

Syed Aaquib Hazari

Turbulence Modelling of Two Phase Stratified Channel and Pipe Flows

Master of Science
Mechanical Engineering

This page is intentionally left blank

Turbulence Modelling of Two Phase Stratified Channel and Pipe Flows

(Sustainable Process Energy Technology)

Submitted in partial fulfilment of the requirements for the degree of

Master of Science

in Mechanical Engineering

By

Syed Aaquib Hazari

Student Number: 4330749

P&E Report Number: 2755

at the Delft University of Technology,

to be defended publicly on Wednesday June 29, 2016 at 10:00 AM.

Supervisor:	Prof. dr. ir. R.A.W.M. Henkes	TU Delft
Thesis committee:	Dr. D.R. van der Heul	TU Delft
	Dr. ir. M.J.B.M. Pourquie	TU Delft
	Ir. M.H.W. Hendrix	TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.



This page is intentionally left blank

Acknowledgements

I would like to take the opportunity to thank a few people who were either directly or indirectly involved in the completion of this project.

Firstly, my supervisor Prof. dr. ir. Ruud Henkes who introduced me to this topic and guided me from its inception to its conception. His invaluable experience and critique made certain that the quality of the work was high. Since I had no daily supervisor he took out time from his busy schedule and ensured we met on a weekly basis. These weekly discussions ensured that goals were met and I was not getting side-tracked. He always urged me to think critically, which helped me to become a better researcher. I thank him for these words of motivation.

There were times when I was confused and lost, however, and that was when dr. ir. Mathieu Pourquie stepped in and showed me the right path. My gratitude and thanks to him. His expertise on CFD software is outstanding and a good chunk of the results were obtained following his advice.

When I began this project it seemed like a vast subject, from turbulence to multiphase flows, from MATLAB to CFD. My immense gratitude goes to Gabriele Chinello, who was my predecessor in this research project. He is a brilliant individual and without his answers and key explanations I could still have been in the quest for results. I also thank him for having a few Skype meetings that answered most of my questions.

I also would like to thank Maurice Hendrix who has given me some valuable tools for post-processing my results and he guided me through the initial stages of using Fluent.

Furthermore, my everlasting gratitude is to all of my friends who were always around to cheer me up during our weekly dinners. Their advice was beneficial to the completion of this thesis. They are like my family away from home. Also a special word of thanks to my colleagues at the university who helped me to remain critical to my work and to make it better.

Finally, I want to thank The Almighty, my parents and my brother who have been of absolute importance during my entire Master study at the TU Delft. Their support and motivation during difficult times has been irreplaceable. I will be forever indebted to them.



This page is intentionally left blank

Abstract

Stratified two phase flow is one of the flow regimes that is of importance in multiphase flow transport through pipelines, such as used for example in the oil and gas industry. Its application also extends to chemical production, energy conversion and food processing. The phenomenon of turbulence further complicates the stratified flow behaviour. Having a simulation tool that accurately predicts the pressure gradient and liquid level in a turbulent channel or pipe flow can lead to better designs of multiphase flow systems.

The common RANS turbulence models (such as $k - \omega$ and $k - \varepsilon$) artificially produce too much turbulence at the liquid-gas interface. Therefore, these models need to be modified such that turbulent viscosity is sufficiently damped at the interface. To ensure this, in the present study the specific dissipation rate (ω) is imposed at the interface. Here ω is an appropriate function of the surface roughness factor (k_s), which represents the effect of interface waves.

The present work is a follow up to a previous research project where a MATLAB tool was developed for the prediction of stratified flow in channels. The first and main objective of the present thesis is to find and test a model for k_s and to apply that to obtain a modified version of the Standard $k - \omega$ (SKW) turbulence model and the Shear Stress Transport (SST) model. MATLAB can be used to find solutions for the channel flow. The simulation results are compared with experimental data. The second objective is to compare the results obtained using the turbulence models in the MATLAB model with predictions using similar models in Fluent. The third objective is to extend this study to a 3D setup of a two phase pipe flow where only the liquid phase is simulated. This is a so-called Segregated Liquid Phase (SLP) simulation. RANS predictions in Fluent are compared with experiments and DNS data. The main conclusions are:

- The calculation of k_s has been automated.
- The predictions of the flow rates for the experimental case by Fabre et al. are better than those for the experimental case by Akai et al.
- MATLAB predictions of the flow rates for the experimental cases are better than those of Fluent. This is because it is difficult to correctly impose an interface condition in Fluent, whereas this is straightforward in MATLAB.

Key Words: Channel Flow, Fluent, MATLAB, Pipe Flow, Stratified Flow, Turbulence Models, Liquid-Gas Interface

This page is intentionally left blank

Table of Contents

1. INTRODUCTION.....	1
1.1 Multiphase Flow.....	1
1.2 Nature of the Flow.....	2
1.3 Research Objectives	4
1.4 Research Approach	6
1.5 Research Outline	7
2. LITERATURE REVIEW.....	8
2.1 Literature Review	8
2.1.1 Experiments	8
2.1.2 Limitations of two phase turbulence models	9
2.2 Channel Flow Experiments	9
2.3 Pipe Flow Experiment.....	11
3. THEORY.....	12
3.1 Multiphase Flow.....	13
3.2 Turbulence.....	14
3.2.1 Baseline (BSL) Model	15
3.2.2 SST Model.....	18
3.3 Interface Correlations.....	20
3.3.1 Charnock Parameter	22
3.3.2 Models for Surface Roughness.....	22
4. MODELLING	24
4.1 MATLAB	24
4.1.1 Sequential Order of the Program	24
4.1.2 Grid.....	26
4.1.3 Discretisation	28
4.1.4 Boundary Conditions	29
4.1.5 Newton-Raphson Scheme.....	29
4.2 ANSYS Fluent	31
4.2.1 DesignModeler 2D	32
4.2.2 ANSYS Meshing 2D	32
4.2.3 Fluent 2D	34
4.2.4 UDF Description.....	35

4.3 ANSYS Fluent 3D.....	36
4.3.1 Geometry and Meshing	37
4.3.2 Simulation Setup.....	38
4.3.3 Segregated Liquid Phase (SLP) Case	39
5. RESULTS.....	42
5.1 Modifications to improve the channel flow results	42
5.1.1 Surface Roughness Results.....	42
5.1.2 Shear Stress Transport Model (SST)	49
5.1.3 Fluent versus MATLAB	51
SKW Results.....	51
Imposing the interface condition in Fluent.....	53
SST Results.....	55
5.2 Channel Flow Results	57
5.2.1 Akai results	57
5.2.2 Fabre results.....	59
5.2 Pipe Flow Results.....	61
6. CONCLUSION	64
6.1 Conclusions	64
6.2 Recommendations	67
6.3 Final Remark	67
Bibliography.....	68
Appendices	71
A. MATLAB Code	71
B. Fluent DEFINE_ADJUST UDF.....	81
C. Fluent DEFINE_PROFILE UDF	82

List of Tables

Table 1.1: $k - \omega$ vs. Modified $k - \omega$ (results taken from Chinello (2015), compared with the channel experiments by Fabre et al. (1987)).....	5
Table 2.1: Experimental results from Fabre et al. (1987) and Akai et al. (1980).....	11
Table 4.1: Coefficients and RHS for equation (23).....	29
Table 4.2: Grid dependence results.....	34
Table 4.3: Fluent parameters for the 2D simulations.....	34
Table 4.4: Specification for 3D cases in Fluent.....	38
Table 5.1: Simulation results compared with Run 250 by Fabre et al.; the prescribed superficial velocities are $V_{SL} = 0.15$ m/s and $V_{SG} = 2.27$ m/s.....	43
Table 5.2: Simulation results compared with Run 400 of Fabre et al.; the prescribed superficial velocities are $V_{SL} = 3.77$ m/s and $V_{SG} = 0.15$ m/s.....	44
Table 5.3: Simulation results compared with Run 600 of Fabre et al.; the prescribed superficial velocities are $V_{SL} = 5.93$ m/s and $V_{SG} = 0.15$ m/s.....	44
Table 5.4: Predictions with the Modified $k - \omega$ model vs the Modified SST model for the channel experiments by Fabre et al. (1987).....	49
Table 5.5: Flow rates (all in the unit m^3/s) for the experiments, and as found with MATLAB and Fluent.....	61

List of Figures

Figure 1.1: Typical flow pattern map of horizontal flow (Weisman, 1983).	2
Figure 1.2: (a). Smoke from a cigarette (Source: Christine Daniloff). (b). Turbulent gases from Mt. Cleveland (Source: Johnson Space Center). (c). Turbulent wingtip vortices (Source: Steve Morris, NY Times).....	3
Figure 1.3: Energy cascading (Davidson, 2004, page 19).....	4
Figure 3.1: Inclined channel flow.....	12
Figure 4.1: Flow chart of the MATLAB model (Method 1).....	25
Figure 4.2: Bi-exponential grid.....	27
Figure 4.3: ANSYS working sequence.....	31
Figure 4.4: Geometry for Run 250 of Fabre et al.(1987).....	32
Figure 4.5: Velocity profile comparisons for different grid sizes.....	33
Figure 4.6: Fluent grid.....	34
Figure 4.7: Sequential order of UDF execution in Fluent.....	36
Figure 4.8: Geometry of 3D cases.....	37
Figure 4.9: Meshing for 3D cases.....	38
Figure 4.10: Cross section of the SLP.....	40
Figure 5.1: Surface roughness modelling for the experimental cases by Fabre et al.....	45
Figure 5.2: The effect of changing k_s values on the velocity profiles. (a). Given pressure drop (b). Given flow rate (c). Non modified velocity prediction.....	47
Figure 5.3: Predictions for the velocity profile compared with the experiments by Fabre et al. (1987) (a). Run 250 (b). Run 400 (c). Run 600.....	48
Figure 5.4: Comparison of various profiles for the liquid and gas phases. (a), (b) Velocity (u), (c), (d) Turbulence Kinetic Energy (k) and (e), (f) Shear stress ($u' v'$) . Run 400 from Fabre et al. (1987).	51
Figure 5.5: Comparison of the velocity profile for Run 400, when using the standard $k - \omega$ model without modification.	52
Figure 5.6: Comparison of the turbulent kinetic energy profile for Run 400, when using the standard $k - \omega$ model without modification	52
Figure 5.7: (a). Run 400 in Fluent using the SKW model without and with interface condition, (b). Run 400 using the SKW model in MATLAB and Fluent with interface condition.....	53
Figure 5.8: Velocity profile with MATLAB and Fluent for Run 400 in the experiments of Fabre et al. 54	
Figure 5.9: Turbulent kinetic energy with MATLAB and Fluent for Run 400 in the experiments of Fabre et al.	55
Figure 5.10: Velocity profile with the SST model for Run 400.....	56
Figure 5.11: (a) ω for SKW case, (b) ω for SST case.....	56
Figure 5.12: Modified SST model applied to Run 2 in the experiments of Akai et al. (1980). Results for the liquid and gas layers. (a), (b) Velocity profiles (c), (d) Turbulent Kinetic Energy.....	58
Figure 5.13: Modified SST model applied to Run 400 in the experiments of Fabre et al. (1987). Results for the liquid and gas layers. (a), (b) Velocity profiles (c), (d) Turbulent Kinetic Energy.....	60
Figure 5.14: Streamwise velocity in the laminar pipe flow case.....	62
Figure 5.15: Streamwise velocity in the turbulent pipe flow case.....	63
Figure 5.16: Spanwise velocity in the turbulent pipe flow simulation.	63

Nomenclature

English Notation	Description	Units
dp/dx	Pressure gradient	Pa/m
g	Acceleration due to gravity	m^2/s
u	Velocity (transport variable)	m/s
h	Height of liquid level	m
k	Turbulent kinetic energy	m^2/s^2
k_s	Surface roughness factor	m
A	Cross sectional area	m^2
H	Height of channel	m
P	Pressure	Pa
Q	Flow rate	m^3/s
Re	Reynolds number	-
U	Bulk velocity	m/s
V	Volumetric flow rate	m^3/s
Greek symbols		
α	Void fraction	-
δ	RMS wave amplitude	m
ϵ	Turbulence dissipation rate	m^2/s^3
θ	Angle of inclination	°
μ	Dynamic viscosity	kg/(ms)
ν	Kinematic viscosity	m^2/s
ρ	Density	kg/m^3

τ	Shear stress	Pa
ω	Specific dissipation rate	s^{-1}
Δ	Difference between	-
Ω	Strain rate tensor	s^{-1}

Subscripts

i	Interface
j	Direction
k	Phase
w	Wall
G	Gas
L	Liquid

Abbreviations

1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
PDM	Pseudo Damping Mechanism
RMS	Root Mean Square
SKE	Standard $k - \epsilon$ Turbulence Model
SKW	Standard $k - \omega$ Turbulence Model
SST	Shear Stress Transport Model

INTRODUCTION

1.1 Multiphase Flow

Multiphase flow can be defined as the presence of multiple immiscible phases, having different chemical and physical properties, which are flowing together in a system. For example, water and gas or water and oil in two-phase flow or water, oil and gas in a three-phase flow. Solid particles can also be considered as one of the phases. For horizontal pipes/channels multiphase flows are broadly classified in various flow regimes: stratified flow, slug flow, annular flow and bubbly flow. The underlying phenomena that distinguish them from single phase flow are primarily related to the interfacial interactions that each phase experiences. Hence, multiphase flow is more complex than a single phase flow.

The prevailing flow regime in a pipe or a channel depends on a number of parameters such as, the volumetric flow rates of gas and liquid, the orientation of the pipe, the pipe diameter and the pressure level. The hatched areas that are shown in the example flow pattern map in Figure 1.1 give an approximation to where the flow patterns changes, and the solid lines are theoretical predictions. Since this thesis focuses on stratified flow in horizontal (or nearly horizontal) channels and pipes (i.e. without/with slight inclination) the discussion will only focus on that flow regime. As shown in Figure 1.1, stratified flow is found for low gas and liquid flow rates.

Stratified flow is important in many industrial applications. An example is multiphase flow transport of gas, oil and water in offshore and onshore pipelines as used in the oil and gas industry. The stratified flow is characterised by a sharp interface between the phases. The pressure drop along the pipe drives the flow, and this pressure drop needs to be sufficient to balance both the wall friction force and the gravity force. For the proper design of multiphase flows there is a need to accurately predict the pressure drop and the amount of liquid (so-called holdup) in the system.

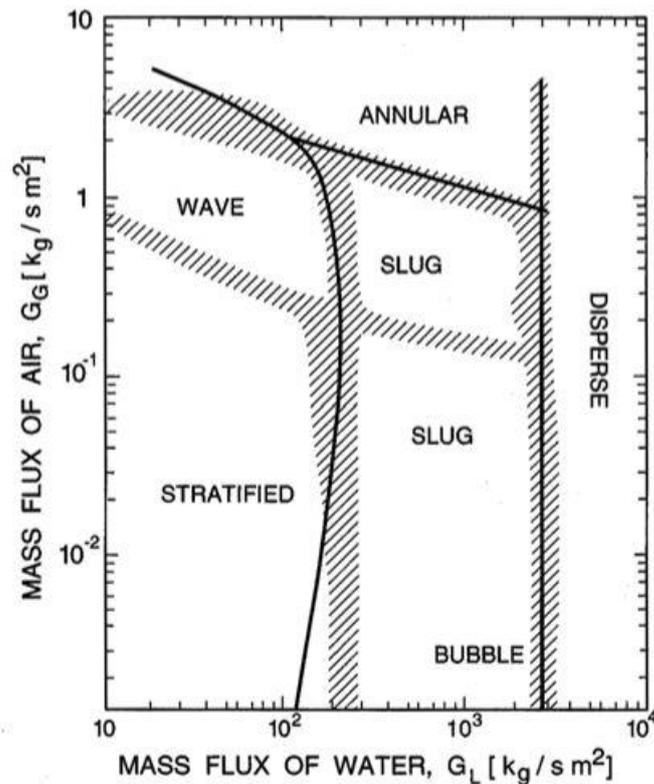


Figure 1.1: Typical flow pattern map of horizontal flow (Weisman, 1983).

1.2 Nature of the Flow

Fluid flow can either be laminar or turbulent, depending on the value of the Reynolds number, which marks the ratio of the inertial force to the viscous force. If the viscous forces dominate the flow, the flow will be laminar. On the other hand, if the inertial forces dominate the flow, the flow will be turbulent.

Laminar flow is a smooth streamlined flow where the fluid layers pass over one another like a deck of playing cards. It is characterised by diffusion and a low value of convection. Laminar flow, for specific cases, can be solved as an exact solution of the Navier Stokes equations. A clear example is blood flow through capillaries, where the velocity is very low and viscosity dominates the flow.

Turbulent flow, on the other hand, is highly irregular and often called ‘chaotic’. For some given initial conditions it is difficult to predict what the flow will be after a stipulated period of time. The longer the time period the more inaccurate the prediction will be. This is exactly why weather prediction is subject to uncertainty. Even with the most powerful supercomputers of today it is impossible to accurately predict turbulent flow for a large scale with a high enough resolution. This is because of the nonlinearity of the advection terms in the Navier Stokes equations.

Figure 1.2(a) shows the onset of turbulence as smoke rises from a cigarette. It starts in the laminar regime but shifts rapidly towards an irregular and chaotic turbulent flow. Figure 1.2(b) is a picture of the volcanic eruption of Mt. Cleveland. All the volcanic ash and gases expelled are highly turbulent. Figure 1.2(c) is the formation of turbulent vortices behind an airplane wing.

