms of the number of cells per meter length of the model, where the longitudinal spacing of each mesh element is held constant. It is important to note that the mesh resolution presented for each of the comparative studies is dependant on the specific pipe diameter used and adjustment is necessary if any comparison between the studies is to be made. For this reason, and in order to provide a consistent comparison across all three experimental studies, the mesh resolution has been defined in terms of a mesh resolution parameter, Υ, equivalent to the mesh resolution multiplied by the specific

pipe diameter used in each of the three case studies shown previously in Table 4.1, i.e;

$$\Upsilon = 0.985 \times mesh\ resolution\,[cells/m] \times D\,[m] \tag{4.2}$$

Table 4.6 shows the five separate mesh resolution parameters used in each of the experimental comparisons, together with the equivalent mesh resolutions generated in cells/m, which have been rounded to the nearest whole number in order to achieve a complete mesh.

Table 4.6: The values of the mesh resolution parameter, Υ and equivalent mesh resolutions used for each experimental comparison.

Υ	Equivalent mesh resolution [cells/m]		
[cells]	Espedal [1998]	Ujang [2003]	Birvalski [2015]
6.25	106	82	125
12.5	212	163	250
25	423	326	500
50	847	652	1000
100	1693	1304	2000

The ranges for the mesh resolution parameter start from a minimum of 6.25 until a maximum of 100, which equates to a substantial reduction when compared to a mesh resolution parameter slightly in excess of 4,670 (equivalent to a mesh resolution of 29k cells per meter) used in the original model from Chapter 3 for the oil/gas flow. Even though all equivalent mesh resolutions considered are far below the original resolution used, a limit of 100 was set for the mesh resolution parameter due to limitations on the processing power available for the purpose of this study, particularly when considering the intent to showcase an applied example using longer flowline sections. For each simulation, the y^+ value was checked in the same manner as the previous chapter and as described in Appendix E.2.

4.4.1 Stratified Flow Comparison

Using the dataset presented previously in Table 4.2, four simulations have been carried out and values of the liquid height and pressure gradient have been simulated using the modified model at five separate mesh resolutions. These simulated results are presented in Figure 4.4 along with both the experimental results of Espedal [1998] and the numerical results of Lo and Fiala [2010]. Error magnitudes of $\pm 10\%$ have been attached to both the experimental pressure gradient and liquid height results of

Espedal [1998] and Figure 4.4 part a) additionally shows the results from the Petalas and Aziz mechanistic model. For each of the four cases considered, it was observed that all flow patterns were stratified wavy, with very slight fluctuations in wave height ($< 10\%$) observable at the phase interface that became more pronounced with each of the four incremental increases in the velocity of the liquid phase.

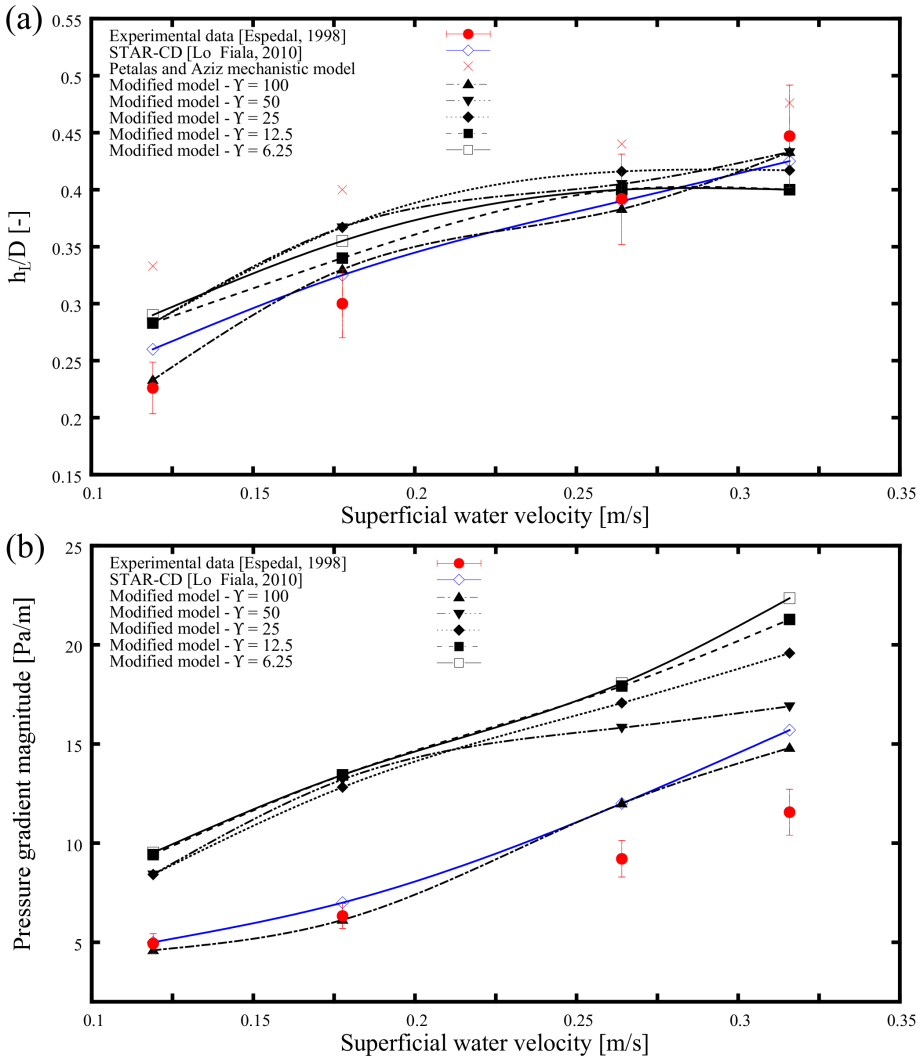


Figure 4.4: The results obtained from the CFD simulations compared to the original experimental dataset from Espedal [1998], with additional results taken from Lo and Fiala [2010] and the Petalas and Aziz mechanistic model; a) liquid height ratio and b) pressure gradient magnitude.

From Figure 4.4 part a), a reasonable correlation can been observed for the comparison of the liquid height results with the experimental data from Espedal [1998] for all mesh resolutions considered. An equivalent mesh resolution of 1693 cells per

meter ($\Upsilon = 100$) has produced results within an error magnitude of 10% for all four of the tests and this resolution was the most comparable to results provided by Lo and Fiala [2010]. All other resolutions have produced error magnitudes within 10% for the two tests at superficial water velocities higher than 0.25 m/s and within 20% for the two tests conducted at liquid velocities lower than this value. The liquid height predictions from the mechanistic model were slightly higher than the simulated and experimental results for each of the different superficial liquid velocities considered.

In contrast to the liquid height comparison, a large deviation can be seen for the pressure gradient results for all equivalent mesh resolutions less than 1693 cells per meter. Similarly, the highest resolution considered was required for the results to be comparable to the results provided by Lo and Fiala [2010], although a significant deviation can be observed from both these results when compared to the experimental data for superficial water velocities above 0.25 m/s. The remaining mesh resolutions exhibit consistent behaviour for the three superficial water velocities less than 0.3 m/s. Above this value, the effect of the mesh resolution is evident, as the accuracy of the simulated values for the pressure gradient increase notably with each incremental increase of the mesh resolution.

The pressure gradient results obtained from the Petalas and Aziz mechanistic model have been omitted from Figure 4.4 part b). The reason for this is that the results obtained from the mechanistic model were exceptionally higher than both the CFD and experimental data, regardless of the value used for the pipe roughness. The specific pressure gradient magnitude results obtained from the mechanistic model were 74.8 Pa/m, 95.5 Pa/m, 152.8 Pa/m and 206.6 Pa/m for the four superficial liquid velocities of 0.1190 m/s, 0.1776 m/s, 0.2460 m/s and 0.3159 m/s, respectively. These results were, however, consistent to the horizontal air/water results presented in the previous chapter. In order for the modified CFD model to achieve the results presented in Figure 4.4 part b), the nutkWallFunction was used as the boundary condition for the turbulence viscosity at the wall patch. This boundary condition is essentially a smooth-wall version of the nutkRoughWallFunction, whereby identical values for the pressure gradient can be obtained in both boundary conditions by setting zero values for the sand grain roughness height (Ks) and roughness constant (Cs) in the nutkRoughWallFunction.

4.4.2 Slug Flow Comparison

4.4.2.1 Visualisation

Using the modified CFD model at the particular two-phase flow considered for the comparison of slug flow ($U_{SG} = 4.64$ m/s, $U_{SL} = 0.611$ m/s), it was observed that the wave height did not actually reach the top of the pipe for any of the mesh resolutions used, suggesting that the flow is still in the stratified wavy flow regime. An example of the visualisation results obtained from the modified CFD model for a mesh resolution parameter of 100 is presented in Figure 4.5. Here, the flow is shown in three separate increments of 5 seconds, starting from an elapsed time of 70 seconds, where the arrow represents the same approximate location on the wave front at each time frame and the lower scale represents the length along the pipe from the entrance in meters.

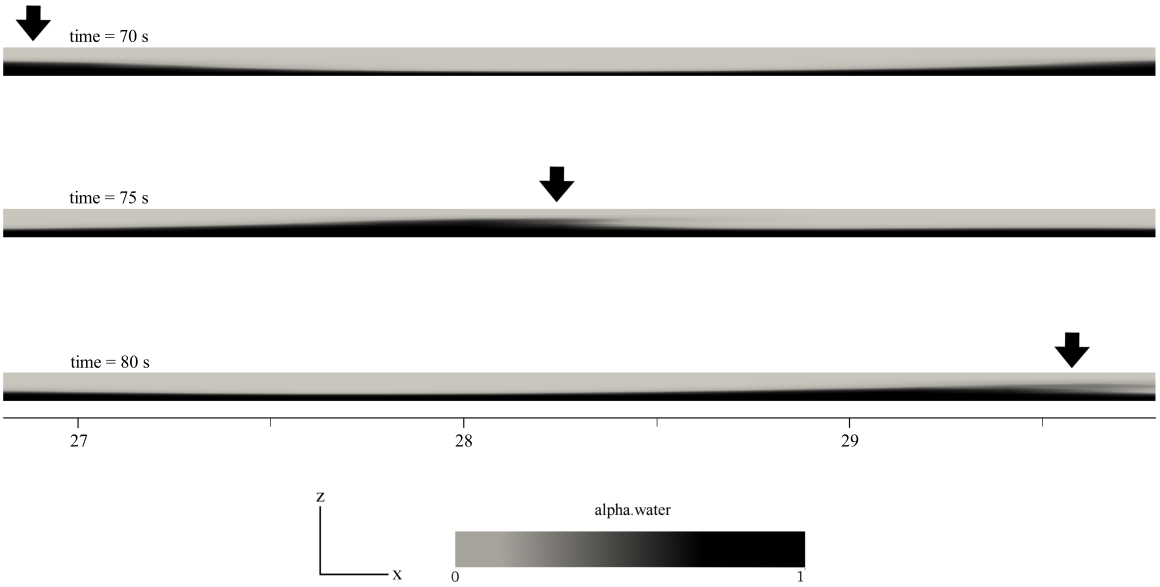


Figure 4.5: Visualisation of slug flow in a 77.9 mm diameter pipe using the modified model at a particular flow ($U_{SG} = 4.64$ m/s, $U_{SL} = 0.611$ m/s).

It is evident from Figure 4.5 that the wave height is not reaching the top of the pipe to create a slug. As the top of the wave front nears the upper wall, the gas velocity in the vicinity increases and causes the upper liquid layer to detach and flow downstream. This would also appear to be the case in the results presented by Lo and Fiala [2010], as evidenced by the data plots for the liquid height ratios at selected vertical planes presented in Appendix C.2. Coincidentally, a comparison with the Petalas and Aziz mechanistic model yields that stratified wavy flow is also predicted and an inspection of the horizontal flow regime map generated by the mechanistic

model and presented in Figure 3.5 of the previous chapter will suggest that the flow considered is most likely in the transition region between stratified and slug flow where discrepancies between the simulated and experimental data are to be expected. It is important to note that a direct comparison from Figure 3.5 cannot be made due to the difference in pipe diameters between the original air/water model to the modified model used here. Nevertheless, a comparison of the slug/wave frequency presented in the following subsection suggests that the development of slug flow is still fairly consistent between the simulated and experimental data.

As no other data for the liquid holdup and pressure gradient relating to this particular test could be found, a comparison of the simulated values for the liquid holdup and pressure gradient to the mechanistic model for each of the mesh resolutions considered is shown in Table 4.7 with associated error magnitudes. These error magnitudes have been derived from a comparison to the predicted values for the liquid holdup and pressure gradient generated from the Petalas and Aziz mechanistic model, which are 0.407 [-] and -724 [Pa/m] respectively.

Table 4.7: The simulated values of liquid holdup and pressure gradient depending on different values of the mesh resolution parameter. Also shown are the respective error magnitudes when compared to the Petalas and Aziz mechanistic model estimations (liquid holdup = 0.407 [-], pressure gradient = -724 [Pa/m]).

Υ [cells]	Liquid holdup [-]	Pressure gradient [Pa/m]	Liquid holdup error magnitude [%]	Pressure gradient error magnitude [%]
6.25	0.478	-488	17.4	48.4
12.5	0.454	-550	11.5	31.6
25	0.432	-645	6.1	12.2
50	0.420	-702	3.2	3.1
100	0.418	-764	2.7	5.2

4.4.2.2 Slug Frequency

The original experimental measurements presented by Ujang [2003] are shown in Figure 4.6. This figure shows the variations in the ratio of the liquid height to the diameter at six selected sampling locations along the length of the pipe over a time period of 50 seconds and is equivalent to a representation of the slug frequency. The sampling locations are defined as the longitudinal distance from the pipe entrance and are listed on the right side of Figure 4.6. In this study, the simulated values of the liquid height at the selected locations have been accessed using Paraview 5.0.0 to plot

the global variables over time using vertical lines to extract data at the symmetric plane of the 3-D half-pipe model (i.e, Y = 0). The specific sampling locations have been selected to correspond directly to the methodology used by Lo and Fiala [2010] and it is noted that the original results from Ujang [2003] feature a total of 14 sampling locations which have not been reproduced here in full.

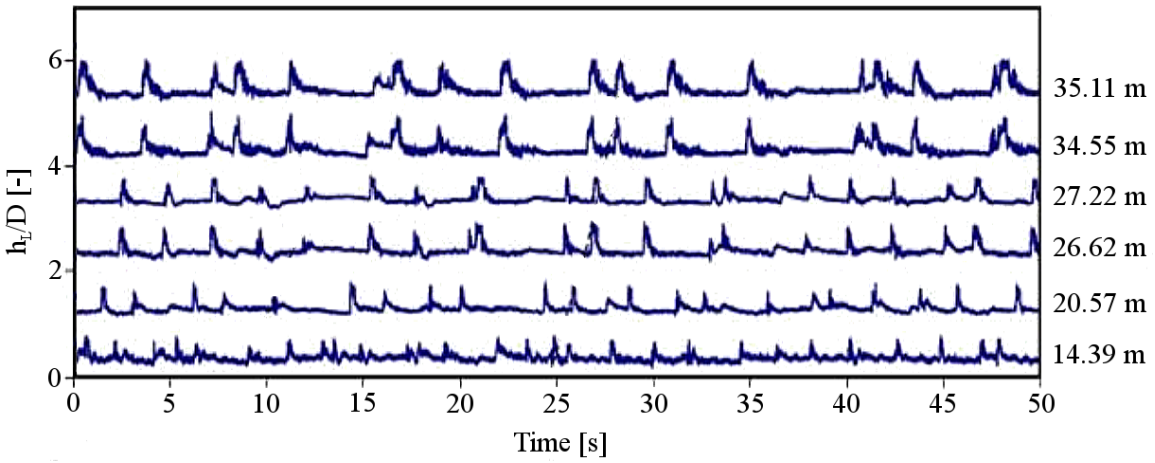


Figure 4.6: The original measurements of the transient slug height measured at multiple distances along the length of the 77.9 mm diameter pipe presented by Ujang [2003] for U_{SG} = 4.64 m/s and U_{SL} = 0.611 m/s (image adapted from Ujang [2003]).

The transient response for the simulated ratios of the liquid height to pipe diameter generated using OpenFOAM® are shown in Figure 4.7 for a mesh resolution of 1304 cells per meter (Υ = 100). It was observed that the two highest mesh resolutions considered produced almost identical results, although greater deviations in the slug height and frequency can be seen for the lower mesh resolutions. For the purpose of clarity, only the highest mesh resolution has been shown in this chapter, where the liquid height results for all other mesh resolutions are presented in Appendix C.1 for comparative purposes. In Figure 4.7, the liquid height was simulated over the same 50 second time period as Ujang [2003]; however, in order to be consistent to the methodology used by Lo and Fiala [2010], this time period begins at an elapsed simulation time of 70 seconds after the commencement of the simulation. The subsequent results presented by Lo and Fiala [2010] are fairly consistent with both the original experimental data and the simulated results presented in this study and have not been reproduced in this chapter; however, these results can be seen in Appendix C.2 for comparative purposes. In order to quantitatively compare the results presented by all three studies for slug flow, the number of water height fluctuations, equivalent

to number of peaks observed over the elapsed time period of 50 seconds have been recorded for each study and are presented in Table 4.8.

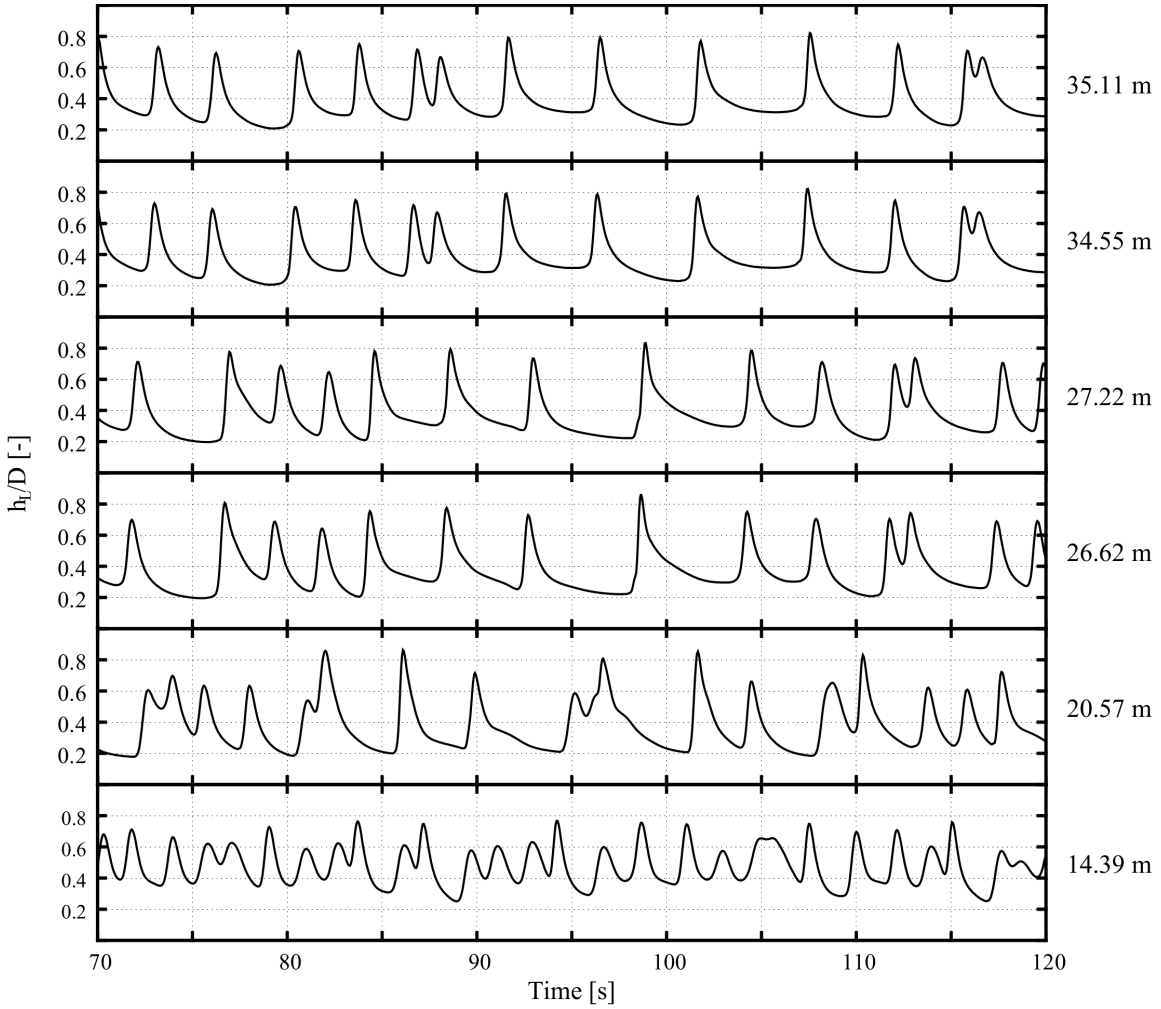


Figure 4.7: The simulated slug flow heights in a 77.9 mm diameter pipe at selected sampling locations using the modified CFD model over a 50 second time period (U_{SG} = 4.64 m/s and U_{SL} = 0.611 m/s).

Table 4.8: A comparison of the number of slug height fluctuations for $U_{SG} = 4.64$ m/s, $U_{SL} = 0.611$ m/s at selected sampling locations for all mesh resolutions considered.

Longitudinal distance [m]	Experimental (Ujang [2003])	Star-CD (Lo and Fiala [2010])	Modified model $\Upsilon = 100$	$\Upsilon = 50$	$\Upsilon = 25$	$\Upsilon = 12.5$	$\Upsilon = 6.25$
35.11	17	13	13	13	6	4	5
34.55	17	13	13	13	6	4	5
27.22	19	13	14	14	7	6	6
26.62	20	13	14	14	7	6	6
20.57	22	14	17	17	9	7	7
14.39	27	18	26	26	12	12	6

4.4.2.3 Slug Length Development

A comparison of the average slug length development for all mesh resolutions is presented in Figure 4.8 with the data from both Ujang [2003] and Lo and Fiala [2010]. Here, the slug length was time-averaged over the same 50 second period used earlier, beginning at an elapsed time of 70 seconds. The representation shown in Figure 4.8 is used to observe the growth rate of the waves generated at the interface throughout the pipe length.

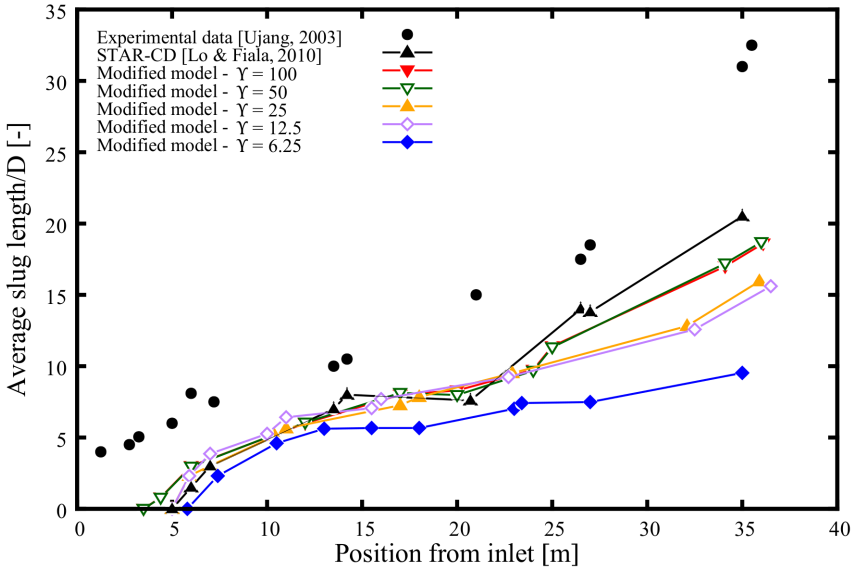


Figure 4.8: The comparative values for the ratio of the average slug length to the pipe diameter (77.9 mm) depending on the position from the inlet (U_{SG} = 4.64 m/s and U_{SL} = 0.611 m/s).

Consistent with the results presented by Lo and Fiala [2010], an initial 6 meters of pipe length is needed for the waves to develop for the equivalent mesh resolution of 82 cells per m ($\Upsilon = 6.25$), which was reduced to approximately 3.5 meters of pipe for the maximum equivalent mesh resolution of 1304 cells per meter ($\Upsilon = 100$). With this taken into account, the wave growth rates for the two highest mesh resolutions were observed to be in good agreement with both datasets from Ujang [2003] and Lo and Fiala [2010] and the wave length consistently increases along the length of the pipe. The following two mesh resolutions ($\Upsilon = 25, 12.5$) still show a fairly good agreement to the experimental data although discrepancies are noticeable towards the pipe exit. Significant deviations can be seen when using the lowest mesh resolution ($\Upsilon = 6.25$) and the trend is not consistent to the other mesh resolutions considered.

4.4.3 Gas-Sweep Comparison

The simulated values of the downward and upward liquid film lengths are presented in Figure 4.9, together with the experimental data from Birvalski [2015] featuring associated ±10% error magnitudes. The results presented in Figure 4.9 correspond to an initial water volume of 300 ml and nine separate inlet air velocities have been used ranging from a minimum of 1 m/s to a maximum of 8.7 m/s. These air velocities have been selected to correspond directly with the results presented by Birvalski [2015] with the exception of the first two, which were not present in the experimental study and have been added to assess how the modified model performs at the onset of compression in the downward liquid length.

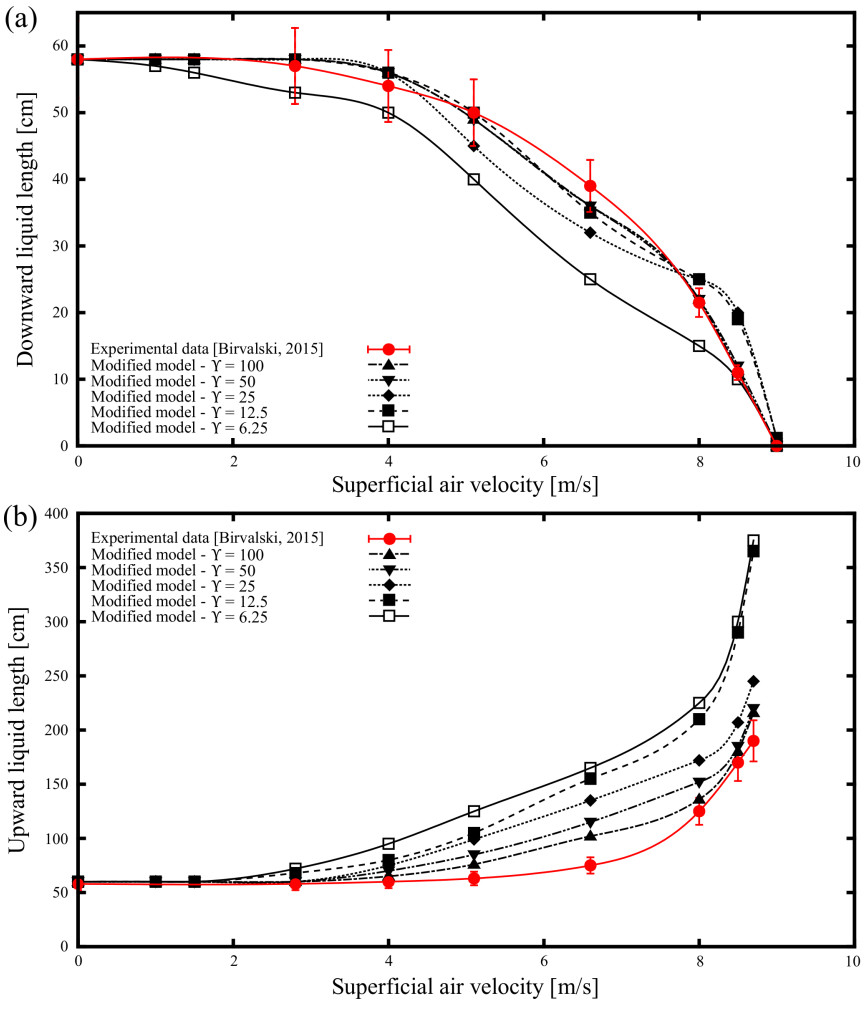


Figure 4.9: The liquid film results obtained from the modified CFD model compared to the experimental results presented by Birvalski [2015] with associated ±10% error magnitudes shown; a) downward liquid length and b) upward liquid length.

For all air velocities considered, the flow was characterised by a slow moving stratified liquid film that was initially pushed up the upward pipe section before settling down again due to the presence of gravity. The simulations were run until a steady state condition was reached before measurements of the downward and upward liquid film lengths were made. As shown in Figure 4.9, the highest mesh resolution considered ($\Upsilon = 100$) has produced the most accurate set of results, with all but two simulations from a total of eighteen having produced liquid film length results within an error magnitude of 10%. The two particular cases that exceeded this error magnitude occurred at superficial air velocities of 5.1 m/s and 6.6 m/s on the upward liquid length segment. The error magnitudes recorded for these two particular tests were 20.6% and 36.0%, respectively.

4.4.4 Computational Time

Each incremental mesh resolution has been compared to both the computational time required to simulate 100 s of elapsed simulation time and the error magnitude for test number FN0503 from the experimental dataset of Espedal [1998]. These results are shown in Figure 4.10. The results for the computational time shown in this figure were generated using a single quad-core i7-4970 processor using parallel processing.

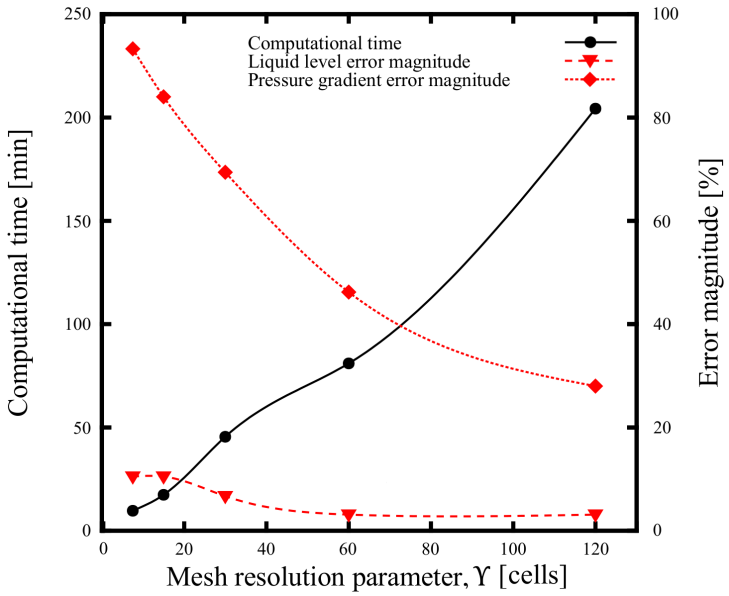


Figure 4.10: A comparison of the relationship between the computational time and the error magnitude to the mesh resolution parameter for test number FN0503 from the experimental dataset presented by Espedal [1998].

As shown in Figure 4.10, the results at the highest mesh resolution considered still show a major deviation in the simulated pressure drop, where an error magnitude slightly above a value of 25% can be observed. This suggests that the maximum equivalent mesh resolution of 1693 cells per m ($\Upsilon = 100$) is still the most appropriate for implementation in the framework. For comparison, Lo and Fiala [2010] used an equivalent mesh resolution of approximately 5,400 cells per meter length ($\Upsilon \approx 310$), which reportedly took 1 day to simulate an elapsed time of approximately 100 seconds using a total of four processors with a constant timestep of 1e-02 seconds, where the results produced were almost identical to the results from the modified CFD model using a mesh resolution parameter of 100.

4.5 Discussion

4.5.1 Stratified Flow Comparison

The results that have been presented in Figure 4.4 for the stratified flow comparison with Espedal [1998] show that the general trend for the liquid level and pressure gradient is consistent for all mesh resolutions, where both the liquid height and pressure gradient are proportional to the superficial water velocity. It is evident that the highest equivalent mesh resolution of 1693 cells per meter ($\Upsilon = 100$) was the most accurate for both the pressure gradient and liquid height and the values obtained from this mesh resolution are the closest to the simulated data of Lo and Fiala [2010]. The four lowest mesh resolutions considered yielded fairly consistent simulated values for the liquid holdup and pressure gradient, with the exception of the pressure gradient at a superficial liquid velocity of 0.3159 m/s (Test FN0503). At this velocity, the differences in the simulated values of the pressure gradient are more readily apparent, with each increment in the mesh resolution contributing to an increased level of accuracy.

The effect of the mesh resolution on the error magnitude for both the liquid height and the pressure gradient for this particular test has been shown previously in Figure 4.10, with an additional comparison to the computational time required to simulate 100 seconds of elapsed simulation time. This figure shows a marked increase in the computational time required for each increment of mesh resolution. Additionally, these results also show that at the highest mesh resolution considered, the simulated pressure gradient still exhibits a relatively high error margin of slightly above a value of 25%, suggesting an even higher equivalent mesh resolution than 1693 cells per meter

is required for higher accuracy. Fortunately, the mesh resolution is dependant on the pipe diameter and can be scaled accordingly when considering larger diameter pipes. For example, when considering a large 30" diameter oil/gas flowline, the equivalent mesh resolution reduces to approximately 133 cells per meter. Thus, for a hypothetical 1 km flowline section the total number of cells required is in the order of 133,000, which is redoubtably still a high value but not entirely unreasonable, especially when compared an equivalent of 6.1M cells that would have been required for the same flowline section using the model originally presented in Chapter 3.

Although the amount of processing power is limited in this study, it is likely that the trend for the amount of computational time required depending on the mesh resolution will continue in approximately the same manner as shown in Figure 4.10, such that the time factor needed for any further increases in mesh resolution may be estimated if any future studies are to be performed using this technique.

4.5.2 Slug Flow Comparison

In comparison to the experimental study of slug flow by Ujang [2003], the representation of the slug flow regime using the modified CFD model was not entirely accurate as the wave height did not reach the top of the pipe, although an inspection of the original results presented by Ujang [2003] in Figure 4.6 shows that true slug flow only occurs near the pipe exit (seen at the 34.55 m and 35.11 m vertical planes), suggesting that the simulated results are not too far out of line. Furthermore, as the flow regime is most likely in a transition region between stratified and slug, a certain amount of discrepancy with the results is to be expected. The inability of the modified CFD model to correctly visualise the slug flow could also be related to the use of the nutkWallFunction boundary condition applied at the wall and subsequent absence of a roughness height parameter. In general, it was observed that the sand grain roughness height had a significant effect on the simulated liquid height values when using the nutkRoughWallFunction, where higher values of liquid height were observed for increased pipe roughness. As the nutkWallFunction assumes the wall to be smooth, a variation will exist when compared to the experimental data derived from a perspex pipe.

Another explanation for the inability for the modified model to capture the wave height may be found in the study of two-phase pipe flow by Knotek et al. [2016], where it was reported that the correct flow patterns of most wavy/slug flows considered could not be generated in OpenFOAM® without the use of a random velocity perturbation

applied to both phases at the inlet. Knotek et al. [2016] also used the SST $k-\omega$ turbulence model with the interFoam solver to resolve two-phase flow in pipes; however, as seen in Figure 4.7 this does not appear to be an issue in this study, where the maximum liquid height ratio is observed to be fairly consistent for vertical planes taken at a longitudinal length of 20.57 m from the pipe entrance and beyond. This observation, combined with the successful generation of slug flow in the previous chapter suggest that it could be more due to the lack of a roughness parameter applied to the pipe wall instead.

Due to the absence of available data for the liquid holdup and pressure gradient present in the study by Ujang [2003], a comparison of the simulated values were made with the Petalas and Aziz mechanistic model. Here, it was shown that little difference exists in the values obtained for the two highest mesh resolution parameters used ($\Upsilon = 50, 100$), and both were in excellent agreement with the mechanistic model. The slug frequency was then compared by examining the fluctuations in the water height at six separate sampling locations along the mid section of the pipe length, over a 50 second time period, starting from an elapsed time of 70 seconds. The specific starting time was selected in order to be consistent with the methodology used by Lo and Fiala [2010]. The results presented in Table 4.8 show identical slug frequency results were observed for the two highest mesh resolutions considered. For the sampling locations at 35.11 m, 34.55 m, 27.22 m and 26.62 m, the simulated frequency at these mesh resolutions was seen to be below that of the experimental data, although closely matched the simulated data from Lo and Fiala [2010]. In contrast, the modified CFD model produced more accurate results for the slug frequency at the 20.57 m and 14.39 m sampling locations when compared to the STAR-CD results from Lo and Fiala [2010]. It is not explicitly known why this was the case and could possibly be due to the numerical implementation of two-phase flow using the VOF solver as both comparisons using OpenFOAM® and STAR-CD appear to generate similar results, although this issue has not been explored in further detail in this study.

The comparison of the slug length development presented in Figure 4.8 shows that the two highest mesh resolutions considered generate comparable results to the experimental data of Ujang [2003] and the simulated data of Lo and Fiala [2010]. In particular, if the development length required by the CFD model to generate the first wave is omitted, the trend line for the slug length development generated from the modified CFD model is highly comparable to the experimental data for these two mesh resolutions. A deviation of this trend is observed for the remaining three mesh

resolutions considered, suggesting that a minimum mesh resolution parameter of 50 is required in order to provide an accurate depiction of the slug length development using the modified CFD model.

4.5.3 Gas-Sweep Comparison

The comparison of the modified CFD model to the experimental study by Birvalski [2015] has shown that all mesh resolutions considered could adequately predict the onset of contraction and elongation of the liquid film in the downward and upward pipe lengths of the low-spot, respectively. This was also the case for the prediction of the gas velocity required to sweep all of the gas from the downward pipe length, which occurred at a velocity slightly above 9 m/s and was accurately predicted using all mesh resolutions. Between these extremities, and for the upward pipe length especially, the effect of the mesh resolution is more readily apparent whereby the accuracy increases substantially due to each incremental increase in resolution. Unsurprisingly, the equivalent mesh resolution of 2000 cells/per meter ($\Upsilon = 100$) was the most accurate and the simulated values obtained from this resolution were within an error magnitude of $\pm 10\%$ for each test of the downward pipe section. For the upward pipe section, the simulated liquid length was within an error magnitude of $\pm 10\%$ for seven of the nine velocities considered. A higher mesh resolution will most likely lead to a higher accuracy for the two remaining velocities outside of these bounds.

A possibility to explain the increased error magnitudes recorded in the upward pipe segment could again be the use of the nutKWallFunction as the wall boundary condition for the turbulence viscosity, which essentially assumes that the pipe walls are smooth. Increasing the wall roughness would act to increase the wall shear stress and could possibly lead to reduced liquid lengths in the upward pipe section; however, as the results presented in Figure 4.9 have shown that the onset of compression and elongation of the liquid film, together with the gas velocity at which the liquid is completely swept from the downward pipe segment has been accurately captured, the effect of of this boundary condition has not been investigated in further detail in this study.

4.5.4 Mesh Resolution Parameter

Although the highest mesh resolution parameter used in this study was 100 due to constraints on computational availability, it is foreseeable that the pressure gradient from the stratified flow comparison, the visualisation of slug flow and the response of

the liquid film in upward pipe segments may be improved further if more processing power becomes available for future studies. Despite this, a mesh resolution parameter of 100 gives excellent results for the liquid height for stratified flow, a relatively good prediction of the frequency and development of the slug flow regime and highly comparable results for response of a stagnant liquid layer during the gas-sweep comparison. Furthermore, in comparison to the resolved boundary layer approach used in Chapter 3, the modified strategy has allowed for the reduction in the total amount of cells by over 97%. As both the hydrate growth models under consideration for use in the optimisation framework by Skovborg and Rasmussen [1994] and Turner et al. [2005] are based upon an estimation of the interfacial area between the phases, it is expected that the modified modelling strategy presented in this chapter will be successful for the intended purpose of constructing an optimisation framework for gas-dominated restart operations involving long flowline sections, provided that any lateral deviation is minimal such that the symmetric boundary condition remains applicable. It is important to note that besides increasing the mesh resolution across the entire pipe length in order to facilitate greater accuracy, some other options do exist and are discussed in the two following subsections.

4.5.5 VOF interface reconstruction algorithms

The results presented by Lo and Fiala [2010] are taken from a corresponding study by Lo and Tomasello [2010], which was conducted in order to assess the capabilities of STAR-CD in simulating multiphase flow and provides more detail of the methodology used to increase the accuracy of the results when compared to the experimental dataset of Espedal [1998]. Lo and Tomasello [2010] recognised that correction of the interface is necessary to ensure that the turbulent viscosity is correctly predicted when using the VOF method for multiphase flow. Without this, the pressure gradient results were observed by Lo and Tomasello [2010] to be even higher than the results generated from any of the mesh resolutions used in this study. In order to achieve more accurate predictions of the liquid height and pressure gradient, two options were considered. The first method used by Lo and Tomasello [2010] was to artificially reduce the turbulent viscosity close to the interface using the following relation;

$$\mu_{int}^{'} = \begin{cases} C_{int}\mu_{int}, & \text{if } 0 < \alpha < 1 \\ \mu_{int}, & \text{otherwise} \end{cases} \tag{4.3}$$

Here, μ_{int} is the turbulent viscosity in the region occupied by the interface, calculated by the standard turbulent models, $\mu_{int}^{'}$ is the modified value of the turbulent viscosity,

C_{int} is an adjustable parameter taken as 0.01 by Lo and Tomasello [2010] and α is a VOF colour function used with STAR-CD, equivalent to the liquid phase fraction used in this study. Using this method, both the simulated values of the pressure gradient and liquid height were much closer to the experimental data recorded by Espedal [1998]. However, this method introduces an error due to the assumption that the turbulent viscosity is negligible in the cells occupied by the interface. Although Lo and Tomasello [2010] found that this error is smaller than the error produced by the standard turbulence models, it was deemed an inelegant method as the region where the turbulent viscosity is damped is wholly dependant on the mesh size.

In the second method, Lo and Tomasello [2010] have introduced an additional source term in the ω equation based originally upon the work of Egorov [2012]. It is these results that are shown in Lo and Fiala [2010] and subsequently compared to in this study. Considering the ω equation;

$$\frac{\partial(\rho\omega)}{\partial t} + \frac{\partial}{\partial x_j}\left[\rho U_j\omega - \left(\mu + \sigma_\omega\frac{\rho k}{\omega}\right)\frac{\partial\omega}{\partial\omega_j}\right] = \frac{\gamma\rho}{k}P - \beta\rho\omega^2 + \frac{\rho\sigma_d}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j} \tag{4.4}$$

The additional source term is used to counteract the destructive term $(-\beta\rho\omega^2)$ in Equation 4.1 and is of the form;

$$\phi\beta\rho\omega_{wall}^2 \tag{4.5}$$

Here, ϕ is the interface area density and is used to activate the source term, where $\phi = 0$ outside the interface region and;

$$\omega_{wall} = B\frac{6\mu}{\beta\rho\Delta n^2} \tag{4.6}$$

From Equation 4.6, Δn represents the grid size in the interface region and B is a tunable parameter. This presents a problem as this method is both explicitly and implicitly dependant on the mesh size, through Δn in Equation 4.6 and through the parameters, B and ϕ respectively. A value of 2500 was selected by Lo and Tomasello [2010] for B in order for the simulated values of the pressure gradient and liquid holdup to correctly correspond to the experimental data by Espedal [1998]. Selection was based on a method of trial and error and was not recommended by Lo and Tomasello [2010]. Furthermore, the dependency of both methods proposed by Lo and Tomasello [2010] on the mesh size near the interface suggest that implementation of this approach is most likely only successful if comparable experimental data is available to give an insight into the most likely location of the interface. In order to alleviate this, Lo and Tomasello [2010] have considered an option to reduce the dependence of the mesh size when using the SST $k-\omega$ turbulence model, which is discussed in the following subsection.

4.5.6 Wall-distance free SST $k-\omega$ turbulence model

Gleize and Burnley [2001] have proposed that the dependence of the wall distance on the SST $k-\omega$ turbulence model may be eliminated by substituting the explicit dependence of the d_n parameter from the coefficients of the original SST $k-\omega$ model with local quantities. In a follow-up study, Masson and Gleize [2004] have constructed a function (F_d) to estimate the distance from the solid boundaries of the domain;

$$F_d = \frac{1}{\omega}\left|\frac{\partial \omega}{\partial y}\right| \approx \frac{1}{y} \tag{4.7}$$

This function is relevant over the entire boundary layer and replaces the $\frac{1}{y}$ parameter in the blending function of the original SST $k-\omega$ model as a means to avoid any explicit dependence of the model from the wall distance. This method, however, assumes that the turbulence equations are integrated inside the boundary layer, implying a sufficiently refined mesh at the interface and wall regions. If the location of the interface is not known, this method becomes infeasible as the entire domain must be finely meshed. Although Lo and Tomasello [2010] have successfully used a tool proposed by Masson and Gleize [2004] to identify the region of influence for the interface, it is evident that this method can still only be used for steady state flows. As the position of the interface is variable for turbulent studies, implementation of this method is not a feasible option for the optimisation framework. An interesting concept would be to couple this methodology to the interDyMFoam solver discussed in the previous chapter, although this is beyond the scope of this study and has not been investigated further.

4.6 Summary and Conclusions

A scalable wall function approach has successfully been implemented for the construction of a modified CFD modelling strategy intended to allow the optimisation framework constructed in this thesis to be applicable for long flowline applications. For this purpose, three CFD models have been validated for stratified flow, slug flow and an additional gas-sweep using experimental data taken from the literature. Five separate increments of mesh resolution, defined as the total number of cells per meter length of the model were used. It was shown than the highest mesh resolution considered ($\Upsilon = 100$) was the most comparable to the experimental data of Espedal [1998] and Birvalski [2015]. The behaviour of the slug height, frequency and length had less dependency on the mesh resolution, whereby the two highest mesh resolutions

considered generated comparable results. A comparison of the computational time required to simulated test FN0503 from the experimental dataset of Espedal [1998] was conducted and showed a marked increase in the computational time required for each incremental mesh resolution. Despite this, the mesh resolutions considered are dependant on the pipe diameter, whereby it is quite possible that this technique can still be successfully employed for longer flowlines with large diameters. Following this study, the following conclusions have been made;

- The scalable wall function approach has successfully shown that a reduction of over 97% can be made in the number of cells required when compared to the resolved boundary layer approach used in Chapter 3, allowing the optimisation framework to be applicable for longer flowline applications.

- Although limited by the amount of processing power available, the highest mesh resolution considered ($\Upsilon = 100$) has yielded excellent results for the liquid height for stratified flow, a relatively good prediction of the frequency and development of the slug flow regime and highly comparable results for response of a stagnant liquid layer during the gas-sweep comparison. As such, it has been concluded that this mesh resolution will be acceptable for use in the optimisation framework.

- Further mesh refinement would enable a more accurate prediction of the pressure gradient during stratified flow.

- Further mesh refinement may not necessarily enable a more accurate prediction of the liquid height and frequency during slug flow, as the results from a mesh resolution parameter of 100 were highly comparable to the results presented by Lo and Fiala [2010] using an equivalent mesh resolution parameter of approximately 310.

- The combined implementation of interFoam, the scalable wall function approach and the SST $k-\omega$ turbulence model has been successful for modelling two-phase flow and can be utilised for further optimisation studies and towards the goal of implementing an optimisation framework applicable all types of pipe geometries and fluid properties.

Chapter 5

Flow Optimisation in a Low-spot Pipe Bend to Reduce the Risk of Hydrate Blockage

In this chapter, a CFD-based optimisation study is performed on a transient, gas-dominated restart operation using the same low-spot pipe geometry and fluid properties featured in the experimental work of Leporcher et al. [2002]. The primary purpose of this study is to determine whether commonly available optimisation methods are suitable for the purpose of avoiding hydrate blockages in flowlines during gas-dominated restart operations. To achieve this, the optimisation study utilises the modified CFD modelling strategy presented in the previous chapter. A modification is made to the interFoam solver in order to capture the interfacial area between the gas and water phases at each time step and provide input to two separate hydrate growth models based upon intrinsic hydrate kinetics and mass transfer, respectively. A genetic algorithm is used on four different types of surrogate models with the goal of minimising the restart gas velocity such that a defined blockage risk is minimised, based upon an estimation of the amount of hydrate growth, coupled with the hydrodynamic characteristics of the flow. Through this study, the most effective DACE and surrogate modelling techniques available using DAKOTA are determined for use in the optimisation framework, a definition of the optimisation problem is proposed and assessed, and the robustness of the coupling between SALOME, OpenFOAM® and DAKOTA is verified. The work presented in this chapter constitutes the culmination of work required to achieve the second part of Objective 3 from this study.

5.1 Problem Description

5.1.1 Description of the Original Experimental Setup Used by Leporcher et al. [2002]

The primary objective of the experimental study conducted by Leporcher et al. [2002] was to investigate whether appropriate restart hydrodynamic conditions could be used to avoid hydrate blockages in flowlines without the usual hydrate prevention techniques such as inhibition or thermal control. For this purpose, a series of experiments were conducted using a flow-loop located at the IFP Research Center at Solaize, France, featuring;

- A single low-spot filled with different amounts of stagnant water prior to commencement of the restart operation (7 L - 36 L).
- Two different restart gas velocities (1 m/s and 0.001 m/s).
- Two different fluid combinations (gas/water and gas/oil/water).

The design of the original experimental flow-loop used by Leporcher et al. [2002] is shown in Figure 5.1, where the the single low-spot section is shown in red. The experimental section of the flow-loop features a 2" diameter pipe of 140 m length and is terminated by a 3 m vertical riser connected to the separator.

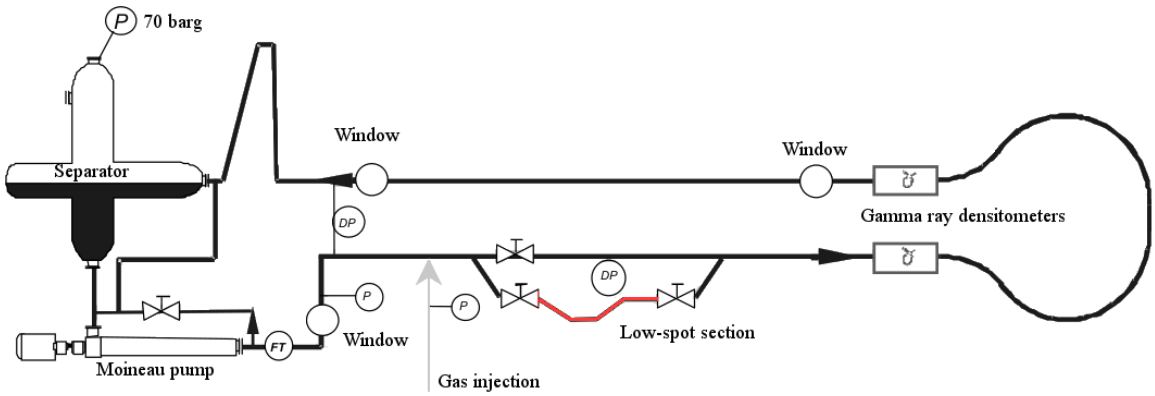


Figure 5.1: The original experimental setup used by Leporcher et al. [2002], featuring a single low-spot section shown in red (adapted from Leporcher et al. [2002]).

As shown in Figure 5.1, the single low-spot section is actually one of two bypass sections attached to the flow-loop. The other bypass section is a 7 m long horizontal section located above the low-spot and is used only for wax deposition measurements

and is not considered in this study. As the specific detail of the connection between these two bypass sections could not be found, it is assumed that the single low-spot connects directly to the horizontal pipe sections. Furthermore, it is assumed that the entire pipe length excluding the low-spot is completely horizontal, whereby the 180° bend shown in Figure 5.1 is not featured.

As the modified CFD modelling strategy presented in the previous chapter is used to emulate the experimental restart operations conducted by Leporcher et al. [2002], this current study will only consider the gas/water fluid combination. Although it is recognised that low-spot sections filled entirely with water are unlikely to occur in real flowline systems, Leporcher et al. [2002] have observed that the presence of an oil phase acts to increase the time taken for a blockage to occur, particularly for low water cut systems. This suggests that the optimal values of the gas restart velocity generated in this study will likely have a conservative bias and will not lead to over predictions in the time for a hydrate blockage to occur. Despite this, the considerations of the variable heights of the stagnant water phase and the effect of the different gas restart velocities are considered. The subsequent geometric and system parameters used in the optimisation study are shown in the following subsection.

5.1.2 Geometric and System Parameters

A representation of the single low-spot flow loop geometry originally presented by Leporcher et al. [2002] and subsequently used in this study is shown in Figure 5.2, together with a definition of the relevant geometric parameters used to create the CFD model.

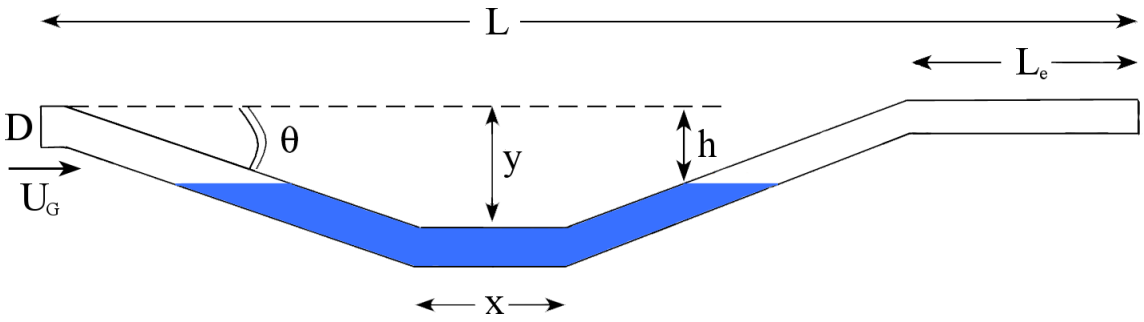


Figure 5.2: The pipe geometry used in the optimisation study, featuring a single low-spot featuring an accumulated water zone shown in blue (adapted from Leporcher et al. [2002]).

The dimensions D, L, L_e, x, y and θ have fixed values taken directly from Leporcher et al. [2002] and are summarised in Table 5.1. The dimension h indicates the height of the accumulated water phase in the low-spot section, represented by the shaded area in Figure 5.2 and is a variable in this study, along with the inlet gas velocity, U_G. For the purposes of a direct comparison with Leporcher et al. [2002], the dimension h is converted to represent the volume of liquid in the low spot section of the pipe, V, whereby h_{min} corresponds to V_{max} with a value of 36 L and h_{max} corresponds to V_{min} with a value of 7 L. The exit length of the pipe, L_e, has been selected as 100 m in order for the total length to be consistent to the setup used by Leporcher et al. [2002].