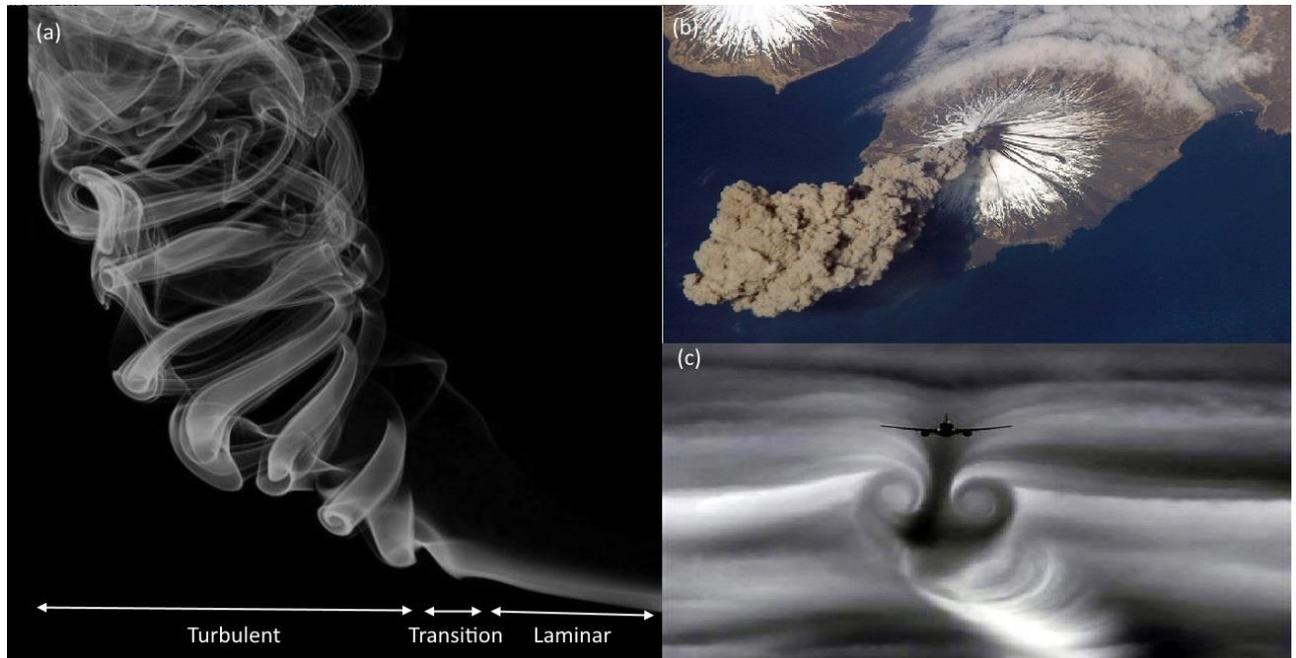


Figure 1.2: (a). Smoke from a cigarette (Source: Christine Daniloff). (b). Turbulent gases from Mt. Cleveland (Source: Johnson Space Center). (c). Turbulent wingtip vortices (Source: Steve Morris, NY Times).

How can one predict turbulence? To answer this question, it is necessary to understand what causes turbulence. Turbulence is characterised by turbulent eddies which exist in the macro scale and the micro (Kolmogorov) scale. The large scale eddies are of the width of the channel and the small scale eddies are of the Kolmogorov scale. The large scale eddies pass their kinetic energy to the smaller scale eddies (without loss of energy), all the way down to the Kolmogorov scale where the energy is released as heat due to viscous dissipation. This is known as the phenomenon of energy cascading as described by Lewis F. Richardson in 1922. Figure 1.3 is a cartoon showing the energy cascading.

Turbulence can be described by the 3-D Navier Stokes equations as a fluctuation of the flow variables. The fluctuation of the velocity (u'_k) is the difference of the total value (u_k) and the mean value (\bar{u}_k), where u_k is the characteristic flow velocity in any dimension:

$$u'_k = u_k - \bar{u}_k$$

When the fluctuation term replaces the velocity in the nonlinear advection term of the Navier Stokes equations a new term, called the turbulent shear stress term, is added. This is a fourth order tensor and closing this requires many simplifications.

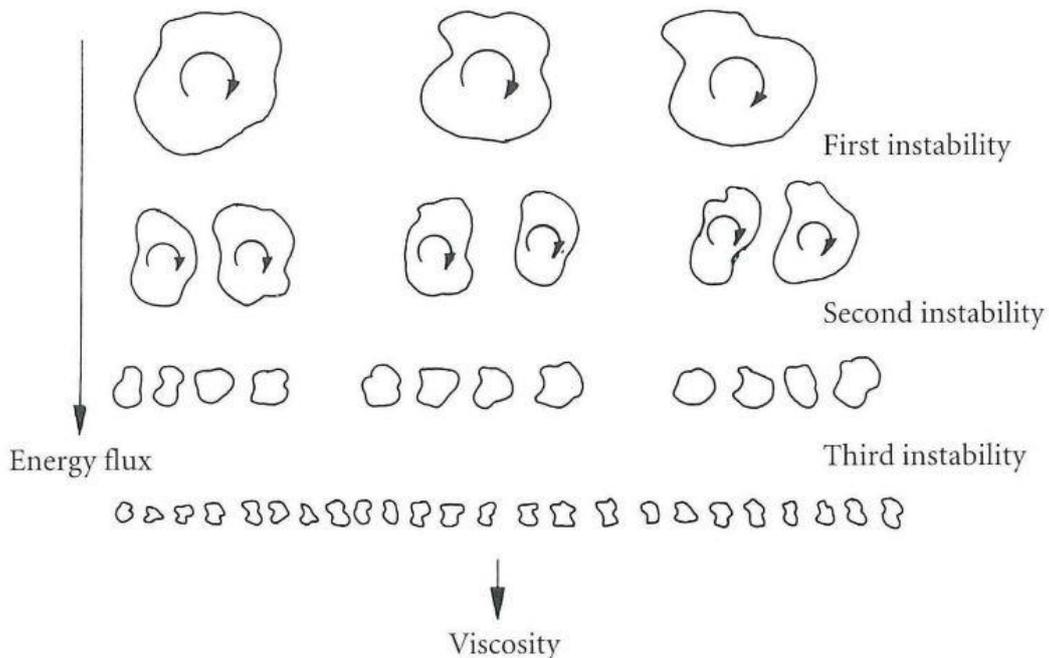


Figure 1.3: Energy cascading (Davidson, 2004, page 19).

The closure of the turbulent shear stress is covered by turbulence modelling. A less accurate, albeit computationally less intensive, method of approximation is based on solving the so-called Reynolds Averaged Navier Stokes (RANS) equations. The RANS equations contain mean values of the Navier Stokes equations and the additional turbulent shear stress tensor. Researchers such as Prandtl, Wilcox, Menter and many others proposed different approximation models for the RANS equations.

The present thesis focuses on the combination of stratified two phase flow and turbulence modelling.

1.3 Research Objectives

An in-house 1D code for the stratified two phase flow has been developed by a previous Master's student Gabriele Chinello (Chinello, 2015) at the University of Pisa in collaboration with TU Delft. This code can calculate the pressure drop and the liquid level of a two phase stratified channel flow for given values of the gas and liquid bulk flow rates or vice versa. The standard $k - \omega$ turbulence equations are solved iteratively and closure coefficients are chosen for the low Reynolds number case. Initially, for a given pressure drop and liquid level the

volumetric flow rates for gas were severely under predicted, and the specific dissipation rate was under predicted in the freestream and near the interface, while it was reasonably accurate near the walls. To overcome this, the slightly-rough Wilcox surface boundary condition was employed at the gas-liquid interface (Wilcox, 2006). Basically a Dirichlet boundary condition for ω was applied. This value was a function of the surface roughness k_s , where surface roughness or sand grain roughness defines how smooth or wavy/rough the interface is.

The results obtained by Chinello (2015) after this interfacial ω boundary condition was imposed were in close agreement with the experimental data from Fabre et al. (1987) as shown in Table 1.1. However, the value for the k_s was tuned manually for each of the experimental runs. This is a clear shortcoming of the method.

The addition of the interface condition improved the resulting predictions for the gas and liquid flow rates significantly, when using the experimental values for the pressure drop and liquid level as input. Table 1.1 shows the comparison of the model predictions before and after the addition of the interfacial boundary condition (Modified $k - \omega$). The predictions are compared with the experiments of Fabre et al. While the liquid flow rates are almost the same for all cases, the gas flow rates are under predicted for the standard $k - \omega$ case. However, the gas flow rates are over predicted by a maximum of 20%.

Table 1.1: $k - \omega$ vs. Modified $k - \omega$ (results taken from Chinello (2015), compared with the channel experiments by Fabre et al. (1987)).

	Experimental Data		Simulation Input		Standard $k - \omega$		Modified $k - \omega$	
	Gas Flow Rate (m^3/s)	Liquid Flow Rate (m^3/s)	Pressure Drop (Pa/m)	Liquid Level (m)	Gas Flow Rate (m^3/s)	Liquid Flow Rate (m^3/s)	Gas Flow Rate (m^3/s)	Liquid Flow Rate (m^3/s)
Run 250	0.0454	0.003	2.1	0.038	0.012	0.003	0.054	0.003
Run 400	0.0754	0.003	6.7	0.0315	0.021	0.003	0.078	0.0035
Run 600	0.118	0.003	14.8	0.0215	0.0367	0.003	0.116	0.003

To obtain this value of ω , however, the simulation model required a value of the surface roughness k_s that was tuned from case to case. The user needs to know beforehand whether the interface is smooth or wavy. This is a major drawback

of this approach. Therefore obtaining a value of k_s without ad-hoc tuning is the first (and main) objective of this research.

The second objective is to improve the accuracy of the results by comparing them with other turbulence models and with results obtained with the CFD tool Fluent. This will be done for the channel configuration.

To extend this study to a realistic scale, modelling must go from 2D to 3D. Therefore the third objective is to create a 3D model in Fluent for a pipe flow and to compare it with relevant experiments. An interesting way to model the liquid phase of a two phase flow would be to segregate the phases and impose the effects of the gas phase at the interface onto the liquid phase. In this way the concept of the Segregated Liquid Phase (SLP) model can be applied.

In summary, this research covers the following tasks for the modelling of two phase stratified liquid-gas flows in channels and pipes:

1. Derive and test a model for interfacial effective roughness (representing the waves) that does not require case-by-case tuning.
2. Implement various turbulence models in the MATLAB code solving the 1D equations in a channel.
3. Compare MATLAB and Fluent results.
4. Compare the model predictions for the channel with experimental data from Akai et al. (1980) and Fabre et al. (1987).
5. Calculate the Segregated Liquid Phase for a pipe in Fluent and compare the predictions with experimental data of Birvalski et al. (2014).

1.4 Research Approach

The software tools used for the 2D channel comparison are MATLAB version 2014(b) and ANSYS Fluent version 16.2. The modelling of k_s can be automated by using surface roughness and wave amplitude correlations available in the literature. Implementing these improvements and validating them with the experimental cases of Akai et al. (1980) and Fabre et al. (1987) is the following step.

To validate the 3D segregated liquid flow case, experimental data from Birvalski et al. (2014) and DNS data from the study of Mehdi Niazi (2014) is taken.

1.5 Research Outline

Chapter 2 discusses the relevant literature and the experiments. Chapter 3 looks into the theory and the model equations for stratified two phase flows and turbulence. Chapter 4 explores the modelling aspects and the simulation setup. Chapter 5 reports the channel and pipe flow results. Chapter 6 provides the conclusions and recommendations.

LITERATURE REVIEW

This chapter is divided into three subsections which deal with:

1. A short description of the relevant literature
2. The channel flow experiments
3. The pipe flow experiments

2.1 Literature Review

The literature available for stratified two phase flows can be split into experimental and modelling studies.

2.1.1 Experiments

A first important experimental study for two phase stratified flow was conducted by Akai et al. (1980). The two fluids are air and mercury flowing in a channel with an 18 mm height. The volumetric flow rates of the two phases were controlled and the resulting pressure drop and liquid level were measured. In the numerical simulations described in the same papers, the turbulence was modelled with a modified version of the low Reynolds number $k - \epsilon$ model by Jones & Launder (1972, 1973). The liquid holdup was imposed in the model by using an empirical correlation for the experimental data, using the Reynolds number of the gas layer. This empirical correlation was tuned to the specific experimental values of the gas and liquid phase Reynolds numbers:

$$\frac{\Delta h}{2} = 7.38 * 10^{-2} - 9.94 * 10^{-6} Re_G + 7.38 * 10^{-10} Re_G^2 \text{ (cm)}$$

$$Re_L = 8.04 * 10^{-3}, 5 * 10^{-3} < Re_G < 1.6 * 10^{-4}$$

Where the Reynolds numbers are based on the height and mean velocity of each phase. Such a correlation cannot be generalised to other flow conditions, which thus is a serious limitation. Another shortcoming of the model used by Akai et al. (1980) is mentioned by Issa (1988): the boundary condition for the dissipation rate at the interface is the same for both smooth and wavy

conditions. This is somewhat doubtful since the values at the interface must be different for the two conditions.

Fabre et al. (1987) also conducted experiments for stratified two phase channel flow. The phases used here were air and water. Results for three cases were reported, referred to as Runs 250, 400 & 600, each having a different gas flow rate and the same liquid flow rate. The interface for Run 250 is smooth, while the interface for Runs 400 and 600 is slightly wavy and completely wavy, respectively.

2.1.2 Limitations of two phase turbulence models

According to Lorencez et al. (1997) the gas flow over the liquid imposes a shear stress which consequently results in the formation of the interfacial waves. The research also highlighted that turbulent eddies are generated at the wall as well as at the interface. This implies that the interface behaves as a rough wall for the gas phase and a moving wall for the liquid phase. Do the regular turbulence models capture these eddies effectively?

Holmås et al. (2005) claimed that the regular turbulence models do not accurately predict the interfacial effects. In their simulations an upward shift of the gas phase velocity was noticed, which was due to an overestimation of the turbulence (e.g. reflected by the turbulent viscosity) at the interface. The liquid profile resembled that of a Couette flow for the same reason. Note that the lack of turbulence damping at the interface gives an asymmetric gas velocity profile, due to the fact that the turbulence is damped in the layer along the top wall but not in the layer along the interface. This gives a higher velocity close to the top wall than close to the interface. Their conclusion was that the present turbulence models cannot accurately predict the effect that the interface has on turbulence for the two phases. Hence there is a need to modify the turbulence models.

It is known that for the gas phase the liquid-gas interface appears as a rough wall and for the liquid phase it appears as a moving wall. Since the turbulence is being overpredicted, there is a need to dampen it at the interface. Wilcox (1998) has defined a boundary condition for slightly rough fixed walls in his low Reynolds $k - \omega$ model.

2.2 Channel Flow Experiments

The experimental cases that will be used for the validation of the simulations in this thesis are Fabre et al. (1987) and Akai et al. (1980).

Fabre et al. (1987) performed experiments in a channel that is 12.6 m long, 0.2 m wide and 0.1 m high and it has an inclination of 0.1%. The length of the channel ensures that the flow is fully developed.

The flow rate of air could be changed, while that of water was kept constant. The corresponding pressure drops and interface heights were measured. The three test cases or runs are shown in Table 2.1 where the interface is smooth for Run 250 and it is rough/wavy for Run 400 and for Run 600.

The existence of a secondary flow was expected since there was a nonzero vertical component of the velocity in the liquid and gas phase. However, this secondary flow was prevalent only in the wavy cases. The non-linearity of the shear stress profiles then proved the existence of transverse motion. The turbulence production was reported to be attenuated near the liquid wall and peaks in the gas phase were found very close to the interface.

The second set of experiments was obtained by Akai et al. (1980), who run experiments for a stratified mercury-air case. The channel was 3.6 m in length, 0.048 m in width and 0.018 m in height. The channel was kept perfectly horizontal implying that there was no streamwise effect of gravity. The strength of the shear stress at the walls was a third of that of the shear stress at the interface.

The gas flow rate was varied while the liquid flow rate remained constant. The pressure drop and liquid level were measured and presented in terms of the Lockhart-Martinelli correlation (1949).

Hanjalic & Launder (1972) conducted experiments for a single phase channel flow with a roughened wall at the bottom. The velocity profile experienced an upward shift compared to the flow in a channel with smooth walls. The velocity profiles in the wavy cases of Fabre et al. and Akai et al. also experienced an upward shift. This indicates that the interface acts as a rough wall.

Table 2.1: Experimental results from Fabre et al. (1987) and Akai et al. (1980).

	$Q_g \left(\frac{m^3}{s} \right)$	$Q_g \left(\frac{m^3}{s} \right)$	$\frac{dp}{dx} \left(\frac{Pa}{m} \right)$	$h \text{ (m)}$		$Q_g \left(\frac{m^3}{s} \right)$	$Q_g \left(\frac{m^3}{s} \right)$	$\frac{dp}{dx} \left(\frac{Pa}{m} \right)$	$h \text{ (m)}$
Fabre et al.	Run 250 (Smooth)				Akai et al.	Run 1 (Slightly Wavy)			
	0.0454	0.003	2.1	0.038		0.005	$4.2 \cdot 10^{-5}$	84.52	0.63
	Run 400 (Slightly Wavy)					Run 2 (Wavy)			
	0.0754	0.003	6.7	0.0315		0.007	$4.2 \cdot 10^{-5}$	154.3	0.54
	Run 600 (Wavy/Rough)					Run 3 (Wavy)			
	0.1187	0.003	14.8	0.0215		0.01	$4.2 \cdot 10^{-5}$	283.652	0.48

2.3 Pipe Flow Experiment

The pipe flow experiment of Birvalski et al. (2014) is important for this thesis. The pipe had an inner diameter of 0.05 m and a length of 10.3 m.

The two phases were air and water. Several cases were reported by them for the pipe flow, with varying roughness of the interface, and varying flow rates. The relevant cases are Case B and Case D. Case B is a laminar flow case with a smooth interface. Case D is a fully turbulent flow case. The flow rate of the liquid is high to ensure a turbulent liquid layer with a non-wavy interface. The experimental results of the Reynolds stresses, scaled with the friction velocity, compared well with experiments conducted for single phase flow.

These experiments have been compared with DNS conducted by Mehdi Niazi (2014) at the TU Delft. The DNS gave accurate results for the streamwise and spanwise velocities, but they showed nonzero values for the time-averaged crosswise velocity at the vertical centreline, which is incorrect. This could have been avoided if the DNS was run for a longer time period to obtain really fully converged statistics for the three velocity components. Note that the spanwise velocity is the vertical height direction, and the crosswise velocity is in the horizontal width direction.

THEORY

Figure 3.1 shows the two phase flow in a channel; (G) denotes the gas phase and (L) the liquid phase. The figure also shows the shear stresses (τ) acting on the walls (w) and the interface (i). The height of the liquid layer is h and that of the channel is H . The flow for both phases is in the same direction. The pressure drop along the channel is balanced by the wall shear stress and gravity force (in the presence of an inclination, denoted by θ).

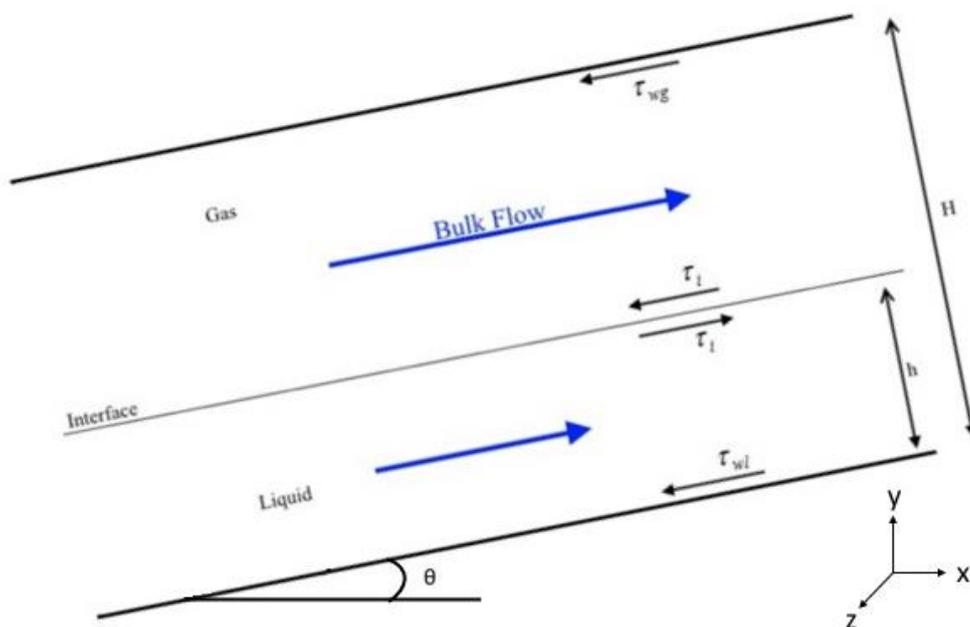


Figure 3.1: Inclined channel flow.

The relevant theory for this project involves the understanding of stratified two phase flow and of turbulence modelling.

3.1 Multiphase Flow

The following quantities are commonly used to describe or model two phase flow in channels or pipes:

1. Holdup fraction (α_k)
2. Superficial velocity (U_{sk})

The subscript s denotes “superficial” and k denotes the phase. The holdup fraction denotes the part of the volume that is occupied by each phase.

The superficial velocity of a phase is defined by:

$$U_{sk} = \frac{\dot{V}_k}{A}$$

Where, \dot{V}_k is the volumetric flow rate of the k^{th} phase and A is cross sectional area of the channel or pipe.

Modelling of the stratified flow can be done by combining the two force balances for each phase, i.e. of the gas (G) and of the liquid (L).

The force balance equations for the gas and liquid phase in a channel are:

$$-\alpha_G \frac{dp}{dx} = \frac{\tau_{wG}}{H} + \frac{\tau_i}{H} + \alpha_G \rho_G g \sin \theta \quad (1)$$

$$-\alpha_L \frac{dp}{dx} = \frac{\tau_{wL}}{H} - \frac{\tau_i}{H} + \alpha_L \rho_L g \sin \theta \quad (2)$$

In a 2D flow, $\alpha_G = (H - h)/H$ and $\alpha_L = h/H$.

Adding both the equations (1) and (2) gives the total pressure gradient across the channel

$$-\frac{dp}{dx} = \frac{\tau_{wG}}{H} + \frac{\tau_{wL}}{H} + (\rho_G \alpha_G + \rho_L \alpha_L) g \sin \theta \quad (3)$$

Note that the sum of the holdup fractions is 1. This equation implies that the driving force along the channel is balanced by the wall shear stresses and by the gravitational force. The interfacial shear stresses cancel each other out since they are in opposite directions.

The interfacial shear can be calculated by subtracting equations (1) and (2)

$$\frac{\tau_{wG}}{\alpha_G H} - \frac{\tau_{wL}}{\alpha_L H} + \frac{\tau_i}{\alpha_G \alpha_L H} - (\rho_L - \rho_G)g \sin\theta = 0 \quad (4)$$

The wall shear stresses can be expressed as

$$\tau_{wk} = \mu_k \frac{\partial u_k}{\partial y} \quad (5)$$

3.2 Turbulence

The theory behind the phenomenon of turbulence will be explained below, starting from the general Navier Stokes equations for single phase flow.

$$\frac{\partial u_i}{\partial x_i} = 0 \quad (6)$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j^2} + g_i \quad (7)$$

with $i, j = 1, 2, 3$

Any variable (ϕ) can be characterised as the sum of its mean value ($\bar{\phi}$) and fluctuating value (ϕ'). Replacing all flow variables with the mean term $\bar{\phi}$ and a fluctuating term ϕ' in equation (7) gives

$$\frac{\partial \bar{u}_i}{\partial t} + u_j \frac{\partial \bar{u}_i}{\partial x_j} = \frac{1}{\rho} \frac{\partial \bar{P}}{\partial x_j} + \frac{\partial}{\partial x_j} \left(\frac{\nu \partial \bar{u}_i}{\partial x_j} - \overline{u'_i u'_j} \right) + g_i \quad (8)$$

Where the extra fluctuating term ($\overline{u'_i u'_j}$) is the Reynolds Stress tensor. Equation (8) is known as the Reynolds Averaged Navier Stokes equation (RANS). To solve the RANS equation closure relations are needed for the Reynolds stresses. This is the area of turbulence modelling. Possible models are:

- Spalart-Allmaras model
- $k - \epsilon$ model
- $k - \omega$ model
- Shear Stress Transport model (SST)
- Reynolds Stress model

Each of these turbulence models has their own benefits and limitations. For this thesis the Standard $k - \omega$ model (SKW) and the $k - \omega$ SST model with low Reynolds number corrections are used.

The SST model is basically a combination of the SKW model and the Standard $k - \epsilon$ model (SKE). It uses blending functions to switch between either of the parent turbulence models. At the walls the $k - \omega$ model is switched on and in the free stream locations the $k - \epsilon$ model is switched on. Essentially, it reduces the drawbacks that both the parent turbulence models have. The $k - \omega$ model cannot accurately predict flow separation, while the $k - \epsilon$ model cannot accurately predict velocity gradients near a wall.

3.2.1 Baseline (BSL) Model

As explained by Menter (1993) the SST model follows from both the parent models (9) - (10) as described below. The BSL model is the basic combination of the SKW and SKE models.

Standard $k - \omega$ model

The kinetic energy and specific dissipation rate transport equations for the SKW are as follows

$$\frac{Dk}{Dt} = \frac{\tau_{ij}}{\rho} \frac{\partial u_i}{\partial x_j} - \beta^* \omega k + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_{k1} \nu_t) \frac{\partial k}{\partial x_j} \right] \quad (9)$$

$$\frac{D\omega}{Dt} = \frac{\gamma_1}{\nu_t} \frac{\tau_{ij}}{\rho} \frac{\partial u_i}{\partial x_j} - \beta_1 \omega^2 + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_{\omega 1} \nu_t) \frac{\partial \omega}{\partial x_j} \right] \quad (10)$$

Standard $k - \epsilon$ model

The kinetic energy and specific dissipation transport equations for the SKE are as follows. Note that actually the dissipation rate ϵ in the original equation has been replaced by $\epsilon = k\omega$, such that an equation for the specific dissipation rate ω results.

$$\frac{Dk}{Dt} = \frac{\tau_{ij}}{\rho} \frac{\partial u_i}{\partial x_j} - \beta^* \omega k + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_{k2} \nu_t) \frac{\partial k}{\partial x_j} \right] \quad (11)$$

$$\begin{aligned} \frac{D\omega}{dt} = & \frac{\gamma_2 \tau_{ij}}{\nu_t \rho} \frac{\partial u_i}{\partial x_j} - \beta_2 \omega^2 + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_{\omega 2} \nu_t) \frac{\partial \omega}{\partial x_j} \right] \\ & + 2\sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \end{aligned} \quad (12)$$

Equations (9) and (10) are multiplied by F_1 and equations (11) and (12) are multiplied by $1 - F_1$ and added. The resulting k and ω equations are:

$$\frac{Dk}{Dt} = \frac{\tau_{ij}}{\rho} \frac{\partial u_i}{\partial x_j} - \beta^* \omega k + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_k \nu_t) \frac{\partial k}{\partial x_j} \right] \quad (13)$$

$$\frac{D\omega}{Dt} = \frac{\gamma}{\nu_t \rho} \frac{\tau_{ij}}{\rho} \frac{\partial u_i}{\partial x_j} - \beta \omega^2 + \frac{\partial}{\partial x_j} \left[(\nu + \sigma_{\omega} \nu_t) \frac{\partial \omega}{\partial x_j} \right] + 2(1 - F_1) \sigma_{\omega 2} \frac{1}{\omega} \frac{\partial k}{\partial x_j} \frac{\partial \omega}{\partial x_j} \quad (14)$$

Where, the model constants $\gamma_1, \beta_2 \dots$ can be represented by ψ with

$$\psi = \psi_1 F_1 + \psi_2 (1 - F_1) \quad (15)$$

Equations (13) and (14) form the BSL SST model. The value of the blending function F_1 dictates whether the model behaves like $k - \omega$ (near wall) or like $k - \epsilon$ (free stream). Here, F_1 is

$$F_1 = \tanh(\theta_1^4)$$

Where θ is

$$\theta_1 = \min \left(\max \left(\frac{\sqrt{k}}{0.09 \omega z}, \frac{500 \nu}{z^2 \omega} \right); \frac{4 \rho \sigma_{\omega 2} k}{CD_{k\omega} z^2} \right)$$

In this expression z is the distance to the nearest wall/surface.

$CD_{k\omega}$ is known as the cross diffusion term and it is expressed as

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

Each of the terms on the RHS of the BSL equations are closed as described below.

Production Terms

The production term for the turbulent kinetic energy is

$$P_k = \overline{-u'_{ij}u'_{ij}} \frac{\partial u_i}{\partial x_j}$$

Where, $\tau_{ij} = \overline{-u'_{ij}u'_{ij}}$. According to the Boussinesq hypothesis

$$P_k = \nu_t S^2$$

Where S is the modulus of mean strain rate tensor.

The production term for the specific dissipation rate is

$$P_\omega = \frac{\gamma}{\nu_t} P_k$$

Dissipation Terms

For the kinetic energy:

$$Y_k = \beta^* \omega k$$

For the specific dissipation rate:

$$Y_\omega = \beta \omega^2$$

Diffusivity Terms

For the kinetic energy:

$$\Gamma_k = \nu + \sigma_k \nu_t$$

For the specific dissipation rate:

$$\Gamma_\omega = \nu + \sigma_\omega \nu_t$$

Where ν_t is the turbulent viscosity term that is a given by

$$\nu_t = \frac{k}{\omega}$$

Model Constants

For the $k - \omega$ model:

$$\begin{aligned} \sigma_{k1} &= 0.5 & \sigma_{\omega1} &= 0.5 & \beta_1 &= 0.075 & \beta^* &= 0.09 \\ \kappa &= 0.41 & \gamma_1 &= \frac{\beta_1}{\beta^*} - \sigma_{\omega1} \frac{\kappa^2}{\sqrt{\beta^*}} & & & & = 0.553 \end{aligned}$$

For the $k - \epsilon$ model:

$$\begin{aligned} \sigma_{k2} &= 1 & \sigma_{\omega2} &= 0.856 & \beta_2 &= 0.0828 & \beta^* &= 0.09 \\ \kappa &= 0.41 & \gamma_2 &= \frac{\beta_2}{\beta^*} - \sigma_{\omega2} \frac{\kappa^2}{\sqrt{\beta^*}} & & & & = 0.44035 \end{aligned}$$

3.2.2 SST Model

Menter et al. (2003) have revised the BSL model. This has changed the formulation of the turbulent viscosity, another blending function F_2 was introduced, and the original closure coefficients are replaced by low Reynolds number corrections.

Production Terms

After numerous calculations Menter et al. (2003) modified the production terms, where a production limiter was added to both the production terms of the k and ω equations.

The P_k term changes to

$$P_k = \min(P_k, 10\beta^* \omega k)$$

The P_ω term changes to

$$P_\omega = \min(P_\omega, 10\beta^* \omega k)$$

Viscosity Term

The expression for the shear stress as resulting from Bradshaw's assumption is shown below. In this assumption the shear stress is taken proportionally to the turbulent kinetic energy:

$$\tau = \rho a_1 k$$

According to the two equation models, the ratio of the production to the dissipation of kinetic energy governs the shear stress in a flow as shown in the expression below.

$$\tau = \rho \sqrt{\frac{\text{Production}_k}{\text{Dissipation}_k}} a_1 k$$

To satisfy Bradshaw's assumption the ratio of production to dissipation should be equal to 1. However, in adverse pressure flows the value of the production is much larger the dissipation. This leads to an incorrect behaviour. Therefore, in order to satisfy Bradshaw's assumption, the eddy viscosity should be

$$\nu_t = \frac{a_1 k}{\max(a_1 \omega; \Omega F_2)}$$

Where the maximum function in the denominator prevents the occurrence of a value of 0. The viscosity changes to the eddy viscosity assumption in one case and to Bradshaw's assumption in the other.

Here $a_1 = 0.31$,

$$\Omega = \sqrt{2\Omega_{ij}\Omega_{ij}}$$

Where Ω_{ij} is mean strain rate tensor.

F_2 is the second blending function denoted by

$$F_2 = \tanh(\theta_2^2)$$

Where θ_2 is

$$\theta_2 = \max\left(2\frac{\sqrt{k}}{0.09\omega y}; \frac{500\nu}{z^2\omega}\right)$$

The same definition of z carries over from equation (14).

The only change to the blending function of F_1 from the BSL to the SST is in the expression of its cross diffusion term. In the SST, the second term becomes 10^{-10} instead of 10^{-20} as shown below.

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

Boundary Conditions

Wall

At a stationary wall, all turbulent quantities become 0, with the exception of the boundary condition for ω :

$$\omega = 10 \frac{6\nu}{\beta_1(\Delta z)^2} \quad \text{at } y=0$$

Where Δz is the distance between the wall and the first inner grid point.

In the viscous sublayer however, we have:

$$\omega = \frac{6\nu}{\beta_1 z^2}$$

Other Constants

For the $k - \omega$ model:

$$\sigma_{k1} = 1.176 \quad \sigma_{\omega1} = 2 \quad \beta_1 = 0.075$$

$$a_1 = 0.31 \quad \gamma_1 = 5/9$$

For the $k - \epsilon$ model:

$$\sigma_{k2} = 1 \quad \sigma_{\omega2} = 1.168 \quad \beta_2 = 0.0828$$

$$\gamma_2 = 0.44$$

3.3 Interface Correlations

Wilcox Low Re $k-\omega$ model

The standard $k - \epsilon$ model (i.e. with low-Reynolds number modifications) cannot accurately predict what happens close to the walls, since the turbulent kinetic energy goes to zero and the value of ϵ tends to infinity. To overcome this, wall functions had been proposed by many researchers, like Spalding (1961), Shih et al. (1999), Nichols and Nelson (2004) and others. However, these wall functions are not always accurate and mostly grid dependent. The need to accurately predict turbulence near the walls has led to the introduction of $k - \omega$ model. Building on the studies of Kolmogorov (1942) and Saffman (1970), Wilcox (1998) made the $k - \omega$ model robust by adding low Reynolds

number correlations. Also, the advantage that the $k - \omega$ equation has over the $k - \epsilon$ model is that the value of ω can be arbitrarily specified at the surface. Hence, incorporating surface effects like roughness becomes rather easy.

In his low Reynolds number modification Wilcox proposed that

$$\omega = u_{\tau_i}^2 \frac{S_R}{\nu_k} \quad \text{at } y = 0$$

Where u_{τ_i} is the wall friction velocity and ν_k is the viscosity of the considered phase. ω is the specific dissipation rate and its value at the wall depends on the non-dimensionalized surface roughness height k_s^+ which is a function of the sand grain roughness or surface roughness k_s :

$$k_s^+ = \frac{u_{\tau_i}^2 k_s}{\nu_k}$$

$$S_R = \left(\frac{200}{k_s^+} \right)^2 \quad \text{if } k_s^+ \leq 5$$

$$S_R = \frac{100}{k_s^+} + \left[\left(\frac{200}{k_s^+} \right)^2 - \frac{100}{k_s^+} \right] e^{5-k_s^+} \quad \text{if } k_s^+ > 5$$

If $k_s^+ \leq 5$ the surface is considered to be almost smooth or slightly rough.

The boundary condition for slightly rough surfaces proposed by Wilcox (2006) is

$$\omega = \frac{40000\nu_k}{k_s^2} \quad \text{at } y = 0 \quad (16)$$

Equation (16) was indeed incorporated in the previous MATLAB code of Chinello (2015). Here it was also used as a condition at the liquid-gas interface, where the “surface wall roughness” now in fact represents the “interface waviness”. However, the limitation of this approach is that the value of k_s needs to be specified (estimated) for each case. Moreover, k_s has an SI unit of metre (i.e. it is not dimensionless). To improve its predictability, the code must be able to calculate the value of the surface roughness (interface waviness) using existing models. These models will be discussed below.

3.3.1 Charnock Parameter

As mentioned by Charnock (1955) the magnitude of roughness of ocean surface waves can be defined by a β parameter value which is dimensionless:

$$k_s = \beta * \frac{u_{\tau_i}^2}{g} \quad (m) \quad (17)$$

Where β is the Charnock Parameter, g is the acceleration due to gravity and u_{τ_i} is the friction velocity at the interface, which is defined as

$$u_{\tau_i} = \sqrt{\frac{\tau_{iG}}{\rho_G}}$$

Here, τ_{iG} is the shear stress at the gas side of the liquid-gas interface which is obtained by solving the momentum equation in the gas phase (1).

β is a dimensionless quantity and its values ranges from 0.36 to 1.05. According to Line et al. (1996) the higher the values of β , the deeper the ocean waves. There are different definitions of β given in the literature. For example, Espedal (1998) has used an additional factor of 30 which significantly reduces the value of β . Fitreman and Rosant (1981) have tested empirically that the value of β in formulation (17) for pipe flows ranges from 0.1-0.2 for smooth to wavy flows.

Although using Equation (17) resolves the issue of having a dimensional input parameter, it is left to the user to know beforehand whether the flow is smooth or wavy. Hence, in an effort to completely automate our simulation model literature from Oliemans (1987), Fernandez-Flores (1984) and Cohen & Hanratty (1968) has been studied.

3.3.2 Models for Surface Roughness

Taking ideas from the Charnock modification for interfaces, various researchers have derived their own definitions of the surface roughness (interface waviness), such as Cohen & Hanratty (1968), Fernandez-Flores (1984), Fitreman & Rosant (1981), Oliemans (1987), Baker & Gabb (1988) and Srichai (1994).

Cohen & Hanratty (1968) have shown that the surface roughness to mimic the interface can be given as a function of the root mean square of wave fluctuations, Δh :

$$k_s = 3 \sqrt{2} \Delta h \quad (m)$$

Fernandez-Flores (1984) specified another relation for the surface roughness, which was dependant on: the root-mean-square (RMS) wave fluctuations, the friction velocity at the interface, the gas phase viscosity ν_g and the hydraulic diameter D_{hG} :

$$k_s = \delta \sqrt{2 \left[1 - \left(\frac{56\nu_g}{u_{\tau i} D_{hG}} \right)^2 \right]}$$

This expression was formulated for pipe flows, but it can also be applied to channel flows.