Table 5.1: The geometric parameters used in the optimisation study.

Parameter	Description	Value
D	Pipe diameter	0.0508 [m]
L	Total horizontal length	140 [m]
L_e	Exit length	100 [m]
x	Low-spot length	3 [m]
y	Low-spot depth	1.1 [m]
θ	Low-spot inclination	9 [°]

Using the compositional gas data provided by Leporcher et al. [2002], the gas density has been derived using the AGA8 guidelines [Starling and Savidge, 1992] at the reported pressure of 70 barg and system temperature of 4°C. The gas viscosity has been calculated using Sutherland's law. The pipe roughness has been derived from an assumption that the original experimental flow-loop used by Leporcher et al. [2002] consisted of commercially available standard steel sections commonly found in piping applications. The complete set of derived system parameters used in the computational analysis is shown in Table 5.2 and the original gas composition used by Leporcher et al. [2002] is presented in Appendix E.5.

Table 5.2: The derived system parameters used in the optimisation study.

Parameter	Value
Gas density	62.37 $[kg/m^3]$
Gas kinematic viscosity	1.73e-07 $[m^2/s]$
Water density	1000 $[kg/m^3]$
Water kinematic viscosity	1e-06 $[m^2/s]$
Interfacial tension	7.24e-02 $[N/m]$
Pipe roughness (absolute)	4.57e-04 $[m]$
System pressure	70 [barg]
System temperature	4 [°C]

5.2 Numerical Method

Using the scalable wall function approach presented in the previous chapter, the two-phase flow of gas and water has been described here using the interFoam solver included with OpenFOAM® 2.3.x. A 3-D half-pipe CFD model has been constructed using SALOME 7.4.0 and implementation of the model follows the same manner that was used in the comparison to the experimental study of Birvalski [2015]. The maximum cell resolution considered from the previous study ($\Upsilon = 100$) was used here with the same mesh cross section, such that a total of 280,000 cells were used throughout the domain. This study also re-uses the SST $k - \omega$ turbulence model with the same numerical schemes and boundary conditions as presented in the previous chapter, although the nutkRoughWallFunction is again used for the boundary condition of the turbulence viscosity at the pipe walls, on account of the steel pipe sections used.

5.3 Methodology

5.3.1 Implementation of the Hydrate Growth Model

Following the work of Di Lorenzo et al. [2014] and Aman et al. [2016] for gas-dominated flow, the hydrate growth rate in this study based upon intrinsic kinetics has been estimated using the model developed by Turner et al. [2005]. However, due to the variable restart gas velocities considered in this study, the effect of mass transfer will likely also have a prominent role in the amount of hydrate growth observed; as such, the model by Skovborg and Rasmussen [1994] has been used in conjunction

with the kinetic model, whereby the total hydrate growth is a summation of the estimations generated from both models. This differs from both studies by Aman et al. [2016] and Di Lorenzo et al. [2014], where both the kinetic and mass transfer limited models were compared directly to experimental data as standalone models.

Instead, in this study the fitting parameter introduced to the kinetic model by Turner et al. [2005] to account for mass and heat transfer limitations has been reduced from a value of 1 to a value of 0.5 in order to account for the additional hydrate growth estimated using the mass transfer model. In the absence of experimental data, the correct value of the fitting constant is unknown and adjustment of this parameter will be necessary. As a value of 0.5 was also used by Di Lorenzo et al. [2014] to force the kinetic model to agree with the experimental flowloop data, it is expected that the combined hydrate growth estimations presented in this study will be slightly conservative.

Using the kinetic hydrate model, the growth rate is based upon the kinetic rate of crystallisation; whereby the chemical driving force is represented in terms of the degree of subcooling (ΔT_{sub}). In contrast, in the mass transfer limited model the expression for the hydrate growth rate is assumed to be governed by the transport of gas molecules through the liquid phase to the hydrate interface. Using a combination of these models, the expression for the total hydrate growth rate ($\Delta n/\Delta t$) in mol/s is;

$$\frac{\Delta n}{\Delta t} = (\chi + \lambda)A_i \tag{5.1}$$

where A_i is gas/water interfacial area and,

$$\chi = uK_1 exp\big(-\frac{K_2}{T}\big)\Delta T_{sub}\frac{1}{W_g} \qquad \text{(kinetic model)} \tag{5.2}$$

$$\lambda = \sum_i k_i(c_i - c_i^{eq}) \qquad \text{(mass transfer model)} \tag{5.3}$$

From Equation 5.2, K_1 and K_2 are kinetic constants, T is the system temperature and W_g is the average molecular weight of the hydrate forming components. The values for the kinetic constants, K_1 and K_2 are derived from the work of Englezos et al. [1987] and are taken as 2.608×10^{16} [kg m^{-2}K^{-1}s^{-1}] and K_2 = 13,600 [K], respectively. The values T, ΔT_{sub} and W_g have been assumed constant at 4 [°C], 11 [°C] and 0.018 [kg mol^{-1}], respectively, which is consistent with the information provided by Leporcher et al. [2002]. Lastly, the constant, u, is the fitting parameter

used to account for mass and heat transfer limitations with a value of 0.5, as mentioned previously.

From Equation 5.3, the summation is over all of the hydrate forming gas species, where the variable k_i represents the mass transfer coefficient of species i in the liquid phase, and c_i and c_i^{eq} represent the gas concentration of species i in the liquid phase, with and without hydrate present, respectively. The method of implementation follows that shown in Di Lorenzo et al. [2014], where in particular, the mass transfer coefficient is a function of the friction velocity, v_f and the Schmidt number, Sc_i, based upon the correlation proposed by Shaw and Hanratty [1977] as follows;

$$k_i = 0.0889 \cdot v_f \cdot Sc_i^{-0.704} \tag{5.4}$$

On account of the zero-net liquid flow (ZNLF) condition caused by the stagnant water region, a small difference arises on the manner on which the interfacial friction factor is estimated, which is used for the calculation of the friction velocity in Equation 5.4 as follows;

$$v_f = \left(\frac{f \cdot \rho_g}{2 \cdot \rho_l} \right)^{0.5} v_g \tag{5.5}$$

From Equation 5.5, f is the interfacial friction factor, ρ_g and ρ_l are the densities of the gas and liquid phases, respectively, and v_g is the gas velocity. Instead of the Beggs and Brill [1973] method used by Di Lorenzo et al. [2014] to calculate the friction factor, the method proposed by Amaravadi et al. [1998] was used instead, where the friction factor is based upon the amplitude of the waves formed on the upward pipe segment at steady state conditions. As a result, the range of values for the interfacial friction factor used in this study were from a minimum value of 0.0142 to a maximum 0.03, depending on the gas velocity. This is opposed to a value of 0.0248 used by Di Lorenzo et al. [2014].

For a transient restart operation, Equation 5.1 can be rewritten to take advantage of the instantaneous interfacial area that is extracted from the CFD simulation, such that the molar amount of hydrate growth in the transient phase is the integral of the interfacial area from the simulation commencement at $t = 0$ seconds until t_{ss}, the time at which the steady state flow conditions are reached. In practical terms, steady state flow occurs when all of the water that has been swept from the low-spot of the

pipe exits the computational domain. The molar amount of hydrate growth during the transient phase can thus be written as;

$$\Delta n_{trans} = (\chi + \lambda) \int_0^{t_{ss}} A_i \Delta t \tag{5.6}$$

For gas-dominated restarts where the water phase has not been completely swept from the low-spot at steady state, it is assumed that eventual blockage of the pipe will occur due to hydrate formation. In order to characterise the hydrate blockage risk based on time, Equation 5.1 can be used to express the hydrate growth rate in mol/s during the steady state phase;

$$r_h = \frac{\Delta n_{ss}}{\Delta t} = (\chi + \lambda) A_{ss} \tag{5.7}$$

It follows that the total amount of gas consumption (Δn_{tot}) at any elapsed time, t, after the steady state conditions have been reached can be written as a summation of the transient and steady state gas consumption, i.e.;

$$\Delta n_{tot} = \Delta n_{trans} + r_h t \tag{5.8}$$

Following Aman et al. [2016], the expression for the total amount of molar gas consumption can be expressed in volumetric terms (m^3) through the use of a multiplication factor, ψ, where;

$$\psi = \frac{n_h W_w + W_g}{(n_h + 1)\rho_h} \tag{5.9}$$

Here, n_h is the hydration number, W_w is the molecular weight of water and ρ_h is the density of the formed hydrate. The values used for n_h, W_w and ρ_h are assumed constant at 6, 0.018 [kg mol^{-1}] and 950 [kg m^{-3}], respectively and have been taken from the study by Aman et al. [2016]. Thus, an expression for the total amount of the hydrate growth in cubic meters is;

$$\Delta V_{tot} = \psi \Delta n_{tot} \tag{5.10}$$

As the expression for the hydrate growth is now in volumetric terms, it can be used to determine the thickness and growth rate of the hydrate layer in units of length. In

order to do this, an estimation of the deposition length is required. Ideally, the hydrate growth model could be coupled to a hydrate deposition model, such as described by Jassim et al. [2010]. Hydrate deposition models; however, rely on an estimation of the initial droplet size in order to calculate the distance along the length of pipe that the droplet will eventually deposit. A problem arises as hydrates can form in a range of droplet sizes. In particular, in an example given by Jassim et al. [2010] for a 20 mm diameter pipe, the deposition distance as a function of particle size varies between 0.6 m to 6000 m for particle sizes inputs of 500 μm to 0.1 μm, respectively. This is also consistent with the observations made by Leporcher et al. [2002] that hydrates had formed along the length of the pipe and not at a single location.

It is thus foreseeable that a simple assumption that hydrates form along the entire exit length of the pipe may result in a severe under prediction in the thickness of the hydrate layer, particularly as there are no practical means of accounting for localised buildup of hydrate material. For this reason, the approach taken in this study is to limit the deposition length to a defined wetted length, L_w. The wetted length is defined as the cumulative length along the pipe from two specific points that mark the boundaries of the accumulated water region that is in physical contact with the entire inner pipe circumference. Here, it is assumed that any hydrate material created outside of this region will be transported downstream, which enables the growth rate and subsequent blockage time to be predicted based upon an annular reduction in the aperture size, commonly associated with gas-dominated flow.

An illustration of the method used in this study to calculate the wetted length for a particular flow ($U_G = 0.6$ m/s, $V_i = 14$ L) is shown in Figure 5.3, where "alpha.water" represents the phase fraction of the initial accumulated water phase. Here, the first point is defined at the first instance the water region is in contact with the entire inner wall circumference. This occurs on the left hand side of the accumulated water region prior to commencement of the simulation (t = 0 s) as shown in Figure 5.3 part a). The final location occurs on the right hand side of the accumulated water region during the transient phase (t > 0 s) as the water is pushed along the pipe, as shown in Figure 5.3 part b). Here, as a result of the pipe bend, the water is no longer incident around the entire inner circumference of the pipe at a point where the upward inclined segment becomes horizontal. This is the maximum longitudinal location that the water is incident on the upper pipe wall and it is assumed that no hydrates will deposit on the wall after this location. The defined wetted length is thus the distance along the pipe length between these two derived points.

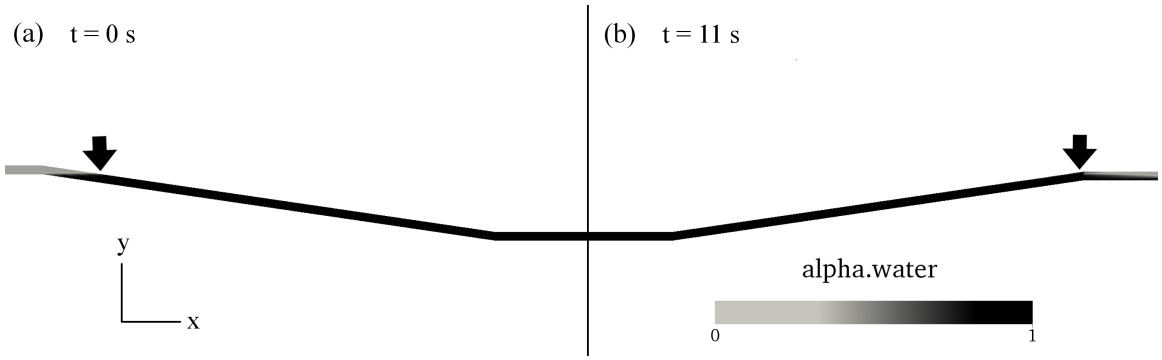


Figure 5.3: Definition of the wetted length as the cumulative length along the pipe from two specific points; a) the initial point is located at the first instance of water on the left hand side of the low-spot and is recorded prior to commencement of the simulation and b) the final point is located on the right hand side of the low-spot at an elapsed simulation time (t > 0 s) as the initial water slug detaches from the upper surface of the inner pipe wall ($U_G = 0.6$ m/s and $V_i = 14$ L).

This approach means that the hydrate growth will most likely be restricted to the pipe bend regions, with the exception of high velocity restarts, where the initial liquid slug traverses a greater length of pipe. Thus, the assumptions made regarding the deposition of the hydrate particles in this study are as follows;

- The total amount of hydrate material formed will deposit along the inner perimeter of the pipe wall, i.e., no sloughing and bulk transportation of agglomerated hydrate particles occurs.
- The total length that defines the area of hydrate deposition is denoted as the wetted length, L_w.
- Based on the deposition area, hydrate growth occurs in a annular fashion from the wetted area of the inner pipe wall in a similar fashion to stenosis.
- No localised buildup of hydrate occurs.

These assumptions will most likely result in a conservative estimate of the hydrate growth and future improvement of the wetted length parameter is necessary. With these assumptions, it is noted that the molar amount of hydrate growth that has occurred during the transient phase can now be written as;

$$\Delta n_{trans} = \frac{1}{\psi}(r_p^2 - r_i^2)\pi L_w \tag{5.11}$$

Here, r_p and r_i represent the radius of the pipe and radius of the internal aperture of the pipe with a deposited hydrate layer respectively. Equation 5.11 can be solved to determine the thickness of the hydrate layer (H_t) at the start of steady state phase, where;

$$H_t = r_p - r_i \tag{5.12}$$

5.3.2 Estimation of a Blockage Time

Based upon the principles of particle driven flow, Guariguata et al. [2012] have shown that blockage of the pipe may occur well before enough hydrate growth occurs to completely close the internal aperture of the pipe. The approach proposed by Guariguata et al. [2012] involves an estimation of the critical throughput of hydrate material for a given aperture that will allow for subsequent buildup to occur. It can be used to determine the critical aperture that will result in a blocked pipe and provide an estimation of the time required for a blockage to occur.

This approach has not been taken here however, as while it is possible to calculate the maximum permissible particle flow rate through a given aperture, the jamming probability per particle is required and can only be calculated via experimental methods. For this reason, a simplified methodology is used in this study by defining a new parameter called the half-block time, t_{hb}. This is simply defined as the time taken for the hydrate layer to restrict the internal aperture size to half the size of the original inner pipe diameter, and is dependent upon the estimated hydrate growth rate during the steady state phase and the new aperture radius, i.e.;

$$t_{hb} = \left(r_i^2 - \left(\frac{r_p}{2} \right)^2 \right) \frac{\pi L_w}{\psi r_h} \tag{5.13}$$

5.3.3 Modification of the interFoam Solver

In order to estimate the molar gas consumption, a modification has been made to the interFoam solver in order to extract the total interfacial area between the gas and water phases throughout the domain after each iteration of the solver. This has been accomplished in the same fashion as the procedure presented by Thiele [2010], where a simplified model was used to calculate the interfacial area based upon the volume of each cell and the value of the liquid phase fraction recorded in each cell. A depiction

of the interfacial area between the two phases in a particular cell has been presented by Thiele [2010] and is reproduced here in Figure 5.4.

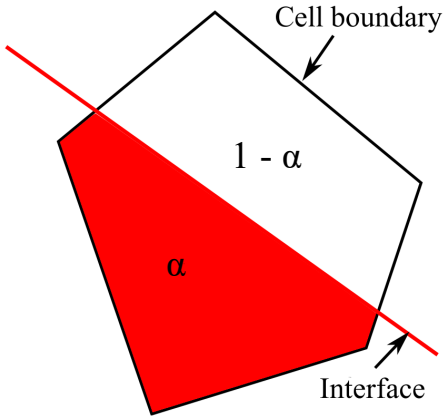


Figure 5.4: A depiction of the interfacial area in a cell as the area between the fluid (α) and the gaseous phase ($1-\alpha$) (adapted from Thiele [2010]).

Using this method, the interfacial area in each cell ($A_{i,cell}$) is given by;

$$A_{i,cell} = \sqrt{3} \cdot V_{cell}^{2/3} \cdot F_a \tag{5.14}$$

Here, V_{cell} is the volume of each cell obtained from the volumetric field data in OpenFOAM®. The area factor, F_a, is a linear function used to scale the interfacial area based upon the liquid phase fraction calculated for each cell and is the same as given by Thiele [2010], i.e:

$$F_a = \begin{cases} 2 \cdot \alpha & 0 \leq \alpha < 0.5 \\ 2 \cdot (1-\alpha) & 0.5 \leq \alpha \leq 1. \end{cases} \tag{5.15}$$

Following this, the total interfacial area at each time step is simply the summation of the area contained in each cell throughout the computational domain, consisting of N cells;

$$A_i = \sum_{i=1}^{N} A_{i,cell} \tag{5.16}$$

This representation of the total interfacial area estimated at each time step is then used as the input for both the kinetic and mass transfer hydrate growth models. The source code used to implement the modified interFoam solver is shown in Appendix D.1.

5.3.4 Definition of the Hydrate Blockage Risk

Prior to any attempts to initiate the optimisation study, it is important to realise that a simple minimisation of both the restart gas velocity (U_G) and the initial water volume (V_i) will cause the optimisation algorithm to identify the point where all water is evacuated from the pipe as the optimum. This approach, however, contradicts observations made by Leporcher et al. [2002] that hydrate blockages did not occur at lower velocities where residual water was still present in the pipe and suggest implementation in such a manner will cause a large over-prediction of the actual optimal restart gas velocity required to minimise the risk of a blockage occurring. Due to this reason, three specific risk zones are defined in this study by the physical presence of water in the upward leg of the pipe at steady state. These risk zones have no direct bearing on the calculated time for a blockage to occur although could be useful for visualisation purposes to aid an operator in determining likely regions of increased blockage risk, depending on the gas restart velocity and amount of accumulated liquid in the flowline prior to restart.

Implementation of the separate risk zones is achieved in OpenFOAM® by using the *probeLocations* utility to capture values of the liquid phase fraction at upper and lower sections of the inclined pipe segment during steady state conditions, which is the most likely location that the liquid phase is likely to remain if the restart gas velocity is not high enough to sweep all of the water held in the low-spot downstream. In the present optimisation study, a total of twelve probes are used on the upward pipe segment and are evenly distributed along the length, where a single pair of probes are located at both the entrance and exit of the upward pipe section. Six of these locations are shown in Figure 5.5, where the vertical distribution of the upper and lower probe locations can also be seen.

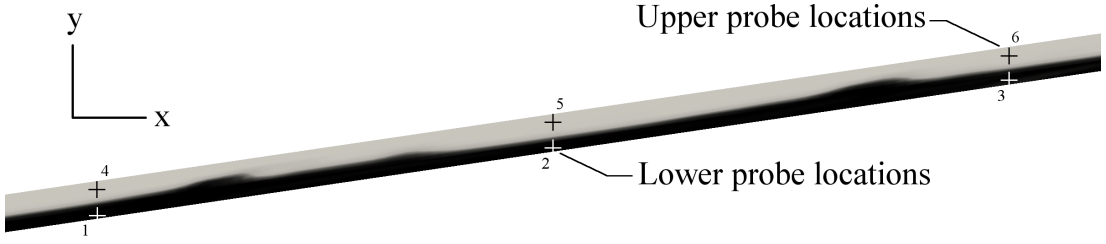


Figure 5.5: The relative distribution of six of the probe locations on the upward leg of the low-spot section.

The values of the liquid phase fraction are summed at the upper and lower probe locations and denoted as two separate objective functions, $\sum \alpha_{upper}$ and $\sum \alpha_{lower}$, respectively. Using this methodology, three risk zones have been identified throughout the design space and are denoted as either Low, Medium or High risk as shown in Figure 5.6.

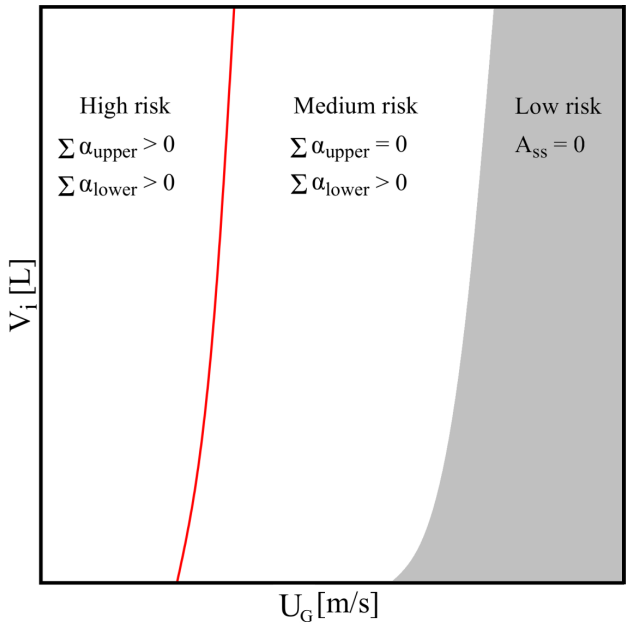


Figure 5.6: The theoretically defined boundaries of low, medium and high risk for hydrate blockage to occur during a gas-dominated restart depending on volume of water in the low-spot section (V_i) and the gas restart velocity (U_G).

The definition of high risk zone depends on the amount of water present in the upward leg of the low-spot section at steady state conditions, as determined by the respective upper and lower probe outputs. As observed by Leporcher et al. [2002], the highest risk of hydrate blockage occurs when gas migrates across a stationary water zone near the upper wall of the pipe. This will happen if the restart gas velocity is too low, resulting in a high liquid level. In this case, the values for both the lower and upper probe sets returned from the simulation would be greater than zero and the amount of hydrate growth from the transient phase is combined with the estimated hydrate growth rate in the steady state phase to yield an estimated time for a blockage to occur, or in this case, the half-block time.

The medium risk zone does not exhibit any gas migrating across a stagnant water phase near the top of the pipe; however, will still contain a certain amount of water

that has not been evacuated completely from the pipe at steady state. The location of this water is expected to remain on the lower wall surface of the upward bend segment. This zone is denoted by the fact that the upper probe set will be zero, while the lower probe set is greater than zero. Similar to the high risk region, the half-block time is used and the same strategy to estimate the amount of hydrate growth in the transient phase and the growth rate in the steady state phase is used.

In the low risk zone, all water initially held in the pipe low-spot is evacuated from the domain during the restart operation, denoted by the fact that the interfacial area returned from the simulation at steady state conditions is zero. Here, the amount of hydrate growth is assessed only during the transient phase and it is assumed that no further growth happens after the water is evacuated from the computational domain.

5.3.5 Optimisation Methodology

Due to the inherently computationally expensive nature of the multiphase CFD simulations, it has been proposed to use a surrogate-based optimisation strategy for the optimisation framework. As mentioned in Chapter 2, surrogate models are constructed using data extracted from the computational model via selected build points in the design space, defined using a DACE technique. An optimisation algorithm is then applied to the constructed surrogate model to search the design space for the global minimum or maximum of the objective function(s), depending on the optimisation criteria. As the aim of the optimisation framework is to find the optimal values of the restart gas velocity depending on a defined level of hydrate blockage risk and the hydrodynamic characteristics of the flow, the available DACE and surrogate modelling techniques selected for use in the framework must be able to function in the most effective and robust manner.

The review of the literature presented in Chapter 2 pertaining to the process of simulation-based optimisation has suggested that the appropriate DACE and surrogate modelling techniques available in the optimisation toolkit can easily be compared to in order to select the most effective for recommendation in the framework. In this study, the most effective DACE technique has been selected based on an assessment of the space-filling properties of the build points over the design space, where an example of this can be found in the work of Franco et al. [2009]. Furthermore, the root-mean-squared (RMS) error metric is used with a leave-one-out 10 fold cross validation technique as a means to determine the robustness, or the "goodness-of-fit" of the surrogate models constructed. The RMS error metric has been selected as the

magnitude of the RMS errors are of the same scale as the sampling data and the cross validation technique has been selected as no new build points are required in addition to those generated from the DACE study. Furthermore, a genetic algorithm has been selected as the optimiser algorithm due to the wealth of literature presented in Chapter 2 that shows that the use of GAs to solve complex engineering problems is well established and highly likely to yield successful results in this study. Finally, based upon a review of the available optimisation toolkits presented in Chapter 2, all optimisation studies have been performed using DAKOTA 6.0 [Adams et al., 2014], where an overview of the coupling between SALOME, OpenFOAM® and DAKOTA is presented in Appendix D.2.

5.3.6 Definition of the Optimisation Problem

As per the definition of hydrate blockage risk in Section 5.3.4, the optimisation problem now involves the minimisation of the liquid phase fraction at all upper probe locations and the maximisation of the liquid phase fraction at all lower probe locations. In order to allow the optimisation algorithm to converge correctly, a total of five objective functions have been used, with the definition of the optimisation problem written as;

$$\begin{aligned} &\text{Maximise:} && L_w, \sum \alpha_{lower} \\ &\text{Minimise:} && \Delta n_{trans}, \frac{\Delta n_{ss}}{\Delta t}, \sum \alpha_{upper} \\ &\text{Subject to:} && V_{min} \leq V_i \leq V_{max} \\ & && U_{min} \leq U_G \leq U_{max} \end{aligned}$$

Here, the maximisation of the wetted length (L_w) will enable the optimisation algorithm to converge to higher velocity cases, which contribute to an increased half-block time as the hydrate growth is distributed along an increased length of pipe. The maximisation of the lower probe set ($\sum \alpha_{lower}$) is used to restrict the restart velocity to cases where liquid is contained within the upper pipe segment of the low-spot. The first two objective functions to be minimised (Δn_{trans} and $\Delta n_{ss}/\Delta t$) are directly related to minimising the amount of hydrate growth during the transient and steady state phases, respectively. The remaining upper probe set ($\sum \alpha_{upper}$) is minimised based upon the observations by Leporcher et al. [2002] that a slow moving gas flow over a stagnant water phase near the top of the pipe constitutes the highest risk of a hydrate blockage to occur. The lower and upper bounds of the initial water volume

(V_{min} and V_{max}) are determined by the amount of water held in the low-spot bend and are taken as 7 L and 36 L, respectively. The lower and upper bounds of the restart gas velocity (U_{min} and U_{max}) are expected to be within a range of 0.1 m/s and 1.5 m/s, respectively. These velocity constraints have been based on the observations made by Leporcher et al. [2002] that a restart velocity of 1 m/s was high enough such that no hydrate blockages occurred. Using this definition of the optimisation problem, the entire work flow used in the optimisation study is presented in Figure 5.7.

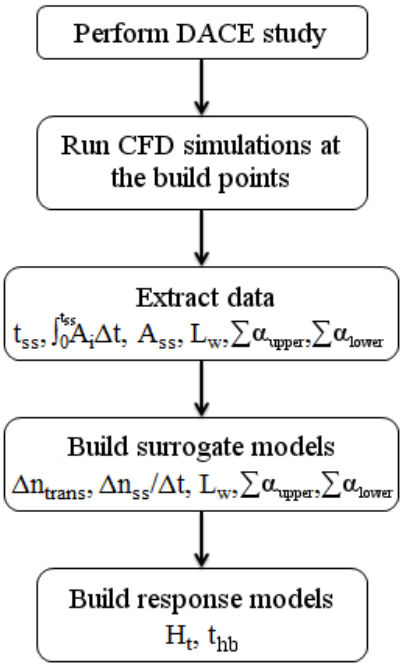


Figure 5.7: Optimisation problem work flow.

As shown in Figure 5.7, the first two stages involve performing the DACE study and running of the CFD simulations at the build points generated. Each of the simulations are run using OpenFOAM® until the steady state conditions are reached and the relevant steady state time (t_{ss}), interfacial area during the transient and steady state conditions ($\int_0^{t_{ss}} A_i \Delta t$ and A_{ss}, respectively), wetted length (L_w), and the probe data ($\sum \alpha_{upper}$ and $\sum \alpha_{lower}$) are extracted. Using the extracted data, five separate surrogate models have been constructed for each of the objective functions.

For visualisation purposes, two response models have then been used to show the thickness of the hydrate layer at a point when the steady state conditions is reached (H_t) and the estimated half-block time (t_{hb}) throughout the entire design space. These

two response surfaces are derived from Equations 5.12 and 5.13, respectively and have been built exclusively using data from the surrogates to project the response over the entire design space. For the purpose of clarity, all of the surrogate and response models have been designated an alphanumeric identification number as shown in Table 5.3.

Table 5.3: Designation of the surrogate (S) and response (R) models used in the optimisation study.

Designation	Model	Description	Source
S1	Δn_{trans}	Transient hydrate growth [mol]	Eq. 5.6
S2	$\Delta n_{ss}/\Delta t$	Steady state hydrate growth rate [mol/s]	Eq. 5.7
S3	L_w	Wetted length [m]	CFD
S4	$\sum \alpha_{upper}$	Upper probe set	CFD
S5	$\sum \alpha_{lower}$	Lower probe set	CFD
R1	H_t	Thickness of the hydrate layer at steady state [m]	Eq. 5.12
R2	t_{hb}	Half-block time [s]	Eq. 5.13

5.4 Results from the Optimisation Study

5.4.1 Validation of the interFoam Modification

In order to ensure that the methodology used to calculate the interfacial area was correct, the interfacial area generated by the modified interFoam solver was first compared to the simulated values of the ratio of the liquid height to pipe diameter (h_L/D) obtained from the four stratified cases used in the comparison to Espedal [1998] from the previous chapter. Here, the interfacial area can be calculated based using the following relation adapted from the work of Taitel and Dukler [1976];

$$A_i = D \cdot L\sqrt{1 - \left(2 \cdot \left(\frac{h_L}{D}\right) - 1\right)^2} \tag{5.17}$$

A summary of the results generated from this comparison is presented in Table 5.4, where the values for the interfacial area obtained from the interFoam solver have been averaged over a period of 40 seconds, from an elapsed time of 40 seconds. The full transient response of the interfacial area obtained from the solver for all of the four flows considered is shown in Figure 5.8.

Table 5.4: A comparison of the total interfacial area (A_i) obtained from the modified interFoam solver to the calculated values of the interfacial area, based upon the simulated ratio of the liquid height to the pipe diameter (h_L/D), obtained from the four stratified wavy cases considered in the previous chapter.

Test number	Simulated h_L/D (CFD) [-]	Calculated A_i (Eq. 5.17) [m^2]	Solver A_i [m^2]	Error mag. [%]
FN0139	0.233	0.457	0.456	0.12
FN0140	0.330	0.508	0.471	7.25
FN0141	0.383	0.525	0.520	0.95
FN0503	0.433	0.535	0.529	1.15

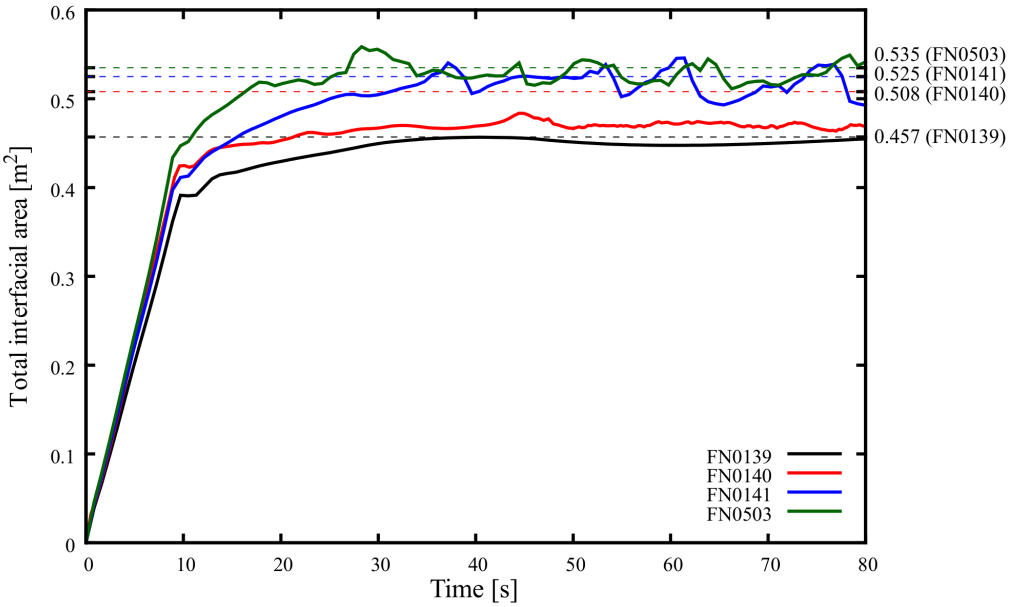


Figure 5.8: The transient response of the total interfacial area obtained from the modified interFoam solver, compared to the interfacial area calculated from the simulated ratio of the liquid height to the pipe diameter obtained from the four stratified wavy cases considered in the previous chapter.

As shown in Figure 5.8, the calculated interfacial area has closely matched with the values generated from the modified solver for all four of the four stratified cases considered. In particular, three out of the four cases have shown excellent results from an elapsed time of 40 s. The remaining case, FN0140 has a slightly increased error margin of 7.25%. It is possible that the increased error margin may be due to size of the vertical mesh spacing or the variations in the vertical mesh spacing throughout the cross-section. Although the error magnitude for this particular case was slightly higher than anticipated, the error margin was deemed acceptable and

no further analysis has been conducted in this study. Furthermore, the transient response of the interfacial area displayed in Figure 5.8 appears to have been captured correctly, whereby the fluctuations observed are directly related to the formation and frequency of waves at the interface, and are influenced by any incremental increase in the superficial liquid velocity. These results suggest that the modified interFoam solver is expected to provide a good approximation of the interfacial area required to estimate the amount of hydrate growth for the stratified/slug flows anticipated during gas-dominated restart operations.

5.4.2 DACE Study

A DACE study was conducted to determine the most appropriate space-filling technique available in DAKOTA for use in the current optimisation study, which will then be recommended for used in the optimisation framework. To achieve this, four of the main DACE techniques available in DAKOTA were compared based upon an inspection of the space-filling characteristics and robustness of the surrogate models generated from three separate sampling sizes of 16, 25 and 49. The DACE techniques include; Latin hypercube sampling (LHS), orthogonal array sampling (OAS), orthogonal array based Latin hypercube sampling (OA-LHS) and Monte Carlo random sampling (MCS). Additional techniques available in DAKOTA, such as the Box-Behnken and Central-Composite designs are typically not suited for multi-objective global optimisation problems as the number of sample points is limited by the number of design variables. For example, the number of build points generated by the Central-Composite and Box-Behnken designs for an optimisation problem involving 2 design variables are 9 and 10 respectively, which is considered to be inadequate for most optimisation problems over global parametric space.

The three separate sampling sizes used were derived from a requirement that the sampling size for the OAS and OA-LHS techniques is determined based on an integer multiple of the number of symbols squared, where the number of symbols must be either 4 or a higher prime number. In this manner, the selected sample sizes of 16, 25 and 49 correspond to symbol numbers of 4, 5 and 7, respectively. For the purpose of consistency, all of the DACE techniques considered use these same three sampling sizes, although it recognised that these sampling sizes may not be the most optimal for the LHS and MCS techniques. The results of the DACE study using all three sampling sizes for each of the four DACE techniques considered is shown in Figure 5.9.

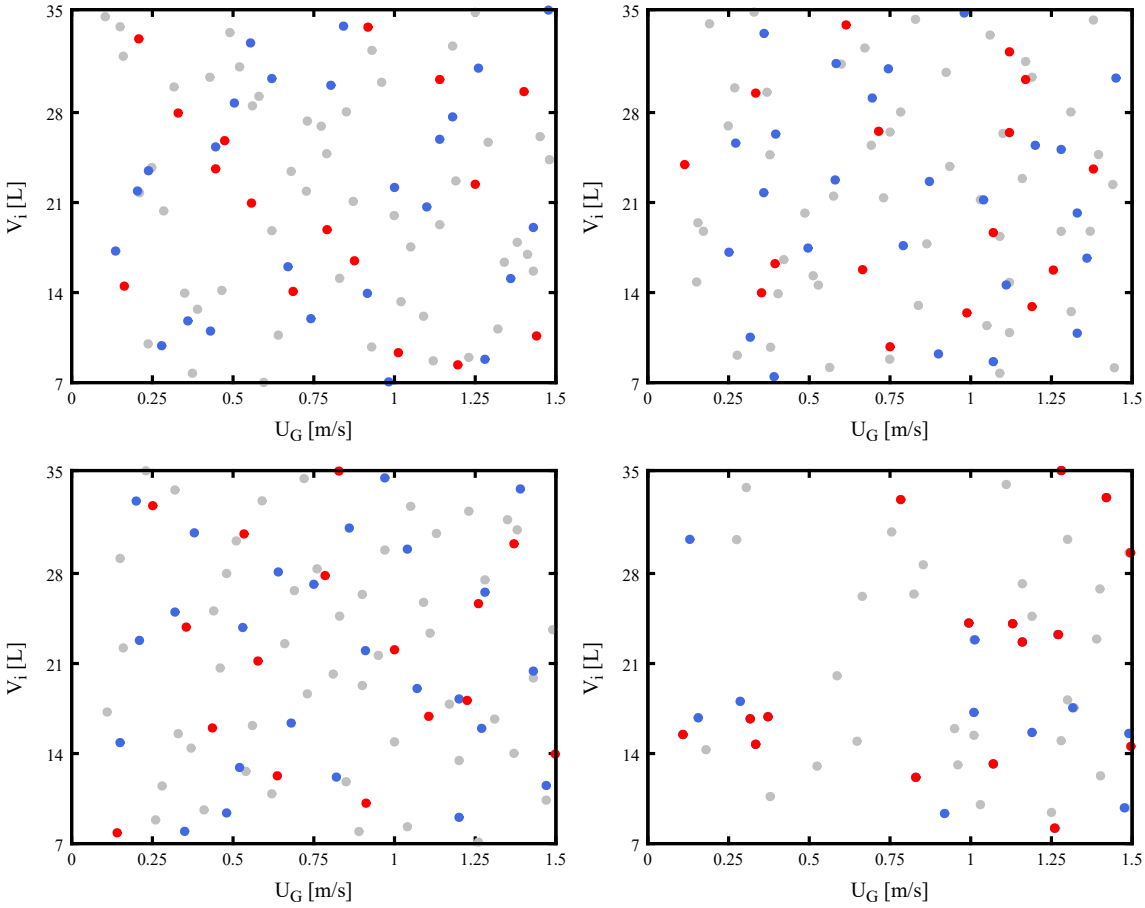


Figure 5.9: The location of the simulation points within the design space obtained from the DACE study for three separate sampling sizes (16 - red, 25 - blue, 49 - grey), using four different sampling techniques; (a) Latin hypercube sampling, (b) Orthogonal array sampling, (c) Orthogonal array based Latin hypercube sampling and (d) Monte Carlo random sampling.

The results shown in Figure 5.9 depict the location of the surrogate build points within the design space bounded by values of the initial water volume and the gas restart velocity. The space-filling metrics for each of the DACE methods have been assessed using the same methodology proposed by Franco et al. [2009], where the Minimal Spanning Tree (MST) criterion developed by Dussert et al. [1986] has been used to determine the mean and standard deviation of the minimum distance between the build points throughout the design space, which is also known as the arc length. According to Franco et al. [2009], designs with higher values of the mean and lower values of the standard deviation of the arc lengths produced better space-filling designs, which were termed "quasi-periodicals". The results from this approach are shown in Table 5.5. Here, the values for the maximum mean and minimum standard

deviation are highlighted in red for each sampling size and the specific MSTs used to generate these results are shown in Appendix D.3.

Table 5.5: The space-filling quality metrics of different DACE methods using the MST technique for the arc length between the build points.

DACE points	DACE method	Mean distance	Std. dev. (Pop.)
16	LHS	6.020	2.399
	OAS	5.887	2.132
	OA-LHS	6.672	1.555
	MCS	5.608	3.213
25	LHS	4.565	1.823
	OAS	4.929	1.384
	OA-LHS	5.034	1.411
	MCS	4.841	3.365
49	LHS	3.060	1.294
	OAS	3.488	1.215
	OA-LHS	3.622	1.108
	MCS	3.131	1.514

The quality metric results using the MST criterion presented in Table 5.7 provide a clear picture of the space-filling properties of each of the techniques considered. Specifically, the data presented here shows that both the OAS and OAS-LHS techniques are similar and exhibit superior space-filling qualities when compared to the standard LHS and MCS techniques which can be confirmed by the visual results presented previously in Figure 5.9. Although, it is expected that either the OAS or OA-LHS techniques will generate comparable surrogate models, the OA-LHS technique has been selected to build the surrogate models used in this study as it appears to produce slightly better space-filling designs than the OAS technique.

5.4.3 CFD Response of the Transient Interfacial Area

CFD simulations were run at the build points generated by the DACE study until steady state conditions were achieved. The time for steady state conditions to occur was estimated using a plot of the transient response of the total interfacial area captured by the solver. An example of the response of the total interfacial area for a selection of gas velocities and initial water volumes is shown in Figure 5.10 until

an elapsed simulation time of 600 s. For the purpose of visualisation, the restart gas velocities and initial water volumes shown in Figure 5.10 are not the same as those generated from the DACE study. Instead, four gas restart velocities of 0.1, 0.5, 1 and 1.5 m/s are used with three separate volumes of initial water volumes of 7, 22 and 36 L.

From Figure 5.10, upon commencement of the simulation, the transient interfacial area rapidly increases in all cases as the initial water volume is pushed up the upward segment of the pipe low-spot. For all of the flows that had enough energy to force an amount of water from the low-spot, a single slug formed and was transported downstream until a certain location (as defined by the wetted length) where the eventual dissipation in wave height caused the flow to become stratified, further increasing the transient interfacial area as the liquid layer in the horizontal pipe segment became increasingly elongated until a point at which the water phase reached the exit of the computational domain. At this point, a reduction in the transient interfacial area is observed until a plateau is reached, indicating the onset of steady state conditions. For these flows, mixing of the gas and water phases was not observed at any locations other than the main wave front and at all other locations the prevailing flow pattern was stratified with no other flow regimes observed.

For gas velocities of 1 m/s or less, water was held in place on the upward pipe segment of the low-spot and low amplitude waves were observed to form at the gas/water interface, with a wave height of typically less than 15% of the pipe diameter. The existence of these waves acted to cause small fluctuations in the transient response of the total interfacial area, which can be readily seen in Figure 5.10. From this figure, it can also generally observed that the maximum interfacial area observed is higher for greater water volumes, although an exception can be found in Figure 5.10 part a) for an initial water volume of 7 L. Here, a restart gas velocity of 0.1 m/s exhibits the lowest total interfacial area recorded during the transient phase until the steady state conditions are reached. This is because none of the initial water volume is evacuated from the low-spot. In contrast, for all water heights shown in Figure 5.10, a restart velocity of 1.5 m/s was observed to be enough to remove all of the water from the low-spot section. Consequently, the steady state area for these cases is zero and it is assumed that the hydrate growth is limited to the transient phase.

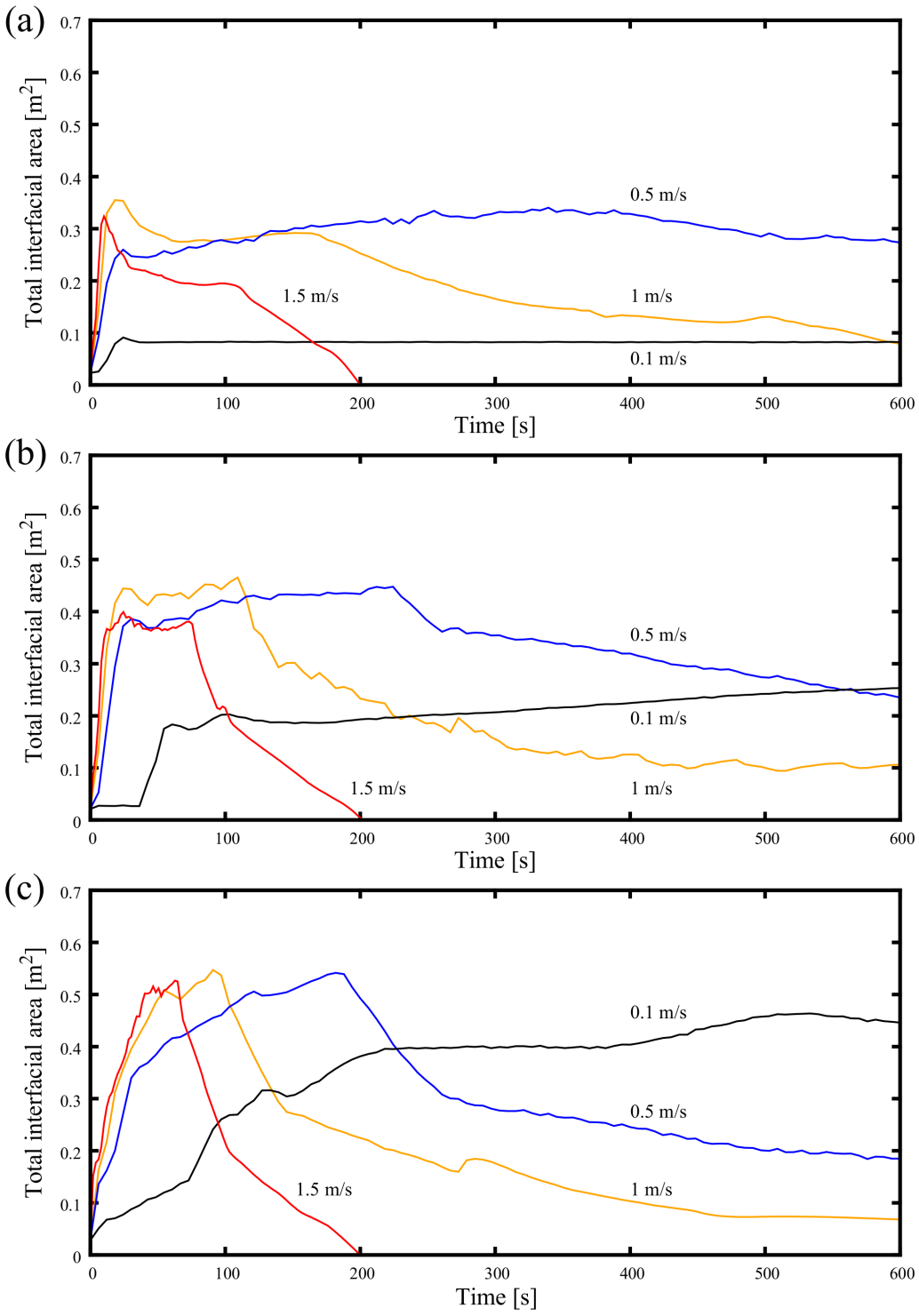


Figure 5.10: The response of the interfacial area obtained from the solver over an elapsed time of 600 s at three separate initial liquid volumes; a) 7 L, b) 22 L and c) 36 L.

5.4.4 Data Extracted from the CFD Simulations

Once the steady state condition was reached and the steady state time identified, the simulation was stopped and the area under the curve ($\int_0^{t_{ss}} A_i \Delta t$) and the steady state interfacial area (A_{ss}) were recorded from the plot of the transient response of the interfacial area. The wetted length (L_w) and the data from the upper and lower probe locations were recorded upon inspection of the CFD model using ParaView 5.0.0. An example of the complete dataset extracted from the CFD simulations is presented in Table 5.6 for 25 build points, where the interfacial area terms have been multiplied by a factor of 2 on account of the 3-D half-pipe mesh used. The dataset extracted using 16 and 49 build points can be found in Appendix D.4.

From Table 5.6, it can be seen that gas restart velocity of 1.28 m/s was the minimum required to completely remove the initial water volume from the low-spot section of the pipe, in this case 26.54 L. Here, the recorded steady state interfacial area is zero and zero values are also recorded for the upper and lower probe sets, resulting in the lowest risk for hydrate blockage to occur, with the amount of hydrate growth based solely upon an estimation during the transient phase. At all velocities lower than 1.28 m/s, a certain amount of initial water has remained in the pipe, contributing to non-zero values recorded for the interfacial area at steady state conditions. For all of these cases, it is expected that eventual blockage of the pipe will occur and in particular, instances in Table 5.6 where non-zero values for the upper probe set are recorded indicate a region of highest blockage risk, as per the definition shown in Figure 5.6 previously.

Table 5.6: The complete dataset of the data extracted from the CFD simulations using the 25 sampling point DACE study.

	CFD Input		Extracted data					
DACE point	U_g [m/s]	V_i [L]	t_{ss} [s]	$2\int_0^{t_{ss}} A_i \Delta t$ [m^{2}s]	$2A_{ss}$ [m^2]	L_w [m]	$\sum \alpha_{upper}$ [-]	$\sum \alpha_{lower}$ [-]
1	1.07	19.06	600	283	0.22	32	0.0	4.0
2	0.21	22.80	6400	4120	0.34	12	6.0	6.0
3	1.20	18.24	350	166	0.13	31	0.0	2.4
4	0.20	33.63	7350	6396	0.34	15	6.0	6.0
5	0.48	9.38	1100	505	0.31	12	1.4	6.0
6	0.86	31.53	370	370	0.19	30	0.0	4.0
7	0.91	22.00	700	325	0.19	27	0.0	4.0
8	0.15	14.85	6350	4125	0.36	10	6.0	6.0
9	0.32	25.00	4200	2436	0.31	24	5.2	6.0
10	0.35	7.95	2500	905	0.30	12	5.1	6.0
11	1.27	15.96	350	146	0.10	32	0.0	2.4
12	0.97	35.43	700	354	0.20	31	0.0	4.0
13	1.20	9.04	350	151	0.10	13	0.0	2.4
14	1.39	34.57	320	143	0.00	35	0.0	0.0
15	0.68	16.37	800	470	0.21	20	4.2	6.0
16	0.52	12.92	900	412	0.22	22	4.4	6.0
17	0.75	27.16	725	465	0.20	28	4.1	6.0
18	1.47	11.52	205	81	0.00	12	0.0	0.0
19	0.82	12.17	750	367	0.19	15	0.0	2.9
20	0.53	23.80	920	443	0.25	26	3.7	6.0
21	1.43	20.41	260	124	0.00	37	0.0	0.0
22	1.04	29.89	580	301	0.20	32	0.0	4.0
23	0.38	31.16	4700	2513	0.33	27	5.0	6.0
24	1.28	26.54	390	170	0.00	33	0.0	0.0
25	0.64	28.12	850	522	0.22	28	0.0	4.3

5.4.5 Surrogate-based Optimisation

Five separate surrogate models (S1 - S5) were constructed for each of the five objective functions shown previously in Table 5.3. The Multi-Objective Genetic Algorithm (MOGA) was used on each surrogate model based upon the maximisation and minimisation criteria for each of the objective functions shown previously in Section 5.3.2. Due to the ease of implementation, all of the available surrogate models for global optimisation in DAKOTA have been compared. The most appropriate has been selected based upon a leave-one-out cross validation strategy using 10 folds with RMS error margins calculated to assess the "goodness-of-fit" of each surrogate model constructed. The available techniques to construct surrogate models in DAKOTA include; Artificial Neural Networks (ANN), the Kriging method, Multivariate Adaptive Regression Splines (MARS) and Moving Least Squares (MLS). Table 5.7 shows the RMS error magnitudes result for all surrogate models using each of the 16, 25 and 49 sampling point datasets. Assessment of the values shown in Table 5.7 is based on the lowest value reported, where the magnitude of the RMS errors shown are of the same scale as the sampling data and the lowest values obtained for each surrogate model have been shown in red.

Table 5.7: Results of the RMS errors obtained for each surrogate model technique available in DAKOTA using the leave-one-out cross validation strategy with 10 folds. The lowest error values are highlighted in red and indicate a higher "goodness-of-fit" rating.

DACE points	Method	S1 (Δn_{trans})	S2 ($\Delta n_{ss}/\Delta t$)	S3 (L_w)	S4 ($\sum \alpha_{upper}$)	S5 ($\sum \alpha_{lower}$)
16	ANN	1736	3.06	41.68	3.81	4.23
	Kriging	924	2.19	14.37	6.41	3.75
	MARS	1050	1.74	15.94	2.90	2.41
	MLS	1391	16.24	56.45	24.13	27.44
25	ANN	2.428	0.395	5.786	1.304	0.913
	Kriging	0.747	0.391	6.207	2.021	0.977
	MARS	3.954	0.371	4.798	1.183	0.800
	MLS	26.081	3.116	58.907	21.552	10.273
49	ANN	3.498	0.248	5.433	1.516	0.872
	Kriging	5.172	0.247	5.129	1.370	0.566
	MARS	2.964	0.247	5.669	1.404	0.740
	MLS	13.905	2.145	35.442	16.837	6.362

From inspection of the RMS errors recorded in Table 5.7, it is immediately apparent that it is not feasible to construct any of the surrogate models using 16 build points, as the recorded RMS results are substantially higher in all but one instance when compared the surrogate models built from 25 and 49 points. In particular, the error magnitudes recorded for the S1 and S2 surrogate models using 16 build points are many orders of magnitude higher than the errors observed using 25 and 49 build points. The RMS error results using 25 and 49 build points show that the MARS technique has outperformed the other techniques for the S2 to S5 surrogate models using 25 build points. However, for the S1 surrogate model, the Kriging method has shown substantially superior results, whereby the recorded RMS error is lower than the value obtained from the MARS technique by a factor of over 5. Using 49 build points, the Kriging method has performed marginally better for all surrogate models with the exception of the first surrogate model, S1, in which the MARS technique has exhibited the lowest RMS error margin.

It is evident from Table 5.8 that the addition of more build points, based upon an incremental rise in the number of symbols when using the OA-LHS technique has only had a marginally positive effect on the robustness of the surrogate models constructed, as the instances of improved robustness in some of the surrogate models are offset by a deterioration of the robustness in other models constructed from the same technique. For example, the use of 49 build points leads to higher RMS error values recorded for the S1 model when using the Kriging method, the S3 and S4 models when using the MARS technique and the S4 model when using the ANN technique.

In light of the expensive nature of the computational simulations, it would seem that it is more appropriate to use 25 build points as a starting point, whereby further refinement of the model can come from manually adding build points to specific regions of interest, or points in the design space where the distance between the original build points is greatest. Subsequently, the Kriging surrogate modelling technique has been used with 25 build points to construct each of the surrogate models from this study. The Kriging method was chosen based upon the substantially superior RMS results for the S1 surrogate and comparable results for the S2 to S5 surrogate models at 25 build points. A representation for each of the constructed S1 to S5 surrogate models throughout the design space is shown in Figure 5.11 parts a) to e).

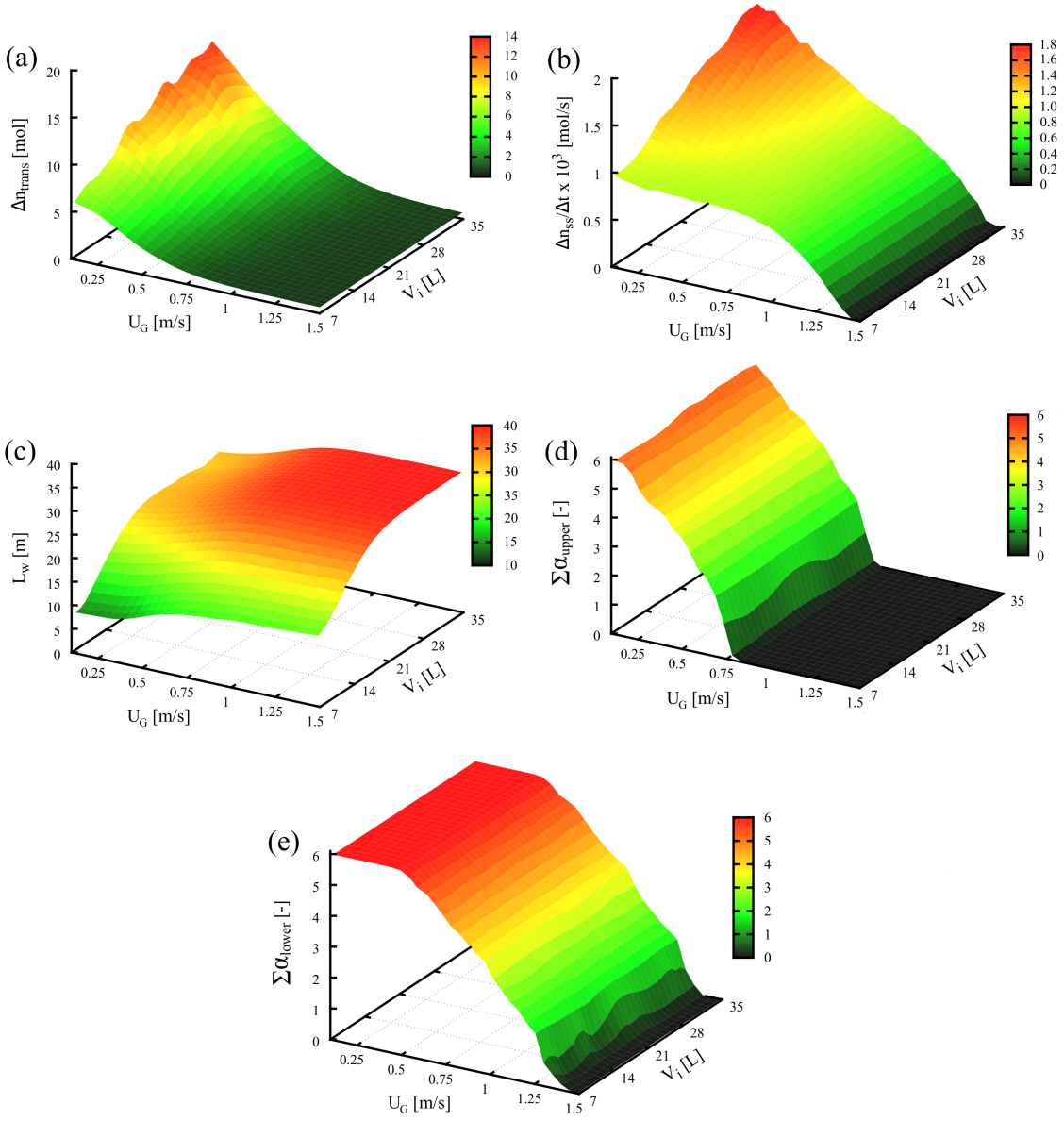


Figure 5.11: A depiction of the S1 - S5 surrogate model responses over the design space; a) molar gas consumption during the transient phase (S1 - Δn_{trans}), b) rate of molar gas consumption during the steady state phase (S2 - $\Delta n_{ss}/\Delta t$), c) wetted length (S3 - L_w), d) summation of the upper liquid phase fraction (S4 - $\sum \alpha_{upper}$) and e) summation of the lower liquid phase fraction (S5 - $\sum \alpha_{lower}$).

For the purposes of visualisation, two additional response surfaces have been constructed to display the results in terms of the thickness of the hydrate layer at steady-state conditions (R1) and the blockage half-time (R2). These two response surfaces are shown in Figure 5.12 parts a) and b), respectively. Following this, the results from the S4 and S5 liquid phase fraction surrogate models shown previously in Figure 5.11 parts d) and e), respectively have been mapped onto the design space in 2D in order to visually represent the three separate regions of hydrate blockage risk defined in Figure 5.6. This depiction is shown in Figure 5.12 part c), along with the location of the evaluation points generated from MOGA.

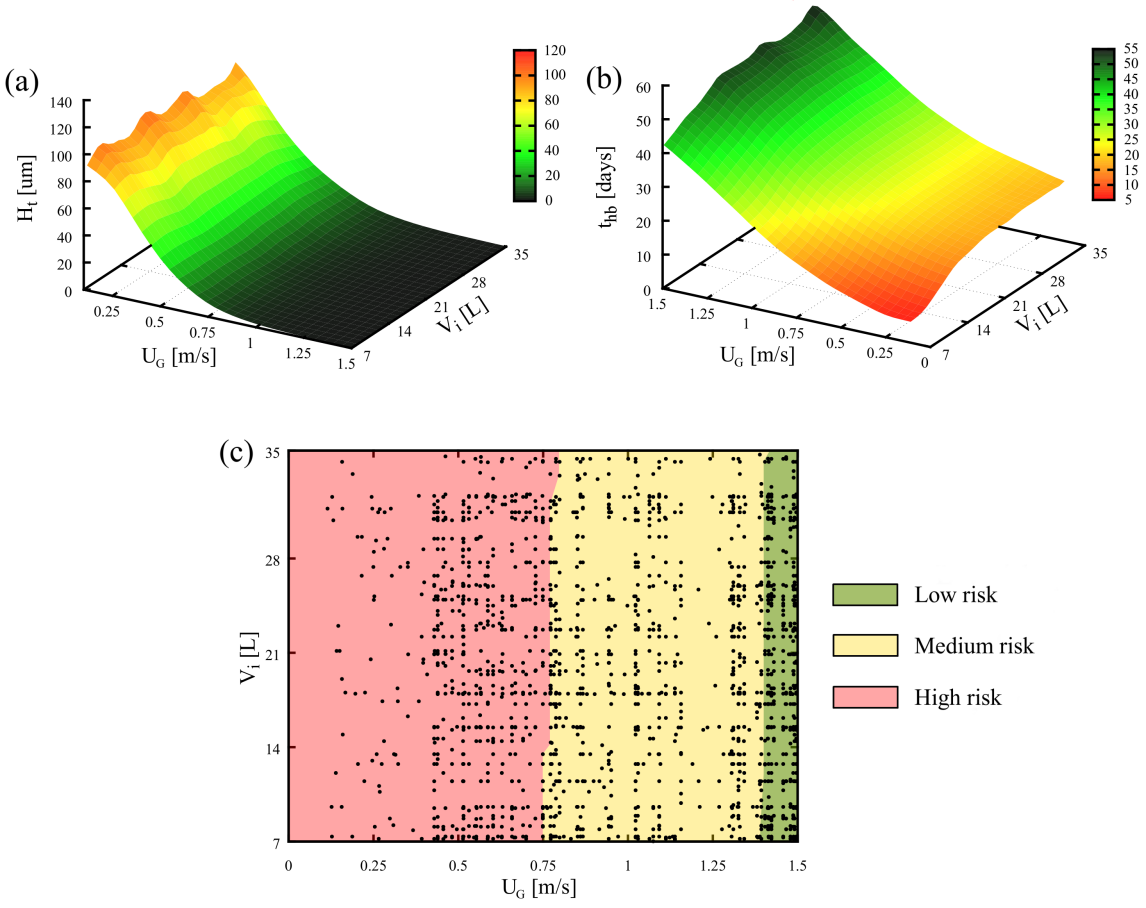


Figure 5.12: The two response surfaces constructed from the surrogate models; where a) the thickness of the hydrate layer at steady state conditions (H1 - H_t) and b) the half-block time (H2 - t_{h-b}). Part c) shows the hydrate blockage risk zones defined by probe data from the surrogate models S4 and S5 in comparison to the evaluation points from MOGA.

5.4.6 Comparison of the Hydrate Models Used

The relative contributions to the S1 and S2 surrogate models from both of the kinetic and mass transfer limited models are presented in Figure 5.13. Here, Figure 5.13 parts a) and b) refer to the amount of hydrate growth in the transient phase and the hydrate growth rate during steady state conditions, respectively, when using the kinetic hydrate model by Turner et al. [2005]. Likewise, the same results are presented in Figure 5.13 parts c) and d) when using the mass transfer limited model by Skovborg and Rasmussen [1994].

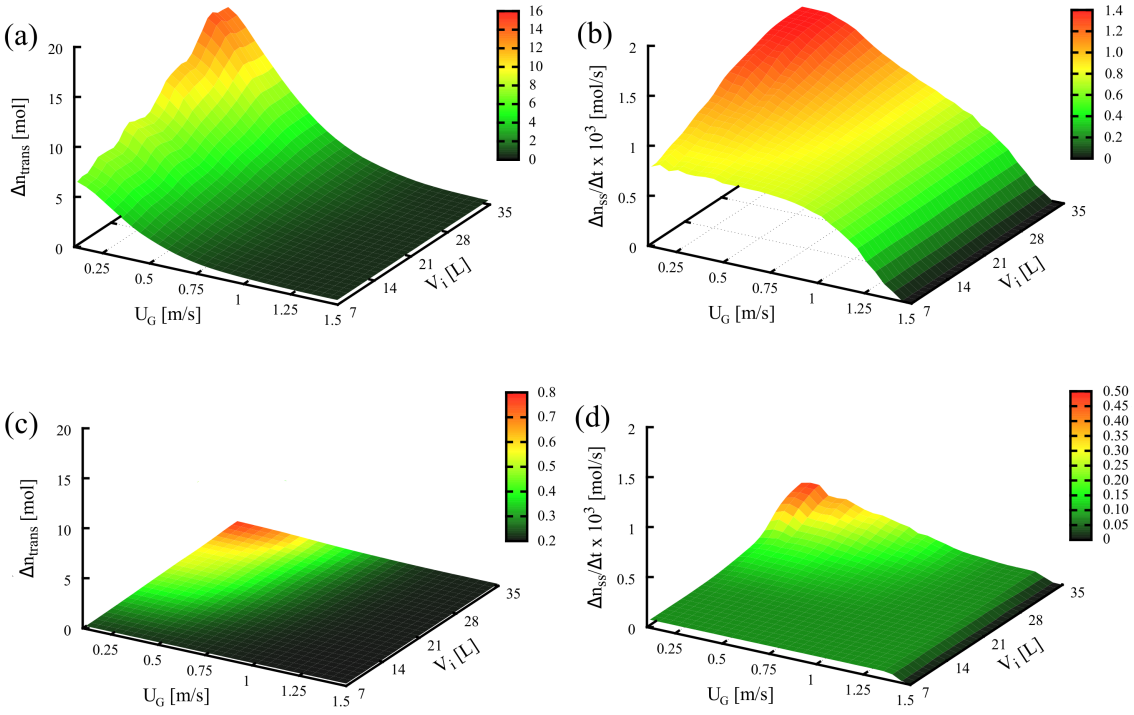


Figure 5.13: A comparison of the contribution to the S1 and S2 surrogate models from both the kinetic and mass transfer limited models, respectively; a) and b) the transient molar gas consumption and the rate of molar gas consumption using solely the kinetic hydrate model, and c) and d) the same results generated using solely the mass transfer limited model.

As the degree of subcooling in this study is assumed to be constant, the amount of growth estimated using the kinetic model is directly proportional to the total interfacial area recorded. This is in contrast to the mass transfer limited model, which is further dependent on the gas restart velocity directly via the mass transfer coefficient in Equation 5.3. As a result, substantially lower values for both the transient hydrate

growth and the steady state hydrate growth rate have been recorded using the mass transfer limited model when compared to the kinetic model at lower gas velocities. This is because instances where the interfacial area is high that typically occur at lower gas velocities are offset by reduced values recorded for the mass transfer coefficient, leading to the substantially lower values recorded for the transient hydrate growth and the steady state hydrate growth rate observable in Figure 5.13.

At gas restart velocities greater than 1 m/s, the difference between each model is reduced substantially although the amount of hydrate growth estimated using the mass transfer model was still typically lower than the kinetic model estimation by a factor of 4. This difference appears to be fairly consistent to the results presented by Aman et al. [2016] for gas-dominated annular flow, although a direct comparison cannot be made on account of the different flow conditions and degree of subcooling used.

5.4.7 Computational Time-scale

In order to assess the capability of the method presented in this study to be used as an practical investigative tool, the time scale required per simulation at each sampling point in the design space has been averaged for all initial water volumes using data obtained from the optimisation study and is presented in Figure 5.14, along with additional results from higher velocities cases until a maximum gas velocity of 4 m/s. Here, all computational time results are based upon the use of a single quad-core i7-4970 processor running in parallel.

The length of computational time required is dictated by the fact that each simulation is run until the steady state conditions are reached, whereby higher gas velocities have greatly reduced transient times that offset the reduction in time-step required for the solution to remain stable, leading to reduced computational times. For an optimisation study using 25 build points, the average time required was approximately 100 hours per simulation, which equates to slightly more than 104 days of processing time using a single workstation. This shows the highly computationally expensive nature of the simulations even when using the modified modelling strategy and highlights the necessity to use parallel computing options for longer flowline sections.

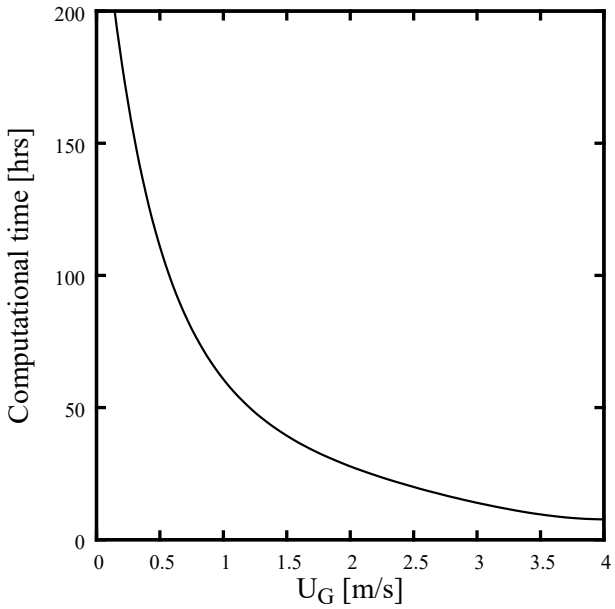


Figure 5.14: The average computational time recorded until steady state conditions are reached, depending on the restart gas velocity.

5.5 Discussion

5.5.1 Modelling Hydrate Growth

The main premise of this study has involved the use of CFD to describe a two-phase gas-dominated restart operation, emulating the experimental setup used by Leporcher et al. [2002], whereby the transient response of the interfacial area has been used to provide input to the kinetic hydrate growth model by Turner et al. [2005] and the mass transfer limited hydrate growth model by Skovborg and Rasmussen [1994]. These two models have been combined to provide an estimation of the amount of hydrate growth during the transient phase, and the hydrate growth rate during steady state conditions. Although it is recognised that authors such as Sloan and Koh [2007] have suggested that the Skovborg and Rasmussen [1994] model is best described as a data-fit containing systematic driving force errors, it has been used in the absence of experimental data as a means to assess the impact that the variable gas restart velocities used in this study may have on the hydrate growth recorded in the steady state and transient phases of a gas-dominated restart.

The comparison of the hydrate growth results obtained from the kinetic and mass transfer limited models presented in Figure 12 of this study suggests that the influence

of the mass transfer model is relatively minor when compared to the magnitude of the estimation obtained from the kinetic model. These results appear to be fairly consistent to the studies by Di Lorenzo et al. [2014] and Aman et al. [2016] for gas-dominated annular flow, although again, it is recognised that authors such as Sloan and Koh [2007] have suggested that intrinsic hydrate kinetics may play a much smaller role for hydrate growth in actual systems when compared to the effects of mass and heat transfer. Nevertheless, as the hydrate models used in this study are based upon the interfacial area between the hydrate forming components and the water phase, extracted from the CFD simulations, it is foreseeable that the approach used in this study could be easily adapted if a more suitable method is available.

The main point of contention with the estimation of the hydrate growth is undoubtedly due to the use of the wetted length to define the deposition length of the hydrate. It is expected that this approach will lead to a conservative estimate of the hydrate layer thickness and has been selected for two reasons; to offset the assumption that no localised hydrate buildup occurs and; in lieu of a more suitable deposition model, as the deposition distance calculated using current models depends on an estimation of the initial diameter of the hydrate particle, with vastly different results returned when compared to small increments of particle diameter input. It is expected that this parameter may be vastly improved with future work, such as the method proposed by Balakin et al. [2016], where CFD was coupled with a population-based method to simulate the agglomeration and deposition of hydrate particles using a turbulent slurry of oil, water and hydrate material.

5.5.2 Modification of the interFoam Solver

The interFoam solver has been modified to calculate the total interfacial area throughout the domain at each timestep. The method of calculation for the maximum interfacial area in each cell is based upon the method used by Thiele [2010]. This method has been validated by comparing the calculated interfacial area to the modified solver output, as shown in Section 5.4.1. The results for three of the four cases considered are highly comparable with error magnitudes $< 2\%$, while the remaining case (FN0140) has an error margin $< 10\%$. It is not explicitly known why this particular test has exhibited an increased error margin; however, this may possibly be attributed to the vertical spacing of each cell in the cross section. As the vertical spacing is not uniform for cells near the inner wall, at certain points the maximum cell volume will change, which would reflect in the value of the interfacial area recorded for that cell.

As a result, variations in the liquid height throughout the pipe may cause amplified variations in the interfacial area recorded by the solver. Another source of error may be due to the slight longitudinal bias for the values of the interfacial area recorded in each cell if the actual gas-water boundary is not horizontal. In this case the interfacial area calculation will definitely not follow the theoretical area calculation from Equation 5.14; however, as the model length to height ratio is large, and it has been shown that the flows are either stratified or slug, it is expected that this should result in only minimal errors and should act to provide a small conservative estimate of the interfacial area.

5.5.3 Hydrate Risk Management Using an Optimisation Approach

An optimisation study has successfully been performed to determine optimal gas restart velocities, depending on the initial water volume and defined hydrate blockage risk. As a means to directly offset the computationally expensive nature of the CFD simulations involved, a surrogate-based optimisation methodology has been used to project the results obtained from the surrogate build points generated using a DACE study over the entire design space, which was bounded by values of the gas restart velocity and initial amount of water held in the pipe. In order to determine the most appropriate surrogate model for use in the present application, four of the different types of surrogate modelling techniques available in DAKOTA were assessed. The build points used to construct the surrogate models were generated using the OA-LHS technique for three sampling sizes of 16, 25 and 49 and the assessment of the robustness for each surrogate model constructed was achieved using a 10-fold leave-one-out cross validation technique.

Although at 25 build points, the MARS technique was shown to be the most robust in terms of the RMS error values recorded for the S2 to S5 surrogate models, the Kriging technique was used due to the vastly superior RMS error values recorded for the S1 surrogate and superior RMS errors recorded at the 49 build points when compared to the other surrogate models as shown in Table 5.7. It is evident that the use of 49 build points has offered only a slight improvement of the RMS errors recorded for most of the surrogate models when compared to the surrogates generated from 25 build points, whereby any improvement observed for a particular model is often offset by an increase in the RMS error values recorded in other surrogate models when using the same surrogate modelling technique. The use of 16 build points was

deemed unacceptable due to the magnitude of the RMS errors recorded in Table 5.7 and is not recommended for use in the framework. Instead, the use of 25 build points is recommended, where additional build points can be added manually to specific regions of interest if further refinement of the surrogate model is required.

The S1 to S3 surrogate models have been used to construct two response surfaces of the half-block time and thickness of the hydrate layer at steady state conditions. The half-block time has been used as current particle jamming models require advance knowledge of the particle diameter and the probability of a blockage occurring, both of which are derived from empirical data. The results from the half-block time response surface presented in Figure 5.12 part b) show that lower velocities contribute to faster times for hydrate blockage to occur, which is consistent with the observations made by Leporcher et al. [2002]. Leporcher et al. [2002] has reported only one applicable blockage time for the air/water system where a blockage occurred at an elapsed time of 4 hours for a restart gas velocity of 1 mm/s. At this gas velocity, bubbles would disperse through the stagnant water region. As a gas restart velocity of 1 mm/s was considered too low for any practical use for a restart operation and bubbling effects would not be captured by the CFD simulation, the minimum value of the gas restart velocity used in this study is greater by a factor of 100. As such, it is expected that the blockage time is a parameter that can be further improved with future studies.

Lastly, the risk profile for a hydrate blockage to occur for the single low-spot pipe geometry used in this study has been generated based upon the hydrodynamic conditions at the upper leg of the low-spot section. The results presented in Figure 11 part c) have shown that the boundaries that define the separate risk zones remain fairly uniform in terms of the restart velocity, regardless of variations in the initial water volume. In particular, it can be seen that a gas restart velocity of approximately 1.3 m/s or greater would be considered "safe" for any amount of initial water volume. This suggests that the computational expense may be minimised by considering only one water level, which would drastically reduce the number of simulations required, although variations may become more pronounced for flowlines featuring multiple low-spots.

5.5.4 Applications for Actual Flowlines

It is apparent that an optimisation approach utilising CFD to describe the flow could potentially offer a greater insight realising a goal of providing hydrate-blockage free restarts operations; however, in order for the optimisation strategy shown in this study to be used for actual flowline applications, a number of important areas must first be considered and addressed appropriately, these include;

- Implementation of pressure driven flow as opposed to velocity to emulate the pressure controlled at the wellhead choke by the operator.
- Implementation of the temperature as an additional variable to cater for changes in the Pressure-Temperature (PT) conditions throughout the flowline.
- Implementation of phase change behaviour within the flowline.
- Investigation of the diameter sizing on the characteristics of the flow.

The first point is not a foreseeable issue as the velocity driven flow used in this study may easily be changed to a pressure driven flow. Using OpenFOAM®, this is accomplished by swapping the boundary conditions for the U and p_rgh fields. Temperature effects can be added to the kinetic model by Turner et al. [2005] by defining the system temperature in Equation 5.2 as an additional variable, and by modifying the interFoam solver to include a thermal transport equation, whereby the interfacial temperature is recorded at each time step and is used in conjunction with the interfacial area as a means of increasing the fidelity of the kinetic model. In addition, thermally driven phase behaviour may potentially be included using interThermalPhaseChangeFoam [Rattner and Garimella, 2014], although the mesh resolution and applicable validation techniques available will undoubtedly be a limiting factor in simulating the phase behaviour accurately. A way to achieve this is possible if prior knowledge of the likely locations for phase change to occur, whereby the mesh resolution can be increased at these locations.

The effect of diameter is a highlighted area for future research as Leporcher et al. [2002] have proposed that larger diameters than the one used in this study may actually be less prone to direct blockage. It is possible that the interfacial behaviour observed in this study does not apply to the larger diameter flowlines used in practice. Larger diameters are expected to have greater liquid loadings where it is likely that bubbling effects may become more significant, leading to increased hydrate formation locally that will not be captured when using the current modelling strategy.

5.6 Summary and Conclusions

In this chapter, the optimisation techniques available in DAKOTA have been employed for the potential application of reducing the risk of hydrate blockage in a flowline during gas-dominated restart operations. Based upon the experimental study of Leporcher et al. [2002], a restart operation has been emulated using CFD for a two-phase flow of gas and water. The hydrate growth models by Turner et al. [2005] and Skovborg and Rasmussen [1994] have been coupled to a modified version of the interFoam solver to determine the amount of hydrate growth during both the transient and steady state phases of the restart operation and surrogate models have been used to project the CFD response generated at the build points over the entire design space. Two response surfaces have been constructed to visualise the results in terms of thickness of the hydrate layer and the defined blockage half-time. Three separate regions of hydrate blockage risk have been categorised here based upon the restart gas velocity and the amount of accumulated water initially held in the pipe. Lastly, a strategy has been suggested to adapt the approach used in this study to produce meaningful results for actual flowline sections. Following the work completed in this chapter, the following conclusions have been made;

- The coupling of SALOME, OpenFOAM® and DAKOTA for the purposes of constructing an optimisation framework has been successful, whereby the transient interfacial area has been extracted from a modified version of the interFoam solver and used to provide an estimation of the hydrate growth.
- The usage of the OA-LHS DACE technique, coupled with the Kriging surrogate modelling technique and MOGA has been successfully demonstrated and will produce acceptable results for the framework.
- The influence of the mass transfer limited hydrate growth model is minor when compared to the kinetic model used, as instances where the interfacial area is high are offset by reduced gas velocities, contributing to lower values of the mass transfer coefficient.
- The proposed optimisation approach has produced results that are consistent with the observations made by Leporcher et al. [2002], whereby low gas velocity restarts were seen to constitute the highest risk of hydrate blockage with reduced blockage times.

- Further improvements are needed to be made in terms of hydrate deposition and particle jamming, which have been simplified in this chapter using the wetted length and half-block time, respectively.
- Further investigations are required to determine the impact of larger pipe diameters and phase change behaviour.

Chapter 6

Applied Optimisation Example: Ormen Lange

In this chapter, the CFD-based optimisation methodology presented in Chapter 5 is further refined through the use of an applied example. The particular example selected is the Ormen Lange natural gas field located offshore of Norway, which has been chosen due to the reportedly challenging nature of providing flow assurance for this field and the gas-dominated characteristics of the two flowlines involved. Through this example, the most appropriate means of applying an optimisation strategy to extended flowline lengths with multiple low-spot bends are investigated for the purpose of constructing the finalised form of the optimisation framework. The work presented in this chapter constitutes the culmination of work required to achieve Objective 4 from this study.

6.1 Problem Description

6.1.1 Ormen Lange Overview

Ormen Lange is a gas-dominated field located offshore of Norway in water depths between 850 to 1100 metres. Gas and condensate from the reservoir flow through four templates located on the seabed to an onshore terminal at Nyhamna via two approximately 120 km long flowlines of 30" diameter. These two flowlines are connected as a loop, allowing de-watering and cleaning pigs to be launched and collected from Nyhamna. After processing at Nyhamna, the gas is exported to the UK using the Langeled pipeline, and can supply up to 20% of the UK's peak gas requirements [Gupta, 2012]. From a flow assurance perspective, Ormen Lange is reportedly one of the most challenging field developments worldwide [Wilson et al., 2004]. The difficulty in providing adequate flow assurance for this field arises from a combination

of subzero ambient temperatures and extremely hilly terrain, including the Storegga slide, a prehistoric slide area with a vertical height of over 200 m. In order to alleviate the risk of hydrate blockage, a constant supply of 60 wt% MEG inhibitor is required during normal production as a means to prevent the formation of hydrates entirely [Holm, 2005].

Flow assurance for the Ormen Lange field is currently facilitated using the proprietary Flow Assurance System (FAS), a comprehensive simulator that is used to provide real-time information about the multiphase flow characteristics throughout the entire subsea production and flowline system [Holmas et al., 2013]. Integral to the FAS is the commercial software FlowManager^{TM} Dynamic, which is used to simulate the transient multiphase flow using a 1-D drift flux model with algebraic slip and friction relations derived from field data, an approach that has been shown to yield accurate predictions of pressure, temperature and flow rates of gas, condensate and water/MEG in the wells, templates, flowlines and slug catchers [Holmas et al., 2013]. An overview of the Ormen Lange field development is shown in Figure 6.1 and the main parameters for the Ormen Lange flowline, obtained from a variety of available sources are presented in Table 6.1.

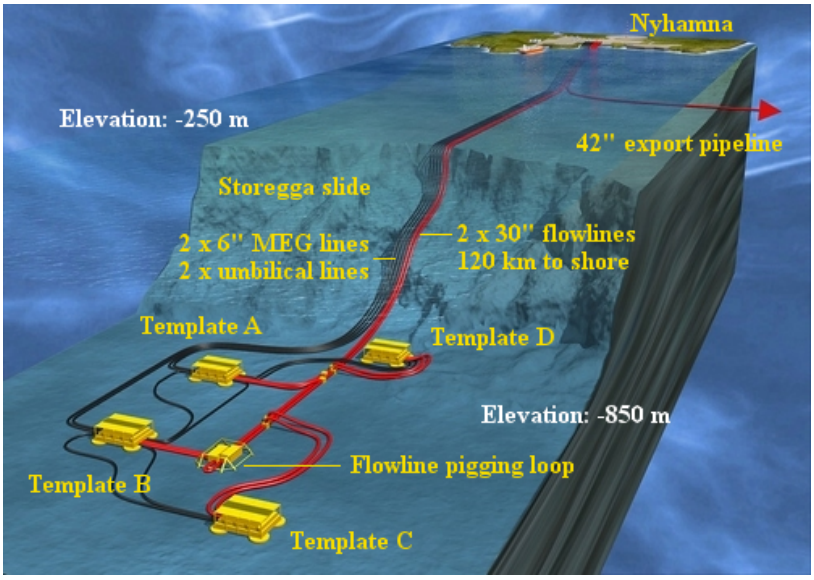


Figure 6.1: An overview of the Ormen Lange field, with the two 30" flowlines shown in red (adapted from Aarvik et al. [2007] with additional input from Gupta [2012]).

Table 6.1: The main parameters of the Ormen Lange field.

Parameter	Value	Source
Operational envelope (per flowline)	20 - 35 [Msm^3/sd]	Holm [2005]
Flowline operating pressure	90 - 255 [bar]	Burns et al. [2008]
Condensate-Gas Ratio (CGR)	1e-04 [Sm^3/Sm^3]	Holm [2005]
Horizontal length	120 [km]	Holm [2005]
Outer pipe diameter	0.762 [m]	Holm [2005]
Pipe thickness	36 [mm]	Røsdal [2008]
Maximum pipe inclination	35 [°]	Holmas et al. [2013]

As the CGR shown in Table 6.1 is low, it is assumed that limited slippage exists between the water and condensate phases and these two phases can be modelled as a single phase, as discussed previously in Section 2.4. In addition, Holm [2005] has reported that the flow is predominately stratified during normal operations, with local slug flow occurring at flows less than 20 Msm^3/sd. This provides a good indication that the modified modelling strategy presented in Chapter 4 can be used to capture the flow and provide a reasonable response of the interfacial area between the gas and water phases during the restart operation. In addition, the usage of a 1-D drift flux model as the basis for the FAS presents an opportunity for the CFD-based optimisation approach proposed in this dissertation to provide greater insight into the transient characteristics of the Ormen Lange flowlines during restart operations, although in the absence of verifiable data, it is recognised that further tuning of the CFD model may be required to achieve the same level of accuracy. In light of this, it is intended that the optimisation approach used in this study is complimentary in nature to the current 1-D drift flux models used in industry, and will be applicable regardless of the simulation tool.

6.1.2 Flowline Profile

The flowline profile used in this study has been obtained from Holm [2005] and is reproduced in Figure 6.2, together with the most likely locations for a hydrate blockage to occur during flowing and non-flowing conditions, as identified by Holm [2005]. This flowline profile has an actual horizontal distance of 118 km, terminating just before the riser section becomes horizontal. The near-vertical elevation change observable in Figure 6.2 at a horizontal distance of approximately 25 km is due to the Storegga slide.

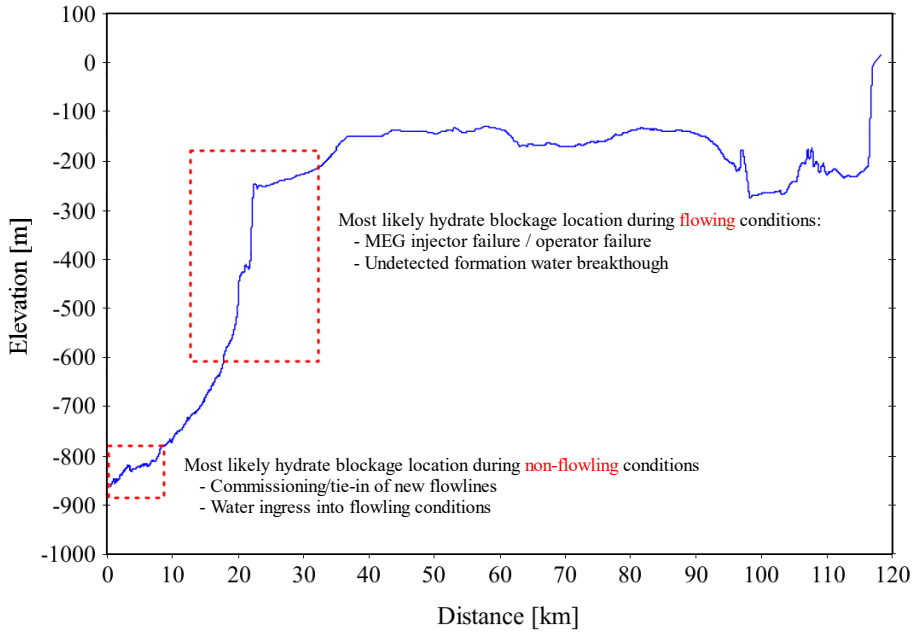


Figure 6.2: The profile of the two Ormen Lange flowlines showing the most likely locations for a hydrate blockage to occur during flowing and non-flowing conditions as identified by Holm [2005] (image adapted from Holm [2005]).

According to Holm [2005], during non-flowing conditions a hydrate blockage will most likely occur due to water ingress at the flowline inlet, which is a possibility during tie-in of new flowline sections and would need to be mitigated through the use of de-watering pigs and chemical inhibition. In contrast, during flowing conditions, hydrate blockages are most likely to occur at the Storegga slide area if either the amount of inhibitor is insufficient or the formation water amount has increased unexpectedly. As this is at the point of the greatest inclination, this is most likely due to the gas flow moving across a stagnant water region as identified in the experimental flowloop study of Leporcher et al. [2002] and used as the premise for defining the hydrate blockage risk in this dissertation. This approach is also used for this particular example, where it is assumed that no water ingress or water breakthrough occurs, and the blockage risk is determined based upon a combination of the amount of hydrate growth and the presence of water in the upward legs of the flowline, as discussed in the previous chapter.

6.1.3 Gas Composition

The gas composition has been obtained from a three-phase flow assurance study of the Ormen Lange flowlines presented by Bratland [2015] and is shown in Table 6.2, whereby the molar fraction for Pentane and heavier components that commonly constitute the condensate and oil phases have been simplified into a "C5+" category.

Table 6.2: The gas composition of the Ormen Lange flowline used in this study, taken from Bratland [2015].

Component	Mass fraction (%)
N2	0.338
CO2	0.418
H20	0.551
C1	91.945
C2	3.430
C3	1.203
iC4	0.268
nC4	0.318
C5+	1.529

6.1.4 Pressure Drop

As the intention for the applied optimisation study is to use pressure driven flow, prior knowledge of the expected pressure drop though the flowline is required to obtain the magnitude of the pressure driving force. The pressure drop for the Ormen Lange flowlines has been presented by Burns et al. [2008] for a range of gas flow rates and is reproduced here in Figure 6.3. In this figure, the recorded field data (represented as point data) has been compared to simulation results from a steady state multiphase model (represented as the solid lines) showing the pressure drops representative of both of the Ormen Lange flowlines at two separate slugcatcher pressures of 90 and 100 bar.

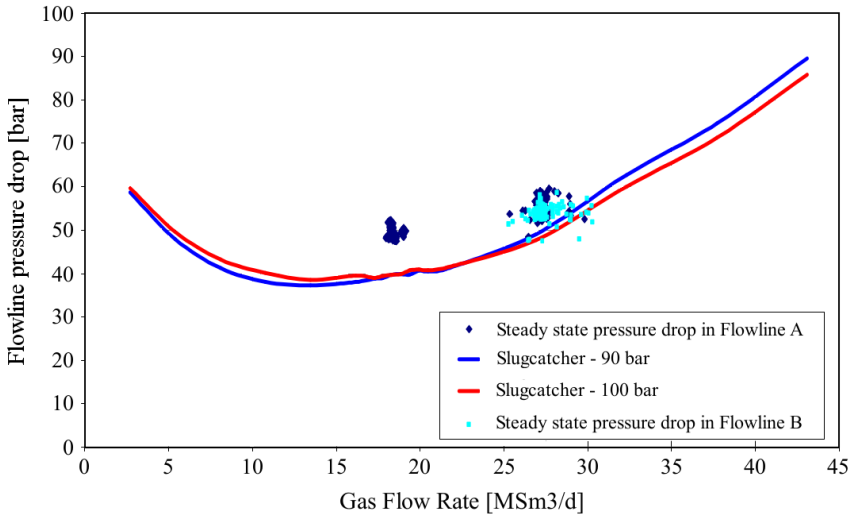


Figure 6.3: Ormen Lange pressure drop data (adapted from Burns et al. [2008]).

As the field data presented in Figure 6.3 has only been shown for a short range of gas flow rates, the results from the steady state model have been used to derive a suitable range of the pressure driving force for use in this study, where the following two assumptions have been made;

- A slugcatcher pressure of 90 bar has been assumed in all cases, which has been taken as the flowline outlet pressure of the CFD model.
- Any increase in the pressure drop contributes to an increase in the gas flow rate, such that gas flow rates less than 15 MSm^3/d are not considered.

Using these assumptions, a pressure drop range of between 40 to 90 bar has been derived from Figure 6.3, corresponding to a range of gas flow rates between 15 to 43 Msm^3/d. Thus, based upon a slugcatcher pressure of 90 bar, an inlet pressure range of between 130 to 180 bar is used in this study.

6.1.5 System Temperature

The system temperature for five separate production flow rates has been presented by Holm [2005] and is shown in Figure 6.4 in comparison to the flowline profile.

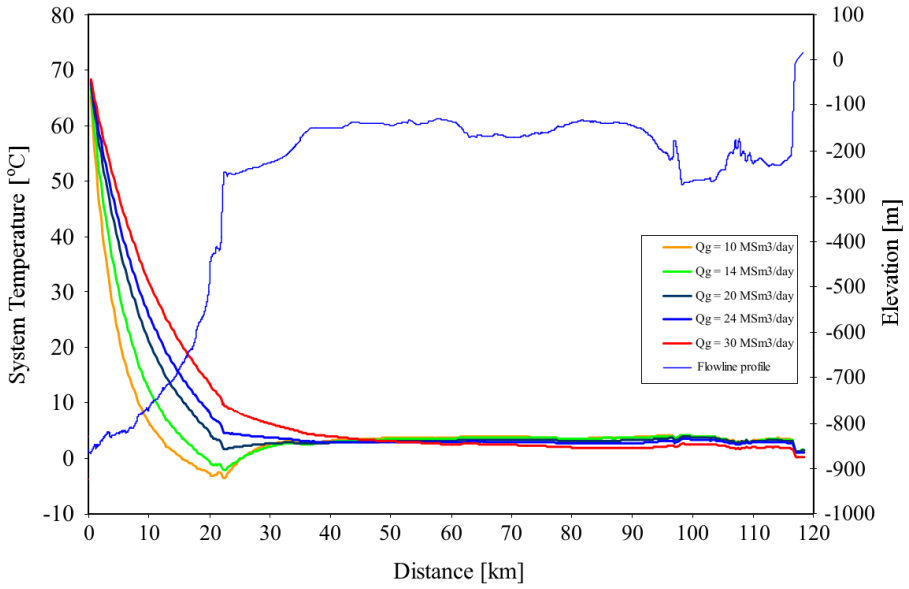


Figure 6.4: The system temperature profile of the Ormen Lange flowlines in comparison to the flowline profile at five separate production flowrates (adapted from Holm [2005]).

From Figure 6.4, the initial temperature at the flowline inlet is approximately 68°C for all flowrates and is governed by the reservoir temperature. As the inlet fluid is transported along the flowline length, the temperature rapidly decreases due to heat transfer to the surrounding water until a horizontal distance of approximately 20 km. During this temperature decrease, a divergence can be seen in the rate of temperature loss depending on the flow rate, whereby faster moving flows are able to maintain heat for a further distance along the flowline when compared to slower moving flows. This divergence in temperature occurs until a horizontal distance of approximately 22 km, whereby the system temperature for all flowrates begins to converge at greater distances until a location slightly above 40 km from the flowline inlet. From this distance until the topside location at 118 km, the temperature is equivalent for all flowrates at approximately 2°C.

The temperature profile shown in Figure 6.4 is during flowing conditions, whereas during an extended shutdown period the entire flowline can be considered to be equivalent to the surrounding ambient temperature at the sea floor, where the ambient sea water temperature for the Ormen Lange field presented by Holm [2005] is reproduced in Figure 6.5.

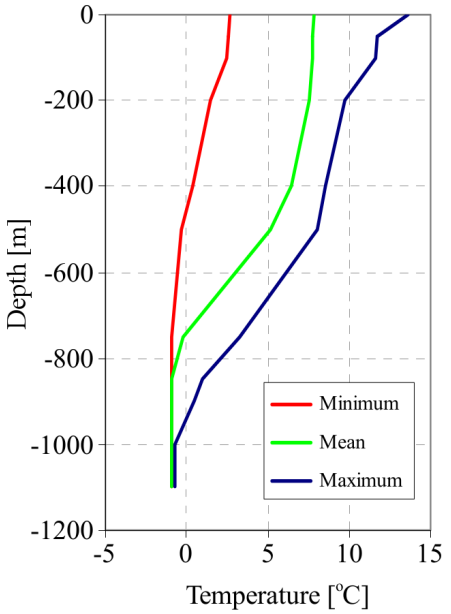


Figure 6.5: The ambient sea water temperature profile the Ormen Lange field (adapted from Holm [2005]).

From Figure 6.5, the coldest temperature experienced is approximately -2°C at the maximum depth of 1100 m. In order to emulate the worst case scenario for hydrate blockage, it is assumed that the system temperature is equivalent to -2°C throughout the entire flowline. As the degree of subcooling will influence both the kinetic and mass transfer hydrate models, it is expected that this assumption will lead to a conservative estimate of the amount of hydrate growth. It is also recognised that a constant temperature will likely cause discrepancies in the properties of the gas phase during the first 40 km of the model length as shown in Figure 6.4. This is due to the heated gas front from the reservoir interacting with the cooled fluid residing in the flowline upon commencement of the restart operation. The complex nature of this thermal interaction has not been considered in this study, although will need to be addressed in future versions of the framework.

6.1.6 Steady State Liquid Holdup

The amount of liquid contained in the Ormen Lange flowlines during steady state operations is quantified using simulations obtained from the FAS. The simulated amount of liquid holdup for a range of production flowrates at three slugcatcher

pressures has been presented by Wilson et al. [2004] and is reproduced here in Figure 6.6.

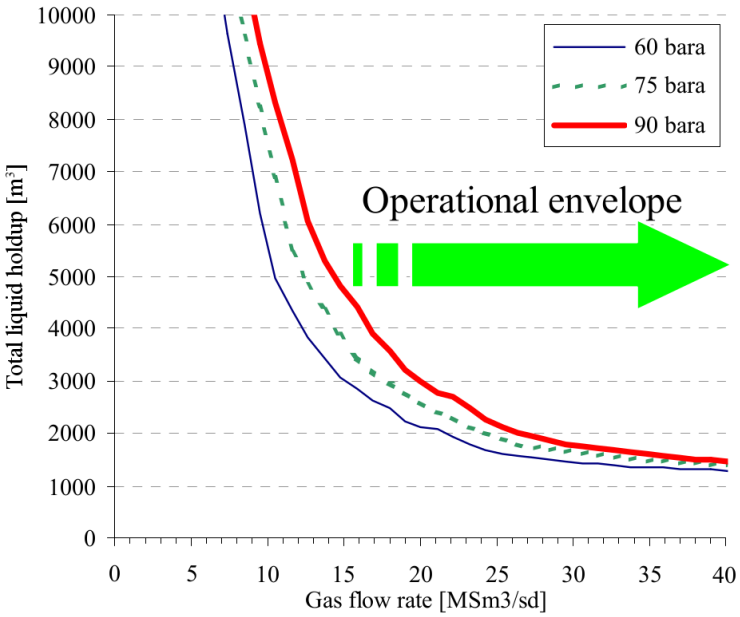


Figure 6.6: The simulated liquid holdup for a range of production flowrates at three slucatcher pressures (adapted from Wilson et al. [2004]).

Using Figure 6.6, for a gas flow rate range between 15 to 43 Msm^3/d, the expected liquid holdup lies within a range of between 5000 m^3 to 1500 m^3, assuming a slugcatcher pressure of 90 bara. By considering the calculated flowline volume is equivalent to approximately 45,500 m^3, an assumed liquid holdup amount between 1500 m^2 and 5000 m^2 corresponds to a liquid volume occupying a range of approximately 3.2 to 11.0% of the flowline volume at normal steady state operations. In this study, a range of between 5 to 25% of the flowline volume has been assumed to represent the amount of accumulated water initially contained within the flowline prior to a restart operation. This range has been extended significantly from the maximum 11.0% expected during normal operating conditions as a means to cater for unforeseen circumstances were the gas flowrate prior to shutdown could be considerably lower, resulting in an increased amount of liquid holdup. Similarly, a lower limit of 5% has been selected as initial water volumes less than this are likely to result in a reduced interfacial area recorded between the gas and water phases, constituting a low risk of hydrate blockage to occur.

6.1.7 Derivation of the System Parameters

In a similar fashion to the previous chapter, the gas composition shown in Table 6.2 can be used to derive the density and viscosity of the gas phase at the system pressure and temperature for use in OpenFOAM®. However, in contrast to the previous chapter where a constant internal pressure and temperature was used to emulate the experimental flowloop setup of Leporcher et al. [2002], the calculation for the correct density and viscosity of the gas phase for an actual flowline becomes increasingly challenging as both the system pressure and temperature are now dependant on the length of the flowline from the reservoir. Although it has been assumed that the temperature remains constant at -2°C, a strategy must be developed in the optimisation framework to include the provision for variations of the density and viscosity of the gas phase depending on the magnitude of the inlet pressure. In order to achieve this, the following steps have been proposed to be added to the general optimisation workflow contained within the finalised optimisation framework;

1. Initiate the DACE study using the approach presented in the previous chapter for a design space bound by a predetermined range of inlet pressures and initial water volumes.

2. Obtain the specific inlet pressures at each of the surrogate build points generated from the DACE study.

3. Estimate the average system pressure of the flowline at each simulation point by assuming a constant slugcatcher pressure, in this case 90 bar.

4. Using the average system pressure, estimate the gas density and viscosity at each build point.

5. Run the CFD simulations at the build points with the updated gas properties and extract the relevant data to complete the DACE study.

Here, the average pressure between the inlet and outlet pressures can be estimated prior to commencement of each simulation by simply using;

$$P_{avg} = \frac{P_{inlet} + P_{outlet}}{2} \tag{6.1}$$

From Equation 6.1, P_{avg} is the average pressure, P_{inlet} is the inlet pressure ranging from 130 - 180 bar and P_{outlet} is the outlet pressure which is fixed at 90 bar as mentioned previously in Section 6.1.4. The gas density can then be estimated using

the AGA8 guidelines presented in Starling and Savidge [1992] for the entire range of expected pressures. For example, at a constant system temperature of -2°C, the variation in gas density calculated using these guidelines has been plotted in Figure 6.7 depending on a range of pressures between 90 to 180 bar.

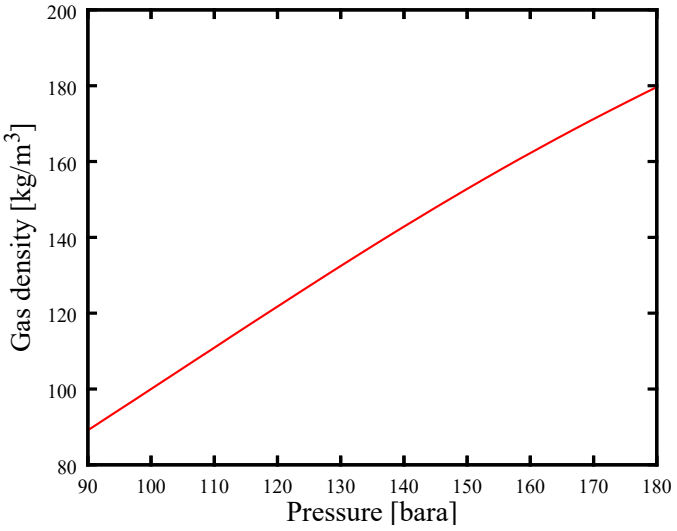


Figure 6.7: Variations in density for the Ormen Lange flowlines calculated using the AGA8 guidelines based on the composition shown in Table 6.2 at a system temperature of -2°.

Based upon the data presented in Figure 6.7, the following second order polynomial has been generated to fit the data as a means to determine the pressure-dependant gas density, ρ_g ($R^2 = 0.999$ for $P = 90$ - 180 bara);

$$\rho_g = -0.0009 \cdot P^2 + 1.3049 \cdot P - 20.982 \tag{6.2}$$

At each sampling point, the average gas density may be obtained from Equation 6.2 by using the average pressure calculated from Equation 6.1. Similarly, in order to estimate the gas kinematic viscosity at each sampling point, the formula proposed by Lee et al. [1966] has been used;

$$\mu_g = K_1 exp(X \rho_g^Y) \tag{6.3}$$

From Equation 6.3, μ_g is the gas viscosity which is converted to a kinematic viscosity prior to use in OpenFOAM®, and K_1, X and Y are constants depending

on the molecular weight and system temperature and are described in further detail in Lee et al. [1966]. Following this procedure, a summary of the derived system parameters is shown in Table 6.3. As the gas properties are now variable depending on the inlet pressure, this table includes the ranges of the gas density calculated using Equation 6.2 at the maximum and minimum average pressures, and the associated ranges for the kinematic viscosities. Furthermore, the pipe roughness shown in Table 6.3 has been derived from an assumption that the flowline comprises of standard steel sections commonly used in piping applications and the interfacial surface tension between the gas and water phases has been taken as the average value reported from the three-phase flow assurance study of the Ormen Lange flowlines by Bratland [2015].

Table 6.3: Derived system parameters.

Parameter	Value
Maximum gas density (@P_{avg} = 135 bar)	138.77 $[kg/m^3]$
Minimum gas density (@P_{avg} = 110 bar)	111.67 $[kg/m^3]$
Maximum gas kinematic viscosity (@P_{avg} = 135 bar)	1.19e-07 $[m^2/s]$
Minimum gas kinematic viscosity (@P_{avg} = 110 bar)	1.32e-07 $[m^2/s]$
Water density	1000 $[kg/m^3]$
Water kinematic viscosity	1e-06 $[m^2/s]$
Interfacial surface tension	6.25e-02 $[N/m]$
Pipe roughness (absolute)	4.57e-04 $[m]$
System temperature	-2 [°C]

6.2 Numerical Method

The numerical method used for the applied example is essentially identical to the method presented in Section 5.2 of the previous chapter, where the interFoam solver included in OpenFOAM® 2.3.x is used to describe the two phase flow of gas and water using the SST k - ω turbulence model at the maximum mesh resolution parameter considered in Chapters 4 and 5 ($\Upsilon = 100$). In order to achieve pressure driven flow in OpenFOAM®, the inlet and outlet pressures for the flowline are assigned fixed values, and the velocity parameter has been assigned a zero gradient at both the inlet and outlet patches. The complete set of boundary conditions used in this analysis is shown in Table 6.4.

Table 6.4: The list of updated OpenFOAM® boundary conditions used for the applied example.

Parameter	Inlet	Outlet	Wall
alpha.water	InletOutlet uniform 0	zeroGradient	zeroGradient
U	zeroGradient	zeroGradient	fixedValue uniform 0
p_rgh	fixedValue	fixedValue	zeroGradient
k	fixedValue	zeroGradient	kqRWallFunction
nut	Calculated uniform 0	Calculated uniform 0	nutkWallFunction
omega	fixedValue	zeroGradient	omegaWallFunction

In addition, SALOME 7.4.0 has been used to generate the meshes used in all simulations and DAKOTA 6.0 has been used to facilitate the optimisation study, where the coupling between OpenFOAM®, SALOME and DAKOTA has been presented in Appendix D.2. Furthermore, it has been assumed that the flowline profile presented in Figure 6.2 includes no lateral deviations, such that the symmetric boundary condition may be used to construct a 3-D half pipe model of the flowline. Due to the existence of multiple low spots that traverse multiple mesh nodes the strategy used to construct the mesh using SALOME has slightly increased in complexity, whereby a full description of the updated meshing strategy for flowlines involving multiple low-spots has been presented in Appendix E.4 and an example of the SALOME code used to generated the Ormen Lange flowline is presented in Appendix B.3.

6.3 Methodology

6.3.1 CFD Modelling Strategy

In principle, the technique used to create the CFD model featured in the previous chapter can simply be applied on a larger scale to encompass the entire length of any particular flowline. However, a single, complete mesh of a flowline involving multiple low-spot bends, each containing accumulated water regions now poses an increasingly complex problem due to the technique used to obtain the interfacial area facilitated through the use of the modified interFoam solver. In particular, as the interfacial area is extracted from the volumetric field data available in OpenFOAM®, the interfacial area is recorded at each timestep for all cells in the computational domain, suggesting that a new method must be introduced to ensure that the interfacial area is correctly distributed to represent the actual hydrate growth that occurs locally in each low-spot section. It is possible that the correct distribution of the interfacial area throughout

the domain may be achieved by defining specific cell zones of the mesh in the same manner that the *setFields* utility in OpenFOAM® is used to define the water regions. In this case, the interfacial area is restricted to regions specified beforehand by the user in accordance with the location of the low-spots and extended to include the likely location of the wetted length parameter.