Another model for stratified wavy flows was proposed by Oliemans (1987) in which the surface roughness is specified as

$$k_s = 3\sqrt{2} 2\delta \quad , 2\delta < h_l$$

$$k_s = 3\sqrt{2} h_l \quad , 2\delta \geq h_l$$

Where δ is the RMS wave amplitude. It is related to the wave amplitude h' according to:

$$\delta = \frac{h'}{\sqrt{2}}$$

Pots et al. (1988) describe an expression for the wave amplitude obtained by fitting the measured amplitude from the experiments carried out by Kordyban (1974). The maximum wave amplitude is 5.06 mm. The modelled wave amplitude (with the unit m) is dependent on the superficial gas velocity U_{Sg} and the interfacial velocity U_{int} :

$$h' = \frac{0.00506}{2} \left(1 + \tanh \left[\frac{(|U_{Sg} - U_{int}| - 4.572)}{1.2192} \right] \right)$$

MODELLING

4.1 MATLAB

MATLAB 2014b was used to solve the 1D code. The results were compared with experimental data from Fabre et al. (1987) and from Akai et al. (1980). The flow chart in Figure 4.1 describes the sequential order in which the numerical model is solved in MATLAB.

4.1.1 Sequential Order of the Program

Before proceeding to give an overview of the MATLAB model, we note that there are two ways to solve for the flow profile:

- Method 1: The gas and liquid flow rates are used as the specified conditions.
- Method 2: The pressure gradient and liquid level are used as the specified conditions.

Method 1 is most natural to simulate industrial applications and lab experiments where flow rates are imposed and the pressure gradient and liquid level/holdup are measured. MATLAB uses the Newton-Raphson iteration scheme to obtain the output. This means that first pressure gradient and liquid level are estimated and the flow rates follow as results. The estimate for the pressure gradient and liquid level is iteratively updated until the desired flow rates are found (within a certain convergence criterion).

Method 2 is computationally faster than Method 1, because there is no need for the root-finding as the values of flow rates are directly obtained as output when using the pressure drop and liquid level as the specified input conditions.

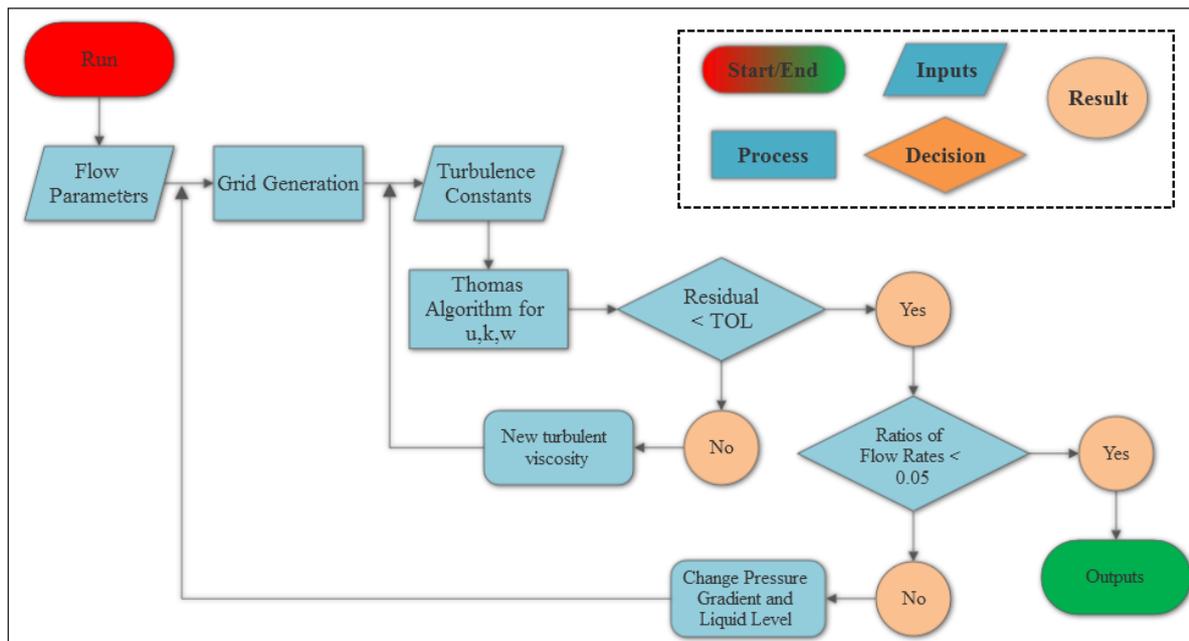


Figure 4.1: Flow chart of the MATLAB model (Method 1).

The following steps are part of the solutions procedure of Method 1:

1. The specified flow parameters are stored and estimated values for the pressure gradient and liquid level are used as input parameters to start the iteration process
2. Grid generation
3. Turbulence constants are defined
4. Transport equations are solved
5. Tolerance condition is checked. If the residuals are within the tolerance limit, proceed to 6. If not, use new values for the turbulence viscosity and solve the transport equations again until the convergence criterion is satisfied.
6. The flow rates obtained from the calculation are compared with the specified values. If the percentage difference is under the desired small limit the solution is found. If not, change the values for the pressure gradient and liquid level (using the Newton-Raphson method) and go back to step 2.

More details on the steps above are given in subsequent sections.

The flow profile is steady and fully developed; this means that there is no velocity component in the y-direction.

In Figure 4.1 it is assumed that the flow rates are specified. The first stage involves entering the input data: flow rates and physical properties of the gas and liquid (i.e. density, viscosity), and estimated values for the pressure gradient

and liquid level. In the next stage the bi-exponential grid is created; details on the grid will be given in section 4.1.2. The transport equations are solved on this grid in the following order: velocity u , turbulent kinetic energy k and specific dissipation rate ω . The Thomas Algorithm is used to solve the transport equations shown in section 4.1.3.

There are four loops: three inner loops for each transport equation and one outer loop to check the tolerance. The inner loops run over all the grid cells. The outer loop runs until either the specified maximum value has been reached or the residuals are under the pre-set value of the tolerance. Here, the tolerance is set as 10^{-5} . If the residuals are less than the tolerance, then the solution output is reported. Further details will be given in section 4.1.4.

Important output quantities include the pressure gradient, the liquid level, the profile of the transport variables along the height of the channel, and the value of the effective roughness at the liquid-gas interface.

4.1.2 Grid

To obtain accurate results at the walls and at the interface there is a need for a locally refined grid. Thereto a bi-exponential grid is used. This is a combination of two exponential grids: one for each phase. The refinement at each wall can be manually altered. Figure 4.2 shows the bi-exponential grid for a channel with a height of 0.1 m and an interface located at $y=0.038$ m.

The ratio between two successive grid cells for one phase, say the gas, is equal to

$$R = \frac{h_{j+1}}{h_j} = e^{\frac{10}{n_y}(\alpha_m - 0.5)}$$

Here, n_y is the total number of grid cells and α_m is the grid refinement variable that has two different values for the upper and lower half of each phase. In this way the size of each cell is related to the size of the first cell according to:

$$h_j = h_1 R^{j-1}$$

The total height of the gas layer is therefore given by

$$H - h = \sum_{j=1}^{n_y} h_j$$

The height of the gas layer in terms of the first cell is

$$H - h = h_1 \sum_{j=1}^{n_y} R^{j-1} = h_1 \frac{1 - R^{n_y}}{1 - R} \quad (18)$$

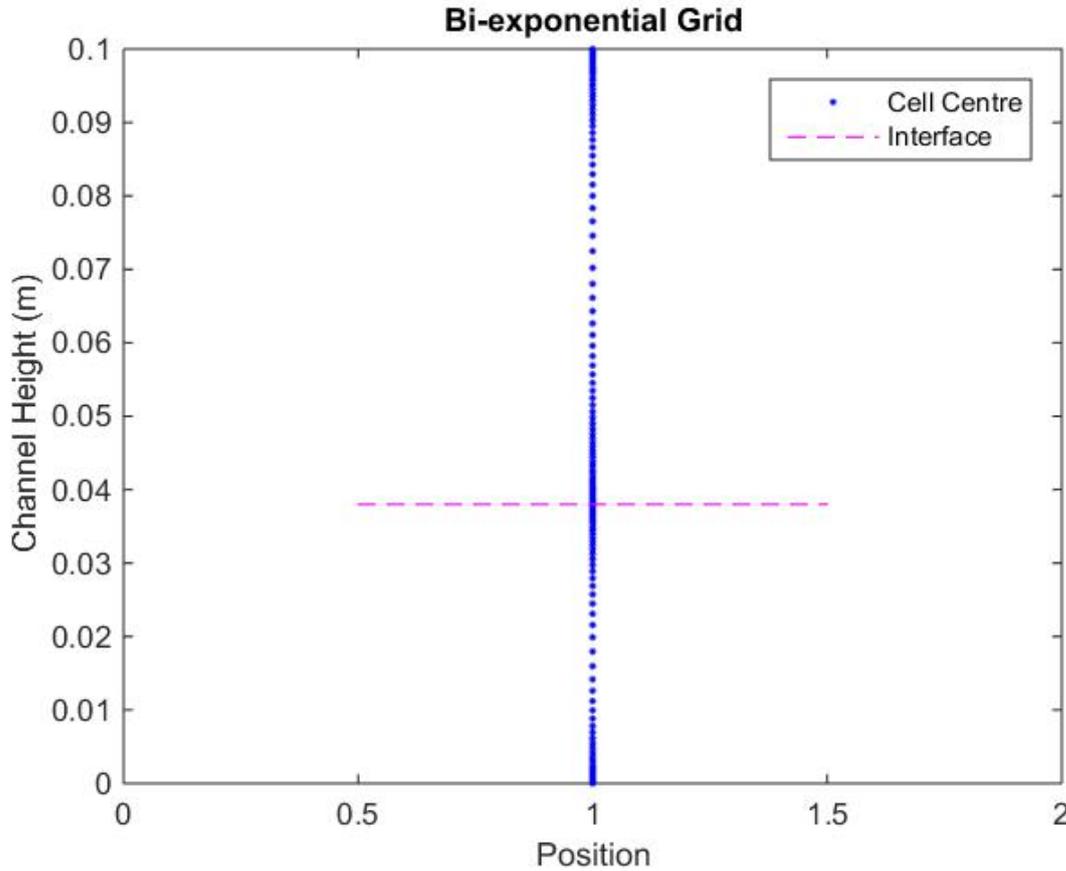


Figure 4.2: Bi-exponential grid.

The following steps have been implemented to create the grid:

- The value of R is calculated based on the number of cells and of the refinement parameter.
- The height of the first cell is calculated from $h_1 = (H - h) \frac{1-R}{1-R^{n_y}}$
- Subsequent heights of every cell can then be calculated from $h_j = h_1 R^{j-1}$
- The cell nodes are then $y_{j+1} = y_j + h_j$
- The cell centres are $y_{c,j} = y_{j-1} + (y_j - y_{j-1})/2$

Similarly, a second exponential grid with a different value of α_m is created for the liquid phase. Both of these grids are combined and therefore form the bi-exponential grid.

4.1.3 Discretisation

The discretisation scheme that is used is the finite difference method for a non-equidistant grid. The second order central differencing scheme is adopted.

$$\left(\frac{\partial\phi}{\partial y}\right)_j = \frac{\phi_{j+1} - \phi_{j-1}}{y_{j+1} - y_{j-1}} \quad (19)$$

Here ϕ is any flow parameter. Finite differences are also used to discretise the diffusion term ζ from $\left(\frac{\partial}{\partial y}\zeta\left(\frac{\partial\phi}{\partial y}\right)\right)$:

$$\left(\frac{\partial}{\partial y}\zeta\left(\frac{\partial\phi}{\partial y}\right)\right)_j = \frac{\zeta\left(\frac{\partial\phi}{\partial y}\right)_{j+\frac{1}{2}} - \zeta\left(\frac{\partial\phi}{\partial y}\right)_{j-\frac{1}{2}}}{y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}} \quad (20)$$

where j denotes the cell centre, and $j + \frac{1}{2}$ and $j - \frac{1}{2}$ denote nodes.

Substitution of eq. (19) into eq. (20) gives

$$\left(\frac{\partial}{\partial y}\zeta\left(\frac{\partial\phi}{\partial y}\right)\right)_j = \frac{\zeta_{j+\frac{1}{2}}\frac{\phi_{j+1} - \phi_j}{y_{j+1} - y_j} - \zeta_{j-\frac{1}{2}}\frac{\phi_j - \phi_{j-1}}{y_j - y_{j-1}}}{y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}}} \quad (21)$$

To iteratively solve the system of equations the Tri Diagonal Matrix Algorithm (TDMA), also known as the Thomas Algorithm, is used:

$$T\phi = RHS$$

Where T is the tri-diagonal matrix, ϕ is the flow variable and RHS is any source term or constant term.

Simplifying (21) results in a system of equations

$$\left(\frac{\partial}{\partial y}\zeta\left(\frac{\partial\phi}{\partial y}\right)\right)_j = a_j\phi_{j-1} + b_j\phi_j + c_j\phi_{j+1} \quad (22)$$

where the coefficients are

$$a_j = \frac{\zeta_{j-\frac{1}{2}}}{(y_j - y_{j-1})(y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}})}$$

$$b_j = -a_j - c_j$$

$$c_j = \frac{\zeta_{j+\frac{1}{2}}}{(y_{j+1} - y_j)(y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}})}$$

$$a_j \phi_{j-1} + b_j \phi_j + c_j \phi_{j+1} = RHS_j \quad (23)$$

To solve this system a two-step method with forward elimination and backward substitution is used.

Table 4.1 shows the values of each of the coefficients in the SST model.

Table 4.1: Coefficients and RHS for equation (23).

ϕ	Γ	b_j^*	RHS_j
u	$\nu + \nu_t$	$-a_j - c_j$	$\left[\frac{1}{\rho} \frac{\partial p}{\partial x} \right]_i$
k	$\nu + \sigma^* \nu_t$	$-a_j - c_j$	$\min \left[\nu_t \left(\frac{u_{j+1} - u_{j-1}}{y_{j+1} - y_{j-1}} \right)^2, 10\beta_j^* k_j \omega_j \right] + Y_k$
ω	$\nu + \sigma \nu_t$	$-a_j - c_j - \beta_j \omega_{j,old}$	$CD_{kw} + \min \left[\alpha_j \left(\frac{u_{j+1} - u_{j-1}}{y_{j+1} - y_{j-1}} \right)^2, 10\beta_j^* k_j \omega_j \right]$

4.1.4 Boundary Conditions

The following boundary conditions were included:

1. Periodic boundary conditions in streamwise direction.
2. The Wilcox (2006) slightly rough wall boundary condition at the interface.
3. A zero value of the turbulent quantities at the wall except for ω which is

$$\omega = 10 \frac{6\nu}{\beta_1 (\Delta z)^2} \quad \text{at the wall}$$

$$\omega = \frac{6\nu}{\beta_1 z^2} \quad \text{in the viscous sublayer}$$

The values for β_1 , Δz and z were given in Chapter 3.

Note that the $y^+ < 5$ for all simulations

4.1.5 Newton-Raphson Scheme

The Newton-Raphson iterative scheme is commonly considered as the fastest iterative root-finding method.

The convergence criterion that was applied in our study is

$$\frac{|Q_{G_{input}} - Q_G|}{Q_{G_{input}}} < 0.0005 \quad \text{and} \quad \frac{|Q_{L_{input}} - Q_L|}{Q_{L_{input}}} < 0.0005$$

The values for the gas and liquid flow rates are:

$$Q_G = W \int_h^{H-h} u \, dy \quad Q_L = W \int_0^h u \, dy \quad (24)$$

Where W is the width of the channel. As long as at least one of the conditions is not satisfied, then the estimate of the pressure gradient and of the liquid level is updated. This is done through calculating derivatives that are part of the Newton-Raphson procedure, which is as follows. First the pressure gradient is slightly disturbed according to

$$\left(\frac{dp}{dx}\right)_{disturbed} = \left(\frac{dp}{dx}\right)_{old} + C \quad (25)$$

Where C is a constant with a small value, chosen as 10^{-3} . The flow is solved again with this new pressure gradient term, whereas the liquid level remains at its old value.

After the flow is solved the partial derivatives of the flow rates with respect to the pressure gradient are stored:

$$\frac{\partial Q_G}{\partial p} \quad \text{and} \quad \frac{\partial Q_L}{\partial p} \quad (26)$$

Hereafter the value for the liquid level is disturbed as

$$h_{disturbed} = h_{old} + C \quad (27)$$

The flow is now recalculated using the disturbed h value and the old pressure gradient. The calculated partial derivatives of the flow rates with respect to the liquid level are stored

$$\frac{\partial Q_G}{\partial h} \quad \text{and} \quad \frac{\partial Q_L}{\partial h} \quad (28)$$

To obtain the real change in the pressure gradient and liquid level the following system of equations is solved

$$\begin{pmatrix} \frac{\partial Q_G}{\partial h} & \frac{\partial Q_L}{\partial h} \\ \frac{\partial Q_G}{\partial p} & \frac{\partial Q_L}{\partial p} \end{pmatrix} \begin{pmatrix} dh \\ dp \end{pmatrix} = \begin{pmatrix} Q_{G_{input}} - Q_G \\ Q_{L_{input}} - Q_L \end{pmatrix} \quad (29)$$

The values of dh and dp are calculated and herewith the old values of pressure gradient and liquid level are updated and used in the next iteration of the

Newton-Raphson scheme. This iterative process is continued until convergence is obtained. The MATLAB code is given in Appendix A.

4.2 ANSYS Fluent

The second software tool that is used for the comparisons is ANSYS Workbench 16.1. The geometry was constructed with the DesignModeler, the meshing was carried out with ANSYS Meshing, and the fluid flow calculations were performed with Fluent. Similar to the previous section a brief overview of the modelling of the channel flow and the pipe flow will be given here.

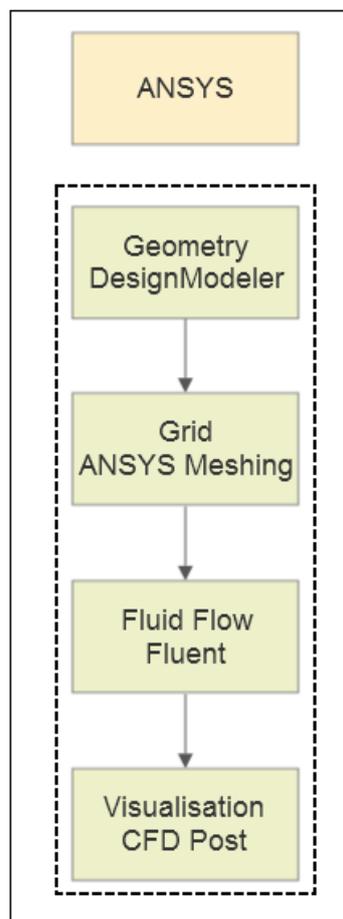


Figure 4.3: ANSYS working sequence.