An alternative approach has been explored in this study, whereby a complete mesh of the flowline has been made in SALOME and the corresponding Python script file has been manipulated to partition the mesh into three segments that are then modelled separately using individual script files in order to represent a more accurate depiction of the localised hydrate growth occurring in regions with low-spots. To emulate the response of the entire flowline, each segment has been modelled consecutively, whereby the outlet pressure and liquid holdup from any preceding segment is used to define the inlet conditions for the following segment. Although this approach also requires user input to define the mesh partitions, the advantage of this technique is that relatively horizontal regions of the mesh that are unlikely to contain any substantial amount of accumulated water both before and during a restart operation may be omitted entirely from the computational analysis, leading to a substantial reduction in the computational time required. Here, it is assumed that any of the water that has been evacuated from any preceding low-spot will pass through an omitted region entirely, provided that the upward inclination of the omitted region is lower than the preceding section. As the hydrate growth is assumed to be confined to a region determined by the wetted length parameter, this approach should provide a representation of the amount of hydrate growth analogous to the method mentioned previously, with the added advantage of fostering faster computational times for the framework. In this study, the three mesh segments of the CFD model used to represent the Ormen Lange flowline are shown in Figure 6.8 in conjunction with two additional regions of the mesh that have been omitted.

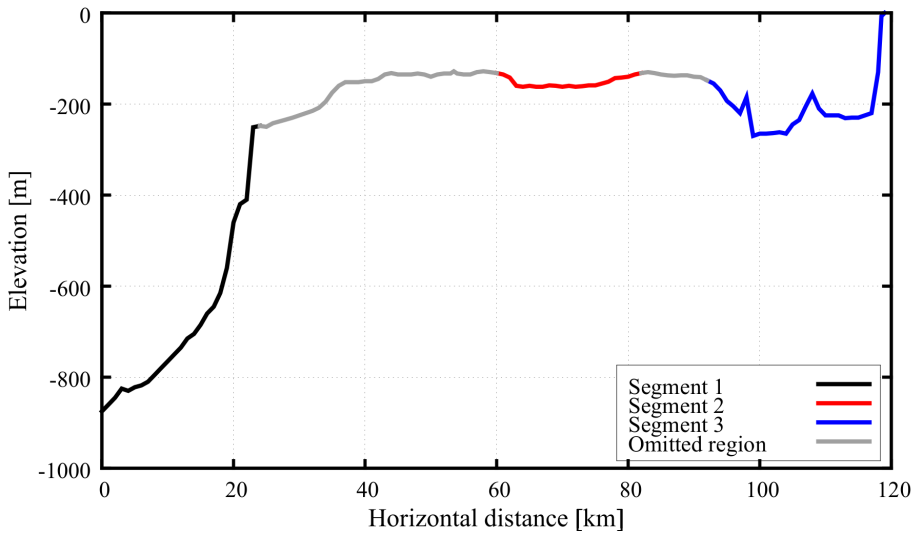


Figure 6.8: The topology of the Ormen Lange flowlines used in the optimisation study showing the three separate CFD model segments and the two regions omitted from the analysis.

From Figure 6.8, each of the three mesh segments used in the current study contains a major low-spot that is the most likely location for accumulated water to reside during an extended shutdown period. Wherever applicable, these mesh segments have been extended to include regions where the wetted length parameter is expected to be located. In addition, the third mesh segment contains a total of three low-spot regions which have been combined due to close proximity. Using this strategy, the exact horizontal distance and mesh cell count for each of the three mesh segments and two omitted regions of the Ormen Lange flowline are shown in Table 6.5.

Table 6.5: The horizontal distance and cell count of the three separate CFD model segments and the two omitted regions used to represent the Ormen Lange flowline.

Segment	Horizontal Distance [km]	Cell Count [M cells]
Segment 1	0 - 24	3.528
Omitted region 1	24 - 60	N/A
Segment 2	60 - 82	3.234
Omitted region 2	82 - 92	N/A
Segment 3	92 - 118	3.822

From Table 6.5, the total cell count for all three mesh segments combined is now

approximately 10.584 M cells, equating to a reduction of the total cell count in excess of 38% when compared to an equivalent mesh of the entire flowline. As the flowline has now been split into three segments that are now modelled separately, the inlet and outlet pressures corresponding to each segment must be defined to provide the pressure driving force for each segment. Likewise, the gas properties for each segment must also be updated using the average pressure for each segment. In this study, a linear relationship between the inlet and outlet pressures has been assumed. Thus, the pressure at any horizontal distance throughout the flowline from the inlet may be calculated as;

$$P(x) = P_{inlet} + \frac{90 - P_{inlet}}{118} \cdot x \tag{6.4}$$

From Equation 6.4, $P(x)$ is the assumed pressure at a particular horizontal location, x. The subsequent inlet and outlet pressures for each respective mesh segment can now be calculated using the horizontal distances specific to each segment presented in Table 6.5. Likewise, the average pressure for each segment is calculated using Equation 6.1 and the updated values of the gas density and viscosity are calculated using Equations 6.2 and 6.3, respectively. As an example, the respective inlet, outlet and average pressures and the updated gas properties for each of the three segments used in this study has been presented in Table 6.6 for an inlet pressure of 180 bar and an outlet pressure of 90 bar.

Table 6.6: An example of how the gas density and viscosity are estimated for each flowline segment (P_{inlet} = 180 bar).

Segment	P_{inlet} [bar]	P_{outlet} [bar]	P_{avg} [bar]	Gas density [kg/m^3]	Gas kinematic viscosity [m^2/s]
1	180.0	161.7	170.9	175.7	1.11e-07
2	134.2	117.5	125.9	129.0	1.23e-07
3	109.8	90.0	99.9	100.4	1.33e-07

6.3.2 Constraints

As mentioned in Section 6.1.4, a pressure inlet range of between 130 - 180 bar has been selected. Likewise, from Section 6.1.6, it has been assumed that the volumetric amount of accumulated water is within a range of between 5 - 25% of the maximum

flowline volume has been assumed, where the maximum flowline volume was calculated to be approximately 45,500 m^3. In this study, this water volume has further been restricted to four major low-spot regions of the flowline, as shown in Figure 6.9 in relation to the two omitted regions of the mesh.

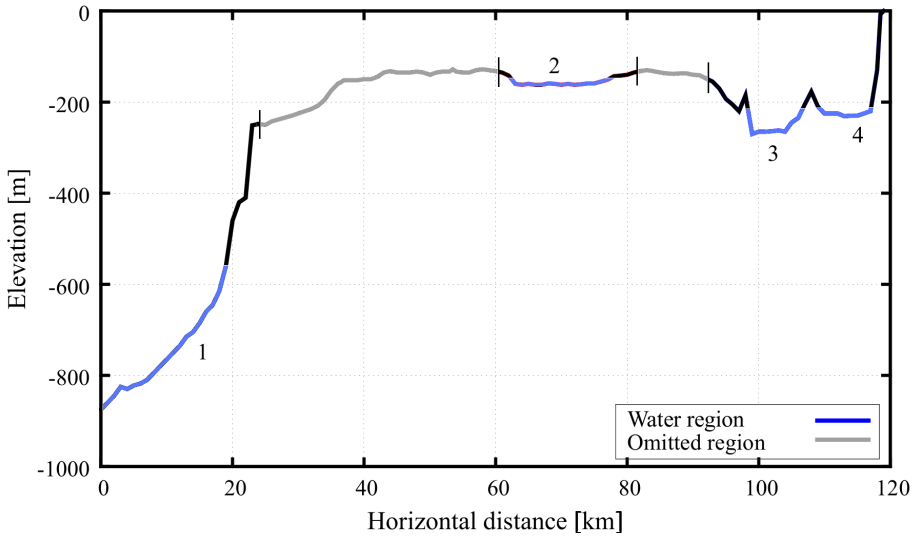


Figure 6.9: The Ormen Lange flowline topology showing the four regions of water accumulation, represented in blue in relation to the omitted regions, represented in grey with vertical lines to indicate the boundaries of the omitted regions

A problem arises here as it is not possible to directly know the distribution of water contained in each of the low-spots using anything other than simulated data, which could not been found in the literature. Due to the absence of available data, and in order to restrict the number of variables used in this study to a total of two, the amount of water contained in each low-spot is assumed to follow a certain distribution that has been kept fixed throughout the optimisation study. This assumed distribution, or weightage is shown in Table 6.7.

Table 6.7: The location and assumed weightage for each water region.

Water region	Location	Weightage [%]
1	Segment 1	50
2	Segment 2	20
3	Segment 3	15
4	Segment 3	15

In order to define regions of the hydrate blockage risk, three sets of probes have been selected to coincide to each of the four initial water regions shown in Figure 6.9 and are extended to encompass the immediate downstream segment of the water region. The distribution of these probes follows the procedure introduced in the previous chapter where a pair of probes are placed at the upper and lower extremities of the pipe. The longitudinal spacing for each probe pair has been kept constant at a distance of 1 km throughout the mesh, where the distribution of each probe set is shown in Table 6.8

Table 6.8: Distribution of the probe sets for each mesh segment.

Probe set	Distribution range [km]	Number of probes
1	0 - 21	42
2	62 - 82	40
3	95 - 118	46

6.3.3 Coupling of the Partitioned Mesh

As mentioned in Section 6.3.1, in order to couple the partitioned mesh to represent the entire flowline, each mesh segment is run consecutively, where the outlet data recorded at the first segment is used as the inlet data for the following segment. During the simulation of the first segment, the liquid phase fraction, which has been specified in OpenFOAM® in this study as the "alpha.water" field has been extracted using probes located at the symmetry plane of the outlet patch. This probe data is accessed using the *probeLocations* utility initiated at each write interval as determined in the *controlDict* file. A total of 10 probes have been used for each outlet patch where the general location of these probes in relation to the symmetry plane and outlet patch is shown in Figure 6.10.

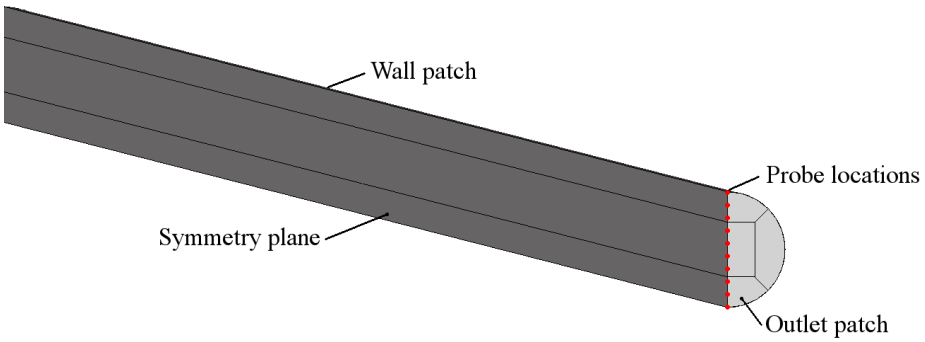


Figure 6.10: The distribution of probes on the outlet patch to record the liquid phase fraction at each designated write interval.

These probes shown in Figure 6.10 are spaced equidistant along the vertical (Y) axis and can be used to derive the liquid height at the outlet patch using the following relation;

$$\frac{h_L}{D} = \frac{\sum \alpha_{probe}}{10} \tag{6.5}$$

The subsequent input for the second segment can be achieved by modifying the alpha.water boundary condition at the inlet patch from an inletOutlet type to a UniformFixedValue type that includes time-dependant variations in the magnitude of alpha.water depending on a specified tabular file, based upon the extracted probe data. As the Inlet patch is the entire face of the 3-D half pipe section, the value of alpha.water in this case is equivalent to the liquid holdup. Thus the liquid height ratio from Equation 6.5 must be converted to the liquid holdup as the inlet for the second segment. This may be achieved in the same manner first introduced in the mechanistic model by Taitel and Dukler [1976], where the liquid holdup, E_L is given by;

$$E_L = \frac{A_L}{A} \tag{6.6}$$

From Equation 6.6, A is the total pipe area and A_L is the liquid area in the cross section, given by;

$$A_L = 0.25 \cdot D^2 \left[\pi - cos^{-1}\left(2 \cdot \frac{h_L}{D} - 1\right) + \left(2 \cdot \frac{h_L}{D} - 1\right)\sqrt{1 - \left(2 \cdot \frac{h_L}{D}\right)^2} \right] \tag{6.7}$$

Similarly, this technique is used to extract the liquid height from the second segment and input the liquid holdup for the third segment. Thus, following the example presented in Table 6.6 showing the updated pressures and gas properties used for each mesh segment according to an inlet pressure of 180 bar, Figure 6.11 shows the subsequent transient response of the liquid holdup recorded using the probes placed at the outlet patches of each of the three segments in conjunction to the subsequent liquid holdup used at the inlet patches to provide a representation of the entire flowline. In order to generate these results, an initial water volume of 25% of the pipe volume has been used with the weightage in each water region as per the values shown in Table 6.7.

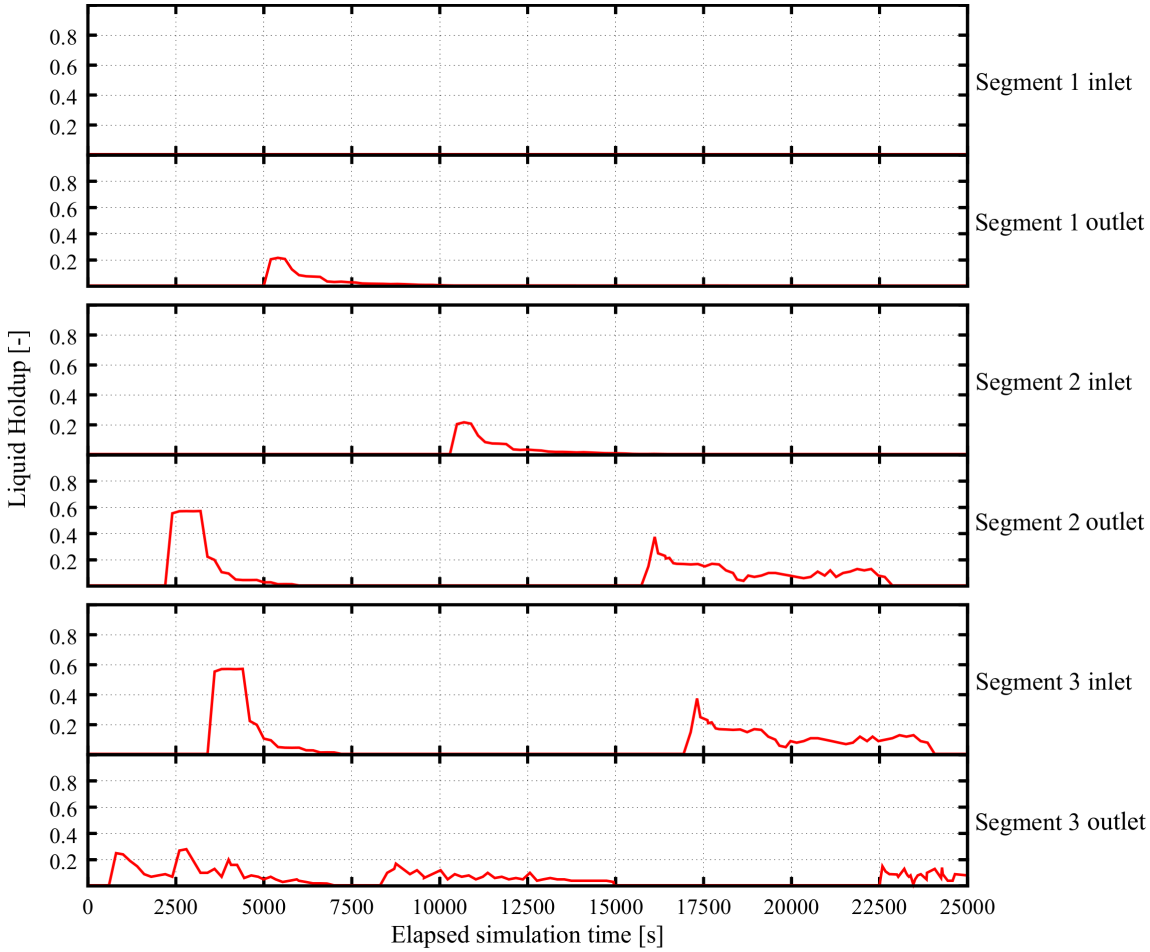


Figure 6.11: An example of the liquid holdup recorded at the inlet and outlet planes for each of the three model segments considered for the Ormen Lange flowline (P_{inlet} = 180 bar, V_i = 25% Pipe volume).

From Figure 6.11, a time delay has been introduced from the elapsed time the liquid holdup exits the first computational domain to the elapsed time the liquid holdup appears at the inlet of the second computational domain. This time delay is based upon the estimated time required for water exiting the computational domain of the first segment to traverse the first omitted region of the mesh. As shown in Figure 6.11, a time delay is also apparent between the outlet of the second mesh segment to the inlet of the third segment, although this delay is noticeably smaller on account of the reduced size of the second omitted region. An estimation of the time required for water to traverse through the two omitted regions has been accomplished by using the same probes located in Figure 6.10 to record the velocity of the water phase at the outlet patch at each of the preceding segments, whereby the approximate time for the water region to pass through the omitted region may be estimated based upon the horizontal distance shown in Table 6.5. In this manner, the liquid velocity is obtained from the probes at each write interval over the entire simulation time; however, the magnitude of the velocity used to estimate the transition time over an omitted region is taken as the average value recorded over the first three write intervals for probes in which non-zero readings of the alpha.water phase fraction are recorded. This average excludes the lowermost probe located on both the symmetry plane and the wall patch as the velocity is assumed to be zero as per the no-slip boundary condition used.

Following from the liquid holdup inlet data presented in Figure 6.11, the subsequent transient interfacial area recorded for each of the three mesh segments is shown in Figure 6.12. In this figure, the blue colour represents the contribution of the interfacial area from the initial water region present in the flowline low-spot as it traverses through each of the three mesh segments of the flowline. Likewise, the yellow and red colours represent the contribution of the interfacial area from the water contained in the low-spots of the second and third mesh segments, respectively.

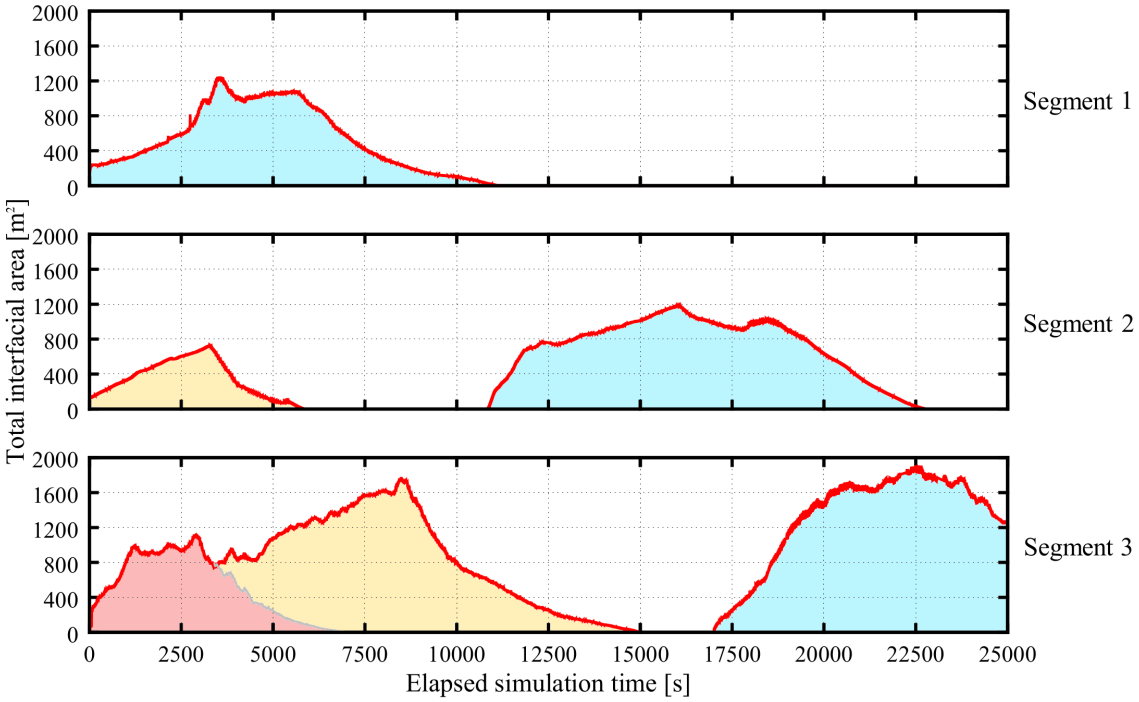


Figure 6.12: The transient interfacial area recorded for each of the three model segments considered for the Ormen Lange flowline (P_{inlet} = 180 bar, V_i = 25% Pipe volume).

As shown in Figure 6.12, the transient interfacial area has been obtained according to each mesh segment, which can be used to calculate the amount of hydrate growth occurring locally in each low-spot via the methodology presented in Section 5.3.1 of the previous chapter

6.3.4 Simulation Cut-off Time

It is important to note that the method used to obtain an estimate of the amount of hydrate growth shown in Section 5.3.1 of the previous chapter requires that the CFD simulations be run until steady state conditions are achieved. This may however, result in the allocation of computational resources to describing flow conditions that are no longer relevant. In particular, in this methodology it is still assumed that any of the inlet gas flow will be inhibited, such that when the gas front from the inlet patch interacts with a slower moving water region, it will likely eliminate the possibility of further hydrate growth by virtue of the inhibitor function. Thus, as the gas phase will reach the flowline exit before the water phase, the flowline will most likely be fully inhibited well before the steady state conditions are reached. In

light of this, a simulation cut-off time, t_{co} has been used in place of the steady state time. The cut-off time is designated as the final elapsed time that hydrate growth is expected to occur, and is given by;

$$t_{co} = \frac{L}{U_G} \tag{6.8}$$

From Equation 6.8, L is the entire flowline length at 118,000 km and for the purpose of simplicity, U_G is assumed to be equivalent to the gas velocity at the flowline entrance, which has been calculated using an empirical relation found in ISO 13703:2000, i.e;

$$U_G = \frac{131 \cdot Z \cdot q_g \cdot T}{D^2 \cdot P} \tag{6.9}$$

From Equation 6.9, Z is the gas compressibility factor, q_g is the gas flow rate in $[m^3/hr]$ at normal conditions, T is the system temperature [K], D is the inner pipe diameter [mm] and P is the operating pressure $[kPa]$. In order to cater for the variable inlet velocity, Equation 6.9 can be used with the correlation between the pressure and gas flow rate presented in Figure 6.3 to present the inlet pressure and estimated inlet velocity in relation the range of expected gas flow rates, as shown in Figure 6.13.

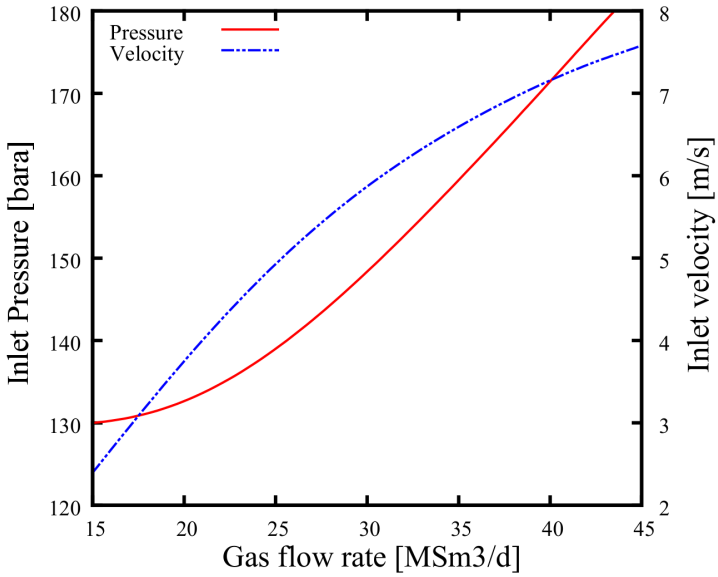


Figure 6.13: The range of inlet pressures and calculated inlet velocities for the Ormen Lange flowlines depending on the gas flow rate for a constant slugcatcher pressure of 90 bar.

Similar to the method used in Equation 6.2 to represent the gas density, the relationship between the inlet pressure and flowrate presented in Figure 6.13 can be represented as a second order polynomial ($R^2 = 0.9714$) of the form;

$$Q = -0.0048 \cdot P^2 + 1.974 \cdot P - 157.96 \tag{6.10}$$

Where Q is the gas flow rate in Msm3/d and requires conversion to normal conditions for use in Equation 6.9. Thus, an estimation for the inlet gas velocity and subsequent simulation cut-off time may be derived from the pressure at each sampling point generated from the DACE study. Though the use of the cut-off time parameter, the hydrate growth will most likely be confined to the transient phase, and the need to record the steady state interfacial area or estimate the half-block time as per the previous chapter is now redundant. Following the example of the interfacial area recorded for an inlet pressure of 180 bar in Figure 6.12, the corresponding inlet gas velocity is calculated using Equation 6.8 to be in excess of 7.4 m/s. Assuming that this value is approximately constant throughout the flowline, the initial gas present at the inlet plane at the commencement of the simulation will reach the flowline exit at a time slightly above 15,900 s. This represents a significant reduction in the simulation time required when considering the interfacial area results presented in Figure 6.12 show that steady state conditions have not yet been achieved for the third segment at an elapsed simulation time of 25,000 seconds.

6.3.5 Optimisation Methodology

Based upon the conclusions presented in the previous chapter, the OA-LHS DACE technique is used to define 25 surrogate build points in a design space bounded by the range of inlet pressures and initial amounts of accumulated water presented in Section 6.3.2. Due to partitioning of the mesh into three segments, the build points derived from the DACE study are used to identify the inlet, outlet and average pressures for three sets of 25 build points corresponding to each of the mesh segments. Thus, a total of 75 simulations are required in order to represent the entire flowline.

Each CFD simulation was run until the designated cut-off time calculated from Equation 6.8, whereby the simulations were stopped and the relevant wetted length and probe data was extracted and used to provide an estimation of the hydrate growth at each simulation point, the Kriging technique has then used to build surrogate models in order to represent the transient amount of hydrate growth, wetted length

and upper and lower probe sets throughout the design space for each mesh segment, where for each of the three flowline segments, the optimisation problem has been defined in a similar manner to Section 5.3.6 of the previous chapter, i.e.,

$$\begin{aligned} &\text{Maximise:} && L_w, \sum \alpha_{lower} \\ &\text{Minimise:} && \Delta n_{trans}, \sum \alpha_{upper} \\ &\text{Subject to:} && V_{min} \leq V_i \leq V_{max} \\ & && P_{min} \leq P_{inlet} \leq P_{max} \end{aligned}$$

In this definition, the inlet pressure constraint is used as opposed to the gas velocity constraint from the previous chapter and the objective function for the steady state hydrate growth rate ($\Delta n_{ss}/\Delta t$) has been removed. The removal of this objective function should have no impact on the convergence of MOGA as both Δn_{trans} and $\Delta n_{ss}/\Delta t$ are designed to converge to cases with higher gas flow rates.

6.3.6 Implementation of the Hydrate Growth Models

The same manner as presented in Section 5.3.1 has been used to estimate the amount of hydrate growth using the kinetic and mass transfer limited hydrate models by Turner et al. [2005] and Skovborg and Rasmussen [1994], respectively. In particular, the values of the constants u, K_1, K_2 and W_g that are used in Equations 5.2 and 5.3 have remained unchanged from the previous chapter. As mentioned in Section 6.1.5, the system temperature, T used in Equation 5.2 has been assumed to be -2°C and in the absence of available data, the hydrate formation temperature has been assumed to be 15°C, such that the degree of subcooling used in Equation 5.2, ΔT_{sub}, is now 17°C. Lastly, the gas velocity used to calculate the mass transfer coefficient in Equation 5.4 has been derived from the inlet gas velocity given by Equation 6.8.

6.4 Results from the Optimisation Study

6.4.1 DACE Study

The OA-LHS DACE technique was used to generate 25 surrogate build points over the design space bounded a range of inlet pressures between 130 to 180 bar and initial water volumes between 5 to 25 % of the flowline volume. The locations for these build points are shown in Figure 6.14.

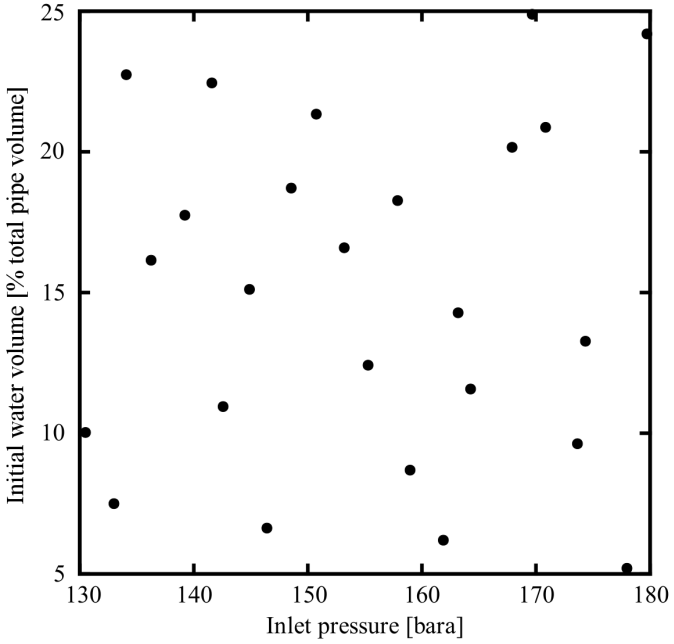


Figure 6.14: The location of the surrogate build points obtained from the DACE study using the OA-LHS technique.

Using the location of the build points in Figure 6.14, the corresponding inlet and outlet pressures for each segment have been calculated using the manner shown in Section 6.3.1 and have been presented in Table 6.9.

Table 6.9: The corresponding inlet and outlet pressures derived for each respective mesh segment using the 25 build points generated from the DACE study.

	DACE output		Segment 1		Segment 2		Segment 3	
DACE point	P_i [bar]	V_i [% pipe volume]	P_i [bar]	P_o [bar]	P_i [bar]	P_o [bar]	P_i [bar]	P_o [bar]
1	174.34	13.27	174.34	165.15	136.37	118.78	110.79	90.00
2	170.84	20.87	170.84	162.36	134.65	117.71	110.02	90.00
3	158.96	8.69	158.96	152.90	128.81	114.09	107.40	90.00
4	164.27	11.57	164.27	157.13	131.42	115.71	108.57	90.00
5	139.22	17.75	139.22	137.18	119.11	108.07	103.05	90.00
6	148.54	18.71	148.54	144.60	123.69	110.91	105.10	90.00
7	155.28	12.42	155.28	149.97	127.00	112.97	106.59	90.00
8	153.19	16.59	153.19	148.30	125.97	112.33	106.13	90.00
9	179.72	24.19	179.72	169.44	139.01	120.42	111.97	90.00
10	169.67	24.89	169.67	161.43	134.08	117.36	109.76	90.00
11	161.88	6.20	161.88	155.23	130.25	114.98	108.04	90.00
12	130.51	10.03	130.51	130.24	114.83	105.41	101.13	90.00
13	157.87	18.27	157.87	152.03	128.28	113.76	107.16	90.00
14	177.97	5.20	177.97	168.04	138.15	119.89	111.59	90.00
15	141.58	22.45	141.58	139.06	120.27	108.79	103.57	90.00
16	146.41	6.63	146.41	142.90	122.64	110.26	104.63	90.00
17	134.08	22.74	134.08	133.08	116.58	106.50	101.92	90.00
18	167.91	20.16	167.91	160.03	133.21	116.82	109.37	90.00
19	163.17	14.28	163.17	156.25	130.88	115.37	108.33	90.00
20	133.00	7.50	133.00	132.22	116.05	106.17	101.68	90.00
21	142.57	10.95	142.57	139.84	120.75	109.09	103.79	90.00
22	144.88	15.11	144.88	141.68	121.89	109.79	104.30	90.00
23	150.74	21.34	150.74	146.35	124.77	111.58	105.59	90.00
24	173.63	9.63	173.63	164.59	136.02	118.57	110.63	90.00
25	136.26	16.15	136.26	134.82	117.65	107.16	102.40	90.00

6.4.2 Data Extracted from the CFD Simulations

CFD simulations were run at the build points generated for each of the three CFD model segments until the designated cut-off time, using the respective inlet and outlet pressures for each segment as shown in Table 6.9. The visualisation of the two-phase flow was largely analogous to that observed in the previous chapter, whereby

upon commencement of the simulation, a single slug formed and was transported downstream until dissipation of the wave height occurred at the wetted length and the flow became stratified. At the cut-off time, the most likely locations for residual water was at the Storegga slide location at an approximate horizontal distance of 20 km from the flowline inlet and the riser base at a horizontal distance of 116 km. The increased liquid holdup at the Storegga slide area would appear to be consistent with the most likely hydrate blockage location during flowling conditions as identified by Holden et al. [2005].

Interestingly, on the upward leg of any of the low-spot locations, there was no water accumulation observed near the upper pipe wall at the simulation cut-off time for any of mesh segments for any of the inlet pressures considered, suggesting that an inlet pressure of 130 bar is high enough such that the high risk zone defined in Figure 5.6 of the previous chapter does not exist in the current design space. In addition, on account of the relatively horizontal section, the second mesh segment did not contain any residual water at the designated cut-off time, indicating that a flowline inlet pressure of 130 bar is capable of removing all accumulated water from this section, thus constituting a low risk for hydrate blockage to occur.

In a similar fashion to the previous chapter, the area under the transient interfacial area plot ($2\int_0^{t_{ss}} A_i \Delta t$), wetted length ($L_w$), and the upper and lower probe sets were recorded ($\sum \alpha_{upper}$ and $\sum \alpha_{upper}$, respectively). At the cut-off time, if no further increase and decrease in the interfacial area was expected the flow was designated steady state. The applicable output at each sampling point for the first, second and third CFD model segment is presented in Tables 6.10, 6.11 and 6.12, respectively.

Table 6.10: The extracted data obtained from the first CFD model segment.

	CFD Input			Extracted data				
DACE point	P_i [bar]	V_i [% pipe volume]	t_{co} [hrs]	S.S. achieved?	$2\int_0^{t_{ss}} A_i \Delta t$ [$\times 10^6$ m^{2}s]	L_w [km]	$\sum \alpha_{upper}$ [-]	$\sum \alpha_{lower}$ [-]
1	174.34	13.27	5.8	Y	9.8	30.0	0	0
2	170.84	20.87	6.0	Y	12.0	29.5	0	0
3	158.96	8.69	9.3	Y	32.5	18.1	0	0
4	164.27	11.57	8.9	Y	81.7	12.5	0	0
5	139.22	17.75	14.3	Y	112.2	10.3	0	13
6	148.54	18.71	9.9	Y	35.6	16.5	0	0
7	155.28	12.42	9.4	Y	86.3	12.1	0	0
8	153.19	16.59	9.4	Y	86.7	12.1	0	0
9	179.72	24.19	5.5	Y	11.5	29.9	0	0
10	169.67	24.89	7.3	Y	12.7	26.2	0	0
11	161.88	6.20	8.6	Y	14.4	23.3	0	0
12	130.51	10.03	16.4	Y	85.1	8.0	0	15
13	157.87	18.27	9.4	Y	33.7	18.1	0	0
14	177.97	5.20	5.6	Y	8.9	25.0	0	0
15	141.58	22.45	10.8	Y	84.0	16.4	0	20
16	146.41	6.63	10.2	Y	57.0	15.9	0	0
17	134.08	22.74	14.9	Y	126.3	10.5	0	30
18	167.91	20.16	7.5	Y	12.7	25.2	0	0
19	163.17	14.28	8.4	Y	28.8	22.7	0	0
20	133.00	7.50	15.6	Y	83.7	8.3	0	15
21	142.57	10.95	10.6	Y	9.1	12.1	0	5
22	144.88	15.11	10.6	Y	9.3	12.1	0	5
23	150.74	21.34	9.9	Y	6.7	16.5	0	0
24	173.63	9.63	5.8	Y	9.3	24.0	0	0
25	136.26	16.15	14.9	Y	114.5	10.0	0	14

Table 6.11: The extracted data obtained from the second CFD model segment.

	CFD Input			Extracted data				
DACE point	P_i [bar]	V_i [% pipe volume]	t_{co} [hrs]	S.S. achieved?	$2\int_0^{t_{ss}} A_i \Delta t$ [$\times 10^6$ m^{2}s]	L_w [km]	$\sum \alpha_{upper}$ [-]	$\sum \alpha_{lower}$ [-]
1	174.34	13.27	5.8	N	5.5	18.0	0	0
2	170.84	20.87	6.0	N	5.7	17.5	0	0
3	158.96	8.69	9.3	N	3.8	14.4	0	0
4	164.27	11.57	8.9	N	10.8	11.4	0	0
5	139.22	17.75	14.3	N	13.9	8.0	0	0
6	148.54	18.71	9.9	N	4.2	13.5	0	0
7	155.28	12.42	9.4	N	10.9	10.5	0	0
8	153.19	16.59	9.4	N	10.8	10.5	0	0
9	179.72	24.19	5.5	N	5.6	18.0	0	0
10	169.67	24.89	7.3	N	3.6	16.8	0	0
11	161.88	6.20	8.6	N	2.4	15.4	0	0
12	130.51	10.03	16.4	N	15.5	8.0	0	0
13	157.87	18.27	9.4	N	4.1	14.6	0	0
14	177.97	5.20	5.6	N	1.9	16.1	0	0
15	141.58	22.45	10.8	N	15.7	12.1	0	0
16	146.41	6.63	10.2	N	4.7	11.0	0	0
17	134.08	22.74	14.9	N	19.4	8.2	0	0
18	167.91	20.16	7.5	N	3.7	16.3	0	0
19	163.17	14.28	8.4	N	4.3	15.4	0	0
20	133.00	7.50	15.6	N	15.1	7.9	0	0
21	142.57	10.95	10.6	N	11.3	10.1	0	0
22	144.88	15.11	10.6	N	12.0	10.1	0	0
23	150.74	21.34	9.9	N	13.6	12.5	0	0
24	173.63	9.63	5.8	N	1.9	16.2	0	0
25	136.26	16.15	14.9	N	14.3	8.1	0	0

Table 6.12: The extracted data obtained from the third CFD model segment.

	CFD Input			Extracted data				
DACE point	P_i [bar]	V_i [% pipe volume]	t_{co} [hrs]	S.S. achieved?	$2\int_0^{t_{ss}} A_i \Delta t$ [$\times 10^6$ m^{2}s]	L_w [km]	$\sum \alpha_{upper}$ [-]	$\sum \alpha_{lower}$ [-]
1	174.34	13.27	5.8	N	14.3	18.1	0	0
2	170.84	20.87	6.0	N	15.0	17.5	0	0
3	158.96	8.69	9.3	N	23.6	14.5	0	0
4	164.27	11.57	8.9	N	26.9	12.0	0	0
5	139.22	17.75	14.3	N	43.2	9.9	0	16
6	148.54	18.71	9.9	N	30.9	14.1	0	0
7	155.28	12.42	9.4	N	30.7	11.5	0	0
8	153.19	16.59	9.4	N	26.1	11.5	0	0
9	179.72	24.19	5.5	N	14.7	18.3	0	0
10	169.67	24.89	7.3	N	16.0	16.2	0	0
11	161.88	6.20	8.6	N	18.9	16.9	0	0
12	130.51	10.03	16.4	N	61.5	10.4	0	18
13	157.87	18.27	9.4	N	27.3	14.5	0	0
14	177.97	5.20	5.6	N	14.4	17.5	0	0
15	141.58	22.45	10.8	N	27.3	12.2	0	15
16	146.41	6.63	10.2	N	26.2	11.8	0	10
17	134.08	22.74	14.9	N	62.8	10.0	0	20
18	167.91	20.16	7.5	N	17.2	16.4	0	0
19	163.17	14.28	8.4	N	19.8	16.0	0	0
20	133.00	7.50	15.6	N	58.3	9.9	0	19
21	142.57	10.95	10.6	N	32.7	11.0	0	6
22	144.88	15.11	10.6	N	36.0	11.0	0	7
23	150.74	21.34	9.9	N	23.9	12.5	0	0
24	173.63	9.63	5.8	N	15.1	17.4	0	0
25	136.26	16.15	14.9	N	45.6	10.4	0	19

6.4.3 Surrogate-based Optimisation

As the upper probe set has been observed to be zero for all cases, it has subsequently been eliminated from the optimisation definition shown in Section 6.3.5. Instead, a total of three surrogate models have been generated using the Kriging technique. Likewise, as the half-block time parameter is no longer used on account of the cut-off

time, only one response surface has been generated, corresponding to the thickness of the hydrate layer at the simulation cut-off time. The designation of the three surrogate models and the single response surface is shown in Table 6.13.

Table 6.13: Updated designation of the surrogate (S) and response (R) models used in the applied optimisation study of the Ormen Lange flowline.

Designation	Model	Description	Source
S1	Δn_{trans}	Transient hydrate growth [mol]	Eq. 5.6
S2	L_w	Wetted length [m]	CFD
S3	$\sum \alpha_{lower}$	Lower probe set	CFD
R1	H_t	Thickness of the hydrate layer at the simulation cut-off time [m]	Eq. 5.12

The surrogate modelling results obtained using the Kriging technique for the S1, S2 and S3 surrogate models are shown for the first, second and third CFD model segments in Figures 6.15, 6.16 and 6.17, respectively.

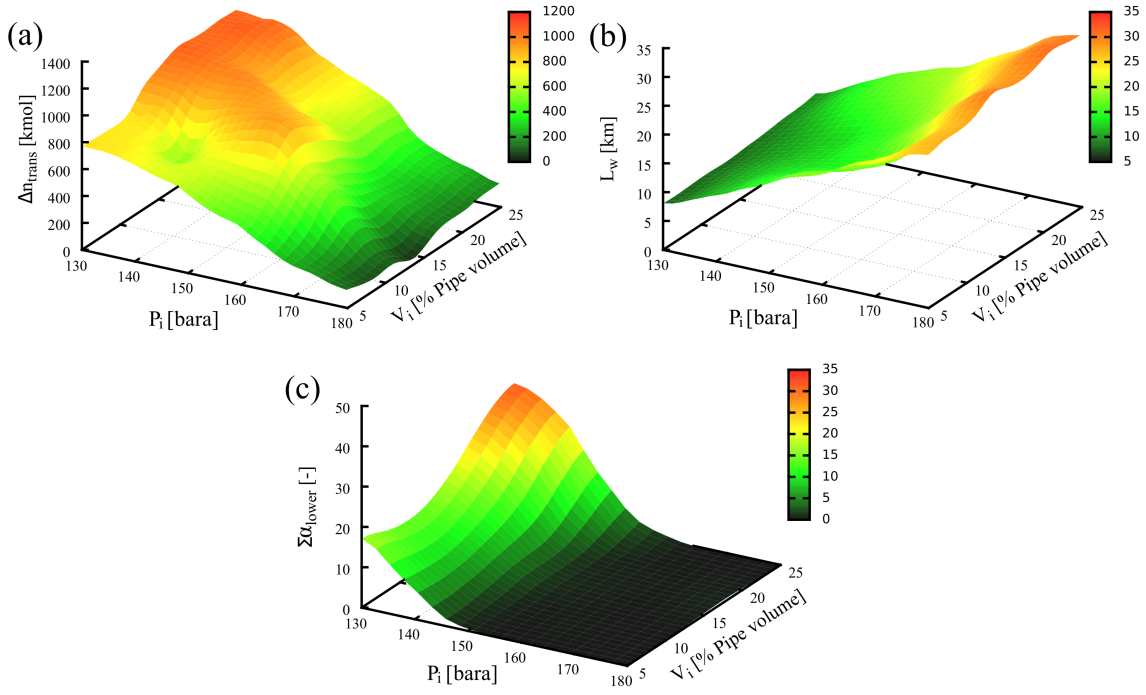


Figure 6.15: The Kriging surrogate model response over the design space for the first CFD model segment of the Ormen Lange flowline where; a) molar gas consumption during the transient phase (S1 - Δn_{trans}), b) wetted length (S2- L_w) and c) summation of the lower liquid phase fraction (S3 - $\sum \alpha_{lower}$).

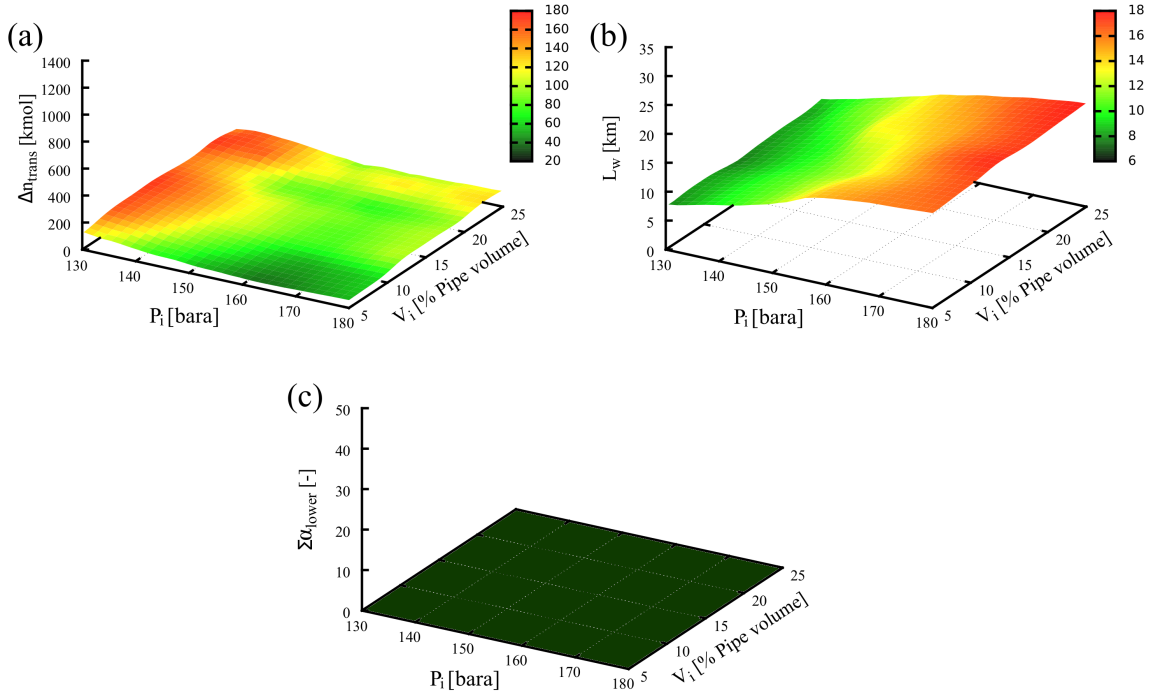


Figure 6.16: The Kriging surrogate model response over the design space for the second CFD model segment of the Ormen Lange flowline where; a) molar gas consumption during the transient phase (S1 - Δn_{trans}), b) wetted length (S2 - L_w) and c) summation of the lower liquid phase fraction (S3 - $\sum \alpha_{lower}$).

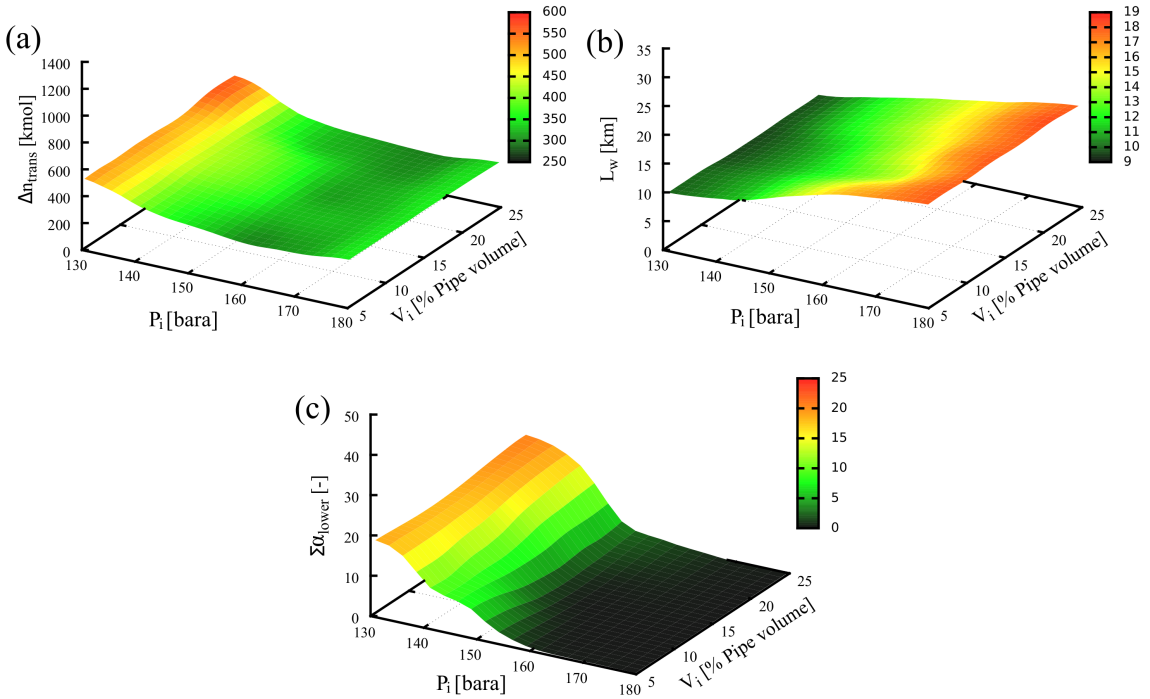


Figure 6.17: The Kriging surrogate model response over the design space for the third and final CFD model segment of the Ormen Lange flowline where; a) molar gas consumption during the transient phase (S1 - Δn_{trans}), b) wetted length (S2 - L_w) and c) summation of the lower liquid phase fraction (S3 - $\sum \alpha_{lower}$).

The R1 response surface generated for each CFD model segment is shown in Figure 6.18 parts a), c) and e) for model segments 1, 2 and 3, respectively. Additionally, Figure 6.16 parts b), d) and f) show the regions of defined blockage risk for model segments 1, 2 and 3 respectively together with the point data generated from MOGA. Like the previous chapter, the S3 surrogate model has been used to define the blockage risk, where the green colour shown in Figure 6.18 represents the low risk zone and the yellow color represents the medium risk zone. On account of the zero values recorded for all of the upper probe locations, a region of high risk does not exist for the range of inlet pressures and water volumes considered.

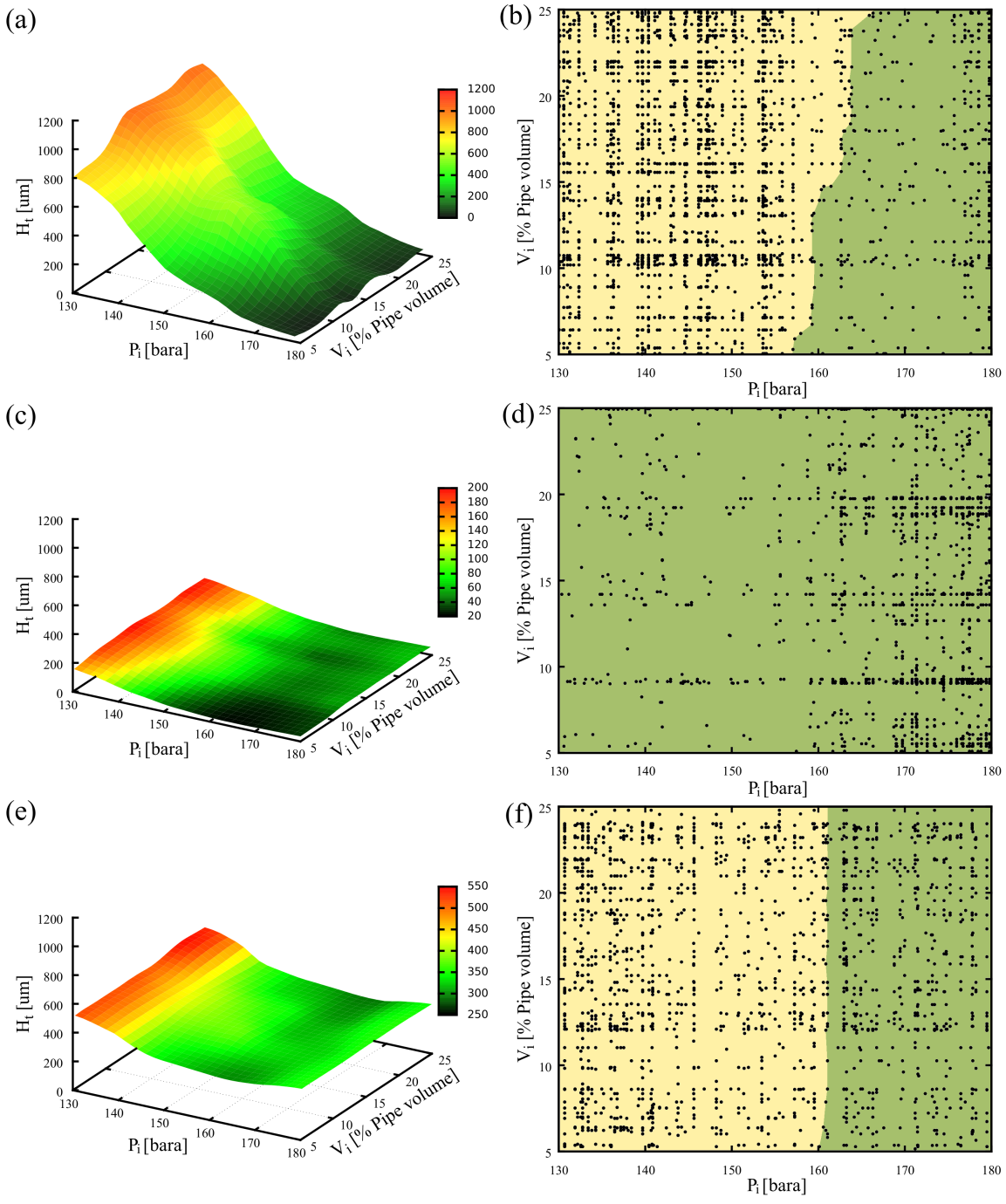


Figure 6.18: The R1 response surface constructed from the surrogate models showing the thickness of the hydrate layer at the simulation cut-off time in parts a), c) and d) for model segments 1, 2 and 3, respectively. Parts b), d) and e) depict the defined regions of hydrate blockage risk in model segments 1, 2 and 3, respectively, together with the evaluation points generated from MOGA

The hydrate blockage risk associated with each model segment shown in Figure 6.18 has been combined to represent the hydrate blockage risk throughout the entire flowline. This representation is shown in Figure 6.19, where the low risk zone is represented by the green coloured area, and the medium risk zone represented by the yellow coloured area.