Figure 4.3 is the workflow that is applicable to all the simulations with ANSYS both for the 2D or 3D simulations. Since Fluent is the main module, from now on the simulations in ANSYS will be referred to as ‘Fluent’ simulations.

The modelling for the 2D channel and for the 3D pipe flow will be discussed separately.

4.2.1 DesignModeler 2D

The DesignModeler is a convenient software tool to construct simple geometries. Using the sketching tool, a 2D channel flow geometry was built. Periodic boundary conditions were applied, and therefore the flow depends only on the vertical (height) coordinate. For the sake of convenience, the Fabre run 250 will be taken as an example. Here, the channel height is 0.1 m and the channel width is 0.2 m. There is also a very small inclination of -0.0572° .

The gas and the liquid region are separated at the interface, which was at a height of 0.038 m. This can be seen in Figure 4.4.

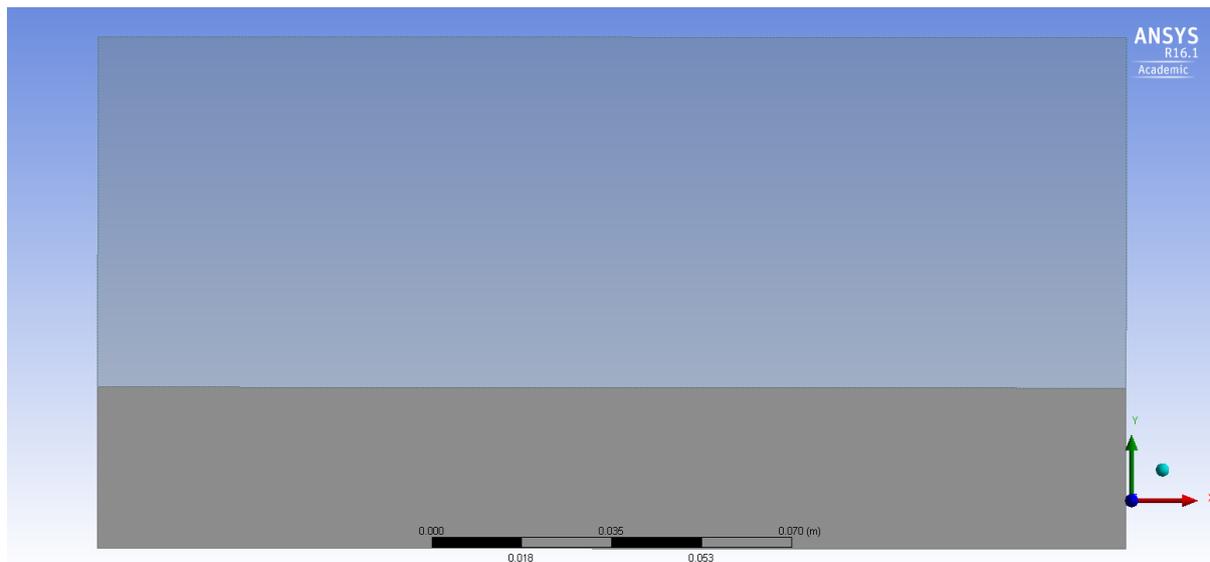


Figure 4.4: Geometry for Run 250 of Fabre et al.(1987)

4.2.2 ANSYS Meshing 2D

The channel grid is constructed with ANSYS meshing. Different grid sizes are tested to ensure grid independence of the results. The problem is essentially a 1D problem, since periodic boundary conditions eliminate the length coordinate. Hence, there are only two cells used along the length coordinate. The height coordinate has grid cells that are distributed with the bi-exponential distribution. This is to ensure an accurate representation of the shear stress at the interface and at the walls, while using the $k - \omega$ boundary conditions. The channel width is not included in the CFD simulation.

To ensure that the results are accurate, it is necessary to verify the grid dependence. Three cases with different grid cells are compared, using 100, 200 and 300 cells. The number of cells is only changed in the height direction, while the number of cells in the streamwise direction remains the same (namely 2).

The example case of Run 250 by Fabre et al. is shown in Figure 4.5, where the standard $k - \omega$ model is used. It is clear that the velocity profiles are in good agreement when using 200 and 300 cells. The use of 200 cells is made as a default choice for all the simulations since this gives a good accuracy at a fair computation time.

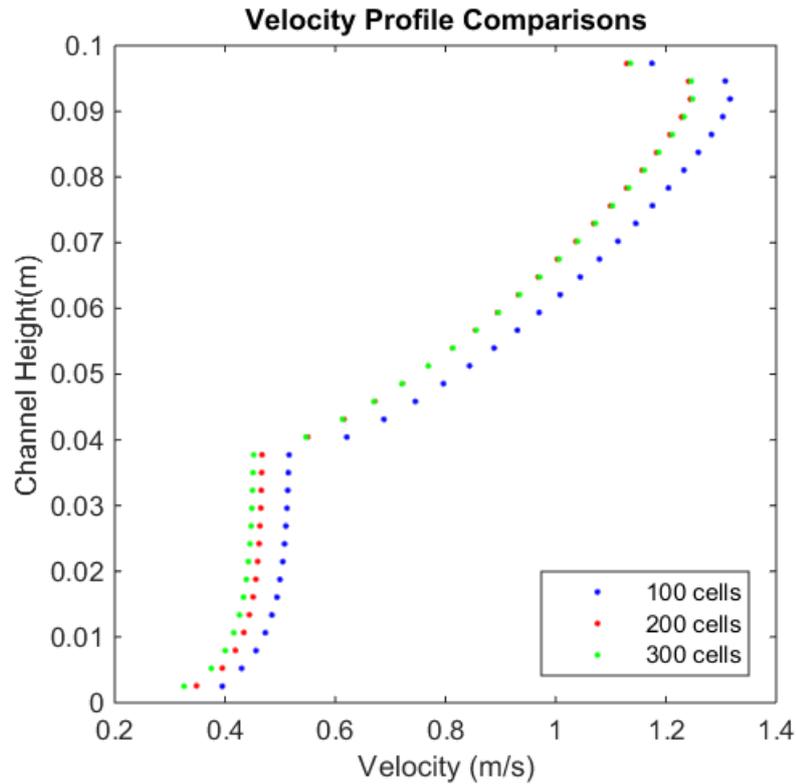


Figure 4.5: Velocity profile comparisons for different grid sizes.

The face meshing scheme and the bi-exponential grid can be seen in Figure 4.6. The decreased grid spacing in the figure marks the walls and interface. The y^+ values close to the walls are under 5, which is required for the $k - \omega$ turbulence models to run smoothly. The total flow rates and the y^+ values are also compared and they are summarized in Table 4.2.



Figure 4.6: Fluent grid.

Table 4.2: Grid dependence results.

Number of Grid Cells	Total Flow Rate [m^3/s]	y+ at top wall	y+ at bottom wall
100	0.38	4	5
200	0.31	1.5	3.5
300	0.32	1.5	4

4.2.3 Fluent 2D

The flow calculations are done in Fluent, using the steady-state, pressure-based solver. The Fluent results will be compared with the MATLAB simulations. The applied model details in Fluent for the 2D simulation are listed in Table 4.3.

Table 4.3: Fluent parameters for the 2D simulations.

Simulation Setup	Choice
Solver	Steady State, Pressure Based
Multiphase	VOF
Phases	Air & Water
Turbulence	<i>Standard $k - \omega$ & SST $k - \omega$</i>
Boundary Conditions	Periodic and Wall boundary
Discretisation Schemes <i>u, k, ω</i>	First Order Upwind
Pressure-Velocity Coupling	Simple
Under Relaxation Factors	Low

The Volume of Fluid (VOF) method can track the interface with a good accuracy. It uses the volume fraction of a phase to identify the location of the interface, and the interface is reconstructed using the compressive method.

Both the turbulence models Standard $k - \omega$ and $k - \omega$ SST are used. Low Reynolds number terms are switched on for a higher accuracy near the walls. Even though there is a small inclination of the channel (for the Fabre case) the effect of gravity is noticeable. Hence, gravity is switched on.

The periodic boundary conditions are prescribed at the inlet and the outlet of the channel. The pressure gradient is then imposed. When periodic boundary conditions are chosen in Fluent, it is not possible to impose flow rates, and instead only the pressure gradient and the liquid level can be imposed. This is a limitation of Fluent. Besides this, regular wall boundary conditions were imposed at the walls, with no slip and zero kinetic energy.

Since there is no advection there is no need for a second order discretisation scheme. At the default setting of the underrelaxation factors (URFs) the residuals did no longer reduce after a certain number of iterations, and they showed a non-converging oscillating behaviour. To ensure that convergence was reached, the URFs for the transport variables had to be reduced by factors ranging between 4 and 8.

Convergence is checked in two ways:

1. Residuals must be below 10^{-6} for all variables.
2. Verification of balance of pressure gradient with wall shear stress and gravity

$$\frac{dp}{dx} = \frac{dp}{dx_{grav}} + \frac{\tau_{walls}}{H} \quad (30)$$

4.2.4 UDF Description

The goal is to compare results with Fluent and MATLAB for the two phase model with the SKW turbulence model. This requires using the same interface condition in Fluent as is used in MATLAB. To enable imposing this interface condition in Fluent the UDF functionality needs to be used.

Any Fluent program can be customised to a certain degree by the user with the help of User Defined Functions (UDFs). For our purposes the UDF should dampen the turbulent viscosity at the interface. This can be done by adding a large ω source term. Another way might be to tweak the ω profile.

DEFINE_SOURCE can add a source term to the transport equations and DEFINE_PROFILE can tweak the profile. Yet another way can be to adjust the value of ω_i at the interface at every iteration through using DEFINE_ADJUST.

Initially all of them seemed to be promising options. However, upon closer evaluation the first two had some drawbacks. In the case of DEFINE_SOURCE, a source term would be added to the entire domain and not just at the interface. The case of DEFINE_PROFILE is incompatible with the application of periodic boundary conditions. The definition of DEFINE_PROFILE states that an initial entry profile can be described for any of the transport variables. But this turns out to be only possible with regular boundary conditions.

In this thesis the DEFINE_ADJUST UDF is used to impose a value of ω at the interface similar to what MATLAB does. The different locations where the UDFs are called in a Fluent program can be seen in Figure 4.7. DEFINE_ADJUST is called at every iteration and the flow variables (velocity, turbulent kinetic energy etc.) are modified accordingly. The UDF is given in Apependix B

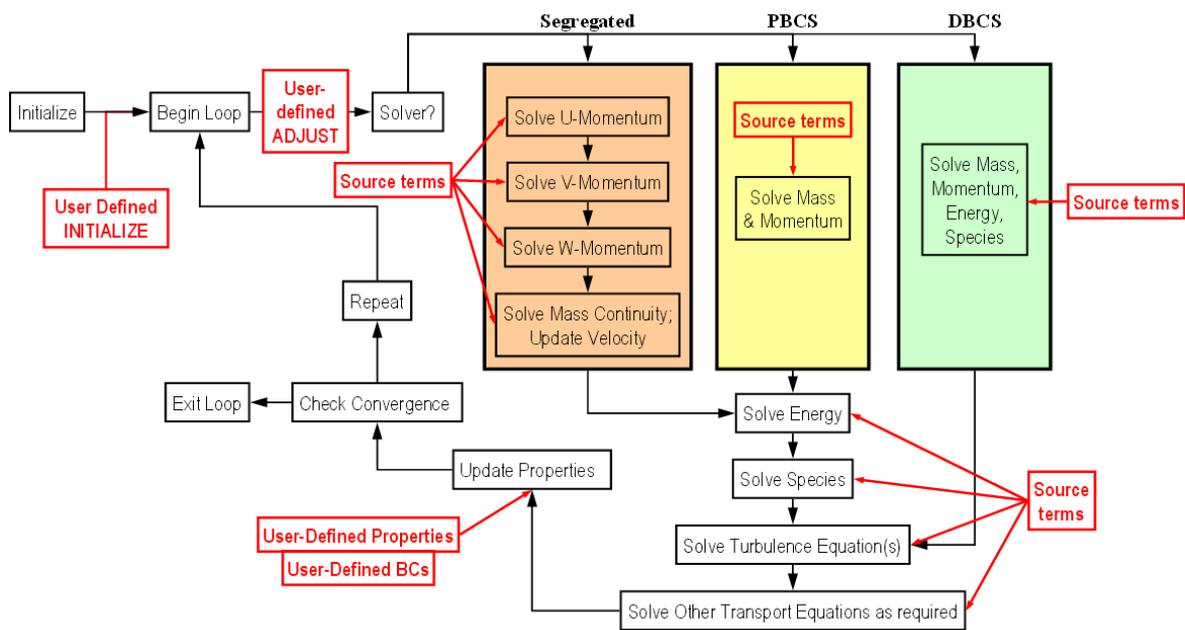


Figure 4.7: Sequential order of UDF execution in Fluent.

4.3 ANSYS Fluent 3D

The simulations conducted in Fluent are compared with the experimental cases of Birvalski et al. (2014) and DNS results of Mehdi Niazi (2014). The RANS simulations with Fluent are only carried out for the liquid layer having a prescribed shear stress at the top wall. This is similar to what was done in the

DNS simulations of Mehdi Niazi. These simulations are called the segregated liquid phase (SLP) simulations.

What follows is the geometrical and meshing description of the simulation setup.

4.3.1 Geometry and Meshing

In the geometry shown in Figure 4.8 the diameter of the horizontal pipe is 50 mm and the height of the horizontal SLP section is 18 mm. The structured geometry has a similar square as used in the corresponding mesh shown in Figure 4.9. This is done using ANSYS Meshing. The streamwise direction in Figure 4.8 is the positive z direction.

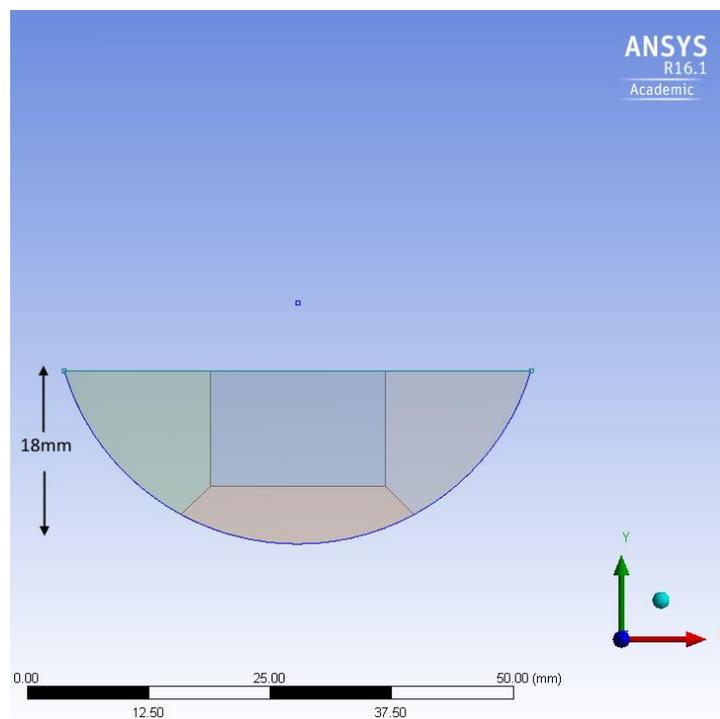


Figure 4.8: Geometry of 3D cases.

The magnitude of the dimension in the streamwise direction is not important, since periodic boundary conditions are used.

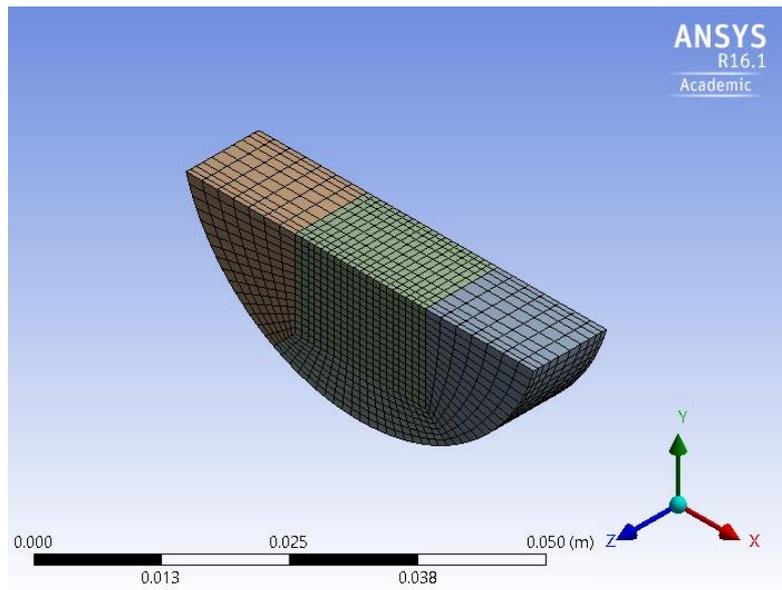


Figure 4.9: Meshing for 3D cases.

4.3.2 Simulation Setup

The Fluent simulation setup for the 3D case is similar to the Fluent 2D case. The specifications for the 3D case are given in Table 4.4.

Table 4.4: Specification for 3D cases in Fluent.

Simulation Setup	Choice for SLP
Solver	Steady State, Pressure Based
Multiphase	n/a
Phase	Water
Turbulence	Standard $k - \omega$
Boundary Conditions	Periodic, Specified Shear & Wall boundary
Discretisation Schemes u, k, ω	Second Order Upwind
Pressure-Velocity Coupling	Simple
Under Relaxation Factors	Default

Periodic boundary conditions were imposed at the inlet and the outlet, using the water flow rate as input for the SLP case. A specified shear stress is included at the top boundary of the computational domain. The convergence criterion is set to 10^{-9} for all variables.

4.3.3 Segregated Liquid Phase (SLP) Case

The Segregated Liquid Phase means that the liquid phase is separated from the pipe and is simulated by imposing the shear stress. This shear stress is actually in balance with the wall shear stress for the gas layer on the top wall of the pipe and the pressure gradient along the gas layer.

Theoretically speaking, for a stratified flow this simulation method for the Segregated Liquid Phase should be possible as long as there are no waves at the interface. The waves created at the interface move at about the same speed as the liquid layer. Therefore, for the liquid layer it appears as if these waves do not exist. However, for the gas layer the waves move at a different velocity. This is also why a ‘Segregated Gas Phase’ case cannot be simulated in the same way.

Previously, DNS simulations were conducted by a Master Student, Mehdi Niazi, (Niazi, 2014), at TU Delft. This simulation considered the SLP case mentioned above. These simulations were compared with the experimental results of Birvalski et al. (2014). Details of the experimental setup and of other related quantities are given below.