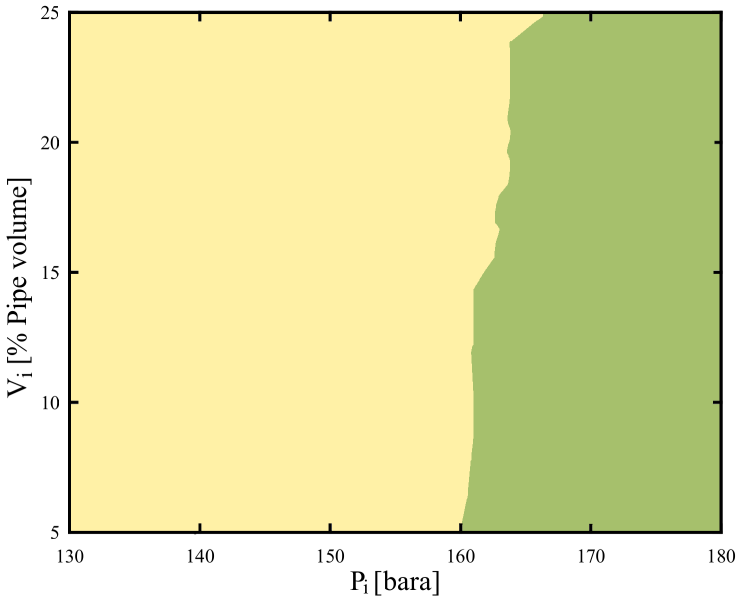


Figure 6.19: The complete risk profile for the Ormen Lange flowlines obtained from combining the risk profile generated for each of the three CFD model segments presented in Figure 6.18 parts b), d) and f), respectively.

6.4.4 Comparison of the Hydrate Models Used

The relative contributions to the S1 surrogate model from both the kinetic and mass transfer limited hydrate growth models are presented in Figure 6.20. In this figure, parts a), c) and e) depict the amount of growth during the transient phase estimated using the kinetic model for mesh segments 1, 2 and 3, respectively. Likewise, Figure 6.20 parts b), d) and f) depict the amount of hydrate growth using the mass transfer limited model for mesh segments 1, 2 and 3, respectively.

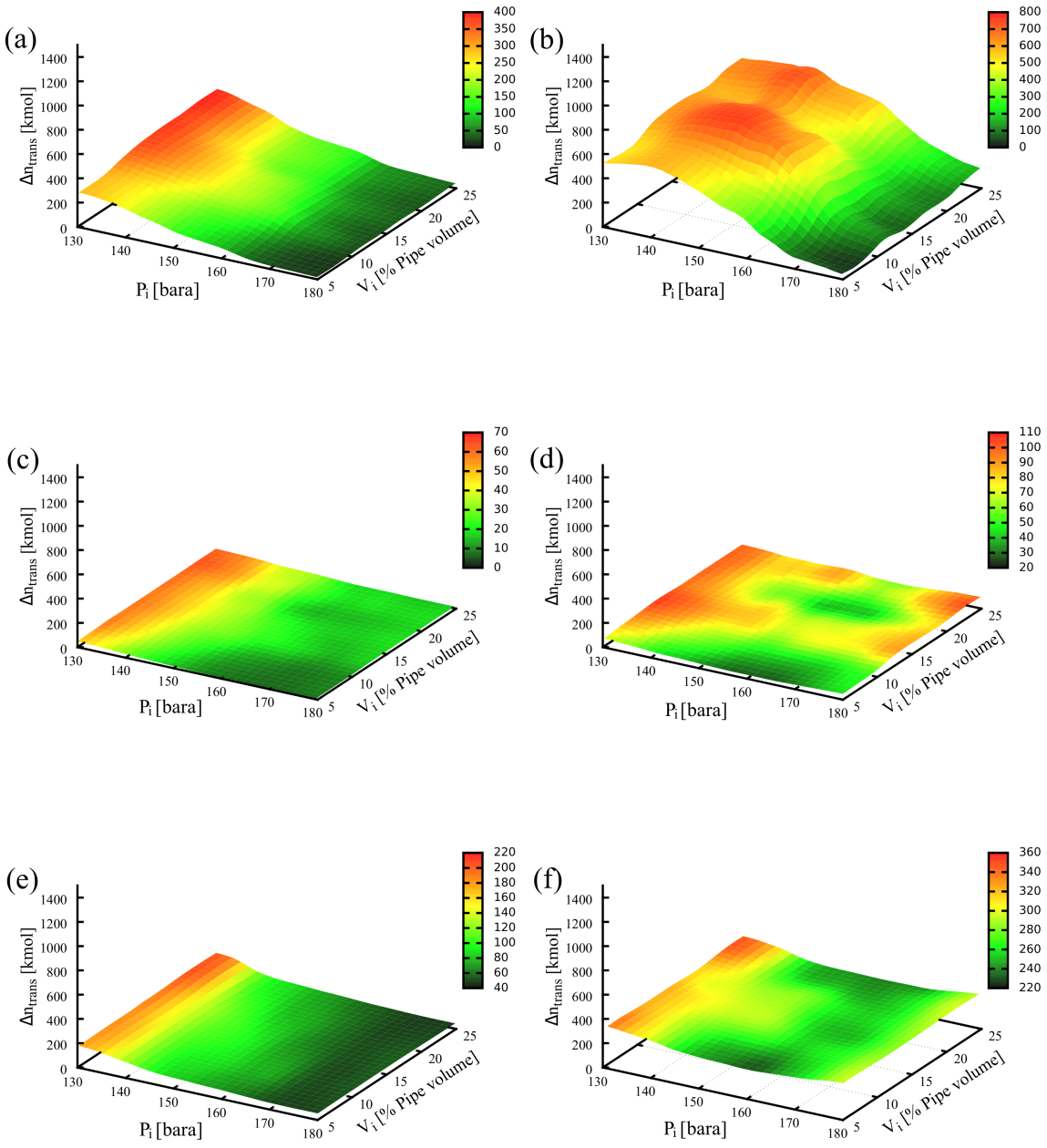


Figure 6.20: A comparison of the amount of hydrate growth predicted by the kinetic and mass transfer limited models; Parts a), c) and e) represent the contribution of the total transient hydrate growth obtained from the kinetic model by Turner et al. [2005] for CFD model segments 1, 2 and 3, respectively and Parts b), d) and f) represent the contribution of the total transient hydrate growth obtained from the mass transfer limited model by Skovborg and Rasmussen [1994] for CFD model segments 1, 2 and 3, respectively.

As readily observable in Figure 6.20, the relative contribution from the mass transfer limited model to the total amount of transient hydrate growth was much greater than the contribution from the kinetic model in all cases. This is in contrast to the results obtained from the optimisation study of the single low-spot bend presented in the previous chapter; however, the results presented in Figure 6.20 appear to support the suggestion by Sloan and Koh [2007] that intrinsic kinetics plays a much smaller role in actual systems when compared to the effects of mass and heat transfer, indicating that the methodology proposed to calculate the hydrate growth for the flowline could be applicable for use in the framework. The increased amount of hydrate growth estimated using the mass transfer limited model occurs due to the higher gas velocity calculated using Equation 6.2 that is used to calculate the mass transfer coefficient as per Equation 5.4. For comparison, the range of gas velocities used in this study was approximately between 2.3 m/s to 7.5 m/s as shown in Figure 6.13, compared to a maximum gas velocity of 1.5 m/s used for the optimisation study of the low-spot pipe bend from Chapter 5.

6.4.5 Computational Time-scale

In a similar manner to the analysis presented in the previous chapter, the time scale required per simulation at each sampling point generated from the DACE study has been averaged over an inlet pressure range from 130 to 180 bar and is presented in Figure 6.21. The computational time results presented in this figure are based upon the use of a single quad-core i7-4790 processor running in parallel until the designated cut-off time and the magnitude of the computational time depicted is the combined simulation time from each of the three segments, representing the estimated total time required to simulate the entire flowline at depending on the inlet pressure.

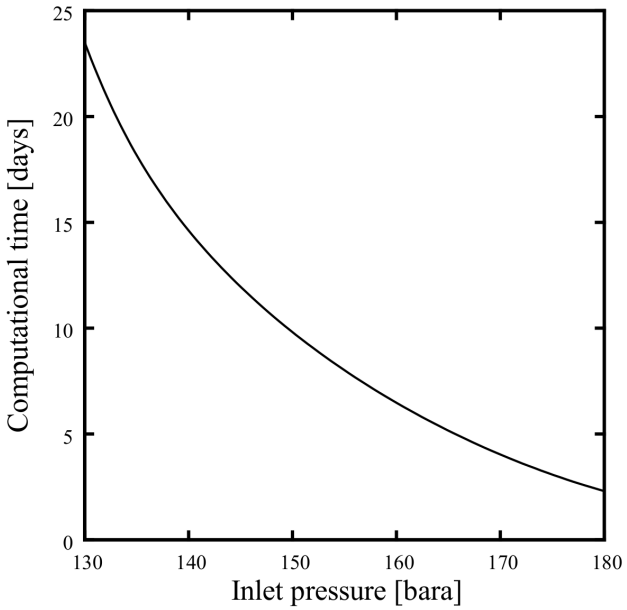


Figure 6.21: The average computational time required until the cut-off time is reached, depending on the inlet pressure using a single quad-core i7-4970 processor running in parallel.

For the optimisation study of the entire flowline using 25 build points, the average time required was approximately 10 days of simulation time per simulation, equating to a total processing time of 250 days when using a single workstation.

6.5 Discussion

In this chapter, an attempt has been made to apply the optimisation technique derived in the previous chapter for the purpose of hydrate blockage mitigation using an actual flowline. It has been intended that through this process, the robustness and usefulness of the finalised framework as an applicable investigate tool will have been strengthened sufficiently such that the framework may be applied in a comprehensive manner to any flowline geometry or fluid properties. The particular example of the Ormen Lange field was selected due to the challenges involved in providing flow assurance for the two flowlines that traverse a long length of hilly terrain in an environment that is highly conducive for the nucleation and growth of hydrates. In addition, the low CGR for this flowline has enabled the gas and condensate phases to be modelled as a single phase when using the two phase interFoam solver. However, in the absence of data available for verification purposes, it is largely unknown how representative

the use of this solver has been to emulate the actual flowline conditions, and further investigations are required to increase the fidelity of the simulation. In particular, although an effort has been made to update the gas properties based on the pressure, the thermal interactions between the gas front emanating from the reservoir, and the effects of phase change behaviour have not been considered.

Using OpenFOAM®, a driving force based upon the expected pressure differential has successfully been applied. This was facilitated based upon the data available in the literature, and it is foreseeable that issues may be encountered in future if the exact pressure profile is not known. Likewise, the initial amount of water was derived based upon the simulated liquid holdup found in the literature, where the upper range has been extended to 25 % of the total flowline volume in order to account for unexpected conditions where the liquid holdup may be significantly higher prior to shutdown. Due to the manner in which the interfacial area is obtained throughout the entire computational domain at each time step, a significant amount of work has been conducted in order to achieve a more representative estimation of the amount of hydrate growth occurring in each low-spot. This has resulted in the partitioning of the mesh into three separate segments whereby a methodology has been proposed to couple the independent mesh segments in order to provide an adequate representation of the entire flowline. Although the methodology used has the advantage of reducing the total number of cells used when compared to an alternative approach involving designated cell zones as mentioned in Section 6.3.1, it has unfortunately made the process much more complex in order to correctly represent the response of an entire flowline using partitioning. In light of this, it is evident that this strategy is not particularly user friendly and has not been recommended for use in the framework. Instead, it is recommended that the option exploring the use of designated cell zones is investigated further in future studies.

Based upon an assumption that during a restart operation, the inlet gas flow will be chemically inhibited, the simulation cut-off time was used to dictate the elapsed time when the inlet gas flow will reach the flowline exit. This time has been estimated based upon an estimation of the inlet gas velocity using Equation 6.9, whereby the cut-off time has replaced the steady state time from the previous chapter and has made the calculation of the hydrate growth rate during steady state conditions given by Equation 5.7, and the subsequent blockage half-time, given by Equation 5.13, redundant. Instead, all simulations have been run until the cut-off time and the

amount of hydrate growth estimated using the combination of the kinetic and mass transfer limited models presented in the previous chapter.

Using DAKOTA, the generation of the 25 surrogate build points using the OA-LHS DACE technique and the subsequent application of the Kriging surrogate modelling technique and MOGA has been successful in projecting the estimations of the transient hydrate growth, wetted length and lower probe sets over the design space bounded by the range of inlet pressures and initial water volumes. Due to the success of these particular optimisation techniques, they have been recommended for implementation in the finalised framework and are expected to provide robust solutions of the system response for any manner of flowline geometry or inlet variable. The three surrogate models of the lower probe set generated for each of the three CFD model segments have been combined in order to represent the hydrate risk profile of the entire flowline, as shown in Figure 6.19. It is expected that the combined hydrate risk profile is particularly useful for a flowline operator as a means to represent a safe operational window based upon a range of inlet pressures that will allow for a reduced risk of hydrate blockages occurring during restart operations without the usage of inhibition techniques. Despite this, it is recognised that the success of this technique depends heavily on the ease of implementation, which must be developed further for the simulation-based optimisation approach to be accepted in industry as a useful investigative tool.

In a similar fashion to the previous chapter, the data extracted from the S1 and S2 surrogate models have been used to construct the response surface showing the estimated thickness of the hydrate layer at the simulation cut-off time for each flowline segment, which is displayed in Figure 6.18 parts a), c) and e), respectively. This results presented in this figure have shown a relatively benign amount of hydrate growth with a maximum of only 1.2 mm recorded for the first flowline segment at the lowest inlet pressure considered. Similar to the previous chapter, the amount of hydrate growth and thickness of the hydrate layer have been estimated based upon the wetted length parameter. The minimal hydrate thickness displayed for each flowline segment in Figure 6.18 suggests that either the hydrate blockage risk is minimal for all of the inlet pressures considered or the approach used to define the deposition length is currently inadequate and must be refined further. As the Ormen Lange flowline is expected to be an extreme example of hydrate growth in an uninhibited flowline, it is most likely the latter, whereby further work is required in order to provide a more accurate depiction of the actual hydrate deposition length.

In contrast to the optimisation study of the single low-spot bend featured in Chapter 5, the mass transfer limited model by Skovborg and Rasmussen [1994] was observed to have a larger influence on the total hydrate growth when compared to the kinetic model by Turner et al. [2005]. This was largely due to the estimation of the gas velocity that was used for the calculation of the mass transfer coefficient using Equation 5.4, whereby the gas velocity corresponding to the maximum pressure used in this study was greater than the gas velocity used in the previous chapter by a factor of 5. Lastly, the computational time involved to complete the optimisation study of the Ormen Lange flowlines was in the order of 250 days for a single workstation. This figure is undoubtedly excessive when compared to current simulation tools that utilise the 1-D drift flux approach that are expected to simulate the same flowline in a manner of hours; however, it is expected that the computational expense using a CFD-based strategy may be mitigated substantially through the use of additional parallel computing options.

6.6 Summary and Conclusions

In this chapter, the optimisation technique used to reduce the risk of hydrate blockages occurring in a single low-spot pipe segment has been further refined through the use of an applied optimisation study featuring an example of one of the two 30" Ormen Lange flowlines. As a means to determine the amount of localised hydrate growth occurring in each major low-spot, the flowline geometry has been partitioned into three separate segments and recombined in order to represent the entire flowline. A new parameter designated the simulation cut-off time has been introduced based upon the estimated time for the inhibited gas flow introduced at the inlet to traverse the entire flowline length. As it was not necessary to run all simulations until steady state conditions were achieved, the optimisation definition has been simplified to exclude the steady state hydrate growth rate, rendering the blockage half-time defined in the previous chapter redundant. Following the work completed in this chapter, the following conclusions have been made;

- The optimisation strategy presented in the previous chapter has been successfully adapted to generate an estimation of the amount of hydrate growth and hydrate risk profile for the Ormen Lange flowline during a restart operation.
- The numerical strategy using OpenFOAM®, SALOME and DAKOTA has been successfully adapted to include pressure driven flows for longer flowline sections,

although further refinement is necessary to reduce the computational cost involved in order to increase the usefulness of this technique as an investigative tool when compared to the traditional 1-D drift flux models used.

- Although the results from the optimisation strategy have predicted benign conditions for the amount of hydrate growth occurring during a simulated restart operation, further improvements are needed to be implemented for the correct modelling of hydrate deposition and particle jamming.

- The strategy used to partition the mesh into separate segments has been successful in; a) allowing for estimates of the localised hydrate growth occurring in major low-spots, and b) reducing the total number of cells in the mesh by over 38% through the usage of omitted regions; however, this implementation comes at a high price due to the increased level of complexity due to the number of steps required to partition the mesh and subsequently combine the results of each segment to represent the entire flowline.

- In light of the added complexity required for the partitioned strategy, it has not been recommended for use in the framework. Instead, it is recommended that future studies explore an additional modification that can be made to the interFoam solver in order to limit the interfacial area recorded to specific cell zones specific by the user and not the entire volumetric field data as currently used.

- Despite the additional complexity, the optimisation concepts first presented in Chapter 5 have been successfully refined in this chapter to yield the amount of hydrate growth during a gas-dominated restart and the hydrate blockage risk for an actual flowline, suggesting these concepts can still be implemented in the optimisation framework.

Chapter 7

The Optimisation Framework

In this chapter, the optimisation approach first introduced in Chapter 5 and refined in Chapter 6 is presented in the finalised form of an optimisation framework capable of accommodating different types of flowline geometries and fluid properties. A text-based user interface (TUI) is shown as a means to couple SALOME, OpenFOAM® and DAKOTA as a black box interface that is useful for a practitioner not familiar with these software packages and the main operations facilitated by the framework are described. The work presented in this chapter constitutes the culmination of work required to achieve the fifth and final objective from this study.

7.1 Scope of the Framework

The original intent of the optimisation framework has been to produce a comprehensive solution for the reduced risk of hydrate blockage for use with any flowline design and any fluid properties. This has evidently not be attained in the current framework format. In particular, although it is possible to use any fluid properties in the current version of the framework, the lack of verification methods for three-phase flow have limited the current version of the framework to two-phase gas-dominated flow with low CGR values to minimise erroneous modelling of the condensate phase. Despite this, it is expected that the optimisation techniques used in the framework will still apply in the event that the interFoam solver is replaced with a three-phase solver to allow modelling of oil-dominated flows.

In addition, the current framework is based upon the 3-D half-pipe modelling strategy defined in Chapter 4 as a means to reduce the amount of cells required. This has limited the current version of the framework to flowlines where the lateral deviation is minimal such that the symmetric boundary condition is not void. This;

however, is not a limiting factor of the framework, whereby a practitioner with suitable computational resources at their disposal could easily utilise the same framework for flowlines that include lateral deviations using a full 3-D mesh.

With these current limitations taken into consideration, the entire internalised workings of the framework are not presented in this chapter. Instead, a description is provided of how the framework is set up in order to facilitate communication between the specific software used and obtain results for the purpose of reducing the hydrate blockage risk in flowlines during restart operations. It is hoped that the information provided in this chapter will aid a future user in setting up a similar framework using an industry standard simulation tool for a more representative depiction of the hydrate growth and blockage risk.

7.2 Description of the Finalised Framework

The finalised version of the framework has been constructed to function as a black box interface. As described in Section 2.1, the framework includes an applicable optimisation toolkit, a simulation tool and a user interface. Based upon a review of the available optimisation toolkits presented in Section 2.3, DAKOTA 6.0 has been selected as the optimisation toolkit used for the framework and provides access to the OA-LHS DACE technique, the Kriging surrogate modelling technique and MOGA. In addition, DAKOTA is used to facilitate communication with OpenFOAM® 2.3.x, to describe the two phase flow via the interFoam solver and SALOME 7.4.0 for the purposes of constructing the CFD model and mesh generation. The optimisation results presented in Chapters 5 and 6 provide ample evidence to suggest that DAKOTA will be successful in providing a robust interface in the event another script-based simulation tool is used to describe the flow in lieu of OpenFOAM®.

Using the optimisation framework, it has been intended that all elements of the optimisation study may be executed from an interface that references a single master folder. Thus, the purpose of the TUI is to manipulate and modify the necessary script files required to perform geometry construction, mesh construction, optimisation properties, and to facilitate running of the simulations and data extraction based upon a defined workflow.

It is noted here that based upon the conclusions derived from Chapter 6, this current version of the framework does not include an option for the user to partition the mesh in order to limit the total interfacial area to the low-spot regions of the mesh.

Although this approach was successful in providing a more accurate representation of the interfacial area in each low-spot at a reduced computational cost, the process involved was deemed too complex and not user-friendly for use in the optimisation framework.

Instead, in order to increase the ease of use, the user has the option to construct a single mesh of the flowline, whereby the interFoam solver will record the interfacial area over the entire domain and appropriate distribution of the interfacial area is up to the user discretion. This is a limitation of the current framework that must be given appropriate attention in future versions of the framework in order for the framework to serve as a useful investigative tool. It is expected that this limitation may be overcome through the use of an additional modification that can be made to the interFoam solver as described in Section 6.3.1.

7.2.1 Framework Workflow

The workflow of the framework involves the following major steps that are accessible through the TUI;

1. The user inputs the desired geometric properties of the flowline and the desired mesh resolution. The framework then facilitates construction of a 3-D half-pipe mesh using SALOME.

2. Additional properties relevant to OpenFOAM® are selected by the user.

3. The user defines the number, location and upper and lower bounds of specific accumulated water regions and the number and location of the upper and lower probe sets to record the liquid phase fraction. The SALOME script file used to create the mesh is updated accordingly by the framework.

4. The user selects the desired sampling size and performs a DACE analysis using the OA-LHS technique from DAKOTA according to the corresponding range of inlet pressures and initial water volumes used to define the design space. The framework will then generate the build points required for the SBO study.

5. Based upon the range of inlet pressures generated at each build point, the user updates the gas properties and simulation cut-off time.

6. CFD simulations are run at the build points generated by the DACE study using the modified interFoam solver and the transient interfacial area, wetted length and relevant probe data is extracted from the simulations by the user.

7. An SBO study is then performed using the Kriging method and MOGA to produce estimations for the amount of hydrate growth during the restart operation, the wetted length, and the upper and lower probe sets over the design space.

8. Based upon the response from the surrogate models, the amount of transient hydrate growth is calculated by the user at each evaluation point generated from MOGA using the approach suggested in Section 5.3.1, generating a response surface showing the thickness of the hydrate layer at the simulation cut-off time.

9. Finally, the evaluation point data from MOGA is used with the surrogate models for the upper and lower probe sets in order to generate the regions hydrate blockage risk over the entire design space as shown in Figure 5.6.

7.2.2 User Interface

The title page of the TUI is shown in Figure 7.1 featuring five headings to perform the required tasks necessary to design, modify and initiate an optimisation study of a flowline. These headings are designed to emulate the workflow presented in Section 7.2.1 and the respective actions that may be taken by the user according to these headings is displayed in Figure 7.2.

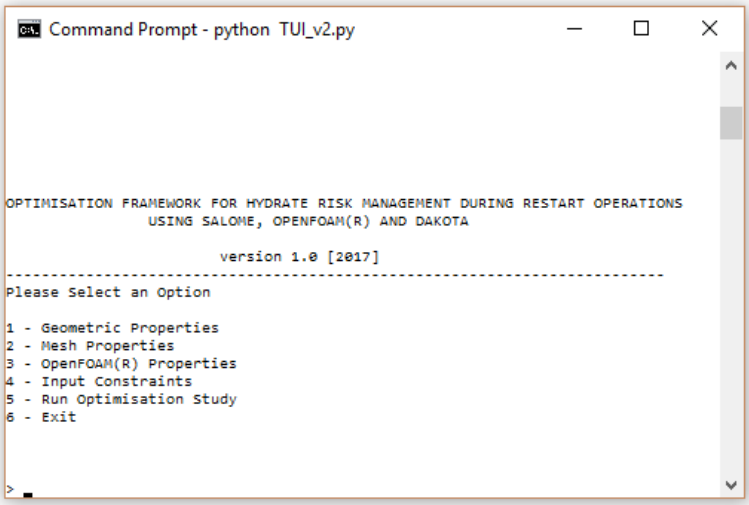


Figure 7.1: The title page of the user interface.

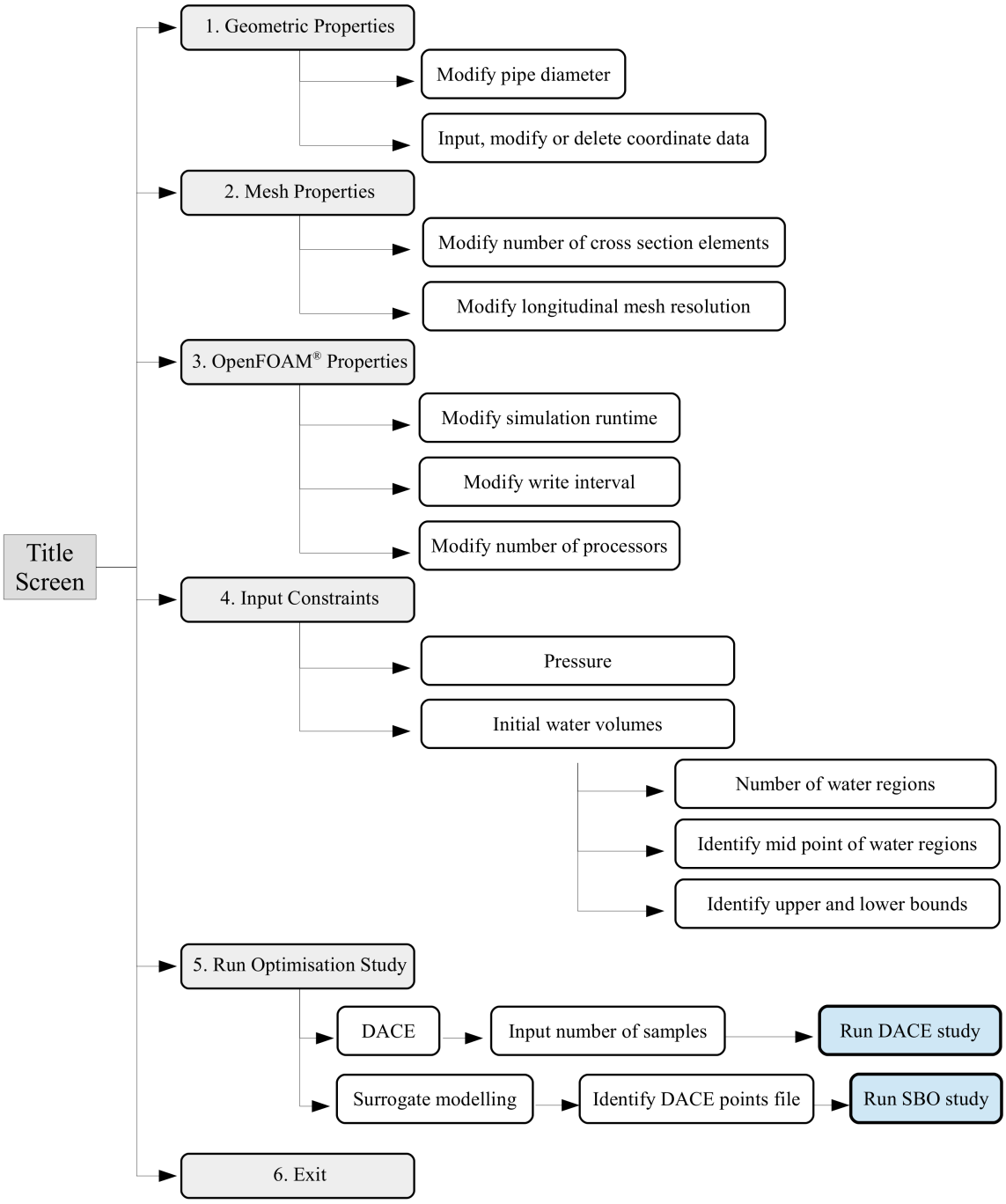


Figure 7.2: The components of the the TUI accessible from the title screen.

7.2.3 Master Folder Structure

The structure of the master folder required to implement the framework is shown in Figure 7.3 and a description of the files contained within all sub-folders of the master folder is presented in Table 7.1.

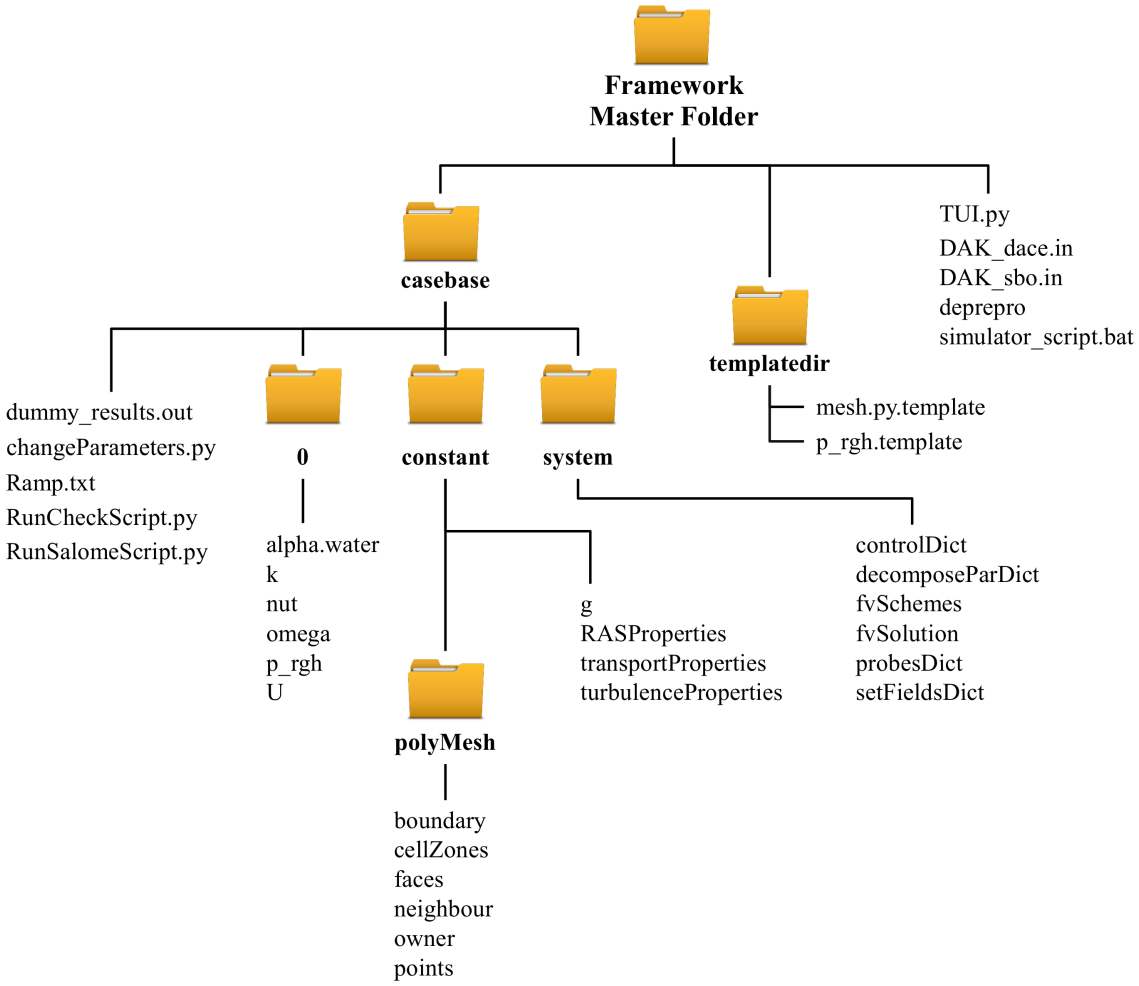


Figure 7.3: The structure of the folder tree used for the framework.

Table 7.1: A description of the files contained within the folder tree.

File	Description
alpha.water	A file to describe the liquid phase fraction field used in interFoam.
boundary	A file used to describe the mesh, automatically created when the ideasUnvToFoam utility is run.
cellZones	A file used to describe the location of the accumulated water regions, automatically created when the ideasUnvToFoam utility is run.
changeParameters.py	A Python script used to update the gas properties and simulation run time at each build point.
controlDict	A dictionary file containing essential parameters for the running of interFoam.
DAK_dace.in	The DAKOTA input file required to run the DACE study.
DAK_sbo.in	The DAKOTA input file required to run the surrogate-based optimisation study.
decomposeParDict	A dictionary file for decomposing a case to run in parallel.
deprepro	A Perl script used by simulator_script.bat to insert realisations of the inlet pressure and water volume into the files located in "templatedir".
dummy_results.out	A dummy results file used by DAKOTA in the event a simulation failure, allowing the continuation of the optimisation study.
faces	A file used to describe the mesh, automatically created when the ideasUnvToFoam utility is run.
fVSchemes	A dictionary file used to assign numerical schemes to the terms used in interFoam, such as the equation derivatives.
fVSolution	A dictionary file used to control the equation solvers, tolerances and algorithms.
g	A file to describe the gravitational acceleration used in interFoam.
k	A file to describe the turbulence kinetic energy used in interFoam
mesh.py.template	The template file created in SALOME showing the designation of the inlet water variable.
neighbour	A file used to describe the mesh, automatically created when the ideasUnvToFoam utility is run.
nut	A file to describe the turbulence viscosity used in interFoam
omega	A file to describe the turbulent dissipation rate in interFoam
owner	A file used to describe the mesh, automatically created when the ideasUnvToFoam utility is run.
points	A file used to describe the mesh, automatically created when the ideasUnvToFoam utility is run.
probesDict	A dictionary file containing the location of the probes
p_rgh	A file to describe the pressure used in interFoam, where p_rgh is the difference between the static and hydraulic pressure.
p_rgh.template	A template file of the p_rgh variable with designation of the inlet pressure as a variable for use with DAKOTA.
Ramp.txt	A text file used to introduce a delay in the inlet pressure, allowing the initial water region to settle.
RASProperties	A dictionary file containing information about the turbulence model.
RunCheckScript.py	A Python script used to check if tangible results are obtained from the solver, if not the dummy results are passed to DAKOTA.
RunSalomeScript.py	A Python script used by simulator_script.bat to open SALOME, construct the geometry and mesh and then close SALOME.
setFieldsDict	A dictionary file used to designate the liquid phase fraction in the water region defined in the *cellZones* file.
simulator_script.nat	A batch file script used as an analysis driver to execute the CFD simulations and extract the probe data for use with DAKOTA.
transportProperties	A dictionary file used to describe the density, viscosity and surface tension of the fluids.
TUI.py	The text user interface, written in Python.
turbulenceProperties	A file used to select laminar or turbulent flow.
U	A file to describe the velocity used in interFoam.

7.3 Framework Operations

This section details the operations conducted by the framework based upon the actions taken by the user using the TUI.

7.3.1 Step 1 - Geometric Properties

During this stage the user inputs the pipe diameter and the coordinate data of the entire flowline. A file is created called *mesh.py.template* in "templatedir". This file emulates a Python dump file generated by SALOME 7.4.0 and features the "GEOM" component. The framework allows for modification of the pipe diameter and coordinate data, whereby the *mesh.py.template* file is updated accordingly. In this file, the total pipe volume is estimated for latter use with the water volume depending on the user input.

7.3.2 Step 2 - Mesh Properties

In this stage the "SMESH" component of the *mesh.py.template* file is created to complete the base mesh in a format recognised by SALOME. The user has the opportunity to modify the number of cross section elements used for the mesh from a default setting of 90 which will generate the mesh cross section shown in Figure 4.3. This will update the number of segments in the SMESH module using the following relation;

$$Number\ of\ cross\ section\ elements = 2.5 \times Number\ of\ segments^2 \tag{7.1}$$

The user can also input the mesh resolution in cells/m. This will update the longitudinal spacing of the mesh on all longitudinal edges, termed the "flow edges" by setting the number of segments in the SMESH component using the following relation;

$$Number\ of\ longitudinal\ segments = \frac{Mesh\ resolution\ [cells/m] \times L\ [m]}{Number\ of\ cross\ section\ elements} \tag{7.2}$$

Where L is the horizontal length of the mesh, obtained from the x coordinate of the final vertex located in the GEOM component of *mesh.py.template*.

7.3.3 Step 3 - OpenFOAM® Properties

In this step, modifications can be made to the OpenFOAM® properties located in the "casebase" folder, which is copied into each work directory when the DACE study is run using DAKOTA. In particular, the user has the option to modify the write interval of the simulation, the number of processors used, and the base properties for the gas and water phases, including the density, viscosity and interfacial surface tension. These modifications will alter the *controlDict* file, *decomposeParDict* file, *simulator_script.bat* file and the *transportProperties* file.

Based upon the numerical method presented in Chapter 4, the usage of the "kOmegaSST" turbulence model, implementation of the gravitational acceleration and the specific numerical schemes located in the system directory have been selected by default. In addition, the simple decomposition type is selected by default.

7.3.4 Step 4 - Input Constraints

During this step, the user identifies the boundaries of the inlet pressure and the number and associated boundaries of the initial water regions in addition to specifying the ranges where the probes will be located. On the surface, the user input is used to modify the upper and lower bounds in the "variables" section of the DAKOTA input files used to run the DACE study and the SBO study, *DAK_dace.in* and *DAK_sbo.in*, respectively. The process of using DAKOTA to facilitate the updated inlet pressures is a straightforward process, whereby the value of the inlet pressure is defined as the 'x1' descriptor in both of the DAKOTA input files and the respective *p_rgh.template* file. The process to update the initial water regions through DAKOTA using a single 'x2' descriptor is much more complicated and involves the following steps;

- The user defines the number of water regions.
- The user defines the horizontal distance (x-coordinate) of the mid-point of each water region. This mid-point is used to create a plane in the *mesh.py.template* specific to each of the water regions.
- For each water region, the user then specifies the horizontal distance of the mid-plane to the upper bound of each of the water region to mark the ending point of each water region. An example of this is shown in Figure 7.4 where

the mid-plane is denoted by the vertical red lines and the upper bound of each water region is depicted by the small vertical black lines.

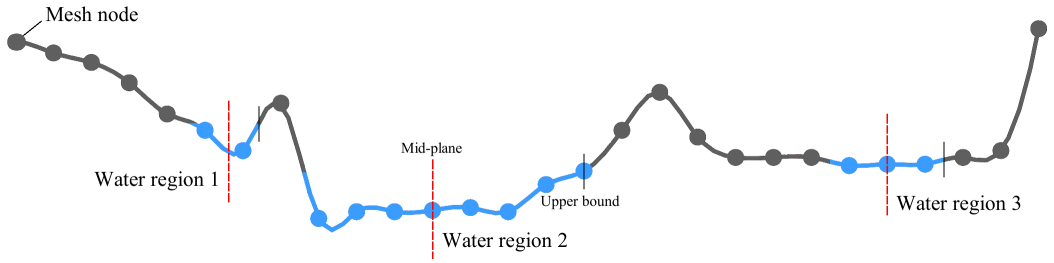


Figure 7.4: An example of the mid-point of three separate water regions featured in a flowline mesh.

- Using the upper bounds, symmetric planes are created in the *mesh.py.template* around the mid-plane of each water region to mark the lower bound of each water region. These additional planes are used to split the flowline geometry into separate sections as described in Appendix E.3 and E.4, whereby this operation will update both the GEOM and SMESH components of this file in the manner shown in Appendix B.3.

- Using the upper and lower bounds for each water region, the approximate water volume in the flowline is estimated together with the weightage of each water region with respect to the total pipe volume estimated by the framework in the first step. The value of the weightage is important as it is assumed to be constant throughout the optimisation study. Based upon the boundaries for each water regions, the initial water volume is calculated in relation to the entire pipe volume and expressed as a percentage. This percentage value constitutes the upper range (maximum volume) of the initial water volume constraint.

- The user now has the option to enter the lower range (minimum volume) of the initial water volume in terms of the percentage of the entire pipe volume.

- The upper and lower ranges for the initial water volume are stored by the framework and are used to modify the location of the upper bound of each water region shown in Figure 7.4 in accordance with the weightage values calculated by the framework in the previous step. This procedure is conducted automatically at each build point generated by the DACE study.

Based upon the location of the upper range of the initial water region, a pair of probes are introduced in each water region at a horizontal interval of 500 m. The bounds are read from the *mesh.py.template* file and the probes are added to the *probesDict* file located in the "system" directory. At each probe interval, the probe pairs are distributed on the symmetric plane in the manner shown in Figure 5.5, whereby the midpoint of the pipe is calculated from the coordinate data in *mesh.py.template* and the upper and lower probes are located at a vertical distance (Y coordinate) of 0.4 $\times$ D from the mid-point. For probes located in near vertical sections, the probe locations must be checked to ensure they lie within the computational domain.

7.3.5 Step 5a - Running the Optimisation Study: DACE

Upon selection of the DACE study, the user has the option to modify the number of build points used to construct the surrogate models. As the OA-LHS DACE technique is used, the current framework enables a selection of either 25 or 49 points. This operation will update both the *DAK_dace.in* and *DAK_sbo.in* files. The DACE study is then run in two parts. In the first part, the DACE study will generate the location of the surrogate build points within the design space without running the simulations at these build points. This is to give the user the opportunity to input the updated properties of the gas density and viscosity and the cut-off time at each sampling point. This operation will update the *transportProperties* file and the "endTime" parameter located in the *controlDict* file in each working directory. This information is placed by the user into a text file called "updateParams.txt" of the following format that is read by the framework;

```
Dens1  Nu1  t_co1
Dens2  Nu2  t_co2
Dens3  Nu3  t_co2
Dens4  Nu4  t_co2
Dens5  Nu5  t_co2
etc...
```

Here, the first column represents the updated gas density values in kg/m^3, the second column represents the updated gas kinematic viscosity in m^2/s and the third column represents the simulation cut-off time in seconds. In addition, the number of rows is equivalent to the number of DACE sampling points used.

Based upon the information provided by the *DAK_dace.in* file, the analysis driver script, *simulator_script.bat* is used to initiate the second part of the DACE study by facilitating interactions between DAKOTA, OpenFOAM® and SALOME in order to

run the simulation at each sampling point. As an example, the following script shows the pre-processing and analysis tasks implemented in the framework via this script.

```
rem --------------
rem PRE-PROCESSING
rem --------------

rem ## Use deprepro to parse information about the variables from the
    ↪ DAKOTA input file into a p_rgh.in file for pressure and mesh.py.in
    ↪ file for the updated mesh for each surrogate build point

perl C:\Framework_Master_Folder\dprepro params.in p_rgh.template p_rgh.in
perl C:\Framework_Master_Folder\dprepro params.in mesh.py.template mesh.py.
    ↪ in

rem --------
rem ANALYSIS
rem --------

echo "***********************************"
echo "Running in directory: "
pwd
echo "***********************************"

rem ## Copy the contents of the casebase folder into each work directory.

cp -r ../casebase/* .

rem ## Copy the contents of p_rgh into each work directory.

cp p_rgh.in 0/p_rgh

rem ## Copy the contents of mesh.py.in into each work directory.

cp mesh.py.in mesh.py

rem ## This python script invokes a subprocess call to initiate the SALOME
    ↪ TUI and opening mesh.py.

python runSalomeScript.py

rem ## allow enough time for SALOME to open the mesh properly.

sleep 60s

rem ## Run the ideasUnvToFoam utility to convert the mesh.unv file into the
    ↪  contents located in the polyMesh subdirectory of each working
```

```
    ↪ directory.

ideasUnvToFoam mesh.unv

rem ## In each work directory, replace the keywords patch in lines 30 and
    ↪ 42 to wall and symmetryPlane. This is because the Inlet, Outlet,
    ↪ Wall and Symmetry Planes are denoted as patch type by default. The
    ↪ lines 30 to 42 are specific to this particular example and need to
    ↪ be changed accordingly.

sed -i "30s/patch/wall/" ./constant/polyMesh/boundary
sed -i "42s/patch/symmetryPlane/" ./constant/polyMesh/boundary

sleep 15s

rem ## run a Python script to update the gas properties and simulation run
    ↪ time located in runParams.txt at each build point

python changeParameters.py

rem ## Use the setFields utility to define the initial water regions.

setFields > log.setFields

rem ## Use the checkMesh utility to check the mesh.

checkMesh > log.checkMesh

rem ## Use decomposePar for parallel processing.

decomposePar > log.decomposePar

rem ## Run the modified interFoam solver ‘‘interFoamA (Appendix D.1) in
    ↪ parallel depending on the number of processors used.

mpiexec -n 8 interFoamA -parallel > log.interFoam
```

Upon completion of the DACE study, a number of "workdir" directories would have been created by DAKOTA corresponding to each build point. Each of these directories will contain the input parameters and the simulation response recorded at each write interval. At each time step, the total interfacial area is generated by the modified interFoam solver and automatically written to a log file called *log.interFoam* next to the phrase "Sum of Interfacial Area". This area can be extracted along with the elapsed simulation time using simple UNIX commands such as;

```
cat log.interFoam | grep "Time =" | cut -d" " -f3 | sed -n "p;N" >>
    ↪ EX_Time1_1
```

```
cat log.interFoam | grep " Sum of Interfacial Area" | cut -d" " -f7 >>
    ↪ Interface_Area1_1
paste EX_Time1_1 Interface_Area1_1 >> Combined1_1
```

From the above example, the simulation time and corresponding interfacial area are written to a file called "Combined1_1" which can be used to produce the plots of the transient interfacial area similar to those shown in Chapters 5 and 6 of this study. Using the plots of the transient interfacial area at each simulation point, the methodology shown in Section 5.3.1 may be used to determine the amount of transient growth occurring until the simulation cut-off time is reached. In the present form of the framework, the wetted length at each simulation point requires a visual assessment which has been achieved in this study using Paraview. At the simulation cut-off time, the mesh will be reconstructed using the reconstructPar utility and the probe data will be accessible in a new file called "alpha.water" located in a newly created sub-folder called "postProcessing". Following extraction of the data and prior to initiation of the surrogate-based optimisation study using DAKOTA, the data is compiled by the user into a text file called "SBOinput.txt" in the form of the following example;

```
174.34  13.27  108.60   30.0  0  0
170.84  20.87  248.12   29.5  0  0
158.96  8.69   508.90   18.1  0  0
164.27  11.57  1189.74  12.5  0  0
139.22  17.75  1061.54  10.0  13 0
etc...
```

Here, the first two columns represent the pressure and initial water volume variables, respectively, and the third, fourth, fifth and sixth columns represent the amount of hydrate growth occurring until the simulation cut-off time, the wetted length, the summation of the lower probe data, and the summation of the upper probe data, respectively. The number of rows in this text file is equivalent to the number of surrogate build points selected.

7.3.6 Step 5b - Running the Optimisation Study: SBO

In this final step, the user can run the SBO study based upon the information contained in *DAK_sbo.in*. This file will refer to "SBOinput.txt" to construct the four surrogate models based upon the definition of the optimisation problem shown in Section 6.3.5. The output generated from the SBO study will be located in a file called "table.out" and will be in the form of the following example;

```
%eval_id interface          x1          x2    obj_fn_1    obj_fn_2    obj_fn_3  obj_fn_4
1 APPROX_INTERFACE 140.5478072 12.96533097 897.6773999 7.061260848 14.91148208  0
2 APPROX_INTERFACE 140.7232887 22.69951476 1134.510632 7.878567755 31.1342381   0
3 APPROX_INTERFACE 141.3245033 18.26456496 1070.994111 7.82287809  22.16587498  0
4 APPROX_INTERFACE 141.5091403 14.79644154 982.9712674 8.028495503 15.91445312  0
5 APPROX_INTERFACE 141.556444  10.1985229  652.2922271 8.425306729 13.88768722  0
etc...
```

Here, 'x1' and 'x2' are the pressure and initial water volume variables as specified in the DAKOTA input file and the following three columns refer to the S1, S2 and S3 objective functions as defined in Table 6.13. The S4 objective function included here corresponds to the upper probe data and the number of rows in this file is equivalent to the number of points required by MOGA to converge to a solution. These points are equivalent to the points shown in Figures 5.12, 6.18 and 6.19 respectively and can be used in conjunction to the surrogate models for the upper and lower probes sets to determine the Pareto curves that mark the boundaries of the theoretical hydrate risk zones as per Figure 5.6.

Finally, the output from the S1 and S2 surrogate models contained in *table.out* can be used to construct the response surface of the thickness of the hydrate layer at the simulation cut-off time using Equation 5.12. This step constitutes the end of the optimisation study and the TUI can be exited. Prior to the initiation of another study, the working directories are to be deleted and the master folder returned to the default structure as shown in Figure 7.3.

Chapter 8

Closure

8.1 Conclusions

In this research, it has been proposed that a simulation-based optimisation approach may serve as a useful investigative tool for the purpose of reducing the risk of hydrate blockages in subsea oil and gas flowlines. Following the presentation of the final form of the optimisation framework, the most pertinent conclusions have been identified as follows;

1. While the CFD-based approach used in this study has been successfully implemented as an integral part of the optimisation framework, the restriction to two-phase flow is a limiting factor in the overall usefulness of this framework. For example, it is unlikely that a low-spot section of a flowline will be filled entirely with water as assumed throughout this dissertation.

2. The coupling of SALOME and OpenFOAM® to the DAKOTA toolkit is appropriate for the purpose of constructing an optimisation framework. The use of the TUI; however, is not strictly necessary for a practitioner already familiar with these programs.

3. The definitions for the optimisation problem used in Chapters 5 and 6 have produced results that are consistent to the observations made by Leporcher et al. [2002], whereby low energy flows constitute the highest risk for a hydrate blockage to occur.

4. The contribution of the total amount of hydrate growth from the mass transfer limited model by Skovborg and Rasmussen [1994] was observed to be greater than the kinetic model by Turner et al. [2005] for the applied example of the

Ormen Lange flowlines shown in Chapter 6. This is in contrast to the observations made in Chapter 5, where the opposite was true. This may seemingly confirm a supposition by Sloan and Koh [2007] that intrinsic kinetics plays a much smaller role in terms of hydrate growth in actual systems when compared to the effects of mass and heat transfer.

5. The strategy to combine the influence of the kinetic hydrate model by Turner et al. [2005] and the mass transfer limited model by Skovborg and Rasmussen [1994] requires further investigation before the results from the framework may be used in confidence to predict the hydrate growth occurring in an actual flowline.

6. The use of the wetted length parameter to predict the hydrate deposition length is a limiting factor regarding the accuracy of the estimation of hydrate growth as the actual deposition and growth of hydrate particles is far more complex. This phenomena must be included in future versions of the framework. Likewise, accurate depictions of the particle driven flow must be considered.

7. Although the methodology used in Chapter 6 to partition the mesh has been successful as a means to limit the interfacial area recorded by the modified solver to localised regions of the mesh and to foster faster computational times via the implementation of omitted regions, it is not feasible for implementation in the framework due to the additional complexity involved. Instead, it is recommended to explore an option that the interFoam solver may be further modified to identify user defined cell zones.

8. It is highly probable that if the current limitations may be overcome, the optimisation framework could serve as a useful investigative tool to aid a flowline operator in determining a range of inlet flow rates or pressures that will result in a reduced risk of hydrate blockage. Furthermore, if the conceptual ideas presented in this dissertation may be refined appropriately, it may be possible to render the use of chemical inhibitors during restart operations redundant.

8.2 Suggestions for Future Work

Upon conclusion of this dissertation, it is evident that the goal of producing an optimisation framework suitable for any flowline geometry and fluid properties has not yet been achieved in its entirety. In particular, the current framework is still limited to gas-dominated flowlines, due to limited verification techniques, and minimal lateral deviations, on account of the symmetric boundary condition used to reduce the computational expense. It is expected that these issues are not insurmountable and may be overcome in future studies. For this purpose, it is foreseeable that the following suggestions may prove helpful to further increase the usefulness of the optimisation framework generated in this study;

1. Improvement of the wetted length parameter to provide a more accurate depiction of the hydrate deposition.

2. Implementation of a solver for three-phase flow with an applicable validation technique. This will enable the framework to be applicable to oil-dominated flowlines.

3. An adaptation of the optimisation concepts used throughout the course of this study to be used with an industry standard tool to describe the flow.

In addition, throughout the course of this study, it is evident that the optimisation techniques used for the current purpose constitute only a small fraction of the inherent capability that an optimisation methodology has in providing innovative solutions to real world problems. For instance, it may be of benefit to pursue an additional optimisation study with variable geometry as a means to generate the optimal flowline design based upon a constrained gas flow rate. As a simple example, the optimisation problem for the same geometry used by Leporcher et al. [2002] that is constrained to an assumed inlet velocity may be rewritten as;

$$\begin{aligned}
&\text{Maximise:} && L_w, \sum \alpha_{lower} \\
&\text{Minimise:} && \Delta n_{trans}, \frac{\Delta n_{ss}}{\Delta t}, \sum \alpha_{upper} \\
&\text{Subject to:} && h = 0m \\
& && 0.1 \leq y \leq 1.1m \\
& && U_G = 1m/s
\end{aligned}$$

In this example, it is assumed that due to operational constraints, the inlet velocity is limited to 1 m/s and the initial water height is set to a maximum allowable in the low-spot. Here, the pipe depth is optimised to allow all water to be extricated from the pipe. The output will show the maximum initial water volume and corresponding pipe depth allowable to enable removal of all accumulated water for a fixed velocity. In a real world scenario, the pipe depth may be altered using subsea rock installation, as shown in Holden et al. [2005].

Appendix A

CFD/Mechanistic Model Validation

Table A.1: Liquid holdup and pressure gradient results obtained from the CFD model presented in Chapter 3 compared to the Petalas and Aziz mechanistic model (PA) for the horizontal air/water flow.