The experimental setup consisted of a pipe that had a radius of 25 mm and the height of the liquid layer is 17.31 mm counted from the bottom of the pipe to the interface. The cross section of the numerical domain is shown in Figure 4.10. Here, a is the height of the liquid level, b is the half the length of interface (the top wall of the computational domain), R is the radius of the pipe and x is the horizontal distance counted from the centre of interface (the top wall of the computational domain).

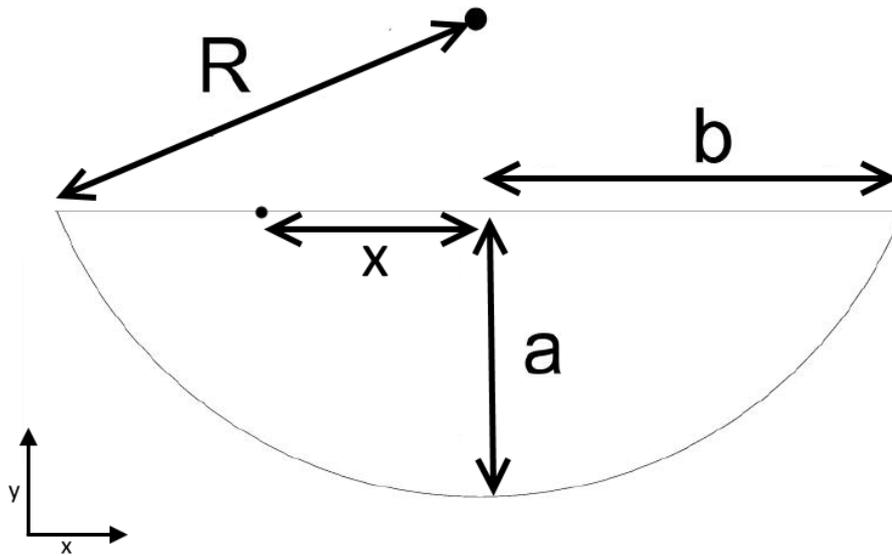


Figure 4.10: Cross section of the SLP.

The experimental interfacial shear stress as reported by Birvalski et al. (2014) is

$$\tau_i = 0.0107 \text{ N/m}^2$$

Therefore, the shear velocity can be given as follows

$$u_{\tau_i} = \sqrt{\frac{\tau_i}{\rho}} = 0.00327 \text{ m/s}$$

The Reynolds number as used by Niazi is as follows

$$Re_{\tau} = \frac{au_{\tau_i}}{\nu} = 54$$

Where a is the height of the liquid layer and ν is the kinematic viscosity of water.

The flow rate of the water was 5 lit/min.

Periodic boundary conditions were imposed at the inlet and outlet, with the given flow rate and a guessed value for the pressure gradient.

The top wall was defined as a moving wall with a shear that is specified as follows:

$$\tau\left(\frac{x}{b}\right) = \begin{cases} \tau_i & , \quad \frac{x}{b} < 0.9 \\ \tau_i \times \left[-100 \left(\frac{x}{b}\right)^2 + 180 \left(\frac{x}{b}\right) - 80\right] & , \quad \frac{x}{b} \geq 0.9 \end{cases} \quad (31)$$

This relation holds for positive x values. For negative x values x should be replaced by $-x$.

This ensures a constant wall shear stress for most of the interface and a parabolically decreasing value when one gets closer to the ends. This shape is added to Fluent using a UDF. The UDF is given in Appendix C.

Some important assumptions to be noted are:

1. Transfer of momentum across the interface occurs only by means of viscous diffusion.
2. No vertical movement at the interface, i.e. $u_y = 0$.

The model was run with DNS for two cases, namely a laminar case and a turbulent case. However, at this Re_τ of 54, the flow is in the transitional regime. The turbulence at this Reynolds number was difficult to simulate. Every simulation seemed to end up in the fully laminar regime. Therefore, Niazi started with a Re_τ of 162, which was completely turbulent, and then reduced the Re_τ down to 54. This trick ensured that the DNS simulations ended up with turbulence.

This is, however, not of importance to our simulations, as these use the RANS equations. This essentially ensures the presence of turbulent eddies at Re_τ of 54.

RESULTS

5.1 Modifications to improve the channel flow results

As mentioned in Chapter 2, the surface roughness factor is imposed at the interface. Note that making this model independent of a user-input value for the surface roughness was one of the goals of this thesis. The value of ω_i is calculated from the S_R expression mentioned in Chapter 2. What follows is the comparison and testing of different models from the literature, including Charnock (1955), Cohen & Hanratty (1968), Fernandez-Flores (1984) and Oliemans (1987).

The experimental cases from Fabre et al. (1987) will be used as the test cases to show the improvements in the modelling.

5.1.1 Surface Roughness Results

We started with the implementation of the Charnock parameter to calculate the value of the surface roughness. The relation of the surface roughness in the Charnock parameter is shown below.

$$k_s = \frac{\beta u_{\tau_i}^2}{g} (m)$$

Where the friction velocity is $u_{\tau_i} = \sqrt{\frac{\tau_{iG}}{\rho_G}}$. The value of the shear stress at the interface is obtained from the pressure drop equation along the pipe, applied to the gas layer.

β is a dimensionless quantity and its value ranges from 0.36 to 1.05.

Next, we use the models given by Cohen & Hanratty (1968), Fernandez-Flores (1984) and Oliemans (1987) to represent the surface roughness. These models are functions of the wave amplitudes. Cohen & Hanratty (1968) and Fernandez-Flores (1984) obtained their expression through experiments and empirical

calculations for channel flows. Oliemans (1987) on the other hand obtained the value of the surface roughness through calculations for pipe flows.

Tables 5.1, 5.2 and 5.3 show the formulae for each of these models for the different cases of Fabre et al. (1987). The predictions with the model of Charnock (1955) give the closest comparison to the experimental values for the pressure drop and liquid level, followed by the Oliemans (1987) model. The Fernandez-Flores (1984) model does not give a close comparison, mainly because of the large under prediction of the surface roughness for all cases.

The simulations were carried out with the MATLAB tool using the standard $k - \omega$ model.

Table 5.1: Simulation results compared with Run 250 by Fabre et al.; the prescribed superficial velocities are $V_{SL} = 0.15$ m/s and $V_{SG} = 2.27$ m/s.

	Experiment	Charnock	Cohen & Hanratty	Fernandez-Flores	Oliemans
Quantity	Run 250	$k_s = \frac{\beta u_{\tau_i}^2}{g} (m)$	$k_s = 3\sqrt{2}\delta (m)$	$k_s = \delta \sqrt{2 \left[1 - \left(\frac{56v_G}{u_{\tau_i} * D_{hg}} \right)^2 \right]} (m)$	$k_s = 3\sqrt{2} \cdot 2\delta$ $2\delta < h_l$ $k_s = 3\sqrt{2} \cdot h_l$ $2\delta \geq h_l$ (m)
Pressure Gradient (Pa/m)	2.1	1.54	1.42	1.4	1.43
Liquid Level (m)	0.038	0.0363	0.0365	0.0363	0.0366
Surface Roughness (m)	5E-4	3.5E-3	3.38E-4	7.8E-5	4.7E-4

Table 5.2: Simulation results compared with Run 400 of Fabre et al.; the prescribed superficial velocities are $V_{SL} = 3.77$ m/s and $V_{SG} = 0.15$ m/s.

	Experiment	Charnock	Cohen & Hanratty	Fernandez-Flores	Oliemans
Quantity	Run 400	$k_s = \frac{\beta u_{\tau_i}^2}{g} (m)$	$k_s = 3\sqrt{2}\delta (m)$	$k_s = \delta \sqrt{2 \left[1 - \left(\frac{56v_G}{u_{\tau_i} * D_{hg}} \right)^2 \right]} (m)$	$k_s = 3\sqrt{2} \cdot 2\delta$ $2\delta < h_l$ $k_s = 3\sqrt{2} \cdot h_l$ $2\delta \geq h_l$ (m)
Pressure Gradient (Pa/m)	6.7	5.2	3.4	3.13	3.53
Liquid Level (m)	0.0315	0.0291	0.0328	0.0335	0.0325
Surface Roughness (m)	0.0154	0.0152	3.1E-3	8E-4	4.2E-3

Table 5.3: Simulation results compared with Run 600 of Fabre et al.; the prescribed superficial velocities are $V_{SL} = 5.93$ m/s and $V_{SG} = 0.15$ m/s.

	Experiment	Charnock	Cohen & Hanratty	Fernandez-Flores	Oliemans
Quantity	Run 600	$k_s = \frac{\beta u_{\tau_i}^2}{g} (m)$	$k_s = 3\sqrt{2}\delta (m)$	$k_s = \delta \sqrt{2 \left[1 - \left(\frac{56v_G}{u_{\tau_i} * D_{hg}} \right)^2 \right]} (m)$	$k_s = 3\sqrt{2} \cdot 2\delta$ $2\delta < h_l$ $k_s = 3\sqrt{2} \cdot h_l$ $2\delta \geq h_l$ (m)
Pressure Gradient (Pa/m)	14.8	13.08	10.7	7	11.6
Liquid Level (m)	0.0215	0.0213	0.0233	0.028	0.022
Surface Roughness (m)	0.018	0.0157	0.0126	4E-3	0.0154

The formula for the RMS wave amplitude δ is obtained from Pots et al. (1988) as mentioned in Chapter 3.

The values for the pressure drop and liquid level predicted by the model of Charnock (1955) are closest to the experiment, but that model applies a value for β that is user-specified. The three values for β were chosen as low, medium and high: 0.39, 0.66 & 0.97. Where low, medium and high corresponded to a smooth, a slightly wavy and a rough interface Berthelsen & Ytrehus (2005). This means that the procedure cannot be automated and it is still left to the user to know beforehand whether the interface should be wavy or smooth. Therefore, the results obtained from the other three models are more valuable in terms of the accuracy of the physics.

The comparison of the surface roughness for each of these models is shown in Figure 5.1. The values shown in the magenta circles represent the experimental values of the gas flow rate for each of the Runs 250, 400 & 600. There is a large underprediction of the value of surface roughness for all the models for the Runs 250 & 400. This results in an underprediction of the gas flow rate as well. The predictions for Run 600 are better than for the other runs. This is because the surface roughness models are known to give better predictions when the gas flow rate increases (see Espedal, 1998). All models give a plateau level for the surface roughness (wave height) at high gas flow rates.

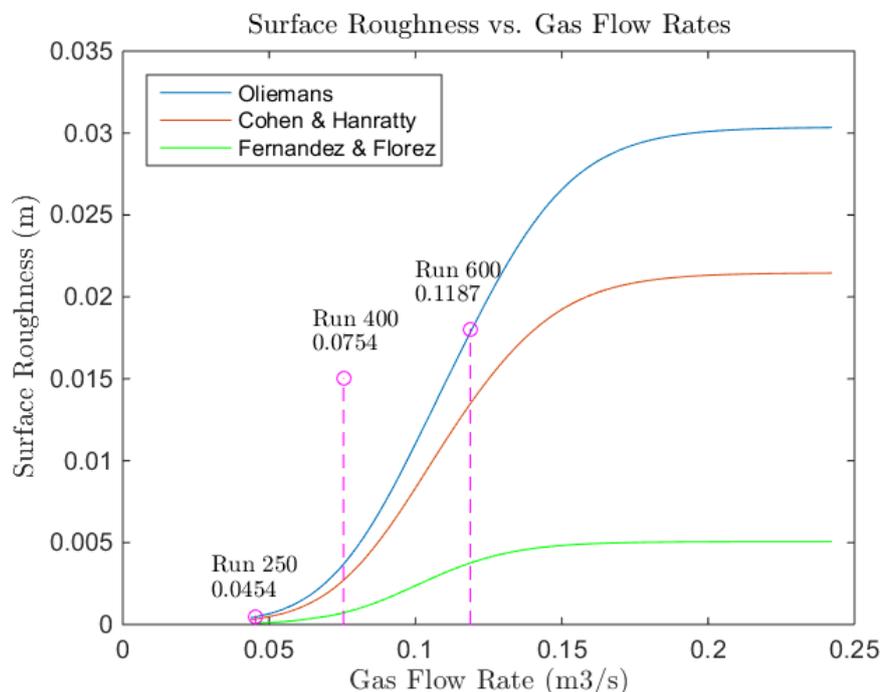


Figure 5.1: Surface roughness modelling for the experimental cases by Fabre et al.

All the models underpredict the amplitude of the interface waves (represented as surface roughness). This thus leads to smaller waves in the models than in the experiments. This gives a too low turbulent viscosity and a too high value of the specific dissipation rate. This gives a too low pressure drop in the

simulations. The inaccurate prediction of the ω_i thus shows why the pressure drop is not the same as in the experiments.

Note that the “MATLAB” & “Fluent” models (SKW & SST) are called Modified MATLAB models. Modified means that the Wilcox (2006) ω_i value is being imposed at the interface. Henceforth, all Modified Models will be prefixed with ‘M’ and all non-modified models will be prefixed with ‘NM’.

A sensitivity analysis for Run 600 was carried out by varying the value of k_s . The predicted velocity profiles are shown in Figure 5.2. The pressure drop and liquid level were fixed in Figure 5.2 (a). The gas and liquid flow rates were fixed in Figure 5.2 (b). Figure 5.2 (c) is a regular prediction of two phase flow, without the Wilcox (1998) modification at the interface. The value of k_s is increased starting from 0.0005 m (blue profile) and ending at 0.0805 m (red profile). The red profiles of 5.2 (a) and (b) look similar to that of 5.2 (c). This shows that the NM SKW model produces excess turbulence at the interface and results in a wavy/rough interface. The following can be concluded from this sensitivity analysis:

1. For the same pressure drop, an increased wave height at the interface gives a smaller gas throughput. This is as expected. Vice versa, a fixed gas flow rate will give more pressure drop if the wave height is increased.
2. The asymmetry in the gas velocity profile increases with increasing height of the interface waves. This is shown by the movement of the location of the velocity maximum to the upper wall when k_s is increased. This is because the upper wall remains smooth whereas the apparent roughness at the interface increases. The boundary conditions thus become more and more asymmetric for increasing k_s .

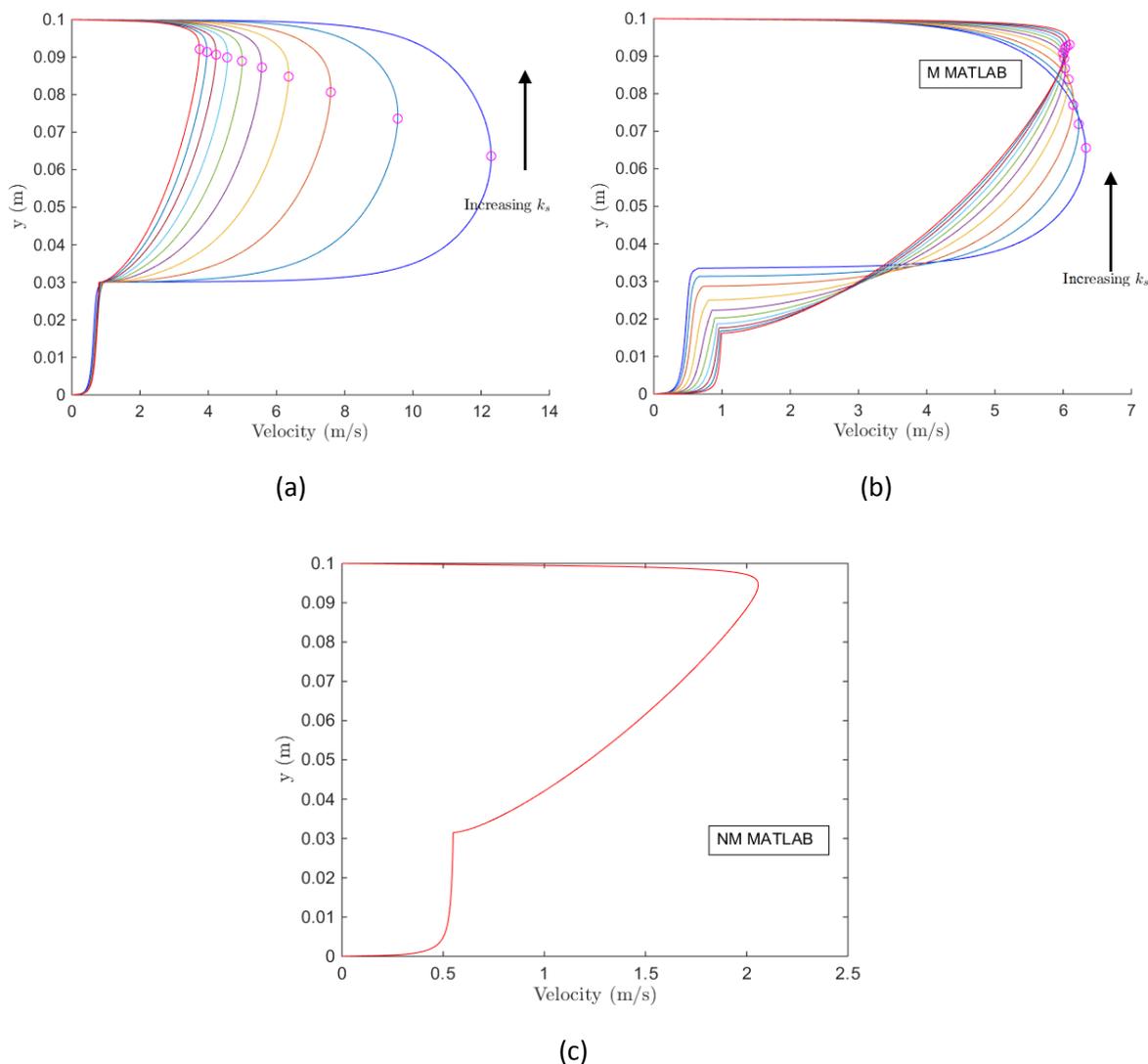


Figure 5.2: The effect of changing k_s values on the velocity profiles. (a). Given pressure drop (b). Given flow rate (c). Non modified velocity prediction

The predicted velocity profiles, using various correlations for the wave height at the interface (surface roughness) are shown in Figure 5.3. While the predicted velocities of the liquid phase are in good agreement with the experiments, the velocities of the gas phase show considerable deviations. Since the interface is smooth for Run 250 the surface roughness effect is not significant. The gas throughput measured in Run 250 is higher than in the simulations. This is because of the finite width of the channel which gives higher velocities at the centre plane and lower values close to the side walls. For the wavy cases (i.e. Run 400 and Run 600) the velocity profiles have a rather gradual shape, while the experiments show a clear peak. This difference in shape is due to the overprediction of the turbulence at the interface, which is due to a too low prediction of the amplitude of the waves at the interface. The relatively larger waves in the experiments reduce the gas velocity at the interface side whereas

larger velocities are found for the gas along the upper wall. This leads to the asymmetric profile with the peak in the experimental velocity profile.