Simulation number	U_{SG} [m/s]	U_{SL} [m/s]	Liquid holdup (PA)	Pressure gradient (PA) [Pa/m]	Liquid holdup (CFD)	Pressure gradient (CFD) [Pa/m]	Flow Pattern
1	1.5	0.3	0.393	-123.2	0.378	-154.4	SW
2	3	0.3	0.278	-213.7	0.261	-292	SW
3	6	0.3	0.188	-401.4	0.174	-508	SW
4	12	0.3	0.126	-785	0.106	-796	SW
5	18	0.3	0.105	-891	0.125	-1050	AN
6	1.5	0.6	0.497	-304.2	0.548	-328	SW
7	3	0.6	0.362	-495	0.384	-475	SW
8	6	0.6	0.245	-918	0.261	-1224	SW
9	12	0.6	0.16	-1839	0.174	-1570	SW
10	18	0.6	0.146	-1770	0.164	-2611	AN
11	1.5	3	0.702	-2150	0.738	-1360	SL
12	3	3	0.552	-3011	0.597	-2644	SL
13	6	3	0.394	-5605	0.378	-4620	SL
14	12	3	0.318	-8355	0.307	-12097	FR
15	18	3	0.283	-10396	0.239	-19638	FR
16	1.5	12	0.917	-20539	0.919	-25226	DB
17	3	12	0.86	-22385	0.845	-20058	DB
18	6	12	0.732	-27927	0.715	-28335	DB
19	12	12	0.619	-35630	0.621	-43743	DB
20	18	12	0.556	-42523	0.524	-53652	FR

Table A.2: Liquid holdup and pressure gradient results obtained from the CFD model presented in Chapter 3 compared to the Petalas and Aziz mechanistic model (PA) for the +10° inclined air/water flow.

Simulation number	U_{SG} [m/s]	U_{SL} [m/s]	Liquid holdup (PA)	Pressure gradient (PA) [Pa/m]	Liquid holdup (CFD)	Pressure gradient (CFD) [Pa/m]	Flow Pattern
1	1.5	0.3	0.377	-811	-0.378	-655	SL
2	3	0.3	0.257	-738	0.261	-607	SL
3	6	0.3	0.239	-648	0.217	-837	FR
4	12	0.3	0.159	-693	0.154	-840	AN
5	18	0.3	0.114	-943	0.115	-1414	AN
6	1.5	0.6	0.45	-1103	0.402	-925	SL
7	3	0.6	0.312	-1159	0.307	-1121	SL
8	6	0.6	0.24	-1307	0.261	-1125	SL
9	12	0.6	0.202	-1364	0.154	-3155	AN
10	18	0.6	0.154	-1859	0.0975	-4180	AN
11	1.5	3	0.708	-3363	0.645	-3136	SL
12	3	3	0.564	-3976	0.475	-5113	SL
13	6	3	0.404	-6173	0.402	-5474	SL
14	12	3	0.32	-8819	0.307	-11742	FR
15	18	3	0.286	-10710	0.261	-8175	FR
16	1.5	12	0.913	-22229	0.884	-25325	DB
17	3	12	0.846	-24253	0.845	-33074	DB
18	6	12	0.745	-27983	0.715	-27365	DB
19	12	12	0.638	-34022	0.645	-40511	DB
20	18	12	0.548	-43540	0.5	-64376	FR

Table A.3: Liquid holdup and pressure gradient results obtained from the CFD model presented in Chapter 3 compared to the Petalas and Aziz mechanistic model (PA) for the -10° declined air/water flow.

Simulation number	U_{SG} [m/s]	U_{SL} [m/s]	Liquid holdup (PA)	Pressure gradient (PA) [Pa/m]	Liquid holdup (CFD)	Pressure gradient (CFD) [Pa/m]	Flow Pattern
1	1.5	0.3	0.199	-20	0.174	-9.66	SW
2	3	0.3	0.168	-186.7	0.154	-128	SW
3	6	0.3	0.133	-467.2	0.115	-329.6	SW
4	12	0.3	0.101	-929.3	0.0975	-864	SW
5	18	0.3	0.0989	-850.7	0.0803	-1150.2	AN
6	1.5	0.6	0.325	-33.9	0.3072	-51.1	SW
7	3	0.6	0.267	-342.5	0.284	-303.7	SW
8	6	0.6	0.203	-897.2	0.217	-721	SW
9	12	0.6	0.144	-1930	0.154	-1453	AN
10	18	0.6	0.141	-1690	0.125	-1942	AN
11	1.5	3	0.696	-958	0.662	-1671	SL
12	3	3	0.543	-2136	0.597	-1851	SL
13	6	3	0.384	-5137	0.28	-3671	SL
14	12	3	0.31	-8217	0.286	-6743	FR
15	18	3	0.277	-10355	0.239	-12414	FR
16	1.5	12	0.91	-19279	0.884	-22607	DB
17	3	12	0.83	-22062	0.845	-31794	DB
18	6	12	0.731	-26675	0.738	-20433	DB
19	12	12	0.6	-36151	0.59	-39235	DB
20	18	12	0.555	-41753	0.5	-46842	FR

Table A.4: Liquid holdup and pressure gradient results obtained from the CFD model presented in Chapter 3 compared to the Petalas and Aziz mechanistic model (PA) for the horizontal oil/gas flow.

Simulation number	U_{SG} [m/s]	U_{SL} [m/s]	Liquid holdup (PA)	Pressure gradient (PA) [Pa/m]	Liquid holdup (CFD)	Pressure gradient (CFD) [Pa/m]	Flow Pattern
1	1.5	0.3	0.31	-165	0.28	-82.3	AN
2	3	0.3	0.199	-339	0.19	-185.7	AN
3	6	0.3	0.118	-596	0.13	-551.6	AN
4	12	0.3	0.065	-2209	0.094	-1692.2	AN
5	18	0.3	0.045	-3182	0.064	-3655	SL
6	1.5	0.6	0.44	-317	0.42	-108.3	AN
7	3	0.6	0.29	-586	0.28	-266	AN
8	6	0.6	0.187	-1276	0.20	-682.9	AN
9	12	0.6	0.109	-3120	0.12	-2020	AN
10	18	0.6	0.0769	-5701	0.091	-4493	AN
11	1.5	3	0.7	-2500	0.70	-782	SL
12	3	3	0.608	-3693	0.58	-1120	SL
13	6	3	0.458	-5896	0.42	-1952.1	SL
14	12	3	0.315	-8600	0.30	-4177.6	AN
15	18	3	0.214	-14250	0.22	-7522	AN
16	1.5	12	0.889	-9015	0.81	-7650	DB
17	3	12	0.8	-11886	0.81	-8509	DB
18	6	12	0.667	-15131	0.7	-11764	DB
19	12	12	0.5	-20850	0.56	-23367	DB
20	18	12	0.42	-27405	0.49	-23954	FR

Table A.5: Liquid holdup and pressure gradient results obtained from the CFD model presented in Chapter 3 compared to the Petalas and Aziz mechanistic model (PA) for the $+10^{\circ}$ inclined oil/gas flow.

Simulation number	U_{SG} [m/s]	U_{SL} [m/s]	Liquid holdup (PA)	Pressure gradient (PA) [Pa/m]	Liquid holdup (CFD)	Pressure gradient (CFD) [Pa/m]	Flow Pattern
1	1.5	0.3	0.33	-849	0.37	-840	SL
2	3	0.3	0.23	-825	0.21	-773	SL
3	6	0.3	0.16	-1084	0.13	-972	AN
4	12	0.3	0.092	-2248	0.075	-1922	AN
5	18	0.3	0.069	-4196	0.065	-3791	AN
6	1.5	0.6	0.43	-1070	0.52	-1015	SL
7	3	0.6	0.31	-1066	0.33	-912	SL
8	6	0.6	0.22	-1376	0.18	-1216	AN
9	12	0.6	0.13	-2722	0.12	-2998	AN
10	18	0.6	0.10	-5019	0.11	-4527	AN
11	1.5	3	0.71	-2653	0.76	-2136	SL
12	3	3	0.55	-2909	0.61	-2113	SL
13	6	3	0.44	-3822	0.43	-2510	SL
14	12	3	0.33	-6497	0.32	-5261	AN
15	18	3	0.24	-10546	0.25	-9094	AN
16	1.5	12	0.89	-12907	0.87	-9468	DB
17	3	12	0.80	-15946	0.80	-13120	DB
18	6	12	0.66	-19652	0.70	-15365	DB
19	12	12	0.50	-28589	0.57	-22138	DB
20	18	12	0.42	-41158	0.50	-36980	FR

Table A.6: Liquid holdup and pressure gradient results obtained from the CFD model presented in Chapter 3 compared to the Petalas and Aziz mechanistic model (PA) for the -10° inclined oil/gas flow.

Simulation number	U_{SG} [m/s]	U_{SL} [m/s]	Liquid holdup (PA)	Pressure gradient (PA) [Pa/m]	Liquid holdup (CFD)	Pressure gradient (CFD) [Pa/m]	Flow Pattern
1	1.5	0.3	0.46	-1030	0.37	-865	SW
2	3	0.3	0.31	-923	0.29	-760	AN
3	6	0.3	0.20	-1112	0.22	-1080	AN
4	12	0.3	0.11	-2195	0.12	-1968	AN
5	18	0.3	0.076	-4093	0.067	-3863	AN
6	1.5	0.6	0.56	-1247	0.52	-1118	SL
7	3	0.6	0.41	-1103	0.42	-1236	AN
8	6	0.6	0.27	-1441	0.23	-1379	AN
9	12	0.6	0.15	-2686	0.13	-2585	AN
10	18	0.6	0.11	-4970	0.10	-4590	AN
11	1.5	3	0.76	-2661	0.72	-2350	SL
12	3	3	0.59	-2951	0.57	-2550	SL
13	6	3	0.46	-3831	0.40	-3720	SL
14	12	3	0.33	-6497	0.31	-6120	AN
15	18	3	0.24	-10546	0.20	-9596	AN
16	1.5	12	0.89	-12907	0.92	-10598	DB
17	3	12	0.80	-15946	0.85	-11390	DB
18	6	12	0.66	-19652	0.60	-17654	DB
19	12	12	0.50	-28589	0.51	-26580	DB
20	18	12	0.42	-41158	0.46	-37243	FR

Appendix B

SALOME Script Files

This Appendix presents the Python script files to automatically generate 3-D half-pipe meshes using SALOME 7.4.0 for the following cases:

- Original model used for air/water flow in Chapter 3.
- Modified model used for the comparison with Birvalski [2015] in Chapter 4.
- Modified model used for the third segment of the Ormen Lange flowline in Chapter 6.

These three examples encompass the meshing strategy used for all meshes presented in this study. In order to use these examples for optimisation studies with DAKOTA, the Python mesh file is first exported as a .UNV file, which is then converted for use in OpenFOAM® using the *ideasUnvToFoam* utility. To do this, the last two lines of all Python files presented here must be replaced with the following code:

```
## EXPORT MESH
print "exporting the mesh....."
retexp = Mesh_1.ExportUNV(path)  ## Note the name of the mesh used here

if retexp == 0:
    print "ERROR: problem when exporting the mesh!"
else:
    print ':) mesh exported in %s.' % path
    pass

## TO CLOSE SALOME PORTS
from killSalomeWithPort import killMyPort
killMyPort(os.getenv('NSPORT'))

print "Port killed: %s" % os.getenv('NSPORT')
```

B.1 3-D Half-pipe - Original Model

```
### This file is a modified version of the code generated automatically by
    ↪ SALOME v7.4.0 with dump python functionality

import sys
import salome
salome.salome_init()
theStudy = salome.myStudy

import salome_notebook
notebook = salome_notebook.NoteBook(theStudy)
sys.path.insert( 0, r'C:/WorkDirectory')

### GEOM component

import GEOM
from salome.geom import geomBuilder
import math
import SALOMEDS
import os

path = os.getcwd() + "/mesh.unv"
geompy = geomBuilder.New(theStudy)

O = geompy.MakeVertex(0, 0, 0)
OX = geompy.MakeVectorDXDYDZ(1, 0, 0)
OY = geompy.MakeVectorDXDYDZ(0, 1, 0)
OZ = geompy.MakeVectorDXDYDZ(0, 0, 1)

## Define initial and final vertices and construct Line_1
Vertex_1 = geompy.MakeVertex(0, 0, 0)  ## First coordinate
Vertex_2 = geompy.MakeVertex(4, 0, 0)  ## Final coordinate
Line_1 = geompy.MakeLineTwoPnt(Vertex_1, Vertex_2)

## Make cross section for extrusion along Line_1
Divided_Disk_1 = geompy.MakeDividedDisk(0.026, 2, GEOM.SQUARE)  ## Pipe
    ↪ radius
Rotation_1 = geompy.MakeRotation(Divided_Disk_1, OX, 45*math.pi/180.0)
Face_1 = geompy.MakeFaceHW(0.1, 0.1, 3)
Extrusion_1 = geompy.MakePrismVecH(Face_1, OY, -0.05)
Cut_1 = geompy.MakeCutList(Rotation_1, [Extrusion_1], True)
Plane_1 = geompy.MakePlaneLCS(None, 0.1, 1)
Partition_1 = geompy.MakePartition([Cut_1], [Plane_1], [], [], geompy.
    ↪ ShapeType["SHELL"], 0, [], 0)
Vertex_3 = geompy.MakeVertex(0, 0.022, 0)
Vertex_4 = geompy.MakeVertex(0, 0, 0.022)
```

```
Vertex_5 = geompy.MakeVertex(0, 0, -0.022)
Arc_1 = geompy.MakeArc(Vertex_4, Vertex_3, Vertex_5)
Extrusion_2 = geompy.MakePrismVecH2Ways(Arc_1, OX, 0.05)
Partition_2 = geompy.MakePartition([Partition_1], [Extrusion_2], [], [],
    ↪ geompy.ShapeType["SHELL"], 0, [], 0)

## Create pipe by extruding Partition_2 along Line_1
Pipe_1 = geompy.MakePipe(Partition_2, Line_1)

## Define patches from GUI selection
Inlet_Air = geompy.CreateGroup(Pipe_1, geompy.ShapeType["FACE"])
geompy.UnionIDs(Inlet_Air, [158, 172, 38, 4, 62])
Inlet_Water = geompy.CreateGroup(Pipe_1, geompy.ShapeType["FACE"])
geompy.UnionIDs(Inlet_Water, [110, 144, 178, 202, 86])
Symmetry = geompy.CreateGroup(Pipe_1, geompy.ShapeType["FACE"])
geompy.UnionIDs(Symmetry, [14, 185, 205, 45, 69, 93])
Wall = geompy.CreateGroup(Pipe_1, geompy.ShapeType["FACE"])
geompy.UnionIDs(Wall, [127, 162, 192, 55])
Flow_Edge = geompy.CreateGroup(Pipe_1, geompy.ShapeType["EDGE"])
geompy.UnionIDs(Flow_Edge, [16, 18, 23, 28, 47, 52, 71, 76, 95, 100, 122,
    ↪ 124, 129, 134, 187, 189])
Outlet_Water = geompy.CreateGroup(Pipe_1, geompy.ShapeType["FACE"])
geompy.UnionIDs(Outlet_Water, [174, 208, 140, 106, 168, 34, 58, 82, 198,
    ↪ 154])
geompy.DifferenceIDs(Outlet_Water, [174, 208, 140, 106, 168, 34, 58, 82,
    ↪ 198, 154])
geompy.UnionIDs(Outlet_Water, [168, 174, 34, 58, 82])
geompy.DifferenceIDs(Outlet_Water, [168, 174, 34, 58, 82])
geompy.UnionIDs(Outlet_Water, [106, 140, 154, 198, 208])
Outlet_Air = geompy.CreateGroup(Pipe_1, geompy.ShapeType["FACE"])
geompy.UnionIDs(Outlet_Air, [168, 174, 34, 58, 82])
Radial = geompy.CreateGroup(Pipe_1, geompy.ShapeType["EDGE"])
geompy.UnionIDs(Radial, [112, 117, 180, 40, 44, 126, 136, 191, 49, 54])
geompy.DifferenceIDs(Radial, [112, 117, 180, 40, 44, 126, 136, 191, 49,
    ↪ 54])
geompy.UnionIDs(Radial, [117, 180, 40, 136, 191, 49])
Lower = geompy.CreateGroup(Pipe_1, geompy.ShapeType["EDGE"])
geompy.UnionIDs(Lower, [112, 44, 126, 54])
geompy.DifferenceIDs(Radial, [117, 180, 40, 136, 191, 49])
geompy.UnionIDs(Radial, [117, 40, 136, 49])
geompy.DifferenceIDs(Lower, [112, 44, 126, 54])
geompy.UnionIDs(Lower, [191, 180])
Diag = geompy.CreateGroup(Pipe_1, geompy.ShapeType["EDGE"])
geompy.UnionIDs(Diag, [112, 44, 54, 126])

### SMESH component
```

```
import SMESH, SALOMEDS
from salome.smesh import smeshBuilder

smesh = smeshBuilder.New(theStudy)
Mesh_1 = smesh.Mesh(Pipe_1)
Regular_1D = Mesh_1.Segment()
Nb_Segments_1 = Regular_1D.NumberOfSegments(8)
Nb_Segments_1.SetDistrType( 0 )
Quadrangle_2D = Mesh_1.Quadrangle(algo=smeshBuilder.QUADRANGLE)
Hexa_3D = Mesh_1.Hexahedron(algo=smeshBuilder.Hexa)
Nb_Segments_2 = smesh.CreateHypothesis('NumberOfSegments')
status = Mesh_1.AddHypothesis(Regular_1D,Radial)
status = Mesh_1.AddHypothesis(Nb_Segments_2,Radial)
[ SubMesh_1 ] = Mesh_1.GetMesh().GetSubMeshes()
status = Mesh_1.AddHypothesis(Regular_1D,Diag)
status = Mesh_1.AddHypothesis(Nb_Segments_2,Diag)
[ SubMesh_1, SubMesh_2 ] = Mesh_1.GetMesh().GetSubMeshes()
status = Mesh_1.AddHypothesis(Regular_1D,Lower)
status = Mesh_1.AddHypothesis(Nb_Segments_2,Lower)
[ SubMesh_1, SubMesh_2, SubMesh_3 ] = Mesh_1.GetMesh().GetSubMeshes()
Nb_Segments_3 = smesh.CreateHypothesis('NumberOfSegments')
status = Mesh_1.AddHypothesis(Regular_1D,Flow_Edge)
status = Mesh_1.AddHypothesis(Nb_Segments_3,Flow_Edge)
[ SubMesh_1, SubMesh_2, SubMesh_3, SubMesh_4 ] = Mesh_1.GetMesh().
    ↪ GetSubMeshes()
Inlet_Air_1 = Mesh_1.GroupOnGeom(Inlet_Air,'Inlet_Air',SMESH.FACE)
Inlet_Water_1 = Mesh_1.GroupOnGeom(Inlet_Water,'Inlet_Water',SMESH.FACE)
Symmetry_1 = Mesh_1.GroupOnGeom(Symmetry,'Symmetry',SMESH.FACE)
Wall_1 = Mesh_1.GroupOnGeom(Wall,'Wall',SMESH.FACE)
Outlet_Water_1 = Mesh_1.GroupOnGeom(Outlet_Water,'Outlet_Water',SMESH.FACE)
Outlet_Air_1 = Mesh_1.GroupOnGeom(Outlet_Air,'Outlet_Air',SMESH.FACE)
aCriteria = []
aCriterion = smesh.GetCriterion(SMESH.EDGE,SMESH.FT_FreeFaces,'=',0,SMESH.
    ↪ FT_Undefined,SMESH.FT_Undefined,1e-007)
aCriteria.append(aCriterion)
[ Inlet_Air_1, Inlet_Water_1, Symmetry_1, Wall_1, Outlet_Water_1,
    ↪ Outlet_Air_1 ] = Mesh_1.GetGroups()

## Define number of segments in region next to wall defined by Arc_1
Nb_Segments_2.SetNumberOfSegments( 10 )

## Define type of grading and scale factor for cross sectional elements
    ↪ next to wall
Nb_Segments_2.SetDistrType( 1 )
Nb_Segments_2.SetScaleFactor( 0.1 )
Nb_Segments_2.SetReversedEdges( [ 117, 40, 136, 49 ] )
Nb_Segments_2.SetObjectEntry( "0:1:1:48" )
```

```
## Define number of longitudinal segments
Nb_Segments_3.SetNumberOfSegments( 500 )
Nb_Segments_3.SetDistrType( 0 )

## Compile mesh
isDone = Mesh_1.Compute()
[ SubMesh_1, SubMesh_2, SubMesh_3, SubMesh_4 ] = Mesh_1.GetMesh().
    ↪ GetSubMeshes()
[ Inlet_Air_1, Inlet_Water_1, Symmetry_1, Wall_1, Outlet_Water_1,
    ↪ Outlet_Air_1 ] = Mesh_1.GetGroups()
[ SubMesh_3, SubMesh_4, SubMesh_1, SubMesh_2 ] = Mesh_1.GetMesh().
    ↪ GetSubMeshes()
SubMesh_1 = Mesh_1.GetSubMesh( Radial, 'SubMesh_1' )
SubMesh_2 = Mesh_1.GetSubMesh( Diag, 'SubMesh_2' )
SubMesh_3 = Mesh_1.GetSubMesh( Lower, 'SubMesh_3' )
SubMesh_4 = Mesh_1.GetSubMesh( Flow_Edge, 'SubMesh_4' )

## Set mesh name to 'Mesh_1'
smesh.SetName(Mesh_1.GetMesh(), 'Mesh_1')

if salome.sg.hasDesktop():
  salome.sg.updateObjBrowser(1)
```

B.2 Birvalski [2015] Low-spot Bend Geometry

```
### This file is a modified version of the code generated automatically by
    ↪ SALOME v7.4.0 with dump python functionality

import sys
import salome

salome.salome_init()
theStudy = salome.myStudy

import salome_notebook
notebook = salome_notebook.NoteBook(theStudy)
sys.path.insert( 0, r'C:/WorkDirectory')

### GEOM component

import GEOM
from salome.geom import geomBuilder
import math
import SALOMEDS
import os
```

```
path = os.getcwd() + "/mesh.unv"
geompy = geomBuilder.New(theStudy)

O = geompy.MakeVertex(0, 0, 0)
OX = geompy.MakeVectorDXDYDZ(1, 0, 0)
OY = geompy.MakeVectorDXDYDZ(0, 1, 0)
OZ = geompy.MakeVectorDXDYDZ(0, 0, 1)

## Define pipe vertices and construct Curve_1
Vertex_0 = geompy.MakeVertex(0, 0, 0)
Vertex_1 = geompy.MakeVertex(10, 0, 0)
Vertex_2 = geompy.MakeVertex(20, 0, -0.29)
Vertex_3 = geompy.MakeVertex(30, 0, 0)
Vertex_4 = geompy.MakeVertex(40, 0, 0)
Divided_Disk_1 = geompy.MakeDividedDisk(0.0254, 2, GEOM.SQUARE)  ## Pipe
    ↪ radius
Rotation_1 = geompy.MakeRotation(Divided_Disk_1, OX, 45*math.pi/180.0)
Plane_1 = geompy.MakePlaneLCS(None, 0.1, 3)
Extrusion_1 = geompy.MakePrismVecH(Plane_1, OY, -0.05)
Cut_1 = geompy.MakeCutList(Rotation_1, [Extrusion_1], True)
Curve_1 = geompy.MakePolyline([Vertex_0, Vertex_1, Vertex_2, Vertex_3,
    ↪ Vertex_4], False)

## Define lower and upper bounds of the water region and create Curve 1-3
Curve_1_edge_5 = geompy.GetSubShape(Curve_1, [5])
Vertex_L = geompy.MakeVertexOnCurve(Curve_1_edge_5, 0.5)
Curve_1_edge_7 = geompy.GetSubShape(Curve_1, [7])
Vertex_U = geompy.MakeVertexOnCurve(Curve_1_edge_7, 0.5)

Curve_INLET = geompy.MakePolyline([Vertex_0, Vertex_1, Vertex_L], False)
Curve_WR = geompy.MakePolyline([Vertex_L, Vertex_2, Vertex_U], False)
Curve_OUTLET = geompy.MakePolyline([Vertex_U, Vertex_3, Vertex_4], False)

## Construct Pipe 1 and create Inlet, Wall, Symmetry & Flow Edges groups
    ↪ via GUI selection
Pipe_Inlet = geompy.MakePipe(Cut_1, Curve_INLET)
Inlet = geompy.CreateGroup(Pipe_Inlet, geompy.ShapeType["FACE"])
geompy.UnionIDs(Inlet, [120, 4, 58, 95])
Inlet_01 = geompy.CreateGroup(Pipe_Inlet, geompy.ShapeType["FACE"])
geompy.UnionIDs(Inlet_01, [116, 141, 54, 91])
Symmetry = geompy.CreateGroup(Pipe_Inlet, geompy.ShapeType["FACE"])
geompy.UnionIDs(Symmetry, [125, 70, 130, 51, 65, 48])
Wall = geompy.CreateGroup(Pipe_Inlet, geompy.ShapeType["FACE"])
geompy.UnionIDs(Wall, [110, 38, 135, 113, 138, 43])
FE_01 = geompy.CreateGroup(Pipe_Inlet, geompy.ShapeType["EDGE"])
geompy.UnionIDs(FE_01, [77, 23, 132, 102, 82, 18, 30, 72, 67, 127, 40, 16,
    ↪ 45, 35, 25, 107])
```

```
## Construct Pipe 2 and create Inlet, Wall, Symmetry, Flow Edges & Water
    ↪ Region groups via GUI selection
Pipe_WR = geompy.MakePipe(Inlet_O1, Curve_WR)
WR_O1 = geompy.CreateGroup(Pipe_WR, geompy.ShapeType["FACE"])
geompy.UnionIDs(WR_O1, [4, 95, 132, 58])
WR_O2 = geompy.CreateGroup(Pipe_WR, geompy.ShapeType["FACE"])
geompy.UnionIDs(WR_O2, [141, 54, 128, 91])
Symmetry_1 = geompy.CreateGroup(Pipe_WR, geompy.ShapeType["FACE"])
geompy.UnionIDs(Symmetry_1, [72, 122, 65, 138, 125, 135])
Wall_1 = geompy.CreateGroup(Pipe_WR, geompy.ShapeType["FACE"])
geompy.UnionIDs(Wall_1, [79, 33, 117, 28, 82, 112])
FE_WR = geompy.CreateGroup(Pipe_WR, geompy.ShapeType["EDGE"])
geompy.UnionIDs(FE_WR, [69, 64, 18, 81, 25, 32, 35, 37, 116, 119, 121, 9,
    ↪ 30, 76, 84, 101, 114, 74, 104, 67, 45, 16, 109, 23, 40])
geompy.DifferenceIDs(FE_WR, [69, 64, 18, 81, 25, 32, 35, 37, 116, 119, 121,
    ↪  9, 30, 76, 84, 101, 114, 74, 104, 67, 45, 16, 109, 23, 40])
geompy.UnionIDs(FE_WR, [69, 18, 25, 35, 119, 30, 76, 114, 74, 104, 67, 45,
    ↪ 16, 109, 23, 40])
Water_Region = geompy.CreateGroup(Pipe_WR, geompy.ShapeType["SOLID"])
geompy.UnionIDs(Water_Region, [93, 56, 2, 130])

## Construct Pipe 3 and create Inlet, Wall & Symmetry groups via GUI
    ↪ selection
Pipe_Outlet = geompy.MakePipe(WR_O2, Curve_OUTLET)
Outlet_O1 = geompy.CreateGroup(Pipe_Outlet, geompy.ShapeType["FACE"])
geompy.UnionIDs(Outlet_O1, [95, 4, 58, 120])
Outlet = geompy.CreateGroup(Pipe_Outlet, geompy.ShapeType["FACE"])
geompy.UnionIDs(Outlet, [141, 54, 116, 91])
Symmetry_2 = geompy.CreateGroup(Pipe_Outlet, geompy.ShapeType["FACE"])
geompy.UnionIDs(Symmetry_2, [130, 33, 110, 28, 113, 125])
Wall_2 = geompy.CreateGroup(Pipe_Outlet, geompy.ShapeType["FACE"])
geompy.UnionIDs(Wall_2, [75, 100, 138, 135, 105, 80])
FE_O2 = geompy.CreateGroup(Pipe_Outlet, geompy.ShapeType["EDGE"])
geompy.UnionIDs(FE_O2, [16, 18, 23, 25, 30, 35, 40, 45, 67, 72, 77, 82,
    ↪ 102, 107, 127, 132])

### SMESH component

import SMESH, SALOMEDS
from salome.smesh import smeshBuilder

smesh = smeshBuilder.New(theStudy)
Mesh_1 = smesh.Mesh(Pipe_Inlet)
Regular_1D = Mesh_1.Segment()

## Number of cross sectional segments
```

```
Nb_Segments_1 = Regular_1D.NumberOfSegments(5)
Nb_Segments_1.SetDistrType( 0 )
Quadrangle_2D = Mesh_1.Quadrangle(algo=smeshBuilder.QUADRANGLE)
Prism_3D = Mesh_1.Prism()
status = Mesh_1.AddHypothesis(Quadrangle_2D,Inlet)
Projection_2D = Mesh_1.Projection2D(geom=Inlet_O1)
Source_Face_1 = Projection_2D.SourceFace(Inlet,None,None,None,None,None)
isDone = Mesh_1.Compute()
[ SubMesh_1, SubMesh_2 ] = Mesh_1.GetMesh().GetSubMeshes()
Mesh_2 = smesh.Mesh(Pipe_WR)
status = Mesh_2.AddHypothesis(Nb_Segments_1)
status = Mesh_2.AddHypothesis(Regular_1D)
status = Mesh_2.AddHypothesis(Quadrangle_2D)
status = Mesh_2.AddHypothesis(Prism_3D)
status = Mesh_2.AddHypothesis(Quadrangle_2D,WR_O1)
Source_Face_2 = smesh.CreateHypothesis('ProjectionSource2D')
Source_Face_2.SetSourceFace( WR_O1 )
Source_Face_2.SetSourceMesh( None )
Source_Face_2.SetVertexAssociation( None, None, None, None )
status = Mesh_2.AddHypothesis(Projection_2D,WR_O2)
status = Mesh_2.AddHypothesis(Source_Face_2,WR_O2)
isDone = Mesh_2.Compute()
[ SubMesh_3, SubMesh_4 ] = Mesh_2.GetMesh().GetSubMeshes()
Mesh_3 = smesh.Mesh(Pipe_Outlet)
status = Mesh_3.AddHypothesis(Nb_Segments_1)
status = Mesh_3.AddHypothesis(Regular_1D)
status = Mesh_3.AddHypothesis(Quadrangle_2D)
status = Mesh_3.AddHypothesis(Prism_3D)
status = Mesh_3.AddHypothesis(Quadrangle_2D,Outlet_O1)
Source_Face_3 = smesh.CreateHypothesis('ProjectionSource2D')
Source_Face_3.SetSourceFace( Outlet_O1 )
Source_Face_3.SetSourceMesh( None )
Source_Face_3.SetVertexAssociation( None, None, None, None )
status = Mesh_3.AddHypothesis(Projection_2D,Outlet)
status = Mesh_3.AddHypothesis(Source_Face_3,Outlet)
isDone = Mesh_3.Compute()
[ SubMesh_5, SubMesh_6 ] = Mesh_3.GetMesh().GetSubMeshes()
Inlet_1 = Mesh_1.GroupOnGeom(Inlet,'Inlet',SMESH.FACE)
Outlet_1 = Mesh_3.GroupOnGeom(Outlet,'Outlet',SMESH.FACE)
Water_Region_1 = Mesh_2.GroupOnGeom(Water_Region,'Water_Region',SMESH.
    ↪ VOLUME)
Nb_Segments_FE1 = smesh.CreateHypothesis('NumberOfSegments')

## Input number of longitudinal segments between vertices
Nb_Segments_FE1.SetNumberOfSegments( 400 )    ## Resolution: 2000 cells/m
Nb_Segments_FE1.SetDistrType( 0 )
status = Mesh_1.AddHypothesis(Regular_1D,FE_O1)
```

```
status = Mesh_1.AddHypothesis(Nb_Segments_FE1,FE_01)
isDone = Mesh_1.Compute()
[ SubMesh_1, SubMesh_2, SubMesh_7 ] = Mesh_1.GetMesh().GetSubMeshes()
[ Inlet_1 ] = Mesh_1.GetGroups()
status = Mesh_2.AddHypothesis(Regular_1D,FE_WR)
status = Mesh_2.AddHypothesis(Nb_Segments_FE1,FE_WR)
isDone = Mesh_2.Compute()
[ SubMesh_3, SubMesh_4, SubMesh_8 ] = Mesh_2.GetMesh().GetSubMeshes()
[ Water_Region_1 ] = Mesh_2.GetGroups()
status = Mesh_3.AddHypothesis(Regular_1D,FE_02)
status = Mesh_3.AddHypothesis(Nb_Segments_FE1,FE_02)
isDone = Mesh_3.Compute()
[ SubMesh_5, SubMesh_6, SubMesh_9 ] = Mesh_3.GetMesh().GetSubMeshes()
[ Outlet_1 ] = Mesh_3.GetGroups()
Symmetry_3 = Mesh_1.GroupOnGeom(Symmetry,'Symmetry',SMESH.FACE)
Wall_3 = Mesh_1.GroupOnGeom(Wall,'Wall',SMESH.FACE)
Symmetry_4 = Mesh_2.GroupOnGeom(Symmetry_1,'Symmetry',SMESH.FACE)
Wall_4 = Mesh_2.GroupOnGeom(Wall_1,'Wall',SMESH.FACE)
Symmetry_5 = Mesh_3.GroupOnGeom(Symmetry_2,'Symmetry',SMESH.FACE)
Wall_5 = Mesh_3.GroupOnGeom(Wall_2,'Wall',SMESH.FACE)
Compound_Mesh_1 = smesh.Concatenate([Mesh_1.GetMesh(), Mesh_2.GetMesh(),
    ↪ Mesh_3.GetMesh()], 1, 1, 1e-005)
[ Inlet_2, Symmetry_6, Wall_6, Water_Region_2, Outlet_2 ] = Compound_Mesh_1
    ↪ .GetGroups()
SubMesh_1 = Mesh_1.GetSubMesh( Inlet, 'SubMesh_1' )
SubMesh_2 = Projection_2D.GetSubMesh()
SubMesh_3 = Mesh_2.GetSubMesh( WR_01, 'SubMesh_3' )
SubMesh_4 = Mesh_2.GetSubMesh( WR_02, 'SubMesh_4' )
SubMesh_5 = Mesh_3.GetSubMesh( Outlet_01, 'SubMesh_5' )
SubMesh_6 = Mesh_3.GetSubMesh( Outlet, 'SubMesh_6' )
SubMesh_7 = Mesh_1.GetSubMesh( FE_01, 'SubMesh_7' )
SubMesh_8 = Mesh_2.GetSubMesh( FE_WR, 'SubMesh_8' )
SubMesh_9 = Mesh_3.GetSubMesh( FE_02, 'SubMesh_9' )

## Set names of Mesh objects
smesh.SetName(Compound_Mesh_1.GetMesh(), 'Compound_Mesh_1')

if salome.sg.hasDesktop():
 salome.sg.updateObjBrowser(1)
```

B.3 Ormen Lange - Third Segment

```
### This file is a modified version of the code generated automatically by
    ↪ SALOME v7.4.0 with dump python functionality

import sys
import salome
import salome_notebook

salome.salome_init()
theStudy = salome.myStudy
notebook = salome_notebook.NoteBook(theStudy)
sys.path.insert( 0, r'C:/WorkDirectory')

# Define functions to determine Wall IDs & Edge IDs

def wall_id(Pipe_ID, curveThreshold):
  AllFaces = geompy.SubShapeAll(Pipe_ID,geompy.ShapeType["FACE"])
  keptFaces = []
  for face in AllFaces:
    curvature = geompy.MaxSurfaceCurvatureByPoint(face, Vertex_0)
    if curvature < curveThreshold:
      keptFaces.append(face)
  WallFaces = geompy.GetSubShapesIDs(Pipe_ID, keptFaces)
  return (WallFaces)

def edge_id(Pipe_ID):
  AllEdges = geompy.SubShapeAll(Pipe_ID, geompy.ShapeType["EDGE"])
  keptEdges = []
  for edge in AllEdges:
    length = geompy.BasicProperties(edge)[0]
    if length > Radius*2:
      keptEdges.append(edge)
  FlowEdges = geompy.GetSubShapesIDs(Pipe_ID, keptEdges)
  return (FlowEdges)

### GEOM component

import GEOM
from salome.geom import geomBuilder
import math
import SALOMEDS
import os

path = os.getcwd() + "/mesh.unv"
geompy = geomBuilder.New(theStudy)
```

```
O = geompy.MakeVertex(0, 0, 0)
OX = geompy.MakeVectorDXDYDZ(1, 0, 0)
OY = geompy.MakeVectorDXDYDZ(0, 1, 0)
OZ = geompy.MakeVectorDXDYDZ(0, 0, 1)

## Pipe verticies
Vertex_0 = geompy.MakeVertex(0, 0, 0)
Vertex_1 = geompy.MakeVertex(500, 0, 0)
Vertex_2 = geompy.MakeVertex(1000, 0, -8)
Vertex_3 = geompy.MakeVertex(2000, 0, -15)
Vertex_4 = geompy.MakeVertex(3000, 0, -30)
Vertex_5 = geompy.MakeVertex(4000, 0, -53)
Vertex_6 = geompy.MakeVertex(5000, 0, -65)
Vertex_7 = geompy.MakeVertex(6000, 0, -80)
Vertex_8 = geompy.MakeVertex(7000, 0, -45)
Vertex_9 = geompy.MakeVertex(8000, 0, -130)
Vertex_10 = geompy.MakeVertex(9000, 0, -125)
Vertex_11 = geompy.MakeVertex(10000, 0, -125)
Vertex_12 = geompy.MakeVertex(11000, 0, -124)
Vertex_13 = geompy.MakeVertex(12000, 0, -122)
Vertex_14 = geompy.MakeVertex(13000, 0, -125)
Vertex_15 = geompy.MakeVertex(14000, 0, -105)
Vertex_16 = geompy.MakeVertex(15000, 0, -95)
Vertex_17 = geompy.MakeVertex(16000, 0, -65)
Vertex_18 = geompy.MakeVertex(17000, 0, -37)
Vertex_19 = geompy.MakeVertex(18000, 0, -70)
Vertex_20 = geompy.MakeVertex(19000, 0, -85)
Vertex_21 = geompy.MakeVertex(20000, 0, -85)
Vertex_22 = geompy.MakeVertex(21000, 0, -85)
Vertex_23 = geompy.MakeVertex(22000, 0, -91)
Vertex_24 = geompy.MakeVertex(23000, 0, -90)
Vertex_25 = geompy.MakeVertex(24000, 0, -90)
Vertex_26 = geompy.MakeVertex(25000, 0, -85)
Vertex_27 = geompy.MakeVertex(26000, 0, -80)
Vertex_28 = geompy.MakeVertex(27000, 0, 10)
Vertex_29 = geompy.MakeVertex(28000, 0, 140)
Vertex_30 = geompy.MakeVertex(28500, 0, 140)

## Construct half-pipe cross section
Radius = 0.36
curveThreshold = Radius*1.1
Divided_Disk_1 = geompy.MakeDividedDisk(Radius, 2, GEOM.SQUARE)
Rotation_1 = geompy.MakeRotation(Divided_Disk_1, OX, 45*math.pi/180.0)
Plane_1 = geompy.MakePlaneLCS(None, 1, 3)
Extrusion_1 = geompy.MakePrismVecH(Plane_1, OY, -1)
Cut_1 = geompy.MakeCutList(Rotation_1, [Extrusion_1], True)
```

```
## Pipe 1 Solid
Curve_1 = geompy.MakePolyline([Vertex_0, Vertex_1, Vertex_2, Vertex_3,
    ↪ Vertex_4, Vertex_5, Vertex_6], False)
Pipe_1 = geompy.MakePipe(Cut_1, Curve_1)
Inlet = geompy.CreateGroup(Pipe_1, geompy.ShapeType["FACE"])
Inlet_List = geompy.GetShapesOnPlaneIDs(Pipe_1, geompy.ShapeType["FACE"],
    ↪ OX, GEOM.ST_ON)
geompy.UnionIDs(Inlet, Inlet_List)
Symmetry = geompy.CreateGroup(Pipe_1, geompy.ShapeType["FACE"])
Symmetry_List = geompy.GetShapesOnPlaneIDs(Pipe_1, geompy.ShapeType["FACE"
    ↪ ], OY, GEOM.ST_ON)
geompy.UnionIDs(Symmetry, Symmetry_List)
P1_O = geompy.CreateGroup(Pipe_1, geompy.ShapeType["FACE"])
P1_O_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_1, geompy.ShapeType
    ↪ ["FACE"], OX, Vertex_6, GEOM.ST_ON)
geompy.UnionIDs(P1_O, P1_O_List)
Wall = geompy.CreateGroup(Pipe_1, geompy.ShapeType["FACE"])
Wall_List = wall_id(Pipe_1, curveThreshold)
geompy.UnionIDs(Wall, Wall_List)
FE_P1 = geompy.CreateGroup(Pipe_1, geompy.ShapeType["EDGE"])
Pipe_1_Edge = edge_id(Pipe_1)
geompy.UnionIDs(FE_P1, Pipe_1_Edge)

## Pipe 2 Solid
Curve_2 = geompy.MakePolyline([Vertex_6, Vertex_7], False)
Pipe_2 = geompy.MakePipe(P1_O, Curve_2)
P2_I = geompy.CreateGroup(Pipe_2, geompy.ShapeType["FACE"])
P2_I_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_2, geompy.ShapeType
    ↪ ["FACE"], OX, Vertex_6, GEOM.ST_ON)
geompy.UnionIDs(P2_I, P2_I_List)
Symmetry_1 = geompy.CreateGroup(Pipe_2, geompy.ShapeType["FACE"])
Symmetry_1_List = geompy.GetShapesOnPlaneIDs(Pipe_2, geompy.ShapeType["FACE
    ↪ "], OY, GEOM.ST_ON)
geompy.UnionIDs(Symmetry_1, Symmetry_1_List)
Wall_1 = geompy.CreateGroup(Pipe_2, geompy.ShapeType["FACE"])
Wall_1_List = wall_id(Pipe_2, curveThreshold)
geompy.UnionIDs(Wall_1, Wall_1_List)
P2_O = geompy.CreateGroup(Pipe_2, geompy.ShapeType["FACE"])
P2_O_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_2, geompy.ShapeType
    ↪ ["FACE"], OX, Vertex_7, GEOM.ST_ON)
geompy.UnionIDs(P2_O, P2_O_List)
FE_P2 = geompy.CreateGroup(Pipe_2, geompy.ShapeType["EDGE"])
Pipe_2_Edge = edge_id(Pipe_2)
geompy.UnionIDs(FE_P2, Pipe_2_Edge)
Water_Region_0 = geompy.CreateGroup(Pipe_2, geompy.ShapeType["SOLID"])
WR_0_List = geompy.SubShapeAllIDs(Pipe_2, geompy.ShapeType["SOLID"])
geompy.UnionIDs(Water_Region_0, WR_0_List)
```

```
## Pipe 3 Solid
Curve_3 = geompy.MakePolyline([Vertex_7, Vertex_8, Vertex_9, Vertex_10,
    ↪ Vertex_11], False)
Pipe_3 = geompy.MakePipe(P2_0, Curve_3)
P3_I = geompy.CreateGroup(Pipe_3, geompy.ShapeType["FACE"])
P3_I_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_3, geompy.ShapeType
    ↪ ["FACE"], OX, Vertex_7, GEOM.ST_ON)
geompy.UnionIDs(P3_I, P3_I_List)
Symmetry_2 = geompy.CreateGroup(Pipe_3, geompy.ShapeType["FACE"])
Symmetry_2_List = geompy.GetShapesOnPlaneIDs(Pipe_3, geompy.ShapeType["FACE
    ↪ "], OY, GEOM.ST_ON)
geompy.UnionIDs(Symmetry_2, Symmetry_2_List)
P3_0 = geompy.CreateGroup(Pipe_3, geompy.ShapeType["FACE"])
P3_0_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_3, geompy.ShapeType
    ↪ ["FACE"], OX, Vertex_11, GEOM.ST_ON)
geompy.UnionIDs(P3_0, P3_0_List)
Wall_2 = geompy.CreateGroup(Pipe_3, geompy.ShapeType["FACE"])
Wall_2_List = wall_id(Pipe_3, curveThreshold)
geompy.UnionIDs(Wall_2, Wall_2_List)
FE_P3 = geompy.CreateGroup(Pipe_3, geompy.ShapeType["EDGE"])
Pipe_3_Edge = edge_id(Pipe_3)
geompy.UnionIDs(FE_P3, Pipe_3_Edge)

## Pipe 4 Solid
Curve_4 = geompy.MakePolyline([Vertex_11, Vertex_12, Vertex_13, Vertex_14,
    ↪ Vertex_15], False)
Pipe_4 = geompy.MakePipe(P3_0, Curve_4)
P4_I = geompy.CreateGroup(Pipe_4, geompy.ShapeType["FACE"])
P4_I_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_4, geompy.ShapeType
    ↪ ["FACE"], OX, Vertex_11, GEOM.ST_ON)
geompy.UnionIDs(P4_I, P4_I_List)
Symmetry_3 = geompy.CreateGroup(Pipe_4, geompy.ShapeType["FACE"])
Symmetry_3_List = geompy.GetShapesOnPlaneIDs(Pipe_4, geompy.ShapeType["FACE
    ↪ "], OY, GEOM.ST_ON)
geompy.UnionIDs(Symmetry_3, Symmetry_3_List)
P4_0 = geompy.CreateGroup(Pipe_4, geompy.ShapeType["FACE"])
P4_0_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_4, geompy.ShapeType
    ↪ ["FACE"], OX, Vertex_15, GEOM.ST_ON)
geompy.UnionIDs(P4_0, P4_0_List)
Wall_3 = geompy.CreateGroup(Pipe_4, geompy.ShapeType["FACE"])
Wall_3_List = wall_id(Pipe_4, curveThreshold)
geompy.UnionIDs(Wall_3, Wall_3_List)
FE_P4 = geompy.CreateGroup(Pipe_4, geompy.ShapeType["EDGE"])
Pipe_4_Edge = edge_id(Pipe_4)
geompy.UnionIDs(FE_P4, Pipe_4_Edge)
Water_Region_1 = geompy.CreateGroup(Pipe_4, geompy.ShapeType["SOLID"])
```

```
WR_1_List = geompy.SubShapeAllIDs(Pipe_4, geompy.ShapeType["SOLID"])
geompy.UnionIDs(Water_Region_1, WR_1_List)

## Pipe 5 Solid
Curve_5 = geompy.MakePolyline([Vertex_15, Vertex_16, Vertex_17, Vertex_18,
    ↪ Vertex_19, Vertex_20, Vertex_21, Vertex_22], False)
Pipe_5 = geompy.MakePipe(P4_O, Curve_5)
P5_I = geompy.CreateGroup(Pipe_5, geompy.ShapeType["FACE"])
P5_I_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_5, geompy.ShapeType
    ↪ ["FACE"], OX, Vertex_15, GEOM.ST_ON)
geompy.UnionIDs(P5_I, P5_I_List)
Symmetry_4 = geompy.CreateGroup(Pipe_5, geompy.ShapeType["FACE"])
Symmetry_4_List = geompy.GetShapesOnPlaneIDs(Pipe_5, geompy.ShapeType["FACE
    ↪ "], OY, GEOM.ST_ON)
geompy.UnionIDs(Symmetry_4, Symmetry_4_List)
P5_O = geompy.CreateGroup(Pipe_5, geompy.ShapeType["FACE"])
P5_O_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_5, geompy.ShapeType
    ↪ ["FACE"], OX, Vertex_22, GEOM.ST_ON)
geompy.UnionIDs(P5_O, P5_O_List)
Wall_4 = geompy.CreateGroup(Pipe_5, geompy.ShapeType["FACE"])
Wall_4_List = wall_id(Pipe_5, curveThreshold)
geompy.UnionIDs(Wall_4, Wall_4_List)
FE_P5 = geompy.CreateGroup(Pipe_5, geompy.ShapeType["EDGE"])
Pipe_5_Edge = edge_id(Pipe_5)
geompy.UnionIDs(FE_P5, Pipe_5_Edge)

## Pipe 6 Solid
Curve_6 = geompy.MakePolyline([Vertex_22, Vertex_23, Vertex_24, Vertex_25,
    ↪ Vertex_26], False)
Pipe_6 = geompy.MakePipe(P5_O, Curve_6)
P6_I = geompy.CreateGroup(Pipe_6, geompy.ShapeType["FACE"])
P6_I_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_6, geompy.ShapeType
    ↪ ["FACE"], OX, Vertex_22, GEOM.ST_ON)
geompy.UnionIDs(P6_I, P6_I_List)
Symmetry_5 = geompy.CreateGroup(Pipe_6, geompy.ShapeType["FACE"])
Symmetry_5_List = geompy.GetShapesOnPlaneIDs(Pipe_6, geompy.ShapeType["FACE
    ↪ "], OY, GEOM.ST_ON)
geompy.UnionIDs(Symmetry_5, Symmetry_5_List)
P6_O = geompy.CreateGroup(Pipe_6, geompy.ShapeType["FACE"])
P6_O_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_6, geompy.ShapeType
    ↪ ["FACE"], OX, Vertex_26, GEOM.ST_ON)
geompy.UnionIDs(P6_O, P6_O_List)
Wall_5 = geompy.CreateGroup(Pipe_6, geompy.ShapeType["FACE"])
Wall_5_List = wall_id(Pipe_6, curveThreshold)
geompy.UnionIDs(Wall_5, Wall_5_List)
FE_P6 = geompy.CreateGroup(Pipe_6, geompy.ShapeType["EDGE"])
Pipe_6_Edge = edge_id(Pipe_6)
```

```
geompy.UnionIDs(FE_P6, Pipe_6_Edge)
Water_Region_2 = geompy.CreateGroup(Pipe_6, geompy.ShapeType["SOLID"])
WR_2_List = geompy.SubShapeAllIDs(Pipe_6, geompy.ShapeType["SOLID"])
geompy.UnionIDs(Water_Region_2, WR_2_List)

## Pipe 7 Solid
Curve_7 = geompy.MakePolyline([Vertex_26, Vertex_27, Vertex_28, Vertex_29,
    ↪ Vertex_30], False)
Pipe_7 = geompy.MakePipe(P6_0, Curve_7)
P7_I = geompy.CreateGroup(Pipe_7, geompy.ShapeType["FACE"])
P7_I_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_7, geompy.ShapeType
    ↪ ["FACE"], OX, Vertex_26, GEOM.ST_ON)
geompy.UnionIDs(P7_I, P7_I_List)
Symmetry_6 = geompy.CreateGroup(Pipe_7, geompy.ShapeType["FACE"])
Symmetry_6_List = geompy.GetShapesOnPlaneIDs(Pipe_7, geompy.ShapeType["FACE
    ↪ "], OY, GEOM.ST_ON)
geompy.UnionIDs(Symmetry_6, Symmetry_6_List)
Outlet = geompy.CreateGroup(Pipe_7, geompy.ShapeType["FACE"])
Outlet_List = geompy.GetShapesOnPlaneWithLocationIDs(Pipe_7, geompy.
    ↪ ShapeType["FACE"], OX, Vertex_30, GEOM.ST_ON)
geompy.UnionIDs(Outlet, Outlet_List)
Wall_6 = geompy.CreateGroup(Pipe_7, geompy.ShapeType["FACE"])
Wall_6_List = wall_id(Pipe_7, curveThreshold)
geompy.UnionIDs(Wall_6, Wall_6_List)
FE_P7 = geompy.CreateGroup(Pipe_7, geompy.ShapeType["EDGE"])
Pipe_7_Edge = edge_id(Pipe_7)
geompy.UnionIDs(FE_P7, Pipe_7_Edge)

### SMESH component

import SMESH, SALOMEDS
from salome.smesh import smeshBuilder

smesh = smeshBuilder.New(theStudy)
LongSegments = 6800  ## Longitudinal segments

## Mesh 1
Mesh_1 = smesh.Mesh(Pipe_1)
Regular_1D = Mesh_1.Segment()
Nb_Segments_1 = Regular_1D.NumberOfSegments(5)
Nb_Segments_1.SetDistrType( 0 )
Quadrangle_2D = Mesh_1.Quadrangle(algo=smeshBuilder.QUADRANGLE)
Prism_3D = Mesh_1.Prism()
status = Mesh_1.AddHypothesis(Quadrangle_2D,Inlet)
Projection_2D = Mesh_1.Projection2D(geom=P1_0)
Source_Face_1 = Projection_2D.SourceFace(Inlet,None,None,None,None,None)
Nb_Segments_P1 = smesh.CreateHypothesis('NumberOfSegments')
```

```
Nb_Segments_P1.SetNumberOfSegments( LongSegments )
Nb_Segments_P1.SetDistrType( 0 )
status = Mesh_1.AddHypothesis(Regular_1D,FE_P1)
status = Mesh_1.AddHypothesis(Nb_Segments_P1,FE_P1)
isDone = Mesh_1.Compute()
[ SubMesh_1, SubMesh_2, SubMesh_3 ] = Mesh_1.GetMesh().GetSubMeshes()
Inlet_1 = Mesh_1.GroupOnGeom(Inlet,'Inlet',SMESH.FACE)
Symmetry_7 = Mesh_1.GroupOnGeom(Symmetry,'Symmetry',SMESH.FACE)
Wall_7 = Mesh_1.GroupOnGeom(Wall,'Wall',SMESH.FACE)

## Mesh 2
Mesh_2 = smesh.Mesh(Pipe_2)
status = Mesh_2.AddHypothesis(Nb_Segments_1)
status = Mesh_2.AddHypothesis(Regular_1D)
status = Mesh_2.AddHypothesis(Quadrangle_2D)
status = Mesh_2.AddHypothesis(Prism_3D)
status = Mesh_2.AddHypothesis(Quadrangle_2D,P2_I)
Source_Face_2 = smesh.CreateHypothesis('ProjectionSource2D')
Source_Face_2.SetSourceFace( P2_I )
Source_Face_2.SetSourceMesh( None )
Source_Face_2.SetVertexAssociation( None, None, None, None )
status = Mesh_2.AddHypothesis(Projection_2D,P2_O)
status = Mesh_2.AddHypothesis(Source_Face_2,P2_O)
Nb_Segments_P2 = smesh.CreateHypothesis('NumberOfSegments')
Nb_Segments_P2.SetNumberOfSegments( LongSegments )
Nb_Segments_P2.SetDistrType( 0 )
status = Mesh_2.AddHypothesis(Regular_1D,FE_P2)
status = Mesh_2.AddHypothesis(Nb_Segments_P2,FE_P2)
isDone = Mesh_2.Compute()
[ SubMesh_4, SubMesh_5, SubMesh_6 ] = Mesh_2.GetMesh().GetSubMeshes()
Symmetry_8 = Mesh_2.GroupOnGeom(Symmetry_1,'Symmetry',SMESH.FACE)
Wall_8 = Mesh_2.GroupOnGeom(Wall_1,'Wall',SMESH.FACE)
Water_Region_3 = Mesh_2.GroupOnGeom(Water_Region_0,'Water_Region',SMESH.
    ↪ VOLUME)

## Mesh 3
Mesh_3 = smesh.Mesh(Pipe_3)
status = Mesh_3.AddHypothesis(Nb_Segments_1)
status = Mesh_3.AddHypothesis(Regular_1D)
status = Mesh_3.AddHypothesis(Quadrangle_2D)
status = Mesh_3.AddHypothesis(Prism_3D)
status = Mesh_3.AddHypothesis(Quadrangle_2D,P3_I)
Source_Face_3 = smesh.CreateHypothesis('ProjectionSource2D')
Source_Face_3.SetSourceFace( P3_I )
Source_Face_3.SetSourceMesh( None )
Source_Face_3.SetVertexAssociation( None, None, None, None )
status = Mesh_3.AddHypothesis(Projection_2D,P3_O)
```

```
status = Mesh_3.AddHypothesis(Source_Face_3,P3_0)
Nb_Segments_P3 = smesh.CreateHypothesis('NumberOfSegments')
Nb_Segments_P3.SetNumberOfSegments( LongSegments )
Nb_Segments_P3.SetDistrType( 0 )
status = Mesh_3.AddHypothesis(Regular_1D,FE_P3)
status = Mesh_3.AddHypothesis(Nb_Segments_P3,FE_P3)
isDone = Mesh_3.Compute()
[ SubMesh_7, SubMesh_8, SubMesh_9 ] = Mesh_3.GetMesh().GetSubMeshes()
Symmetry_9 = Mesh_3.GroupOnGeom(Symmetry_2,'Symmetry',SMESH.FACE)
Wall_9 = Mesh_3.GroupOnGeom(Wall_2,'Wall',SMESH.FACE)

## Mesh 4
Mesh_4 = smesh.Mesh(Pipe_4)
status = Mesh_4.AddHypothesis(Nb_Segments_1)
status = Mesh_4.AddHypothesis(Regular_1D)
status = Mesh_4.AddHypothesis(Quadrangle_2D)
status = Mesh_4.AddHypothesis(Prism_3D)
status = Mesh_4.AddHypothesis(Quadrangle_2D,P4_I)
Source_Face_4 = smesh.CreateHypothesis('ProjectionSource2D')
Source_Face_4.SetSourceFace( P4_I )
Source_Face_4.SetSourceMesh( None )
Source_Face_4.SetVertexAssociation( None, None, None, None )
status = Mesh_4.AddHypothesis(Projection_2D,P4_0)
status = Mesh_4.AddHypothesis(Source_Face_4,P4_0)
Nb_Segments_P4 = smesh.CreateHypothesis('NumberOfSegments')
Nb_Segments_P4.SetNumberOfSegments( LongSegments )
Nb_Segments_P4.SetDistrType( 0 )
status = Mesh_4.AddHypothesis(Regular_1D,FE_P4)
status = Mesh_4.AddHypothesis(Nb_Segments_P4,FE_P4)
isDone = Mesh_4.Compute()
[ SubMesh_10, SubMesh_11, SubMesh_12 ] = Mesh_4.GetMesh().GetSubMeshes()
Symmetry_10 = Mesh_4.GroupOnGeom(Symmetry_3,'Symmetry',SMESH.FACE)
Wall_10 = Mesh_4.GroupOnGeom(Wall_3,'Wall',SMESH.FACE)
Water_Region_4 = Mesh_4.GroupOnGeom(Water_Region_1,'Water_Region',SMESH.
    ↪ VOLUME)

## Mesh 5
Mesh_5 = smesh.Mesh(Pipe_5)
status = Mesh_5.AddHypothesis(Nb_Segments_1)
status = Mesh_5.AddHypothesis(Regular_1D)
status = Mesh_5.AddHypothesis(Quadrangle_2D)
status = Mesh_5.AddHypothesis(Prism_3D)
status = Mesh_5.AddHypothesis(Quadrangle_2D,P5_I)
Source_Face_5 = smesh.CreateHypothesis('ProjectionSource2D')
Source_Face_5.SetSourceFace( P5_I )
Source_Face_5.SetSourceMesh( None )
Source_Face_5.SetVertexAssociation( None, None, None, None )
```

```
status = Mesh_5.AddHypothesis(Projection_2D,P5_0)
status = Mesh_5.AddHypothesis(Source_Face_5,P5_0)
Nb_Segments_P5 = smesh.CreateHypothesis('NumberOfSegments')
Nb_Segments_P5.SetNumberOfSegments( LongSegments )
Nb_Segments_P5.SetDistrType( 0 )
status = Mesh_5.AddHypothesis(Regular_1D,FE_P5)
status = Mesh_5.AddHypothesis(Nb_Segments_P5,FE_P5)
isDone = Mesh_5.Compute()
[ SubMesh_13, SubMesh_14, SubMesh_15 ] = Mesh_5.GetMesh().GetSubMeshes()
Symmetry_11 = Mesh_5.GroupOnGeom(Symmetry_4,'Symmetry',SMESH.FACE)
Wall_11 = Mesh_5.GroupOnGeom(Wall_4,'Wall',SMESH.FACE)

## Mesh 6
Mesh_6 = smesh.Mesh(Pipe_6)
status = Mesh_6.AddHypothesis(Nb_Segments_1)
status = Mesh_6.AddHypothesis(Regular_1D)
status = Mesh_6.AddHypothesis(Quadrangle_2D)
status = Mesh_6.AddHypothesis(Prism_3D)
status = Mesh_6.AddHypothesis(Quadrangle_2D,P6_I)
Source_Face_6 = smesh.CreateHypothesis('ProjectionSource2D')
Source_Face_6.SetSourceFace( P6_I )
Source_Face_6.SetSourceMesh( None )
Source_Face_6.SetVertexAssociation( None, None, None, None )
status = Mesh_6.AddHypothesis(Projection_2D,P6_0)
status = Mesh_6.AddHypothesis(Source_Face_6,P6_0)
Nb_Segments_P6 = smesh.CreateHypothesis('NumberOfSegments')
Nb_Segments_P6.SetNumberOfSegments( LongSegments )
Nb_Segments_P6.SetDistrType( 0 )
status = Mesh_6.AddHypothesis(Regular_1D,FE_P6)
status = Mesh_6.AddHypothesis(Nb_Segments_P6,FE_P6)
isDone = Mesh_6.Compute()
[ SubMesh_16, SubMesh_17, SubMesh_18 ] = Mesh_6.GetMesh().GetSubMeshes()
Symmetry_12 = Mesh_6.GroupOnGeom(Symmetry_5,'Symmetry',SMESH.FACE)
Wall_12 = Mesh_6.GroupOnGeom(Wall_5,'Wall',SMESH.FACE)
Water_Region_5 = Mesh_6.GroupOnGeom(Water_Region_2,'Water_Region',SMESH.
    ↪ VOLUME)

## Mesh 7
Mesh_7 = smesh.Mesh(Pipe_7)
status = Mesh_7.AddHypothesis(Nb_Segments_1)
status = Mesh_7.AddHypothesis(Regular_1D)
status = Mesh_7.AddHypothesis(Quadrangle_2D)
status = Mesh_7.AddHypothesis(Prism_3D)
status = Mesh_7.AddHypothesis(Quadrangle_2D,P7_I)
Source_Face_7 = smesh.CreateHypothesis('ProjectionSource2D')
Source_Face_7.SetSourceFace( P7_I )
Source_Face_7.SetSourceMesh( None )
```

```
Source_Face_7.SetVertexAssociation( None, None, None, None )
status = Mesh_7.AddHypothesis(Projection_2D,Outlet)
status = Mesh_7.AddHypothesis(Source_Face_7,Outlet)
Nb_Segments_P7 = smesh.CreateHypothesis('NumberOfSegments')
Nb_Segments_P7.SetNumberOfSegments( LongSegments )
Nb_Segments_P7.SetDistrType( 0 )
status = Mesh_7.AddHypothesis(Regular_1D,FE_P7)
status = Mesh_7.AddHypothesis(Nb_Segments_P7,FE_P7)
isDone = Mesh_7.Compute()
[ SubMesh_19, SubMesh_20, SubMesh_21 ] = Mesh_7.GetMesh().GetSubMeshes()
Symmetry_13 = Mesh_7.GroupOnGeom(Symmetry_6,'Symmetry',SMESH.FACE)
Outlet_1 = Mesh_7.GroupOnGeom(Outlet,'Outlet',SMESH.FACE)
Wall_13 = Mesh_7.GroupOnGeom(Wall_6,'Wall',SMESH.FACE)

## Compound Mesh
Compound_Mesh_1 = smesh.Concatenate([Mesh_1.GetMesh(), Mesh_2.GetMesh(),
    ↪ Mesh_3.GetMesh(), Mesh_4.GetMesh(), Mesh_5.GetMesh(), Mesh_6.GetMesh
    ↪ (), Mesh_7.GetMesh()], 1, 1, 1e-005)
[ Inlet_2, Symmetry_14, Wall_14, Water_Region_6, Outlet_2 ] =
    ↪ Compound_Mesh_1.GetGroups()
SubMesh_1 = Mesh_1.GetSubMesh( Inlet, 'SubMesh_1' )
SubMesh_2 = Projection_2D.GetSubMesh()
SubMesh_3 = Mesh_1.GetSubMesh( FE_P1, 'SubMesh_3' )
SubMesh_4 = Mesh_2.GetSubMesh( P2_I, 'SubMesh_4' )
SubMesh_5 = Mesh_2.GetSubMesh( P2_O, 'SubMesh_5' )
SubMesh_6 = Mesh_2.GetSubMesh( FE_P2, 'SubMesh_6' )
SubMesh_7 = Mesh_3.GetSubMesh( P3_I, 'SubMesh_7' )
SubMesh_8 = Mesh_3.GetSubMesh( P3_O, 'SubMesh_8' )
SubMesh_9 = Mesh_3.GetSubMesh( FE_P3, 'SubMesh_9' )
SubMesh_10 = Mesh_4.GetSubMesh( P4_I, 'SubMesh_10' )
SubMesh_11 = Mesh_4.GetSubMesh( P4_O, 'SubMesh_11' )
SubMesh_12 = Mesh_4.GetSubMesh( FE_P4, 'SubMesh_12' )
SubMesh_13 = Mesh_5.GetSubMesh( P5_I, 'SubMesh_13' )
SubMesh_14 = Mesh_5.GetSubMesh( P5_O, 'SubMesh_14' )
SubMesh_15 = Mesh_5.GetSubMesh( FE_P5, 'SubMesh_15' )
SubMesh_16 = Mesh_6.GetSubMesh( P6_I, 'SubMesh_16' )
SubMesh_17 = Mesh_6.GetSubMesh( P6_O, 'SubMesh_17' )
SubMesh_18 = Mesh_6.GetSubMesh( FE_P6, 'SubMesh_18' )
SubMesh_19 = Mesh_7.GetSubMesh( P7_I, 'SubMesh_19' )
SubMesh_20 = Mesh_7.GetSubMesh( Outlet, 'SubMesh_20' )
SubMesh_21 = Mesh_7.GetSubMesh( FE_P7, 'SubMesh_21' )
smesh.SetName(Compound_Mesh_1.GetMesh(), 'Compound_Mesh_1')

if salome.sg.hasDesktop():
 salome.sg.updateObjBrowser(1)
```