As the flow rate increases the model of Oliemans (1987) gives a better prediction of the velocity profile. In general, the model of Oliemans (1987) gives slightly more accurate results than the other two models and therefore this will be considered as the baseline model for the forthcoming simulations.

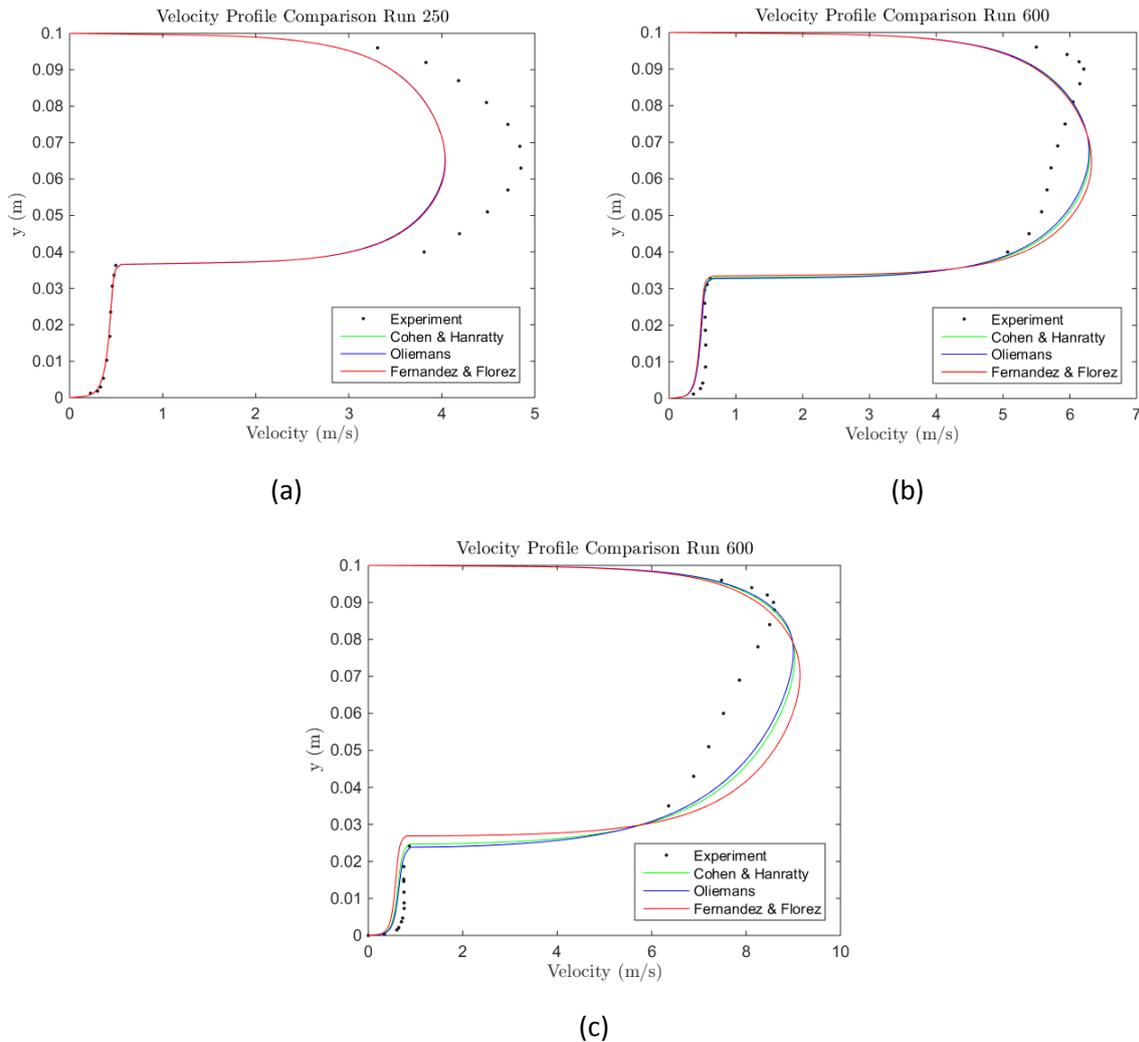


Figure 5.3: Predictions for the velocity profile compared with the experiments by Fabre et al. (1987) (a). Run 250 (b). Run 400 (c). Run 600.

In comparison to the previous model of Chinello (2014), the new simulation model is now more accurate in terms of calculating the value of the surface roughness (amplitude of interface waves). However, there are still remarkable differences between the simulations with the standard $k - \omega$ model and the experiments. This means that there is room for further improvement. Therefore, the next step is to test another turbulence model, which is the SST model.

5.1.2 Shear Stress Transport Model (SST)

The SST model combines both of the well-known RANS turbulence models; $k - \omega$ and $k - \epsilon$. It uses the superior capabilities of the $k - \omega$ model to predict the near wall turbulence and it switches to the $k - \epsilon$ model at the free stream. This is done by the use of hyperbolic tan functions, also known as blending functions. The value of the blending function depends on the distance to the nearest wall. Near the walls it is 1 and in the free stream it is in between 0 and 1. If this distance from the nearest wall is too high, then the flow is said to be in the free stream. According to this, the liquid-gas interface is in the free stream. But physically the interface seems to behave as a rough wall for the gas layer. Therefore, we require a kind of near wall modelling also close to the interface. To impose such near wall modelling at the interface, the values of the blending function are set to 1 at the interface.

The SST is modified by imposing a boundary condition for ω , which depends on the roughness / wave amplitude k_s , using the correlation of Oliemans (1987). This is similar to what has been used in the modified SKW model.

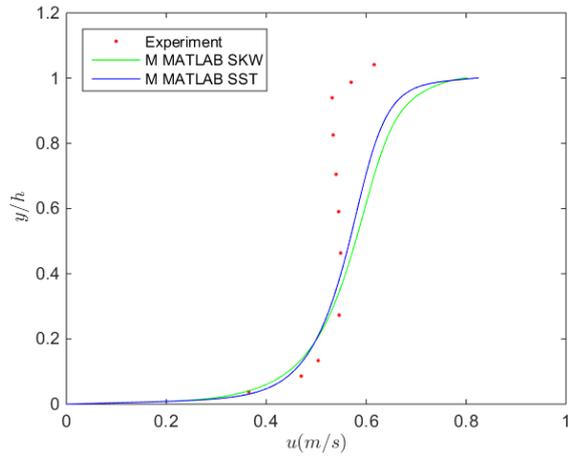
The gas and liquid flow rates are chosen as inputs and the resulting predictions for the pressure gradient and liquid level are compared in Table 5.4. The pressure gradients and liquid levels predicted by the Modified SST model are closer to the experimental values than the predictions with the Modified SKW model.

Table 5.4: Predictions with the Modified $k - \omega$ model vs the Modified SST model for the channel experiments by Fabre et al. (1987).

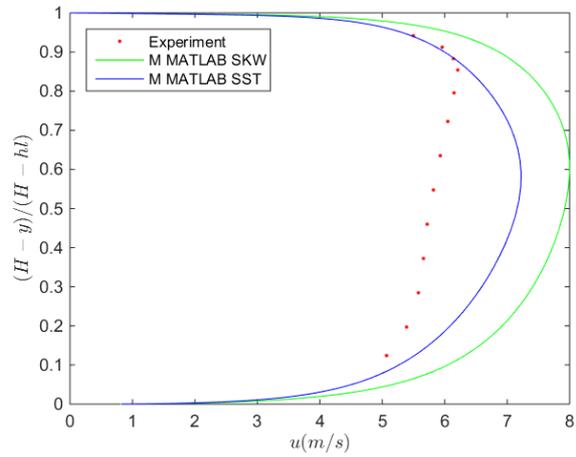
Run	Experiment	Experiment	Modified $k - \omega$		Modified SST	
	dp/dx (Pa/m)	h (m)	dp/dx (Pa/m)	h (m)	dp/dx (Pa/m)	h (m)
Run 250	2.1	0.038	1.43	0.0365	1.8	0.036
Run 400	6.7	0.0315	3.45	0.033	4.2	0.032
Run 600	14.8	0.0215	11.6	0.0225	14.6	0.021

Using the modified SST model (instead of the modified SKW model) gives a considerable improvement in the accuracy of the predictions of the velocity profile of the gas phase; see Figure 5.4(b). The velocity profiles are slightly different due to the different definition of turbulent viscosity. While the turbulent kinetic energy of the liquid layer is almost identical, there is a minor

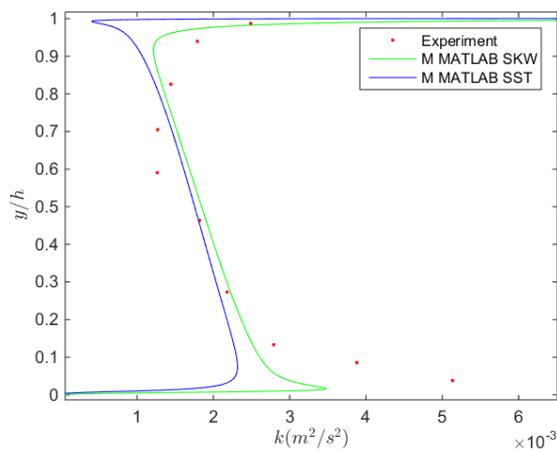
variation in the gas layer, both close to the wall and close to the interface. The reason for this is that the SST model has an extra production limiter term for the transport equation of k . There is a negligible difference between both models for the prediction of the shear stress profiles.



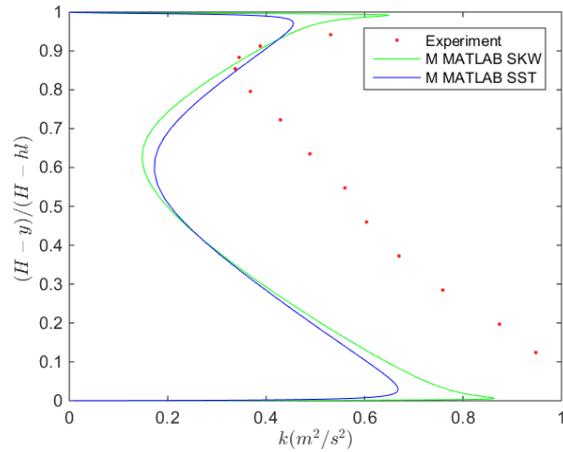
(a)



(b)



(c)



(d)

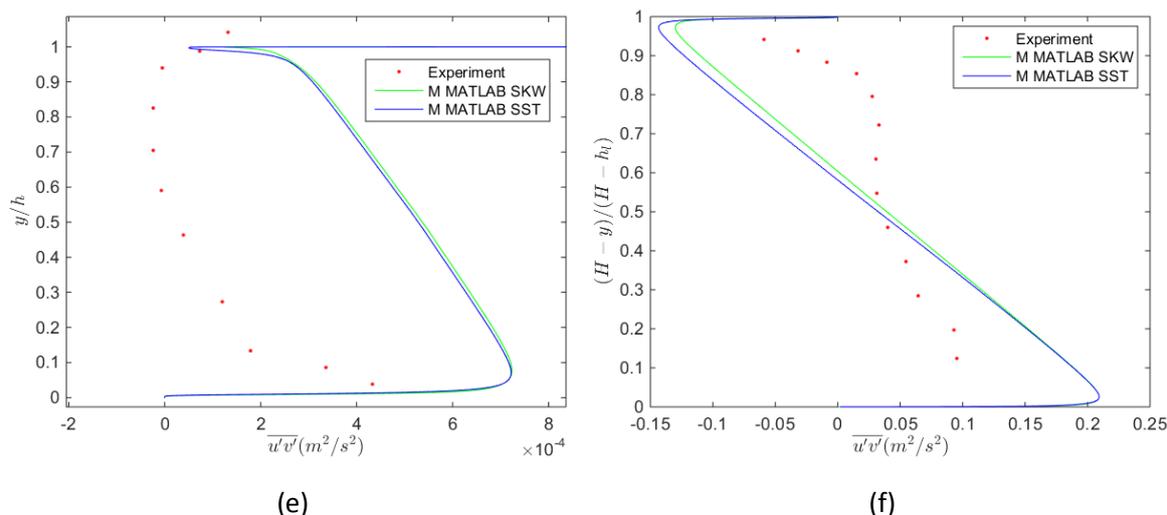


Figure 5.4: Comparison of various profiles for the liquid and gas phases. (a), (b) Velocity (u), (c), (d) Turbulence Kinetic Energy (k) and (e), (f) Shear stress ($\overline{u'v'}$). Run 400 from Fabre et al. (1987).

5.1.3 Fluent versus MATLAB

For a meaningful comparison between the Fluent and MATLAB results, the same interface condition for ω as applied in MATLAB also needs to be used in Fluent. This is possible with the help of the User Defined Functions (UDF). UDFs are short C++ programs that use Fluent macros to modify any flow variable. Here, we use it to modify the value of ω at the interface.

SKW Results

Obtaining agreement between Fluent and MATLAB is the next vital step in this thesis. Fluent, as mentioned by Chinello (2015), cannot accurately predict the two phase flow case for the SKW turbulence model. To obtain matching profiles, Chinello had to switch on the turbulence damping coefficient, which is an option in Fluent. A serious, and in fact unacceptable, limitation of this damping coefficient is that it is grid dependent.

Figures 5.5 and 5.6 show that MATLAB and Fluent give identical profiles for the velocity and turbulent kinetic energy when the SKW model is used without interface condition in MATLAB and with the turbulence damping switched off in Fluent. Essentially, this means that both models are non-Modified.

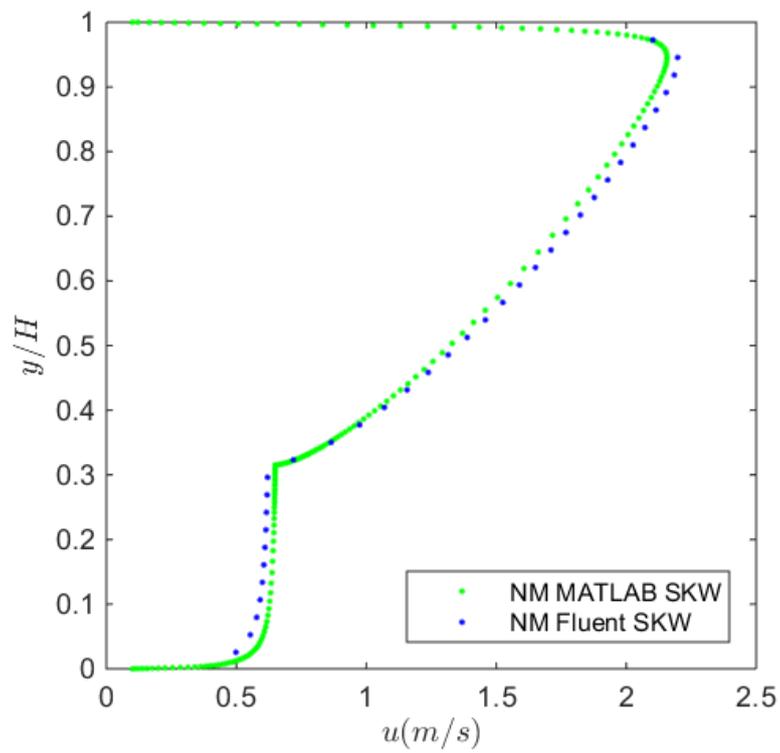


Figure 5.5: Comparison of the velocity profile for Run 400, when using the standard $k - \omega$ model without modification.

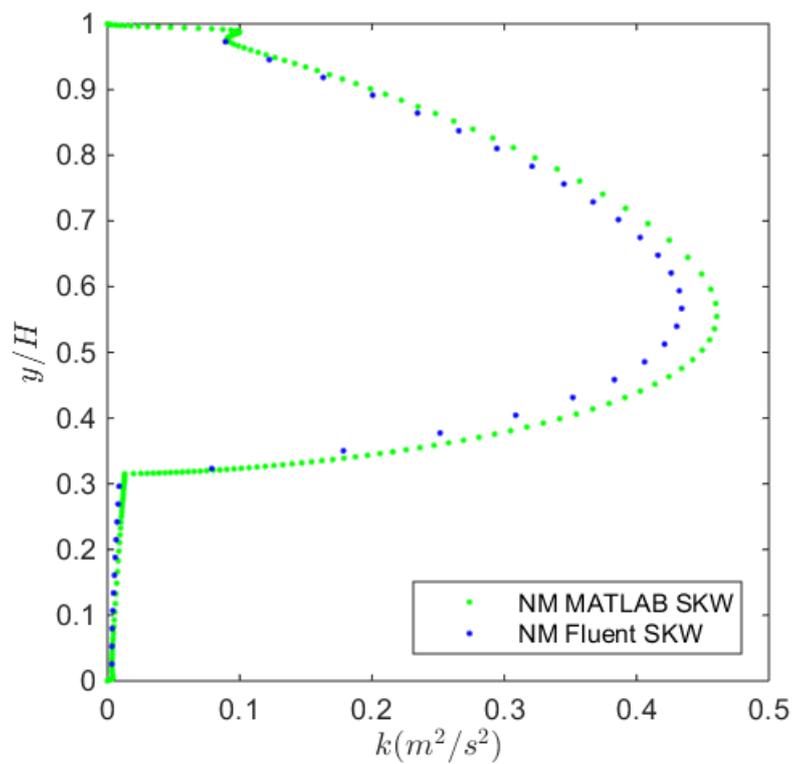


Figure 5.6: Comparison of the turbulent kinetic energy profile for Run 400, when using the standard $k - \omega$ model without modification

Imposing the interface condition in Fluent

For Run 400 the value imposed at the interface according to the output of the Modified MATLAB code was 2200 s^{-1} . This value was added to the UDF in Fluent.

Figure 5.7(a) shows for the velocity profile that due to the UDF the turbulence at the interface in Fluent is indeed being damped (blue curve) which was not the case in the original approach (red curve). This is a considerable improvement. However, when comparing the velocity profiles as obtained with the Modified SKW model in MATLAB and in Fluent, as shown in Figure 5.7(b), there is a clear mismatch between the profiles, even though the imposed value for both is the same, namely 2200 s^{-1} .