Appendix C

Modified CFD Model Validation

C.1 Additional Slug Height Results

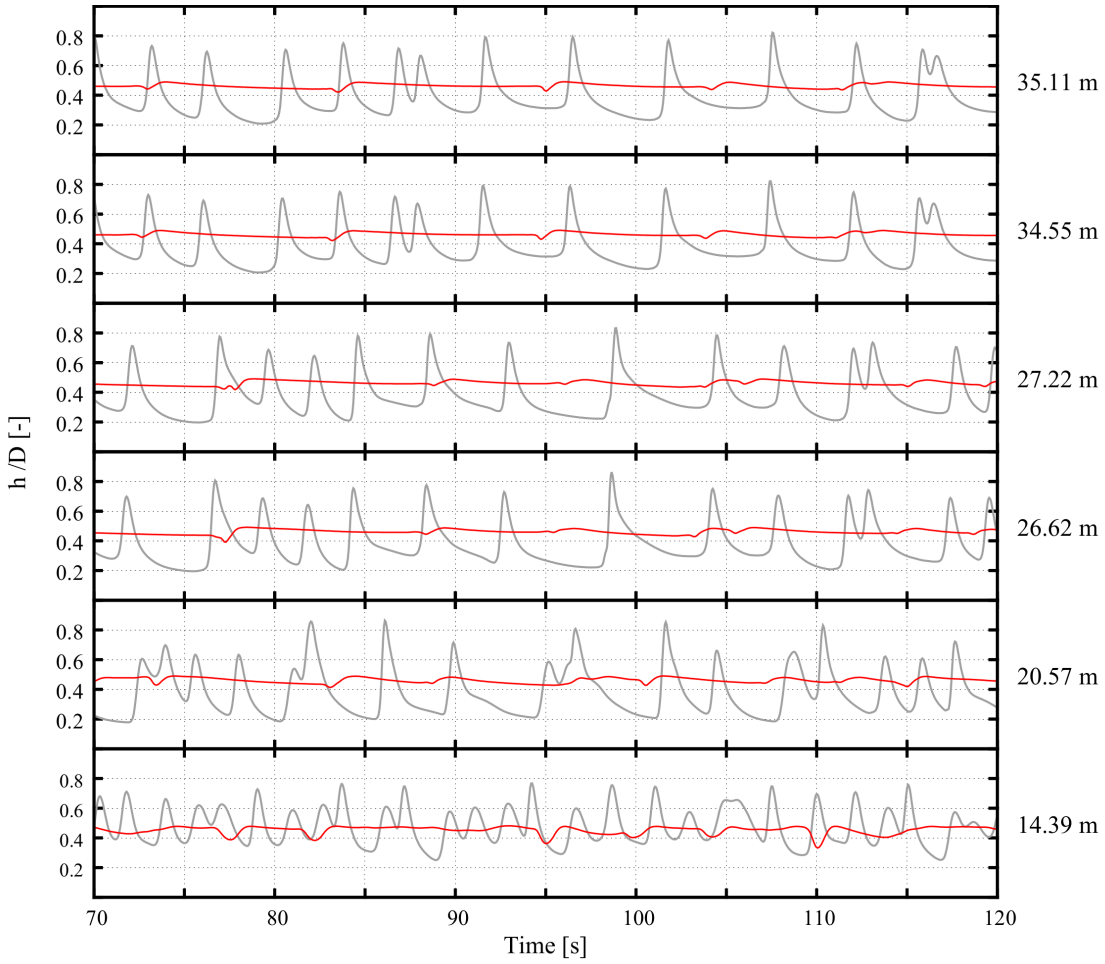


Figure C.1: The simulated slug flow height at selected sampling locations using a mesh resolution parameter of $\chi = 6.25$ (red) compared the maximum mesh resolution parameter considered $\chi = 100$ (grey) over a 50 second time period.

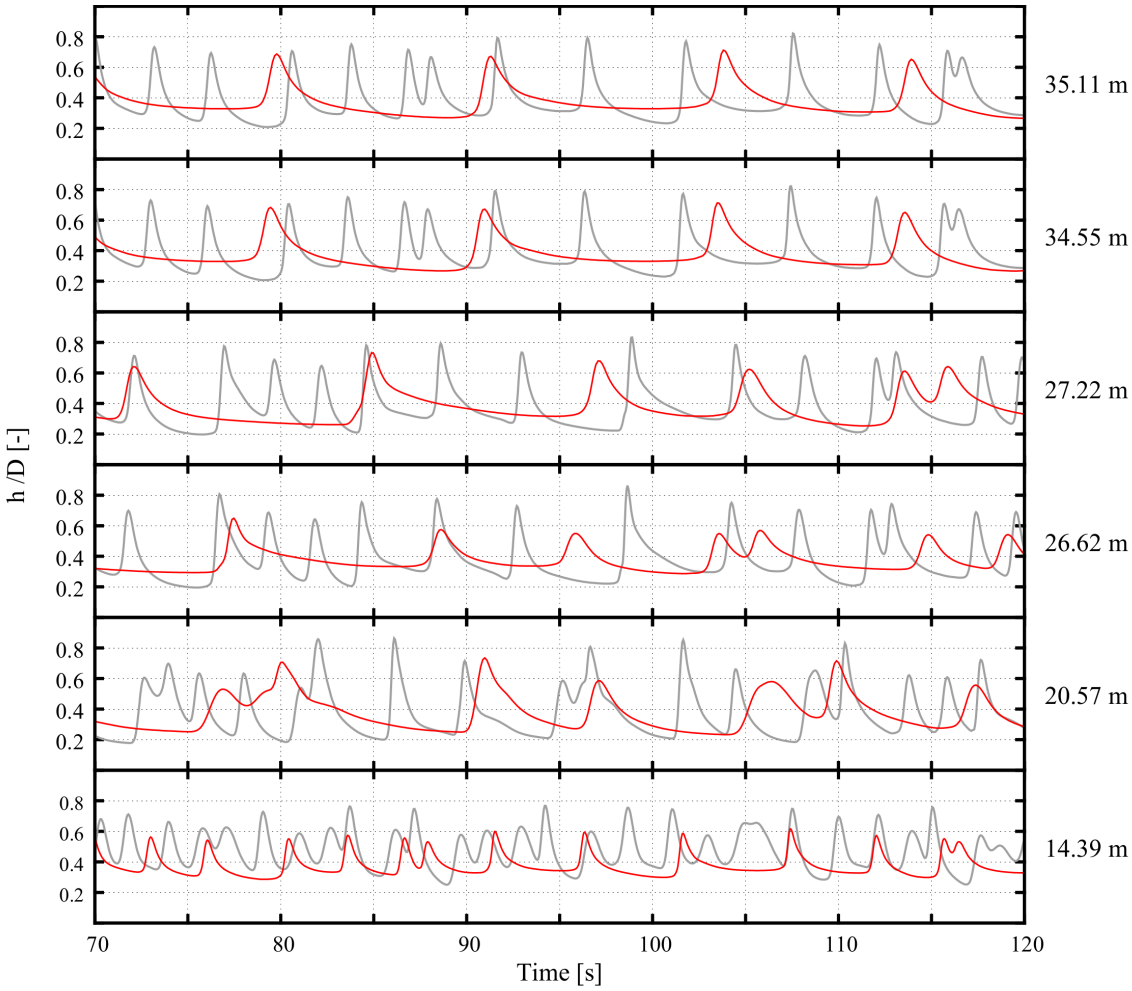


Figure C.2: The simulated slug flow height at selected sampling locations using a mesh resolution parameter of $\chi = 12.5$ (red) compared the maximum mesh resolution parameter considered $\chi = 100$ (grey) over a 50 second time period.

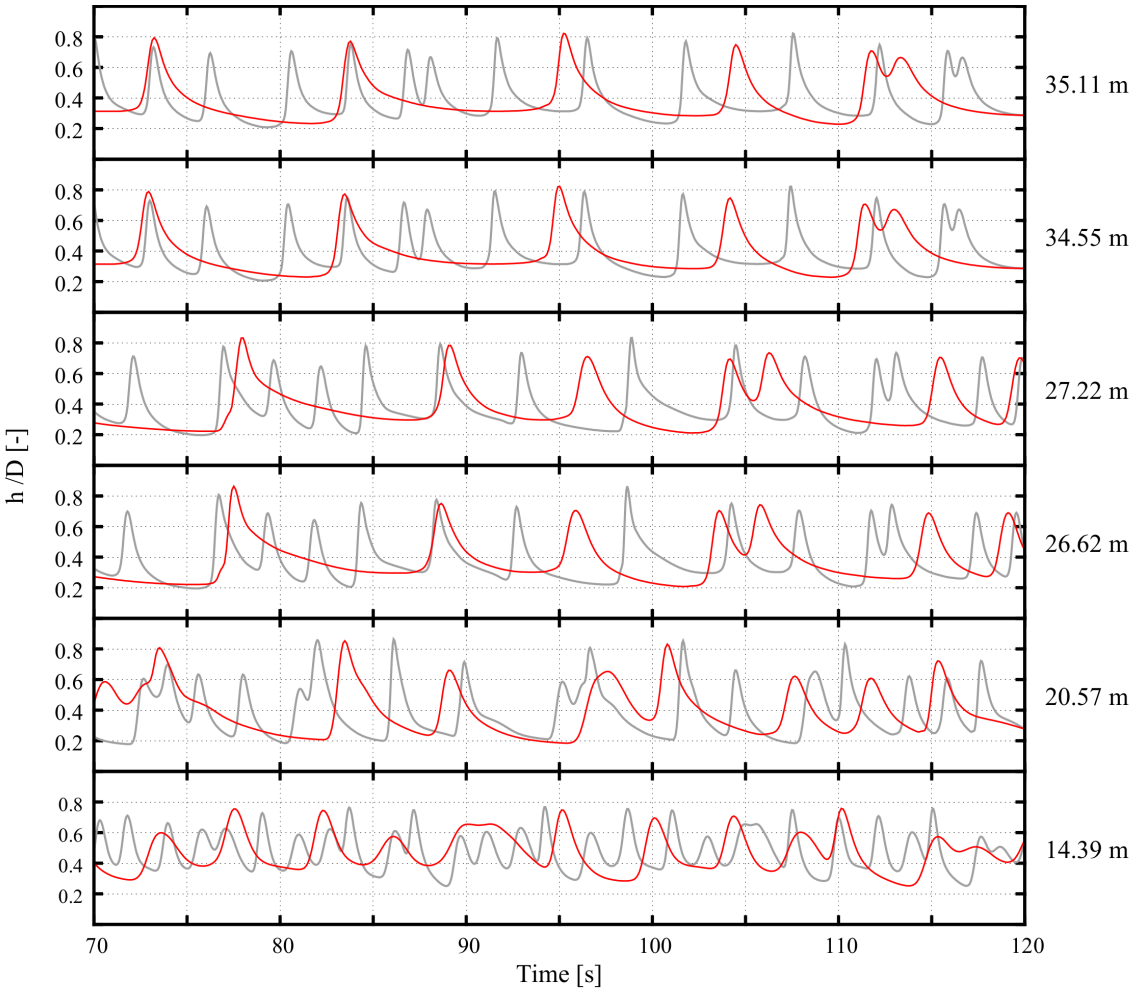


Figure C.3: The simulated slug flow height at selected sampling locations using a mesh resolution parameter of $\chi = 25$ (red) compared the maximum mesh resolution parameter considered $\chi = 100$ (grey) over a 50 second time period.

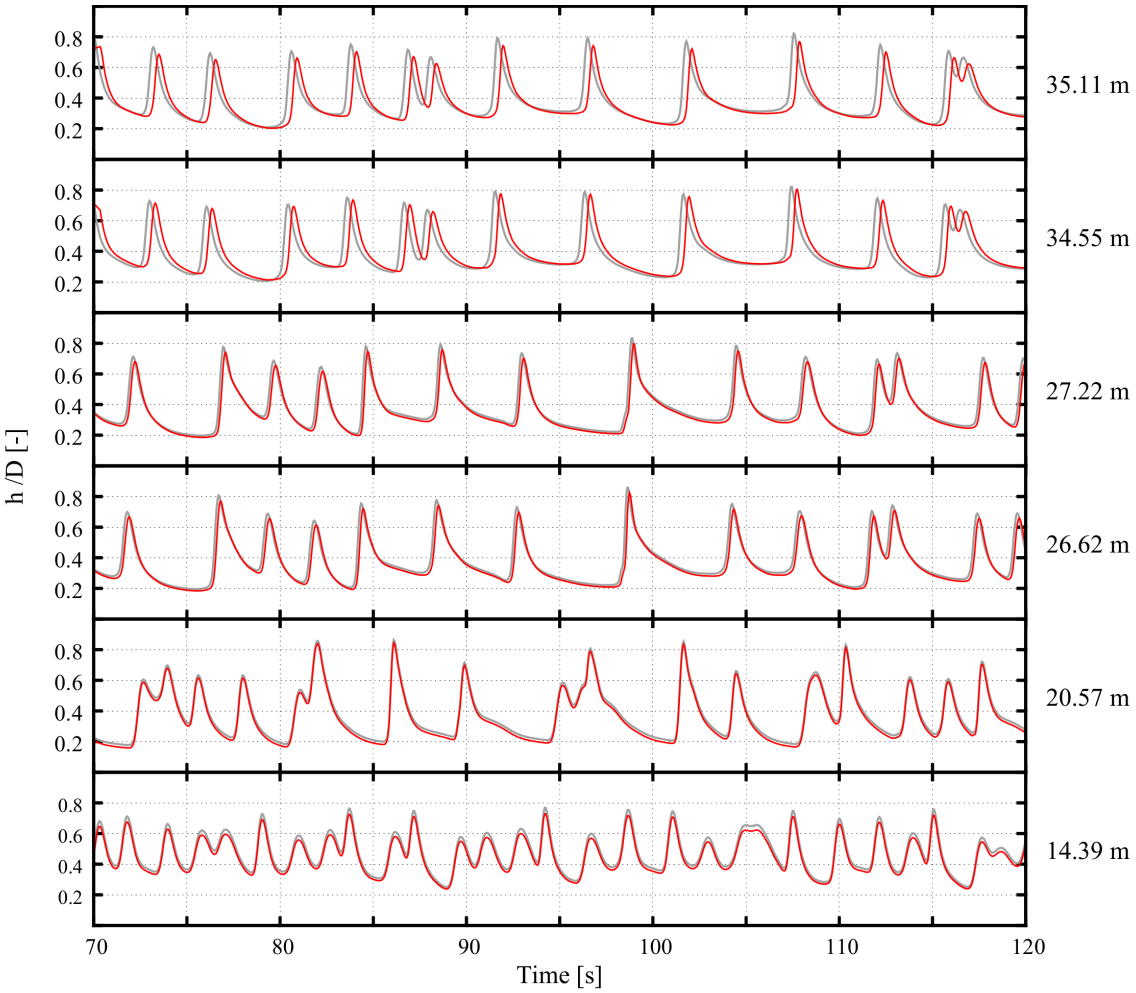


Figure C.4: The simulated slug flow height at selected sampling locations using a mesh resolution parameter of $\chi = 50$ (red) compared the maximum mesh resolution parameter considered $\chi = 100$ (grey) over a 50 second time period.

C.2 Lo and Fiala [2010] Slug Height Results

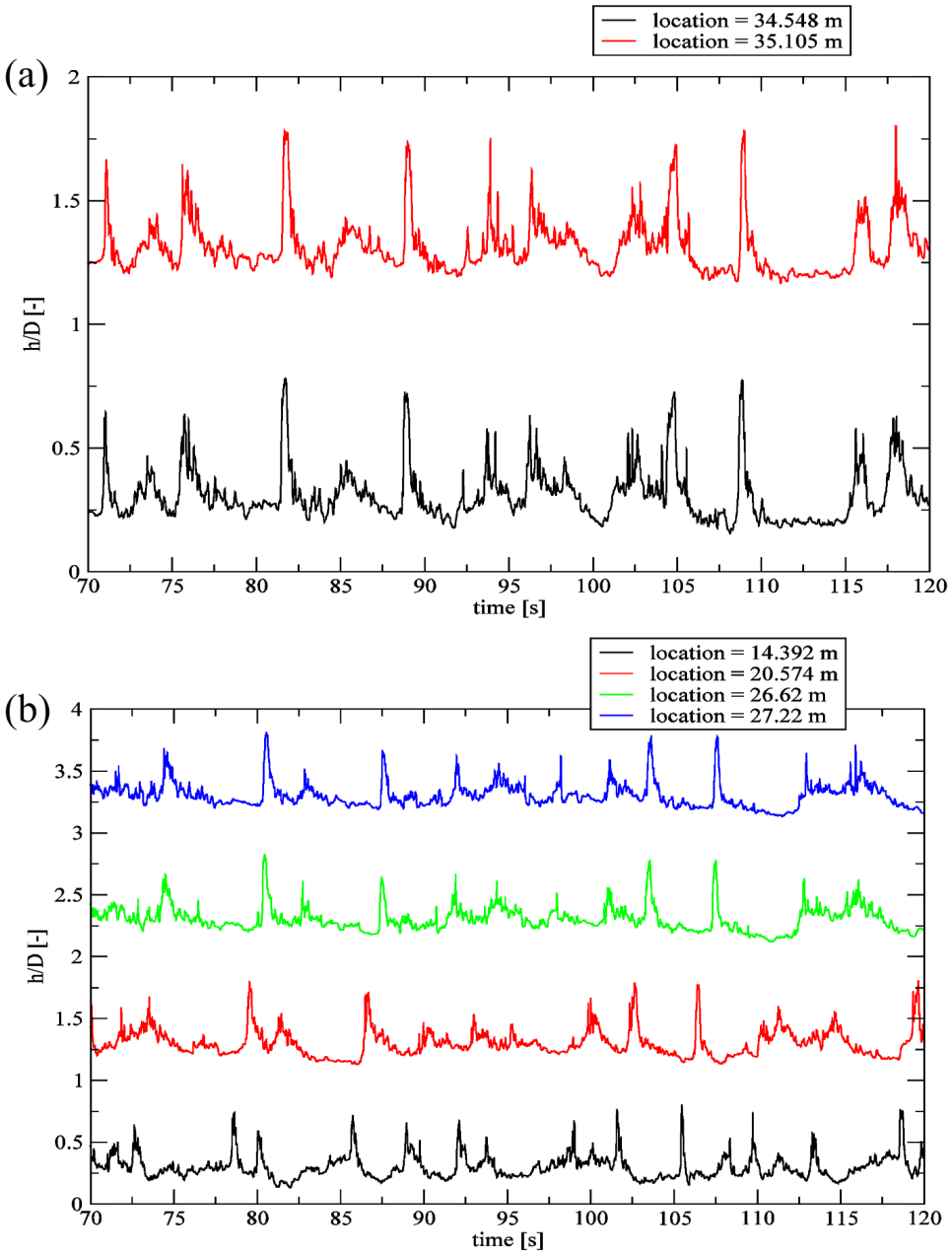


Figure C.5: The simulated slug flow height results from Lo and Fiala [2010] at selected sampling locations; a) 35.105 m and 34.548 and b) 27.22m, 26.62m, 20.574 m and 14.392 m (image adapted from Lo and Fiala [2010]).

Appendix D

Optimisation Study of a Low-Spot Pipe Bend

D.1 Modified interFoam Source Code

```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | Copyright (C) 2011-2014 OpenFOAM Foundation
     \\/     M anipulation  |
-------------------------------------------------------------------------------
License
    This file is part of OpenFOAM.

    OpenFOAM is free software: you can redistribute it and/or modify it
    under the terms of the GNU General Public License as published by the
    Free Software Foundation, either version 3 of the License, or (at
    your option) any later version.

    OpenFOAM is distributed in the hope that it will be useful, but WITHOUT
    ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
    FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
    for more details.

    You should have received a copy of the GNU General Public License
    along with OpenFOAM. If not, see <http://www.gnu.org/licenses/>.

Application
    interFoam

Description
    Solver for 2 incompressible, isothermal immiscible fluids using a VOF
```

```
    (volume of fluid) phase-fraction based interface capturing approach.

    The momentum and other fluid properties are of the "mixture" and a
    single momentum equation is solved.

    Turbulence modelling is generic, i.e. laminar, RAS or LES may be
    selected.

    For a two-fluid approach see twoPhaseEulerFoam.

    This solver has been modified to output the interfacial area between
    the two phases based upon an approach used by Thiele [2010].

\*---------------------------------------------------------------------------*/

#include "fvCFD.H"
#include "CMULES.H"
#include "subCycle.H"
#include "immiscibleIncompressibleTwoPhaseMixture.H"
#include "turbulenceModel.H"
#include "pimpleControl.H"
#include "fvIOoptionList.H"
#include "fixedFluxPressureFvPatchScalarField.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

int main(int argc, char *argv[])
{
    #include "setRootCase.H"
    #include "createTime.H"
    #include "createMesh.H"

    pimpleControl pimple(mesh);

    #include "initContinuityErrs.H"
    #include "createFields.H"
    #include "readTimeControls.H"
    #include "createPrghCorrTypes.H"
    #include "correctPhi.H"
    #include "CourantNo.H"
    #include "setInitialDeltaT.H"

    // * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

    Info<< "\nStarting time loop\n" << endl;

    while (runTime.run())
```

```
{
    #include "readTimeControls.H"
    #include "CourantNo.H"
    #include "alphaCourantNo.H"
    #include "setDeltaT.H"

    runTime++;

    Info<< "Time = " << runTime.timeName() << nl << endl;

    // --- Pressure-velocity PIMPLE corrector loop
    while (pimple.loop())
    {
        #include "alphaControls.H"
        #include "alphaEqnSubCycle.H"

        mixture.correct();

        #include "UEqn.H"
        // --- Pressure corrector loop
        while (pimple.correct())
        {
            #include "pEqn.H"
        }
        if (pimple.turbCorr())
        {
            turbulence->correct();
        }
    }

// Calculating the interfacial area for each cell as per Thiele [2010]

const volScalarField& cellVolume = alpha1 .db().lookupObject<
    ↪ volScalarField>("cellVolu");
volScalarField limitedAlpha1 = min(max(alpha1 , scalar(0)), scalar(1));

const dimensionedScalar
  areaFactor("areaFactor",dimensionSet(0,2,0,0,0,0,0), 0.0);

volScalarField interfaceArea = alpha1 * areaFactor;
volScalarField maxArea = alpha1 * areaFactor;

maxArea = 1.732*pow(cellVolume,(2.0/3.0));
volScalarField IA = (neg(limitedAlpha1-0.5)*maxArea*2.0*limitedAlpha1)
    ↪ + (pos(limitedAlpha1-0.5)*maxArea*(-2.0*( limitedAlpha1 - 1.0)));

// Summation of the interfacial area across the domain
```

```
    scalar IAsum = gSum(IA);

    // Writing at each timestep

    Info<< " Sum of Interfacial Area = " << IAsum << " m^2 "<< endl;

        runTime.write();

        Info<< "ExecutionTime = " << runTime.elapsedCpuTime() << " s"
            << "  ClockTime = " << runTime.elapsedClockTime() << " s"
            << nl << endl;
    }
    Info<< "End\n" << endl;
    return 0;
}

// ************************************************************************
```

D.2 Coupling of OpenFOAM® and SALOME with DAKOTA

A graphical depiction of a generic work flow used for setting up the DAKOTA optimisation study with OpenFOAM® and SALOME is shown in Figure D.1. Here, the Python script file of the mesh created in SALOME is converted into a template file for usage with DAKOTA for the purpose of enabling different heights of water in the pipe, depending on the sampling point generated by the DACE study. The liquid phase is assigned to the mesh using the setFields utility in OpenFOAM®. This template file is referenced to the DAKOTA input file that defines the simulation type, design variables and objective functions. An initial mesh is created in the OpenFOAM® boundary folder along with all simulation variables such as the solver, turbulence model, time step and solution runtime. An additional template file used for the inlet velocity is created here based on the "U" file in the "0" file directory in order to facilitate the required range of inlet velocities. The optimisation study is run using the simulator script file in DAKOTA which is used to call upon the SALOME Text User Interface (TUI) in order to generate the mesh and other simulation variables used in OpenFOAM®. Upon completion of the optimisation study, results are listed in the table out.dat file and associated log files.

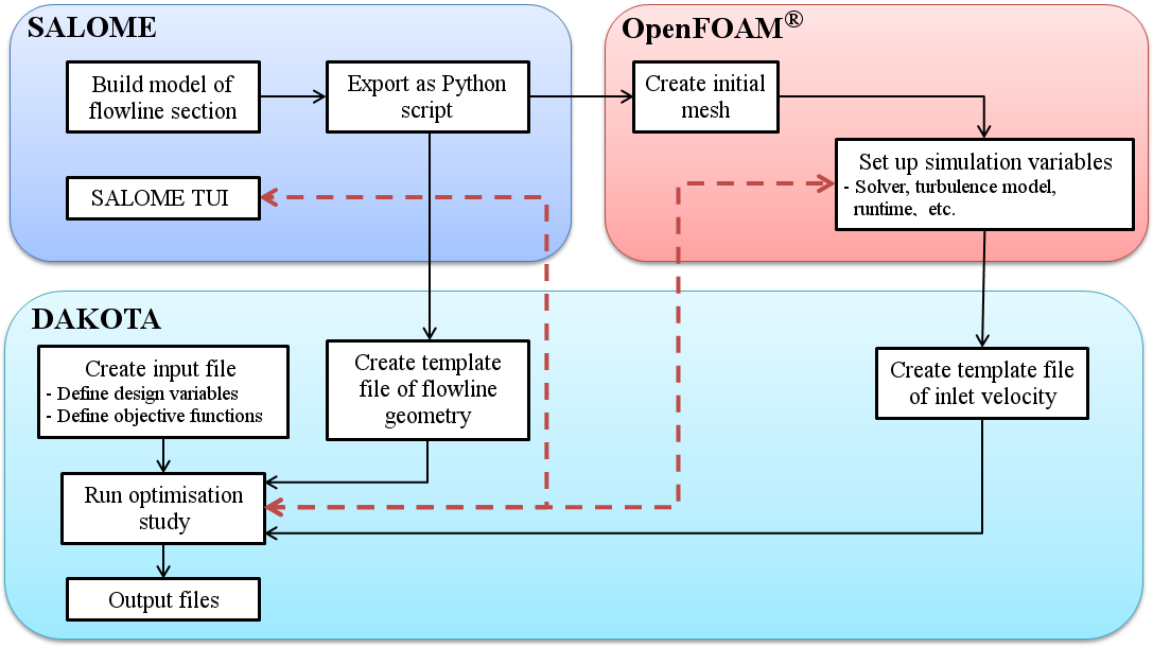


Figure D.1: Coupling of OpenFOAM® and SALOME with DAKOTA.

D.3 MST Results

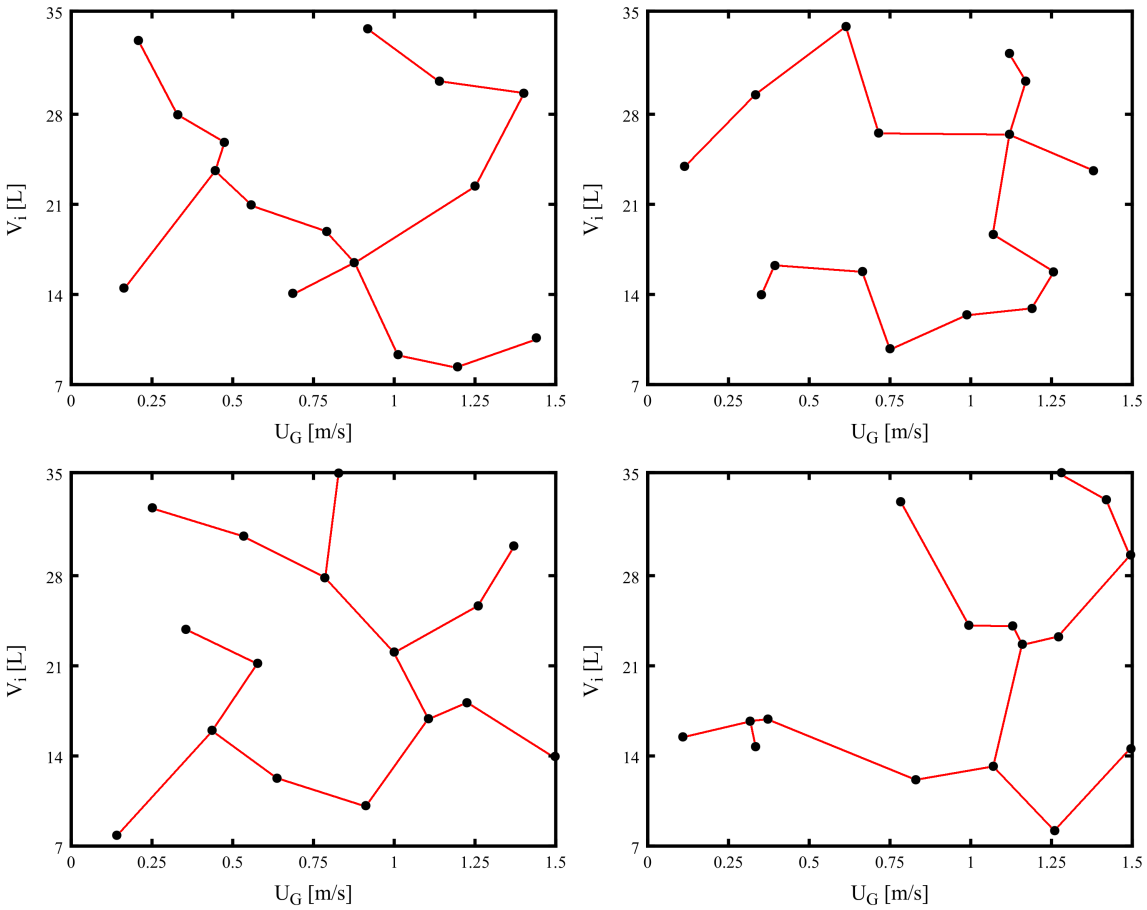


Figure D.2: The MST generated from the build points of the DACE study using a sample size of 16; (a) Latin hypercube sampling, (b) Orthogonal array sampling, (c) Orthogonal array based Latin hypercube sampling and (d) Monte Carlo random sampling.

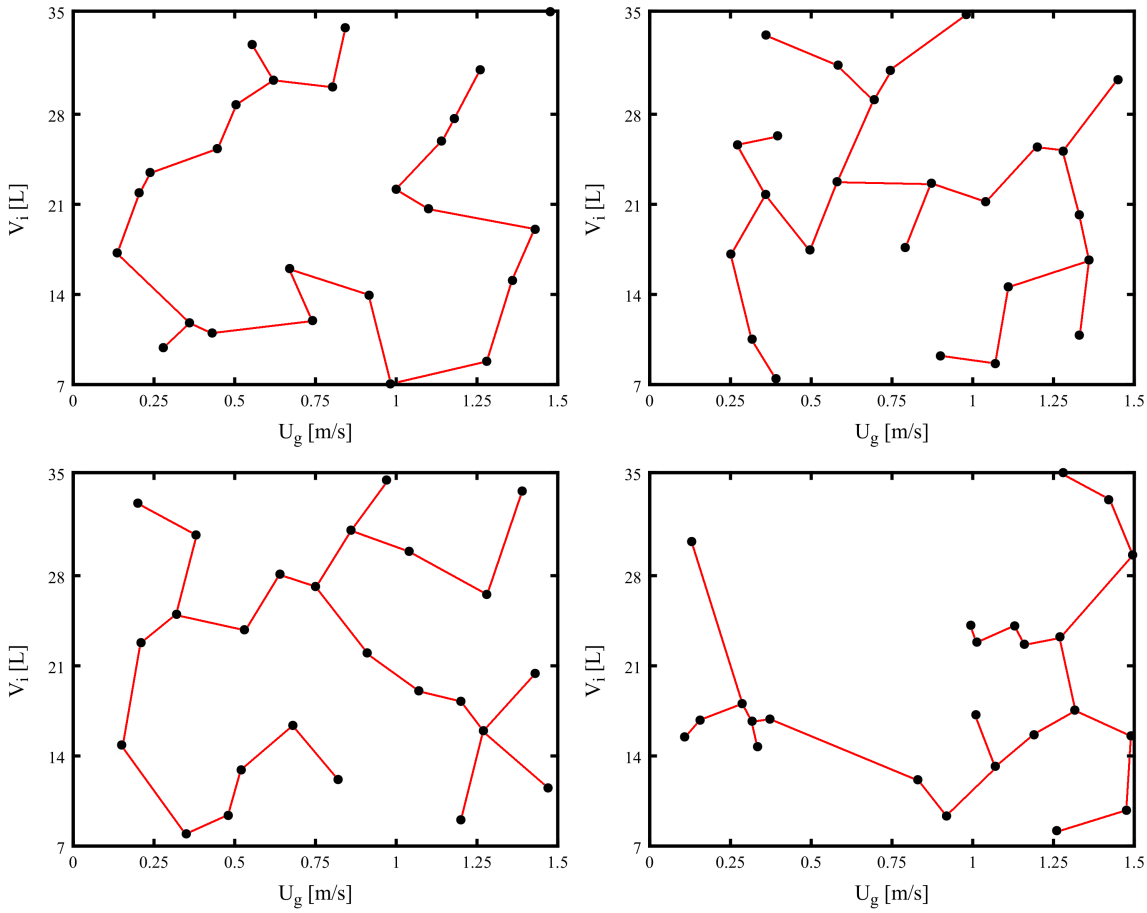


Figure D.3: The MST generated from the build points of the DACE study using a sample size of 25; (a) Latin hypercube sampling, (b) Orthogonal array sampling, (c) Orthogonal array based Latin hypercube sampling and (d) Monte Carlo random sampling.

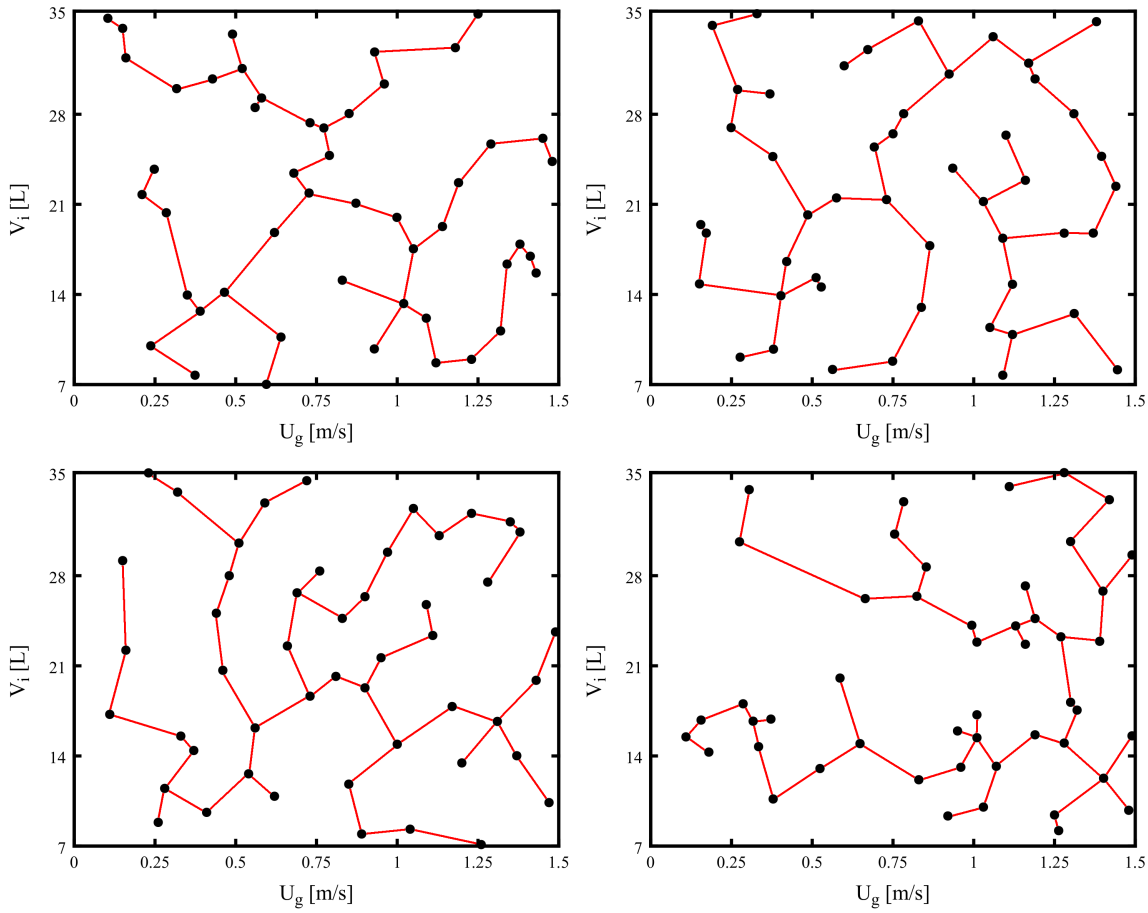


Figure D.4: The MST generated from the build points of the DACE study using a sample size of 49; (a) Latin hypercube sampling, (b) Orthogonal array sampling, (c) Orthogonal array based Latin hypercube sampling and (d) Monte Carlo random sampling.

D.4 Dataset Extracted from the CFD Simulations using 16 and 49 sampling points

Table D.1: The complete dataset of the data extracted from the CFD simulations using the 16 sampling point DACE study.

	CFD Input		Extracted data					
DACE point	U_g [m/s]	V_i [L]	t_{ss} [s]	$2\int_0^{t_{ss}} A_i \Delta t$ [m^{2}s]	$2A_{ss}$ [m^2]	L_w [m]	$\sum \alpha_{upper}$ [-]	$\sum \alpha_{lower}$ [-]
1	0.14	7.84	6400	3967	0.36	10	6.0	6.0
2	0.79	27.84	705	462	0.20	28	4.1	6.0
3	1.26	25.65	400	508	0.22	28	0.0	4.3
4	1.37	30.31	350	155	0.00	35	0.0	0.0
5	0.25	33.27	6995	6270	0.34	15	6.0	6.0
6	1.23	18.13	340	160	0.13	31	0.0	2.4
7	0.91	10.14	750	377	0.19	15	0.0	2.9
8	0.64	12.27	825	363	0.22	22	3.5	6.0
9	0.58	21.19	880	426	0.25	26	3.6	6.0
10	0.53	31.07	935	760	0.24	26	4.1	6.0
11	0.44	15.99	4050	1662	0.32	13	5.0	6.0
12	0.83	35.96	785	440	0.22	28	0.0	3.8
13	1.11	16.89	575	271	0.16	32	0.0	3.1
14	1.49	13.96	169	65	0.00	12	0.0	0.0
15	0.36	23.83	3995	2360	0.30	24	5.2	6.0
16	1.00	22.07	635	298	0.22	32	0.0	4.2

Table D.2: The complete dataset of the data extracted from the CFD simulations using the 49 sampling point DACE study.

	CFD Input		Extracted data					
DACE point	U_G [m/s]	V_i [L]	t_{ss} [s]	$2\int_0^{t_{ss}} A_i \Delta t$ [m^{2}s]	$2A_{ss}$ [m^2]	L_w [m]	$\sum \alpha_{upper}$ [-]	$\sum \alpha_{lower}$ [-]
1	1.38	31.39	315	136	0.00	35	0.0	0.0
2	1.31	16.68	345	151	0.10	32	0.0	2.4
3	0.72	35.38	860	491	0.20	28	0.0	4.3
4	0.56	16.18	890	434	0.22	22	3.2	6.0
5	0.66	22.54	845	484	0.22	28	1.1	4.3
6	1.49	23.63	240	121	0.00	35	0.0	0.0
7	0.37	14.42	4300	1830	0.30	24	5.2	6.0
8	0.16	22.21	6400	4065	0.33	13	6.0	6.0
9	1.43	19.87	255	123	0.00	37	0.0	0.0
10	1.47	10.38	250	80	0.00	12	0.0	0.0
11	1.26	7.12	370	128	0.11	13	0.0	2.4
12	0.32	34.49	4750	2460	0.33	27	5.0	6.0
13	0.69	26.67	860	497	0.22	28	0.0	4.3
14	1.20	13.46	360	363	0.10	13	0.0	2.4
15	0.90	19.29	725	319	0.19	27	0.0	4.0
16	0.51	30.53	940	780	0.24	26	4.2	6.0
17	0.44	25.09	910	613	0.27	26	3.7	6.0
18	0.59	33.65	870	574	0.22	28	0.0	4.3
19	1.09	25.75	585	311	0.20	32	0.0	4.0
20	1.04	8.31	605	137	0.11	32	0.0	3.5
21	0.90	26.37	710	330	0.19	27	0.0	4.0
22	0.26	8.84	2530	1270	0.32	12	5.1	6.0
23	1.37	14.02	350	122	0.10	32	0.0	2.4
24	0.73	18.64	790	445	0.22	20	4.2	6.0
25	0.46	20.65	910	652	0.26	26	3.7	6.0
26	0.15	29.17	7370	4060	0.34	15	1.35	6.0
27	1.05	33.22	590	310	0.20	32	0.0	4.0
28	1.23	32.84	390	164	0.10	33	0.0	1.0
29	0.95	21.63	710	322	0.19	27	0.0	4.0
30	0.54	12.61	895	407	0.22	22	3.2	6.0
31	0.89	7.94	750	348	0.19	15	0.0	2.9
32	0.33	15.54	4560	1786	0.30	15	5.0	6.0
33	0.23	35.98	7350	6672	0.34	15	6.0	6.0
34	0.97	29.82	710	356	0.20	31	0.0	4.0
35	1.13	31.11	580	277	0.20	32	0.0	4.0

	CFD Input		Extracted data					
DACE point	U_G [m/s]	V_i [L]	t_{ss} [s]	$2\int_0^{t_{ss}} A_i \Delta t$ [m^{2}s]	$2A_{ss}$ [m^2]	L_w [m]	$\sum \alpha_{upper}$ [-]	$\sum \alpha_{lower}$ [-]
36	0.85	11.81	760	344	0.19	15	0.0	2.9
37	0.76	28.36	690	461	0.20	30	0.0	4.0
38	0.81	20.18	810	402	0.19	21	4.2	6.0
39	1.17	17.84	340	229	0.20	32	0.0	2.4
40	1.00	14.91	710	303	0.22	27	0.0	4.0
41	1.11	23.36	610	256	0.20	32	0.0	4.0
42	0.28	11.48	2500	3038	0.34	12	5.1	6.0
43	1.35	32.18	320	148	0.00	35	0.0	0.0
44	0.11	17.23	6700	7089	0.36	23	6.0	6.0
45	0.83	24.68	710	561	0.19	30	0.0	4.0
46	0.62	10.87	800	452	0.21	20	4.2	6.0
47	0.41	9.62	1100	765	0.31	12	1.4	6.0
48	1.28	27.5	390	172	0.00	33	0.0	0.0
49	0.48	28	4700	872	0.31	27	5.0	6.0

Appendix E

Miscellaneous

E.1 Description of OpenFOAM® boundary condition terms

fixedValue Applies a fixed value constraint to the patch.

k Turbulent kinetic energy $[m^2 \cdot s^{-2}]$.

kqRWallFunction Provides a suitable condition for the turbulence kinetic energy (k), q (where q = $\sqrt{k}$) and Reynolds stress tensor (R) fields for the case of high Reynolds number flow using wall functions.

nut Turbulence viscosity $[m^2 \cdot s^{-1}]$.

nutkWallFunction provides a turbulent kinematic viscosity condition when using wall functions, based on turbulence kinetic energy.

nutkRoughWallFunction provides a turbulent kinematic viscosity condition when using wall functions for rough walls, based on turbulence kinetic energy. This wall function takes two inputs, the sand grain roughness height (Ks) and the roughness constant (Cs). In this study, Ks has been calculated according to Adams et al. [2012] and Cs is taken as 0.5.

omega Specific turbulent dissipation rate, ω $[s^{-1}]$.

omegaWallFunction Provides a wall function constraint for the specific turbulence dissipation, ω with $\omega = \sqrt{\omega_{vis}^2 + \omega_{log}^2}$, where the parameters ω_{vis} and ω_{log} are the specific turbulence dissipation in the viscous and log regions respectively.

p_rgh The difference between the static pressure (p) and the hydraulic pressure (ρgh), where h is an arbitrary height; i.e, $p\_rgh = p - \rho gh$ $[kg \cdot m^{-1} \cdot s^{-2}]$.

U Velocity $[m \cdot s^{-1}]$.

uniformFixedValue Applies a general fixed value condition with an additional time-varying value dependency.

zeroGradient Applies a zero-gradient from the patch internal field onto the patch faces.

E.2 OpenFOAM® y^+ Calculation

For wall-bounded flows, the non-dimensional wall distance (y^+) is calculated using the following equation;

$$y^+ = \frac{U_\tau y}{\nu} \tag{E.1}$$

Here, y is the distance to the first cell center normal to the wall, ν is the local kinematic viscosity of the fluid and U_τ is the friction velocity. The general consensus is that y^+ values must be in the range of 30 - 300 for the use of wall functions and less than 6 if the boundary layer is to be resolved Guerrero [2014]. Others authours have suggested that the SST $k - \omega$ turbulence model used in Chapter 3 requires a y^+ value of less than 2. Nevertheless, the value of y^+ for the wall patch of the 3-D half-pipe can be accessed in OpenFOAM® using the yPlusRAS utility. Visualisation of the y^+ parameter along the entire wall patch can easily be implemented using Paraview; however, care must be taken to ensure both gas and liquid phases have been accurately represented. This is because the *transportProperties* file may only take one argument for the kinematic viscosity. In order to address this, the yPlusRAS must be run twice (once for each phase) and mapped to the correct position on the pipe wall depending on the value of the corresponding phase fraction.

E.3 SALOME meshing strategy for low-spot pipe bends using hexahedral elements

While the meshing of a straight 3-D half-pipe in SALOME using hexahedral elements, such as the mesh presented in Chapter 3, is a relatively straightforward procedure, the complexity is increased greatly with the incorporation of a pipe bend with an initial water region. This is because a pipe containing an initial water zone cannot be successfully meshed as a single entity in SALOME with hexahedral elements. Without a water zone, the pipe can be easily meshed using 3D extruded elements where the meshed inlet face is the designated "source face" of the outlet face. Here; however,

the pipe must split into separate pipes depending on the number of required water regions. For example, a pipe containing a single water region will be split into 3 separate pipes, with the pipe containing the entire water region in the middle. These pipes are meshed independently of one another and later combined as a compound mesh. The water region is first defined in SALOME by defining a volumetric group of the pipe. Like previous meshes presented in this study, OpenFOAM® compatibility is attained by exporting the mesh as a .UNV file and subsequently converting the .UNV file using the ideasUnvToFoam utility. Following this procedure, the water region is referenced in the cellZones file located in the boundary folder, which contains all relevant information about the water region. The water region is then created in OpenFOAM® using the setFields utility, where the setFieldsDict file references the name of the water region located in the cellZones file.

The manner in which the pipe must be meshed is shown in Figure E.1, where three separate meshes are used and later compiled as a final compound mesh for use in OpenFOAM®. This way, the cross sectional spacing of elements is preserved and no errors are given.

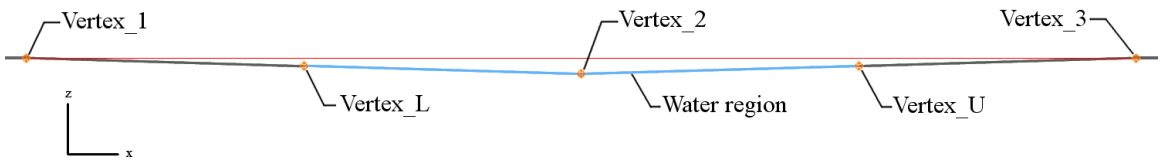


Figure E.1: The vertex locations used to define the low-spot section and initial water region.

The water region itself is defined by the two nodes, denoted here as Vertex_L and Vertex_U which are used to create a pipe. These vertices are produced by the following lines in the Python mesh script:

```
Vertex_L = geompy.MakeVertexOnCurve(Curve_1_edge_5, 0.9)
Vertex_U = geompy.MakeVertexOnCurve(Curve_1_edge_7, 0.1)
```

Here, Curve_1_edge_5 and Curve_1_edge_7 are the segment numbers of Curve_1 from Vertex_1 to Vertex_2 and Vertex_2 to Vertex_3 respectively. The numerical inputs, 0.9 and 0.1 determine the positioning of Vertex_L and Vertex_U along their edges from the preceding node and can be adjusted as necessary. The three curves used to create the separate pipes are defined from the following three lines:

```
Curve_INLET = geompy.MakePolyline([Vertex_ORIGIN, Vertex_1, Vertex_L],
    ↪ False)
Curve_WR = geompy.MakePolyline([Vertex_L, Vertex_2, Vertex_U], False)
Curve_OUTLET = geompy.MakePolyline([Vertex_U, Vertex_3, Vertex_4], False)
```

A couple of possible issues arise here due to the fact that the interface between the adjacent pipes is always vertical. The first issue involves the skewness that can be witnessed for inclined pipe sections. This is actually based upon the nature of the quadilateral elements used and should not pose much of a problem provided the inclination does not deviate significantly from the horizontal, however it is important to note that errors are to be expected if the strategy shown here is used for highly inclined pipes. The second issue involves the volumetric grouping of the initial water region. The water region is initially held within two imaginary vertical barriers defined by the respective pipes as shown earlier. As the simulation commences, the water region will deform due to the presence of gravity until it reaches a steady-state condition. In order to minimise errors and aid the convergence of the water phase fraction, it is important that the water region is allowed to reach steady-state prior to activation of the inlet velocity. This issue is presented graphically in Figure E.2, where two snapshots of the initial water region are shown 10 seconds apart. The lower image shows the steady-state position of the water region due to the influence of gravity. Also shown in this image is the slight skewness exhibited by the elements as mentioned previously.

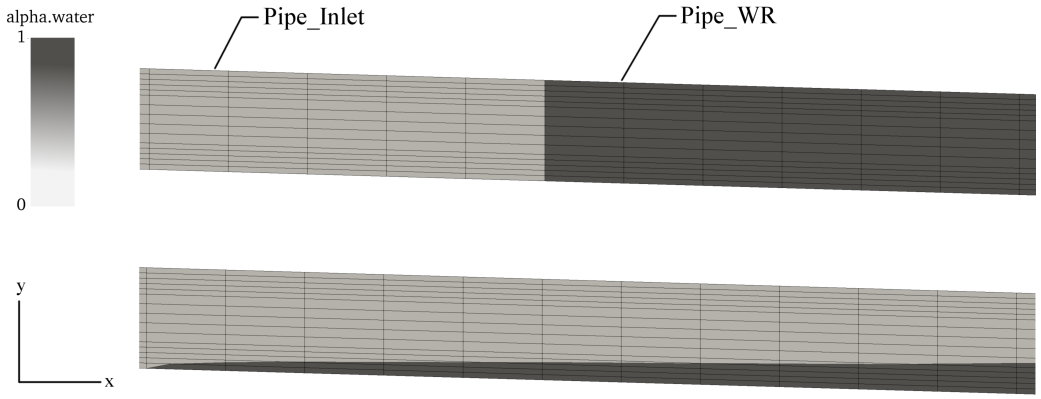


Figure E.2: The SALOME meshing strategy used to define the initial water regions via a vertical plane, whereby the water region is allowed to settle using the *uniformFixedValue* type boundary condition for the inlet velocity.

In order to activate the velocity after the simulation commences (i.e. after t = 0s) the *uniformFixedValue* type boundary condition is used for the inlet velocity. The *U*

file coding for the inlet patch is then as follows:

```
Inlet
{
type           uniformFixedValue;
uniformValue   tableFile;
tableFileCoeffs
{
  dimensions        [0 0 1 0 0];// dimensions
  fileName          "ramp.txt"; // name of data file
  outOfBounds       clamp;      // out-of-bounds handling
  interpolationScheme linear;   // interpolation method
};
  value uniform (0 0 0);
}
```

Here, the referenced "ramp.txt" file contains information about the time and velocity vector. The fully Python script file used in SALOME to generate the low-spot pipe bend used for the comparative study with Birvalski [2015] was shown previously in Appendix B.2.

E.4 SALOME meshing strategy for flowlines with multiple low-spot bends

For longer models built in SALOME, the vertical translation plane that defines the water cut is expected to traverse across multiple node points and the procedure to construct the mesh is increasingly more complex. An example of this can be seen in Figure E.3, where the flowline mesh features three separate water regions, each spanning multiple nodes.

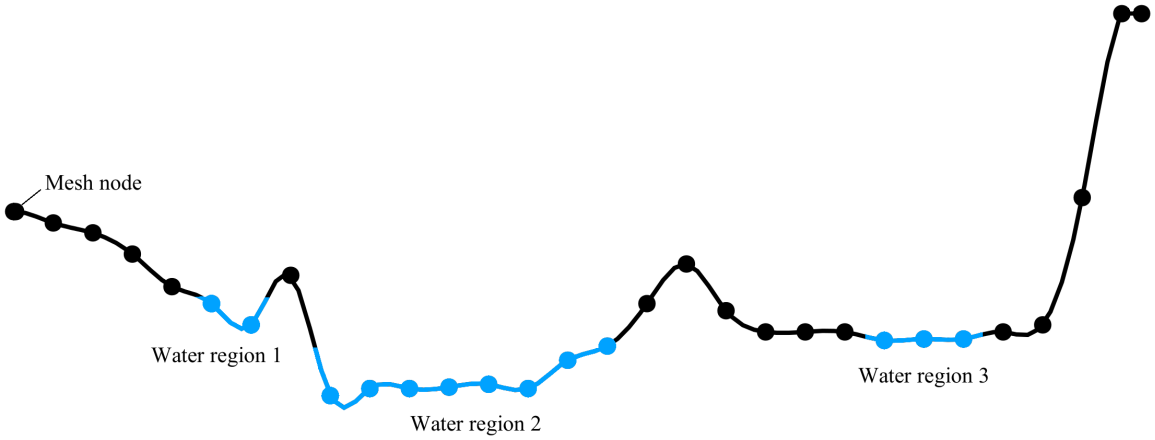


Figure E.3: An example of a long flowline section with multiple water regions containing multiple nodes.

A particular challenge occurs if the water regions change throughout the optimisation study, which will occur if the intial water volume is an initial water volume as seen in Chapter 6. In this case, while the specific node numbering may be easily updated in the script file to facilitate an expanding or contracting water zone, the specific numbering of the "wall", "symmetry" and "flow edge" IDs also need to be updated accordingly in a manner that is not straightforward due to slight variations that may exist depending upon the exact method of construction in SALOME. These instances are indicated by the following lines in the mesh.py.template;

1. *geompy.UnionIDs(Inlet, [num_i])* - For the inlet patch
2. *geompy.UnionIDs(Outlet, [num_o])* - For the outlet patch
3. *geompy.UnionIDs(Symmetry, [num_s])* - For the symmetry patch
4. *geompy.UnionIDs(Wall, [num_w])* - For the wall patch
5. *geompy.UnionIDs(FlowEdge, [num_FE])* - For the longitudinal flow edges

A definition of the Inlet, Wall and Symmetry model faces is shown in Figure E.4 for a flowline segment between two nodes. The flow edges represent a group of all horizontal lines shown in Figure E.4 and is used to vary the longitudinal spacing of the mesh elements, which has been termed the mesh resolution as used in Chapter 4, 5 and 6. The Outlet patch is equivalent to a mirror image of the inlet patch and is located at the final mesh node.

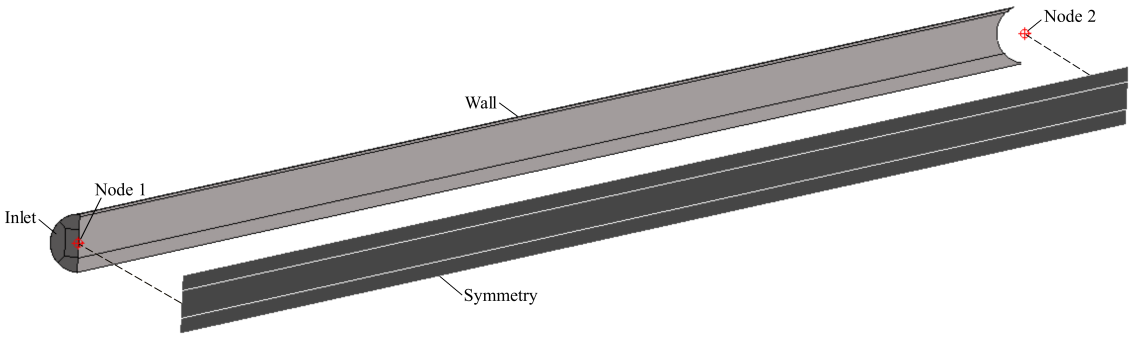


Figure E.4: Definition of the CFD model faces.

The *"num_i"* identifier is a sequence of four numbers that refer to the four faces used to construct the inlet patch. The specific numbering is determined automatically by SALOME during construction of the mesh and this sequence will not change with the expanding or contracting water zones. The identifiers *"num_o"*, *"num_s"*, *"num_w"* and *"num_FE"* are specific sequences of numbers that depend on the location of the of the translation plane used to create the water cut in SALOME and will change accordingly. These sets of numbers usually (but not always) take the form:

$$num\_o = [10 \times n_t + A] \tag{E.2}$$

$$num\_s = [10 \times n_i + B, ..., 10 \times n_t + B] \tag{E.3}$$

$$num\_w = [10 \times n_i, ..., 10 \times n_t] \tag{E.4}$$

$$num\_FE = [10 \times n_{i+1} + C, 10 \times n_{i+1} + D, ..., 10 \times n_t + C, 10 \times n_t + D] \tag{E.5}$$

Here, n_i represents the node number where; $n_i = (1, 2, ..., n_t)$, n_t is the final node number and A, B, C and D are integers dependant upon the cross section and number of nodes contained in the mesh. The number set generated from Equations E.2 to E.5 will follow the same sequences until the location of the vertical translation plane, where a transition will occur, causing the subsequent set of numbers to change. An example of this can be seen by the sequences of numbers presented in the code in Appendix B.2.

It follows that if the translation plane is moved across multiple nodes, the entire sequencing on numbers for each of the three identifiers shown above must be updated correctly to avoid corruption of the mesh when exported to OpenFOAM®. In order to circumvent this issue, two functions are added to the top of the Python code used by SALOME, these two functions are used to automatically determine the identifier numbering for the wall and symmetry patches and are as follows;

```
# Define functions to determine Wall IDs & Edge IDs

# This function is used to generate IDs for all curved edges where
# ''curveThreshold = Radius*1.1''. This is to ensure that only the
# curved wall faces are identified, and not faces that include the
# inlet, outlet and symmetry patches.


def wall_id(Pipe_ID, curveThreshold):
  AllFaces = geompy.SubShapeAll(Pipe_ID,geompy.ShapeType["FACE"])
  keptFaces = []
  for face in AllFaces:
    curvature = geompy.MaxSurfaceCurvatureByPoint(face, Vertex_0)
    if curvature < curveThreshold:
      keptFaces.append(face)
  WallFaces = geompy.GetSubShapesIDs(Pipe_ID, keptFaces)
  return (WallFaces)

# This function is used to generate IDs for all edges, but only if they
# are greater than a certain length (Radius*2). This is to ensure that
# only the longitudinal edges have been selected.

def edge_id(Pipe_ID):
  AllEdges = geompy.SubShapeAll(Pipe_ID, geompy.ShapeType["EDGE"])
  keptEdges = []
  for edge in AllEdges:
    length = geompy.BasicProperties(edge)[0]
    if length > Radius*2:
      keptEdges.append(edge)
  FlowEdges = geompy.GetSubShapesIDs(Pipe_ID, keptEdges)
  return (FlowEdges)
```

These two functions are called in the "GEOM component" of the mesh file to determine the wall IDs and flow edge IDs for the *"num_w"* and *"num_FE"* identifiers in each pipe section. The respective IDs related to the final identifier, *"num_s"* do not need a function call in order to generate the correct ID list. Instead the following command is used;

```
num_s = geompy.GetShapesOnPlaneIDs(Pipe_ID, geompy.ShapeType["FACE"], OY,
    ↪ GEOM.ST_ON)
```

For the inlet IDs;

```
num_i = geompy.GetShapesOnPlaneIDs(Pipe_ID, geompy.ShapeType["FACE"], OX,
    ↪ GEOM.ST_ON)
```

For the outlet IDs;

```
num_o = geompy.GetShapesOnPlaneIDs(Pipe_ID, geompy.ShapeType["FACE"], OX,
    ↪ Vertex_ID, GEOM.ST_ON)
```

Where *"Vertex_ID"* refers to the terminating node on the specific pipe designated by the Pipe_ID. An example for the implementation of this is shown in Appendix B.3, which was used for the third segment of the Ormen Lange mesh used in Chapter 6.

E.5 Gas composition data

Table C.1 presents the molar fractions of the gas used in the study by Leporcher et al. [2002], and subsequently used in Chapter 5 of this study.

Table E.1: Molar fractions presented by Leporcher et al. [2002].

Component	Molar fraction (%)
N2	1.325
CO2	0.393
C1	94.317
C2	2.957
C3	0.679
IC4	0.148
NC4	0.103
C5+	0.078

E.6 Published Works

This research has currently lead to the following publications;

1. A. M. Shuard, H.B. Mahmud and A. J. King. Comparison of Two-Phase Pipe Flow in OpenFOAM with a Mechanistic Model. *IOP Conference Series: Materials Sci. & Eng.*, 121:012018, 2016

2. A. M. Shuard, H.B. Mahmud and A. J. King. An optimization approach to reduce the risk of hydrate plugging during gas-dominated restart operations. *J. Pet. Sci. & Eng.* 156:220-234, 2017

References

A. Aarvik, I. Olsen, K. Vannes, K. Havre, and E. Krogh. Design and development of the Ormen Lange flow assurance simulator. Article presented at the *13th International Conference on Multiphase Production Technology*, Edinburgh, UK, June 13 - 15, 2007.

B. M. Adams, M. S. Ebeida, M. S. Eldred, J. D. Jakeman, L. P. Swiler, J. A. Stephens, D. M. Vigil, T. M. Wildey, W. J. Bohnhoff, and J. P. Eddy. Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: Version 6.3 users manual. Technical report, Sandia National Laboratories, July 2014.

T. Adams, C. Grant, and H. Watson. A simple algorithm to relate measured surface roughness to equivalent sand-grain roughness. *International Journal of Mechanical Engineering and Mechatronics*, 1(2):66–71, 2012.

R. R. Alapati, J. Lee, and D. Beard. Two field studies demonstrate that new LDHI chemistry is effective at high water cuts without impacting oil/water quality. Article presented at the *Offshore Technology Conference*, Houston, Texas, May 5-8, 2008.

Z. M. Aman, M. Di Lorenzo, K. Kozielski, C. A. Koh, P. Warrier, M. L. Johns, and E. F. May. Hydrate formation and deposition in a gas-dominant flowloop: Initial studies of the effect of velocity and subcooling. *Journal of Natural Gas Science and Engineering*, 35(B):1490–1498, 2016.

S. Amaravadi, K. Minami, and O. Shoham. Two-phase zero-net liquid flow in upward inclined pipes: Experiment and modeling. *SPE Journal*, 3(03):253–260, 1998.

P. Andreussi, A. Paglianti, and F. S. Silva. Dispersed bubble flow in horizontal pipes. *Chemical Engineering Science*, 54(8):1101–1107, 1999.

M. Antosiewicz, G. Koloch, and B. Kamiński. Choice of best possible metaheuristic algorithm for the travelling salesman problem with limited computational time: quality, uncertainty and speed. *Journal of Theoretical and Applied Computer Science*, 7(1):46–55, 2013.

J. Arora, O. Elwakeil, A. Chahande, and C. Hsieh. Global optimization methods for engineering applications: a review. *Structural optimization*, 9(3-4):137–159, 1995.

T. Asim, R. Mishra, S. Kaysthagir, and G. Aboufares. Performance comparison of a vertical axis wind turbine using commercial and open source computational fluid dynamics based codes. Presented at the *5th International Conference of Jets, Wakes and Separated Flows conference*, Stockholm, Sweden, June 15 - 18, 2015.

E. L. Axdahl. An automated DAKOTA and VULCAN-CFD framework with application to supersonic facility nozzle flowpath optimization. Technical report, NASA, 2015.

Y. Bai and Q. Bai. *Subsea pipelines and risers*. Elsevier, 2005. ISBN 9780080445663.

G. Bailardi, J. Guerrero, and D. Natali. On the fluid-dynamic design and optimization of sailing yachts' hull using a complete open source framework. Article presented at the *VI International Conference on Computational Methods in Marine Engineering*, Rome, Italy, June 15 - 17, 2015.

V. Balabanov, C. Charpentier, D. Ghosh, G. Quinn, G. Vanderplaats, and G. Venter. Visualdoc: A software system for general purpose integration and design optimization. Article presented at the *9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, Atlanta, USA, September 4-6, 2015.

B. V. Balakin, S. Lo, P. Kosinski, and A. C. Hoffmann. Modelling agglomeration and deposition of gas hydrates in industrial pipelines with combined CFD-PBM technique. *Chemical Engineering Science*, 153:45–57, 2016.

D. Barnea, O. Shoham, and Y. Taitel. Flow pattern transition for downward inclined two phase flow; horizontal to vertical. *Chemical Engineering Science*, 37(5):735–740, 1982.

D. H. Beggs and J. P. Brill. A study of two-phase flow in inclined pipes. *Journal of Petroleum Technology*, 25(05):607–617, 1973.

R. Belt, E. Duret, D. Larrey, B. Djoric, and S. Kalali. Comparison of commercial multiphase flow simulators with experimental and field databases. Article presented at the *15th International Conference on Multiphase Production Technology*, Cannes, France, June 15-17, 2011.

K. Bendiksen. Multiphase flow modelling and its significance for subsea field developments on the NCS. Article presented at the *Tekna Conference on MP Flow Systems*, Oslo, Norway, October 9, 2015.

D. Bestion. The difficult challenge of a two-phase CFD modelling for all flow regimes. *Nuclear Engineering and Design*, 279:116–125, 2014.

M. Birvalski. *Experiments in stratified gas-liquid pipe flow*. PhD thesis, TU Delft, Delft University of Technology, 2015.

C. Blum and A. Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308, 2003.

O. Bratland. Ormen Lange: 3-phase flow assurance study. Technical report, Dr. Ove Bratland Flow Assurance Consulting, 2015.

S. P. Brooks and B. J. Morgan. Optimization using simulated annealing. *The Statistician*, 44(2):241–257, 1995.

T. Buanes, B. Kvamme, and A. Svandal. Computer simulation of CO2 hydrate growth. *Journal of crystal growth*, 287(2):491–494, 2006.

T. Buanes, B. Kvamme, and A. Svandal. Two approaches for modelling hydrate growth. *Journal of mathematical chemistry*, 46(3):811–819, 2009.

C. Burns, S. Lorimer, M. Hartenhof, and T. Vanvik. Start-up and operation of the ormen lange flowlines. Article presented at the *18th International Offshore and Polar Engineering Conference*, Vancouver, Canada, July 6-11, 2008.

G. Capasso. A deterministic method for the multiobjective optimization of electromagnetic devices and its application to pose detection for magnetic-assisted medical applications. Master's thesis, Universtity of Padova, 2015.

M. Cavazzuti. *Optimization Methods*. Springer, 2013. ISBN 9783642311864.

K. Chandrasekar and N. Ramana. Performance comparison of GA, DE, PSO and SA approaches in enhancement of total transfer capability using facts devices. *Journal of Electrical Engineering and Technology*, 7(4):493–500, 2012.

G. Chen, Q. Xiong, P. J. Morris, E. G. Paterson, A. Sergeev, and Y. Wang. OpenFOAM for computational fluid dynamics. Technical report, American Mathematical Society, 2014.

B. Cheng and D. M. Titterington. Neural networks: A review from a statistical perspective. *Statistical Science*, 9(1):2–30, 1994.

L. Chica, R. Pascali, P. Jukes, B. Ozturk, M. Gamino, and K. Smith. Detailed fsi analysis methodology for subsea piping components. Article presented at the *31st International Conference on Ocean, Offshore and Arctic Engineering*, Rio Di Janeiro, Brazil, July 1 - 6, 2012.

R. T. Collins and P. Carr. Hybrid stochastic/deterministic optimization for tracking sports players and pedestrians. Article presented at the *European Conference on Computer Vision*, Zurich, Switzerland, September 6 - 12, 2014.

T. Craft, S. Gant, A. Gerasimov, H. Iacovides, and B. Launder. Wall functions strategies for use in turbulent flow CFD. Article presented at the *12th International Heat Transfer Conference*, Genoble, France, August 18 - 23, 2002.

J. L. Creek, S. Subramanian, and D. A. Estanga. New method for managing hydrates in deepwater tiebacks. Article presented at the *Offshore Technology Conference*, Houston, Texas, May 2-5, 2011.

P. A. de Sampaio, J. L. Faccini, and J. Su. Modelling of stratified gas–liquid two-phase flow in horizontal circular pipes. *International Journal of Heat and Mass Transfer*, 51(11):2752–2761, 2008.

L. Delebecque, E. Sibaud, M. Scocard, C. Rueda, and P. Delbene. Active heating technologies for flowlines in deep water field developments. Article presented at the *Offshore Mediterranean Conference and Exhibition*, Ravenna, Italy, March 25-27, 2009.

A. B. Desamala, A. K. Dasamahapatra, and T. K. Mandal. Oil–water two-phase flow characteristics in horizontal pipeline–a comprehensive CFD study. Article presented at the *8th CUTSE conference*, Miri, Malaysia, December 3-4, 2013.

S. S. Deshpande, L. Anumolu, and M. F. Trujillo. Evaluating the performance of the two-phase flow solver interfoam. *Computational Science & Discovery*, 5(1):1–36, 2012.

M. Di Lorenzo, Z. M. Aman, K. Kozielski, B. W. Norris, M. L. Johns, and E. F. May. Underinhibited hydrate formation and transport investigated using a single-pass gas-dominant flowloop. *Energy & Fuels*, 28(11):7274–7284, 2014.

V. Díaz-Casás, J.-A. Becerra, F. Lopez-Peña, and R. J. Duro. Wind turbine design through evolutionary algorithms based on surrogate CFD methods. *Optimization and Engineering*, 14(2):305–329, 2013.

C. Dussert, G. Rasigni, M. Rasigni, J. Palmari, and A. Llebaria. Minimal spanning tree: A new approach for studying order and disorder. *Physical Review B*, 34(5): 3528, 1986.

N. Dyn, D. Levin, and S. Rippa. Numerical procedures for surface fitting of scattered data by radial functions. *SIAM Journal on Scientific and Statistical Computing*, 7 (2):639–659, 1986.

Y. Egorov. Contact condensation in stratified steam-water flow. Technical report, EVOL-ECORA-D07, 2012.

B. Eisenhower, Z. ONeill, S. Narayanan, V. A. Fonoberov, and I. Mezić. A methodology for meta-model based optimization in building energy models. *Energy and Buildings*, 47:292–301, 2012.

K. Ekambara, R. S. Sanders, K. Nandakumar, and J. Masliyah. CFD modeling of gas-liquid bubbly flow in horizontal pipes: influence of bubble coalescence and breakup. *International Journal of Chemical Engineering*, 2012.

I. R. Ellul. Dynamic multiphase simulation-the state of play. Article presented at the *41st PSIG Annual Meeting*, Bonita Springs, Florida, May 11-14, 2010.

I. R. Ellul, G. Saether, and M. E. Shippen. The modeling of multiphase systems under steady-state and transient conditions-a tutorial. Article presented at the *36th PSIG Annual Meeting*, Palm Springs, California, October 20-22, 2004.

V. Emad. Evaluating the performance of various convection schemes on free surface flows by using interfoam solver. Master's thesis, Eastern Mediterranean University (EMU), 2014.

P. Englezos, N. Kalogerakis, P. Dholabhai, and P. Bishnoi. Kinetics of gas hydrate formation from mixtures of methane and ethane. *Chemical Engineering Science*, 42(11):2659–2666, 1987.

M. Espedal. *An experimental investigation of stratified two-phase pipe flow at small inclinations*. PhD thesis, Norwegian University of Science and Technology, 1998.

J. Eves, V. V. Toropov, H. M. Thompson, N. Kapur, J. Fan, D. Copley, and A. Mincher. Design optimization of supersonic jet pumps using high fidelity flow analysis. *Structural and Multidisciplinary Optimization*, 45(5):739–745, 2012.

C. Fontes, L. Schuenck, and M. Reis. Experimental and computational modelling of gas-liquid annular flow in horizontal tubes. 2008.

A. I. J. Forrester and A. J. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1):50–79, 2009.

J. Franco, O. Vasseur, B. Corre, and M. Sergent. Minimum spanning tree: A new approach to assess the quality of the design of computer experiments. *Chemometrics and intelligent laboratory systems*, 97(2):164–169, 2009.

E. M. Freer, M. S. Selim, and E. D. Sloan. Methane hydrate film growth kinetics. *Fluid Phase Equilibria*, 185(1):65–75, 2001.

M. P. Friedlander and M. Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.

J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 9 (1):1–67, 1991.

R. Gajbhiye and S. Kam. Characterization of foam flow in horizontal pipes by using two-flow-regime concept. *Chemical Engineering Science*, 66(8):1536–1549, 2011.

C. Galusinski and P. Vigneaux. On stability condition for bifluid flows with surface tension: Application to microfluidics. *Journal of Computational Physics*, 227(12): 6140–6164, 2008.

E. Gharaibah, A. Read, and G. Scheuerer. Overview of CFD multiphase flow simulation tools for subsea oil and gas system design, optimization and operation. Article presented at the *Offshore Technology Conference*, Rio de Janeiro, Brazil, October 27 - 29, 2015.

C. A. Gilkeson, V. V. Toropov, H. M. Thompson, M. C. T. Wilson, N. A. Foxley, and P. H. Gaskell. Aerodynamic shape optimization of a low drag fairing for small livestock trailers. Article presented at the *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, British Columbia, Canada, September 10-12, 2008.

A. A. Giunta. Use of data sampling, surrogate models, and numerical optimization in engineering design. Article presented at the *40th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 14-17, 2002.

B. Glaz, P. P. Friedmann, and L. Liu. Surrogate based optimization of helicopter rotor blades for vibration reduction in forward flight. *Structural and Multidisciplinary Optimization*, 35(4):341–363, 2008.

V. Gleize and V. Burnley. A wall-distance free k-w sst turbulence model. Article presented at the *54th annual meeting of the division of fluid dynamics*, San Diego, CA, Nov 18 - 20, 2001.

D. E. Goldberg and J. H. Holland. Genetic algorithms and machine learning. *Machine learning*, 3(2):95–99, 1988.

L. Gomez, O. Shoham, Z. Schmidt, R. Chokshi, A. Brown, and T. Northug. A unified mechanistic model for steady-state two-phase flow in wellbores and pipelines. 1999. Article presented at the *SPE Annual Technical Conference and Exhibition*, Houston, Texas, October 3-6.

N. Goodman and N. Joshi. A tale of two flowlines-paraffin plugging and remediation. Article presented at the *SPE Annual Technical Conference and Exhibition*, New Orleans, Louisiana, September 30 - October 2, 2013.

A. Guariguata, M. A. Pascall, M. W. Gilmer, A. K. Sum, E. D. Sloan, C. A. Koh, and D. T. Wu. Jamming of particles in a two-dimensional fluid-driven flow. *Physical Review E*, 86(6):061311, 2012.

J. Guerrero. Crash introduction to turbulence modeling. Article presented at the *Introductory OpenFoam Course*, University of Genoa, Italy, July 14 - 18, 2014.

N. Gupta. Overview of the Ormen Lange project. University Lecture, Norwegian University of Science and Technology, 2012.

O. Gur, M. Bhatia, W. H. Mason, J. A. Schetz, R. K. Kapania, and T. Nam. Development of framework for truss-braced wing conceptual mdo. *Structural and Multidisciplinary Optimization*, 44(2):277–298, 2011.

S. E. Haaland. Simple and explicit formulas for the friction factor in turbulent pipe flow. *Journal of Fluids Engineering*, 105(1):89–90, 1983.

M. Haghgoo. A comparison of CFD software packages to find the suitable one for numerical modeling of gasification process. Technical report, University of Saskatchewan, 2013.

E. Hammerschmidt. Formation of gas hydrates in natural gas transmission lines. *Industrial Engineering & Engineering Chemistry*, 26(8):851–855, 1934.

T. Hanratty. *Physics of Gas-Liquid Flows*. Cambridge University Press, 2013. ISBN 9781107041202.

R. L. Hardy. Multiquadric equations of topography and other irregular surfaces. *Journal of Geophysical Research*, 76(8):1905–1915, 1971.

D. J. Harvie, M. Davidson, and M. Rudman. An analysis of parasitic current generation in volume of fluid simulations. *Applied Mathematical Modelling*, 30(10): 1056–1066, 2006.

A. Hasan, M. Vuolle, and K. Sirén. Minimisation of life cycle cost of a detached house using combined simulation and optimisation. *Building and Environment*, 43(12): 2022–2034, 2008.

R. Hassan, B. Cohanim, O. De Weck, and G. Venter. A comparison of particle swarm optimization and the genetic algorithm. Article presented at the *1st AIAA multidisciplinary design optimization specialist conference*, Austin, Texas, April 18 - 21, 2005.

T. Hauser, T. A. Johansen, and R. P. LeBeau Jr. Computational optimization of a low reynolds number inflatable airfoil. Article presented at the *20th AIAA Computational Fluid Dynamics Conference*, Honolulu, Hawaii, June 27-30, 2011.

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

S. Høiland, A. E. Borgund, P. Glénat, P. Fotland, K. Askvik, and X. Li. The wetting index: A quantitative measure of indigenous hydrate plugging tendency; flow test validations. Article presented at the *7th International Conference on Gas Hydrates*, Edinburgh, UK, July 17-21, 2011.

O. M. Holden, G. Paulsen, P. Bryn, and T. Marthinsen. Ormen lange pipelines routing and cost-effective seabed preparation. Article presented at the *Fifteenth International Offshore and Polar Engineering Conference*, Seoul, Korea, June 19-24, 2005.

H. Holm. Long gas/condensate tie-backs. Presentation, Rio de Janeiro, Brazil, October 30-31, 2005.

K. Holmas, G. G. Lunde, G. R. Setyadi, P. Angelo, and G. Rudrum. Ormen lange flow assurance system (FAS)-online flow assurance monitoring and advice. Article presented at the *Offshore Technology Conference*, Rio de Janeiro, Brazil, October 29-31, 2013.

K. Holmström, A. O. Göran, and M. M. Edvall. Users guide for tomlab 7, tomlab manual. Technical report, Tomlab Optimization Inc., 2010.

T. Irfansyah, B. Widyoko, G. Gunarwan, and D. Lopez. Simulation of multiphase flows in indonesian pipelines: Comparison of TACITE and OLGA results. Article presented at the *12th International Conference on Multiphase Production Technology*, Barcelona, Spain, May 25 - 27, 2005.

E. Jassim, M. A. Abdi, and Y. Muzychka. A new approach to investigate hydrate deposition in gas-dominated flowlines. *Journal of Natural Gas Science and Engineering*, 2(4):163–177, 2010.

D. Jia. Slug flow induced vibration in a pipeline span, a jumper and a riser section. Article presented at the *Offshore Technology Conference*, Houston, Texas, April 30 - May 3, 2012.

R. Jin, W. Chen, and T. W. Simpson. Comparative studies of metamodelling techniques under multiple modelling criteria. *Structural and Multidisciplinary Optimization*, 23(1):1–13, 2001.

T. A. Johansen. Optimization of a low reynolds number 2-D inflatable airfoil section. Master's thesis, Utah State University, 2011.

F. José, J. N. Carneiro, J. E. Oliveira, M. Sjur, M. Fossen, and S. T. Johansen. CFD simulation of the two-phase flow of different mixtures in a closed system flow wheel. Article presented at the *10th International Conference on CFD in Oil & Gas, Metallurgical and Process Industries*, Trondheim, Norway, June 17-19, 2014.

J. Kenndy and R. Eberhart. Particle swarm optimization. Article presented at the *IEEE International Conference on Neural Networks*, Perth, Australia, November 27 - December 1, 1995.

Z. Khatir, H. Thompson, N. Kapur, V. Toropov, and J. Paton. Multi-objective computational fluid dynamics (CFD) design optimisation in commercial bread-baking. *Applied Thermal Engineering*, 60(1):480–486, 2013.

S. Knotek, A. Fiebach, and S. Schmelter. Numerical simulation of multiphase flows in large horizontal pipes. Article presented at the *17th International Flow Measurement Conference*, Sydney, Australia, September 26 - 29, 2016.

D. Kroshko. Openopt: Free scientific-engineering software for mathematical modeling and optimization. Technical report, 2007.

E. A. Krumrick, E. J. López, and A. G. Camacho. Prediction of the flow regime in liquid-gas flows through straight pipes using computational fluid dynamics. *Mecánica Computacional*, 34(30), 2016.

K. Kumar, J. P. Thaker, and J. Banerjee. Experimental investigations on two-phase flow in horizontal pipe. Article presented at the *Fortieth National Conference on Fluid Mechanics and Fluid Power*, Hamipur, India, December 12-14, 2013.

J. Labeaga and N. H. Omagogeascoa. Two–phase pipeflow simulations with OpenFoam. Master's thesis, Norwegian University of Science and Technology, 2013.

C. Lawrence, J. Nossen, R. Skartlien, O. Skjæraasen, O. Sendstad, J. Hua, L. Wollebæk, L. Liu, K. Hald, and T. Sira. Mechanistic models for three-phase stratified and slug flows with dispersions. Article presented at the *16th International Conference on Multiphase Production Technology*, Cannes, France, June 12-14, 2013.

A. L. Lee, M. H. Gonzalez, and B. E. Eakin. The viscosity of natural gases. *Journal of Petroleum Technology*, 18(08):997–1, 1966.

K. Y. Lee and M. A. El-Sharkawi. *Modern heuristic optimization techniques: theory and applications to power systems*, volume 39. John Wiley & Sons, 2008. ISBN 9780471457114.

M. Lemoël, P. Brown, P. Jean, and B. Shepheard. Design of the world's 1st gravity actuated pipe (GAP) for Murphy's Kikeh deepwater development, East Malaysia. Article presented at the *27th International Conference on Offshore Mechanics and Arctic Engineering*, Estoril, Portugal, June 15-20, 2008.

A. Lenes, J. K. Lervik, H. Kulbotten, A. Nysveen, and A. H. Børnes. Hydrate prevention on long pipelines by direct electrical heating. Article presented at the *15th International Offshore and Polar Engineering Conference*, Seoul, Korea, June 19-24, 2005.

E. Leporcher, J. L. Peytavy, Y. Mollier, J. Sjöblom, and C. Labes-Carrier. Multiphase transportation: hydrate plugging prevention through crude oil natural surfactants. Article presented at the *SPE Annual Technical Conference and Exhibition*, New Orleans, Louisiana, September 27-30, 1998.

E. Leporcher, K. Kinnari, C. Labes-Carrier, P. Maurel, and W. Vandersippe. Multiphase flow: Can we take advantage of hydrodynamic conditions to avoid hydrate plugging during deepwater restart operations? Article presented at the *SPE Annual Technical Conference and Exhibition*, San Antonio, Texas, September 29 - October 2, 2002.

D. Levin. The approximation power of moving least-squares. *Mathematics of Computation of the American Mathematical Society*, 67(224):1517–1531, 1998.

I. Lindmeier, C. Heschl, G. Clauss, and U. Heck. Prediction of the flow around 3d obstacles using open source CFD-software. Article presented at the *Fifth International Symposium on Computational Wind Engineering*, Chapel Hill, USA, May 23-27, 2010.

S. Lo and A. Fiala. Recent progress in CFD modelling of multiphase flow in horizontal and near-horizontal pipes. Article presented at the *Subsea Asia 2010 conference*, Kuala Lumpur, Malaysia, June 9 - 11, 2010.

S. Lo and A. Tomasello. Recent progress in CFD modelling of multiphase flow in horizontal and near-horizontal pipes. Article presented at the *7th North American Conference on Multiphase Technology*, Banff, Canada, June 2 - 4, 2010.

J. Löfberg. YALMIP: A toolbox for modeling and optimization in matlab. Article presented at the *IEEE International Symposium on Computer Aided Control Systems Design*, New Orleans, USA, September 2 - 4, 2004.

A. Mackenzie, A. Lopez, K. Ritos, M. Stickland, and W. Dempster. A comparison of CFD software packages ability to model a submerjed jet. Presented at the *Eleventh International Conference on CFD in the Minerals and Process Industries*, Melbourne, Australia, September 7-9, 2015.

M. B. Malegaonkar, P. D. Dholabhai, and P. R. Bishnoi. Kinetics of carbon dioxide and methane hydrate formation. *The Canadian Journal of Chemical Engineering*, 75(6):1090–1099, 1997.

B. Malone and M. Papay. ModelCenter: an integration environment for simulation based design. Article presented at the *Simulation Interoperability Workshop*, Orlando, Florida, March 15-19, 1999.

J. Martins and C. Marriage. An object-oriented framework for multidisciplinary design optimization. Article presented at the *3rd AIAA multidisciplinary design optimization specialist conference*, Hawaii, USA, April 23, 2007.

E. Masson and V. Gleize. Wall-distance free k-w turbulence model for compressible unsteady flows. Article presented at the *European Congress on Computational Methods in Applied Sciences and Engneering*, Jyvaskyla, Finland, July 24 - 28, 2004.

G. Matheron. Principles of geostatistics. *Economic Geology*, 58(8):1246–1266, 1963.

MATLAB. version 7.10.0 (r2010a). Technical report, The MathWorks Inc., 2010.

Q. H. Mazumder. CFD analysis of single and multiphase flow characteristics in elbow. *Engineering*, 4(4):210–214, 2012.

F. Menter, M. Kuntz, and R. Langtry. Ten years of industrial experience with the SST turbulence model. Article presented at the *Fourth International Symposium on Turbulence, Heat and Mass Transfer*, Antalya, Turkey, October 12-17, 2003.

T. Mochizuki and Y. H. Mori. Clathrate-hydrate film growth along water/hydrate-former phase boundariesnumerical heat-transfer study. *Journal of crystal growth*, 290(2):642–652, 2006.

Y. H. Mori. Estimating the thickness of hydrate films from their lateral growth rates: application of a simplified heat transfer model. *Journal of crystal growth*, 223(1): 206–212, 2001.

S. T. Munkejord, M. J. Mølnvik, J. A. Melheim, I. R. Gran, and R. Olsen. Prediction of two-phase pipe flows using simple closure relations in a 2d two-fluid model. Article presented at the *Fourth International Conference on CFD in the Oil and Gas, Metallurgical & Process Industries*, Trondheim, Norway, June 10-12, 2005.

L. Nardin, K. Sørensen, S. Hitzel, and U. Tremel. modeFRONTIER©, a framework for the optimization of military aircraft configurations. In *MEGADESIGN and MegaOpt-German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design*, pages 191–205. Springer, 2009.

A.-T. Nguyen, S. Reiter, and P. Rigo. A review on simulation-based optimization methods applied to building performance analysis. *Applied Energy*, 113:1043–1058, 2014.

E. O. Obanijesu, V. Pareek, R. Gubner, and M. O. Tade. Hydrate formation and its influence on natural gas pipeline internal corrosion. *Nafta*, 62(5):164–173, 2011.

G. Oddie, H. Shi, L. Durlofsky, K. Aziz, B. Pfeffer, and J. Holmes. Experimental study of two and three phase flows in large diameter inclined pipes. *International Journal of Multiphase Flow*, 29(4):527–558, 2003.

Y. S. Ong, P. B. Nair, and A. J. Keane. Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA journal*, 41(4):687–696, 2003.

OPTIMUS. User's manual. Technical report, Noesis Solutions, 2004.

L. Pan and T. J. Hanratty. Correlation of entrainment for annular flow in horizontal pipes. *International Journal of Multiphase Flow*, 28(3):385–408, 2002.

R. E. Perez, P. W. Jansen, and J. R. Martins. pyopt: a python-based object-oriented framework for nonlinear constrained optimization. *Structural and Multidisciplinary Optimization*, 45(1):101–118, 2012.

N. Petalas and K. Aziz. Stanford multiphase flow database - user's manual. Technical report, Stanford University, 1995.

N. Petalas and K. Aziz. A mechanistic model for multiphase flow in pipes. *Journal of Canadian Petroleum Technology*, 39(06), 2000.

A. S. Ragab, W. Brandstaetter, G. J. Ruthammer, and S. Shalaby. Analysis of multiphase production through hilly terrain pipelines in Matzen Field, Austria by CFD. Article presented at the *SPE Annual Technical Conference and Exhibition*, Denver, Colorado, September 21-24, 2008.

I. Rao, E. D. Sloan, C. A. Koh, and A. K. Sum. Laboratory experiments & modeling for hydrate formation and deposition from water saturated gas systems. Article presented at the *7th International Conference on Gas Hydates*, Edinburgh, UK, July 17 - 21, 2011.

R. V. Rao and V. J. Savsani. *Mechanical design optimization using advanced optimization techniques*. Springer Science & Business Media, 2012.

A. S. Rattner and S. Garimella. Simple mechanistically consistent formulation for volume-of-fluid based computations of condensing flows. *Journal of Heat Transfer*, 136(7):071501, 2014.

M. Ratzel and F. Kocer. Cfd optimization with altair hyperworks. Technical report, Altair Engineering, 2007.

A. Røsdal. Ormen Lange and Langeled. University Lecture, Norwegian University of Science and Technology, 2008.

R. Rzehak and S. Kriebitzsch. Multiphase cfd-simulation of bubbly pipe flow: A code comparison. *International Journal of Multiphase Flow*, 68:135–152, 2015.

C. Sarica and H. Zhang. Development of next generation multiphase pipe flow prediction tools. Technical report, University Of Tulsa, 2006.

R. B. Schnabel. Parallel computing in optimization. Technical report, University of Colorado, 1984.

D. A. Shaw and T. J. Hanratty. Turbulent mass transfer rates to a wall for large schmidt numbers. *AIChE Journal*, 23(1):28–37, 1977.

M. Shengwei. Why isnt open source cfd solution for everyone?, 2013. URL `caewatch.com/why-open-source-cfd-solution-is-not-for-everyone/`.

M. Shippen and W. J. Bailey. Steady-state multiphase flow past, present, and future, with a perspective on flow assurance. *Energy & Fuels*, 26(7):4145–4157, 2012.

T. W. Simpson, T. M. Mauery, J. J. Korte, and F. Mistree. Kriging models for global approximation in simulation-based multidisciplinary design optimization. *AIAA Journal*, 39(12):2233–2241, 2001.

T. W. Simpson, A. J. Booker, D. Ghosh, A. A. Giunta, P. N. Koch, and R.-J. Yang. Approximation methods in multidisciplinary analysis and optimization: a panel discussion. *Structural and Multidisciplinary Optimization*, 27(5):302–313, 2004.

J. Sjöblom, B. Øvrevoll, G. Jentoft, C. Lesaint, T. Palermo, A. Sinquin, P. Gateau, L. Barré, S. Subramanian, and J. Boxall. Investigation of the hydrate plugging and non-plugging properties of oils. *Journal of Dispersion Science and Technology*, 31 (8):1100–1119, 2010.

P. Skovborg and P. Rasmussen. A mass transport limited model for the growth of methane and ethane gas hydrates. *Chemical Engineering Science*, 49(8):1131–1143, 1994.

E. Sloan, C. Koh, A. Sum, A. Ballard, G. Shoup, N. McMullen, J. Creek, and T. Palermo. Hydrates: state of the art inside and outside flowlines. *Journal of Petroleum Technology*, 61(12):89–94, 2009.

E. D. Sloan. A changing hydrate paradigm from apprehension to avoidance to risk management. *Fluid Phase Equilibria*, 228:67–74, 2005.

E. D. Sloan and C. Koh. *Clathrate hydrates of natural gases*. CRC press, 2007. ISBN 9781420008494.

E. D. Sloan, C. A. Koh, and A. Sum. *Natural Gas Hydrates in Flow Assurance*. Gulf Professional Publishing, 2010. ISBN 9781856179461.

M. Smith. *Neural networks for statistical modeling*. John Wiley & Sons, 1993. ISBN 0442013108.

N. Stander, W. Roux, T. Eggleston, and K. Craig. LS-OPT users manual–a design optimization and probabilistic analysis tool for the engineering analyst. Technical report, Livermore Software Technology Corporation, 2007.

K. E. Starling and J. L. Savidge. *Compressibility factors of natural gas and other related hydrocarbon gases*. American Gas Association, Operating Section, 1992.

E. Stenmark. On multiphase flow models in ANSYS CFD software. Master's thesis, Chalmers Universiy of Technology, Sweden, 2013.

R. Storn and K. Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

J. Subash, L. Dykhno, and J. Hudson. Challenges in pigging of subsea gas flowlines. Article presented at the *SPE Annual Technical Conference and Exhibition*, San Antonio, Texas, September 29 - October 2, 2002.

A. Svandal, B. Kvamme, L. Gràànàsy, T. Pusztai, T. Buanes, and J. Hove. The phase-field theory applied to CO2 and CH4 hydrate. *Journal of crystal growth*, 287(2): 486–490, 2006.

P. K. Swamee and A. K. Jain. Explicit equations for pipe-flow problems. *Journal of the hydraulics division*, 102(5):657–664, 1976.

Y. Taitel and A. Dukler. A model for predicting flow regime transitions in horizontal and near horizontal gas-liquid flow. *AIChE Journal*, 22(1):47–55, 1976.

J. Thaker and J. Banerjee. CFD simulation of two-phase flow phenomena in horizontal pipelines using openfoam. Article presented at the *Fortieth National Conference on Fluid Mechanics and Fluid Power*, Hamipur, India, December 12-14, 2013.

R. Thiele. Modeling of direct contact condensation with OpenFOAM. Master's thesis, KTH Royal Institute of Technology, 2010.

D. Tuhus-Dubrow and M. Krarti. Genetic-algorithm based approach to optimize building envelope design for residential buildings. *Building and environment*, 45 (7):1574–1581, 2010.

D. Turner, J. Boxall, S. Yang, D. Kleehammer, C. Koh, K. Miller, E. Sloan, Z. Xu, P. Matthews, and L. Talley. Development of a hydrate kinetic model and its incorporation into the olga2000® transient multiphase flow simulator. Article presented at the *5th International Conference on Gas Hydrates*, Trondheim, Norway, June 13 - 16, 2005.

D. J. Turner, G. Mahadevan, and J. Lachance. Hydrate stable start-up and restart of oil and gas production systems. Article presented at the *Twenty-fifth International Offshore and Polar Engineering Conference*, Hawaii, USA, June 21-26, 2015.

T. Uchida, T. Ebinuma, J. Kawabata, and H. Narita. Microscopic observations of formation processes of clathrate-hydrate films at an interface between water and carbon dioxide. *Journal of crystal growth*, 204(3):348–356, 1999.

P. M. Ujang. *Studies of slug initiation and development in two-phase gas-liquid pipeline flow*. PhD thesis, Imperial College London (University of London), 2003.

C. Vallée, T. Höhne, H.-M. Prasser, and T. Sühnel. Experimental investigation and cfd simulation of horizontal stratified two-phase flow phenomena. *Nuclear Engineering and Design*, 238(3):637–646, 2008.

A. Van der Velden and P. Koch. Isight design optimization methodologies. Technical report, Dassault Systéms, 2010.

F. Vasquez, M. Stanko, A. Vasquez, J. D. Andrate, and M. Asuaje. *Advances in Fluid Mechanics IX*, volume 74, chapter 6. WIT Press, Southhamption, UK, 2012.

M. Vaze and J. Banerjee. Experimental visualization of two-phase flow patterns and transition from stratified to slug flow. *Journal of Mechanical Engineering Science*, 225(2):382–389, 2011.

K. Volk, E. Delle-Case, and D. Estanga. Risk-based restarts of untreated subsea oil and gas flowlines in the GoM. Technical report, Tulsa University, 2007.

G. G. Wang and S. Shan. Review of metamodeling techniques in support of engineering design optimization. *Journal of Mechanical Design*, 129(4):370–380, 2007.

A. Wilson, S. J. Overaa, and H. Holm. Ormen lange-flow assurance challenges. Article presented at the *Offshore Technology Conference*, Houston, Texas, May 3-6, 2004.

K. S. Won and T. Ray. A framework for design optimization using surrogates. *Engineering Optimization*, 37(7):685–703, 2005.

R. A. Worthen and R. Henkes. CFD for the multiphase flow splitting from a single flowline into a dual riser. Article presented at the *17th International Conference on Multiphase Production Technology*, Cannes, France, June 10-12, 2015.

J. Xiao, O. Shonham, and J. Brill. A comprehensive mechanistic model for two-phase flow in pipelines. Article presented at the *SPE Annual Technical Conference and Exhibition*, New Orleans, September 23-26, 1990.

N. Xiong, D. Molina, M. L. Ortiz, and F. Herrera. A walk into metaheuristics for engineering optimization: principles, methods and recent trends. *International Journal of Computational Intelligence Systems, 2015*, 8(4):606–636, June 2015.

X.-S. Yang. *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons, 2010. ISBN 9780470582466.

G. H. Yeoh and J. Tu. *Computational techniques for multiphase flows*. Elsevier, 2009. ISBN 9780080467337.

C. S. Yih. *Fluid Mechanics: a consise introduction to the theory*. McGraw Hill, 1969. ISBN 9780960219001.