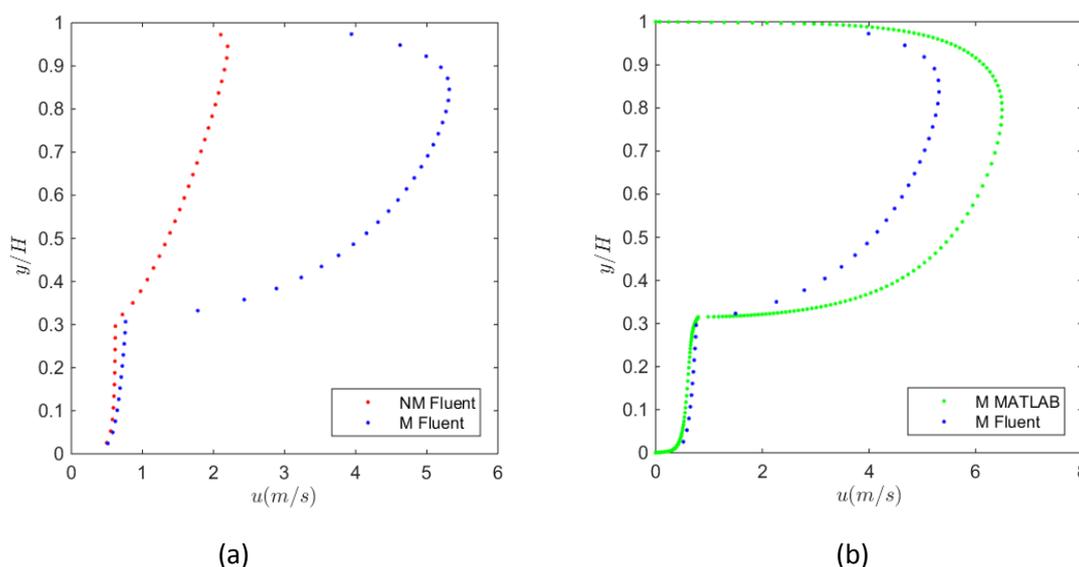


Figure 5.7: (a). Run 400 in Fluent using the SKW model without and with interface condition, (b). Run 400 using the SKW model in MATLAB and Fluent with interface condition.

When the output value of ω at the interface in Fluent is checked, it does not have the value of 2200 s^{-1} as imposed through the UDF, but a lower value of 1076 s^{-1} . This reduction of the interface value is found for all the considered experimental cases in Fluent. The exact reason for this inconsistency is not clearly understood, since the Fluent user manual is not very detailed.

To analyse this issue, it is necessary to understand the definition of DEFINE_ADJUST UDF. This function executes the change/modification to the selected variable at every iteration, before the transport equations are solved. The key word here is 'before'. A new value of ω_i replaces the previous imposed one. This new value is lower than the value that has been imposed. Therefore, using the DEFINE_ADJUST UDF does not give improvement.

An extra check can be made by imposing the reduced interfacial value for ω_i as found with Fluent also in MATLAB. Figures 5.8 and 5.9 show that now the profiles obtained with Fluent and MATLAB are almost identical, which is a significant improvement to the results shown in Figures 5.5 and 5.6. Here, the input value of ω_i for the Modified SKW model in Fluent remains the same, namely 2200 s^{-1} , but that of MATLAB is changed to 1076 s^{-1} .

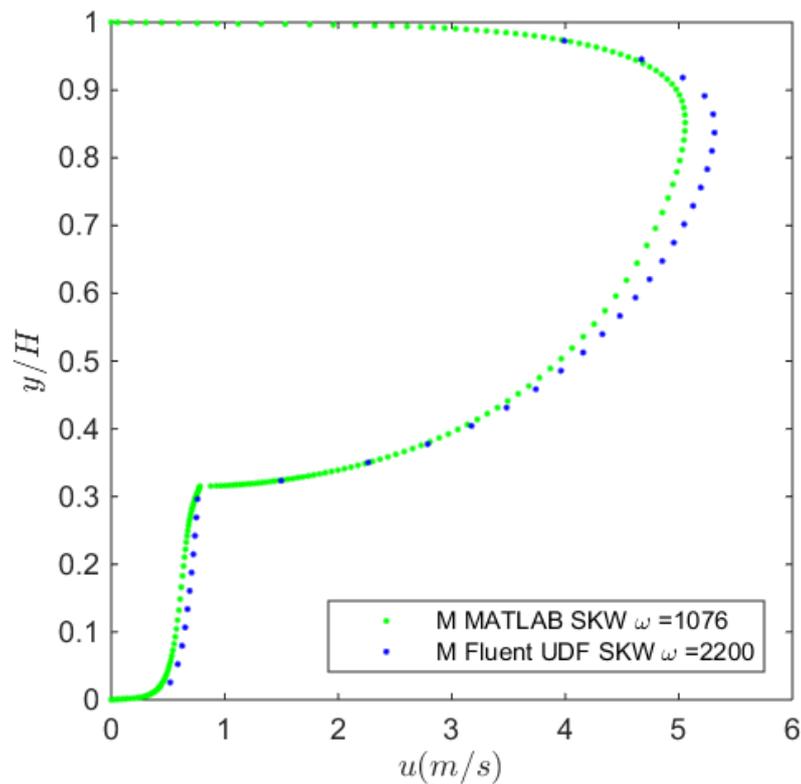


Figure 5.8: Velocity profile with MATLAB and Fluent for Run 400 in the experiments of Fabre et al.

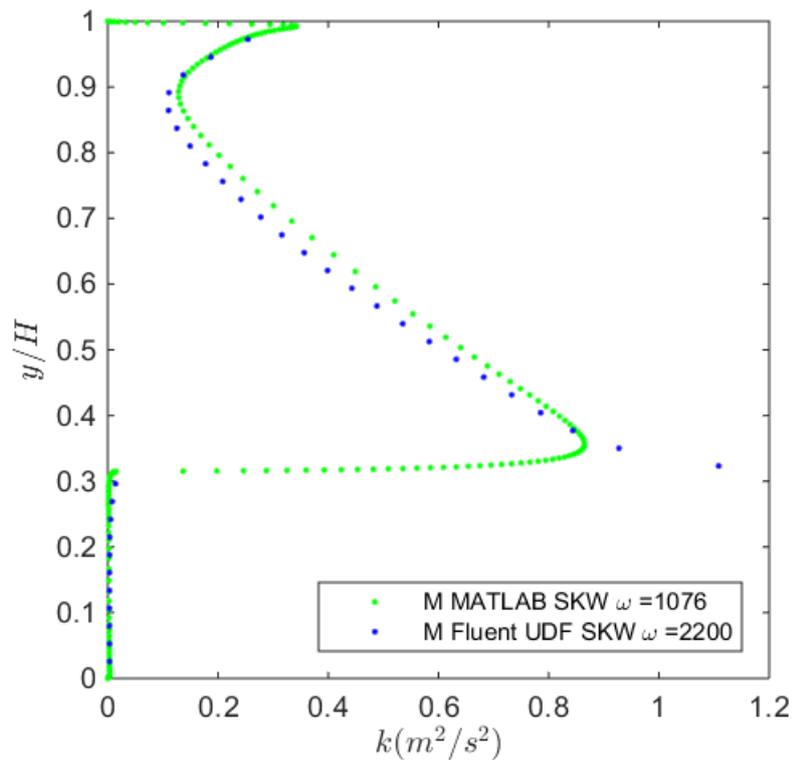


Figure 5.9: Turbulent kinetic energy with MATLAB and Fluent for Run 400 in the experiments of Fabre et al.

SST Results

The two phase SST model in Fluent has some unusual aspects. This model seems to have a built-in turbulence damping mechanism, which will be called the ‘pseudo damping mechanism’ or PDM. Unfortunately, this is not described in the Fluent manual. The PDM damps the turbulence at the interface, which leads to an interface value of ω_i of around 600 s^{-1} .

The effect of this PDM can be seen in Figure 5.10, which shows the results for Run 400 in the experiments of Fabre et al. (1987). Note that neither in MATLAB nor in Fluent an interface value for ω was imposed (i.e. the non-modified version of the SST model is used).

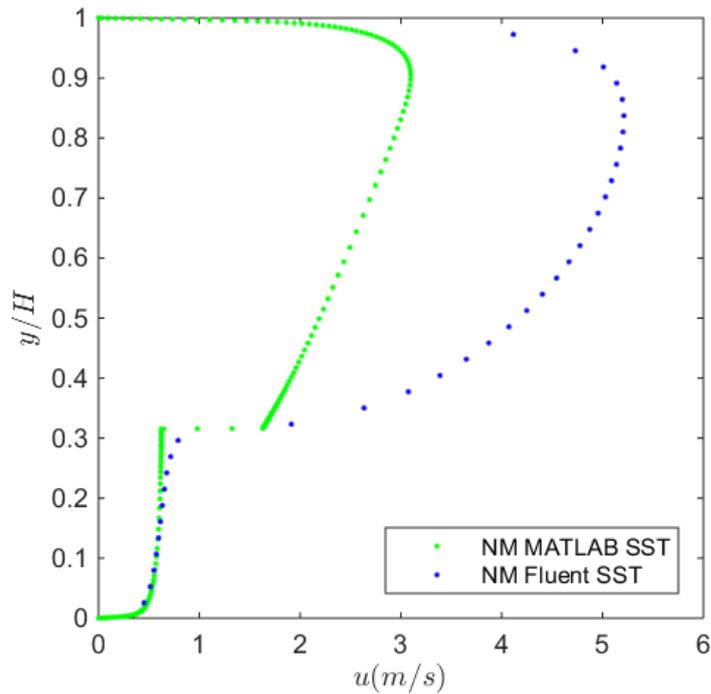


Figure 5.10: Velocity profile with the SST model for Run 400.

The predictions with MATLAB and Fluent are not the same. Difference can only be explained if something in Fluent is causing the turbulence to be damped. This can be investigated further by examining the profiles of the specific dissipation rate ω . Figures 5.11(a) and (b) show the specific dissipation rate with the SKW model and with the SST model, respectively. From Figure 5.11(b) it is clear that Fluent is overpredicting the value of ω and this is indeed causing the damping of the turbulence, leading to a smooth velocity profile.

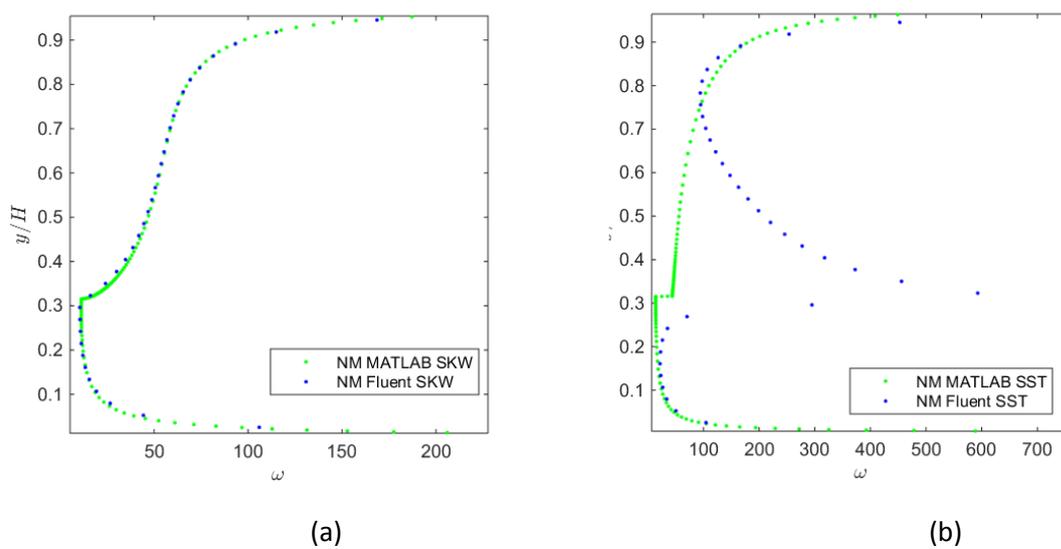


Figure 5.11: (a) ω for SKW case, (b) ω for SST case.

It is unknown why Fluent has this ‘P DM’ only for the SST model and not for the SKW model. There thus is a need to specify an interface condition with a UDF for the SKW model but not with the SST model.

5.2 Channel Flow Results

From the previous sections the following can be concluded:

1. The model of Oliemans (1987) gives more accurate results than other surface roughness models.
2. Using MATLAB, the SST turbulence model gives more accurate results than the SKW model.
3. The interface condition can be changed in Fluent by using the User Defined Function (UDF).

Therefore, further simulations will be made with the best performing turbulence model which is the modified SST model. The modification means imposing the interface boundary condition for ω , using the Oliemans expression for the roughness / wave height. This condition is imposed by using the UDF in Fluent.

That turbulence model will now be applied to an experimental case of Akai et al. (1980), see Figure 5.12, and an experimental case of Fabre et al. (1987), see Figure 5.12. More precisely, the following two cases will be considered:

1. Run 2 - Slightly wavy case, of Akai et al.
2. Run 400 - Wavy case, of Fabre et al.

In the MATLAB and Fluent simulations, the pressure gradient and the liquid level are used as input conditions, and gas and liquid flow rates are calculated as output parameters.

5.2.1 Akai results

The simulations results will be scaled with the same quantities as used by Akai et al. in their experiments. This means that the liquid velocity profile is scaled with the liquid velocity at half the height of the liquid layer:

$$\frac{u}{u_{h=0.003168}} \quad (32)$$

The gas velocity is scaled with the maximum gas velocity:

$$\frac{u}{u_{max}} \quad (32)$$

The turbulent kinetic energy is scaled with the local liquid velocity:

$$\frac{\sqrt{k}}{u_{local}} \tag{34}$$

The local height coordinate for the liquid and the gas layer are scaled as follows:

$$\frac{y}{h} \tag{35}$$

$$\frac{y-h}{H-h} \tag{36}$$

Here y is the height at any location, h is the liquid level, $C = y/h$ and H is the height of the channel.

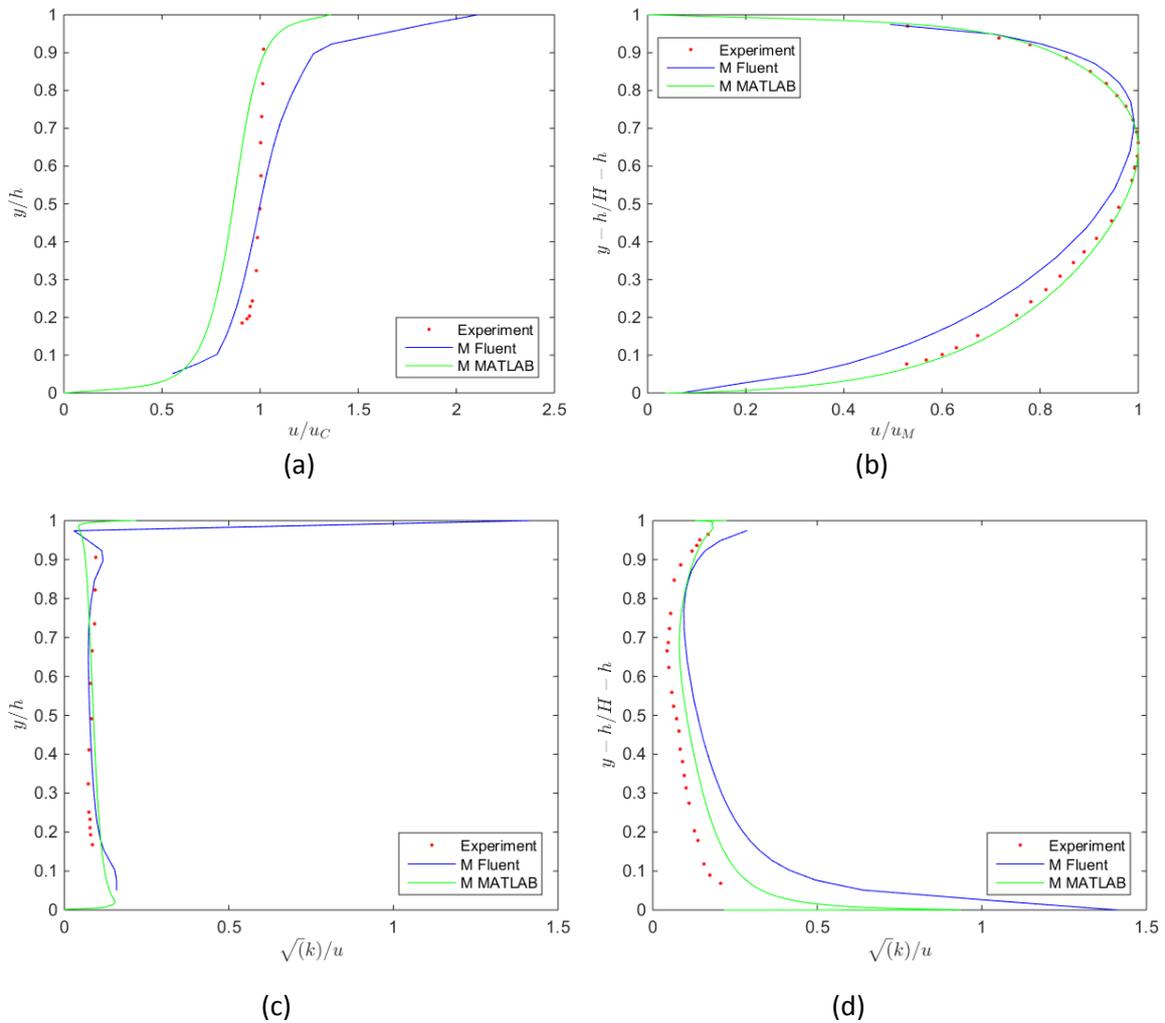


Figure 5.12: Modified SST model applied to Run 2 in the experiments of Akai et al. (1980). Results for the liquid and gas layers. (a), (b) Velocity profiles (c), (d) Turbulent Kinetic Energy.

The predictions for the velocity profile of the gas phase are almost identical to the experimental data, and the same holds true for the turbulent kinetic energy for both phases. The velocity in the liquid layer (as shown for the mid plane of the channel) is underpredicted, but only very slightly. Near the wall and the interface, the velocity in the liquid layer appears to be underpredicted and overpredicted, respectively. It is important to note that the resulting value for ω at the interface is $\omega_i = 14000\text{s}^{-1}$ which resulted from MATLAB and subsequently obtained iteratively for Fluent.

There is good agreement between the predictions by MATLAB and Fluent for the velocity profiles in both phases. The agreement for the turbulent kinetic energy in the gas phase is less good, while there is good agreement for k in the liquid layer.

5.2.2 Fabre results

The results for the Fabre case were scaled in the same way as the Akai results.

For MATLAB the interface value for ω is $\omega_i = 3300\text{ s}^{-1}$. In Fluent a higher value of $\omega_i = 15000\text{ s}^{-1}$ has to be imposed at the UDF to find an actual output value of $\omega_i = 2480\text{ s}^{-1}$, which was considered to be sufficiently close to the MATLAB value.

The trends in the prediction results for the case from Fabre et al. (1987) are similar to those of Akai et al. (1980). The turbulent kinetic energy in Figure 5.13(c) (d) is in good agreement with the experiments for both the phases. The gas phase velocity profiles in MATLAB and Fluent are almost identical, and close to the experiments. Also for the liquid layer the MATLAB and Fluent results are close, but there is some deviation for the experimental profile.

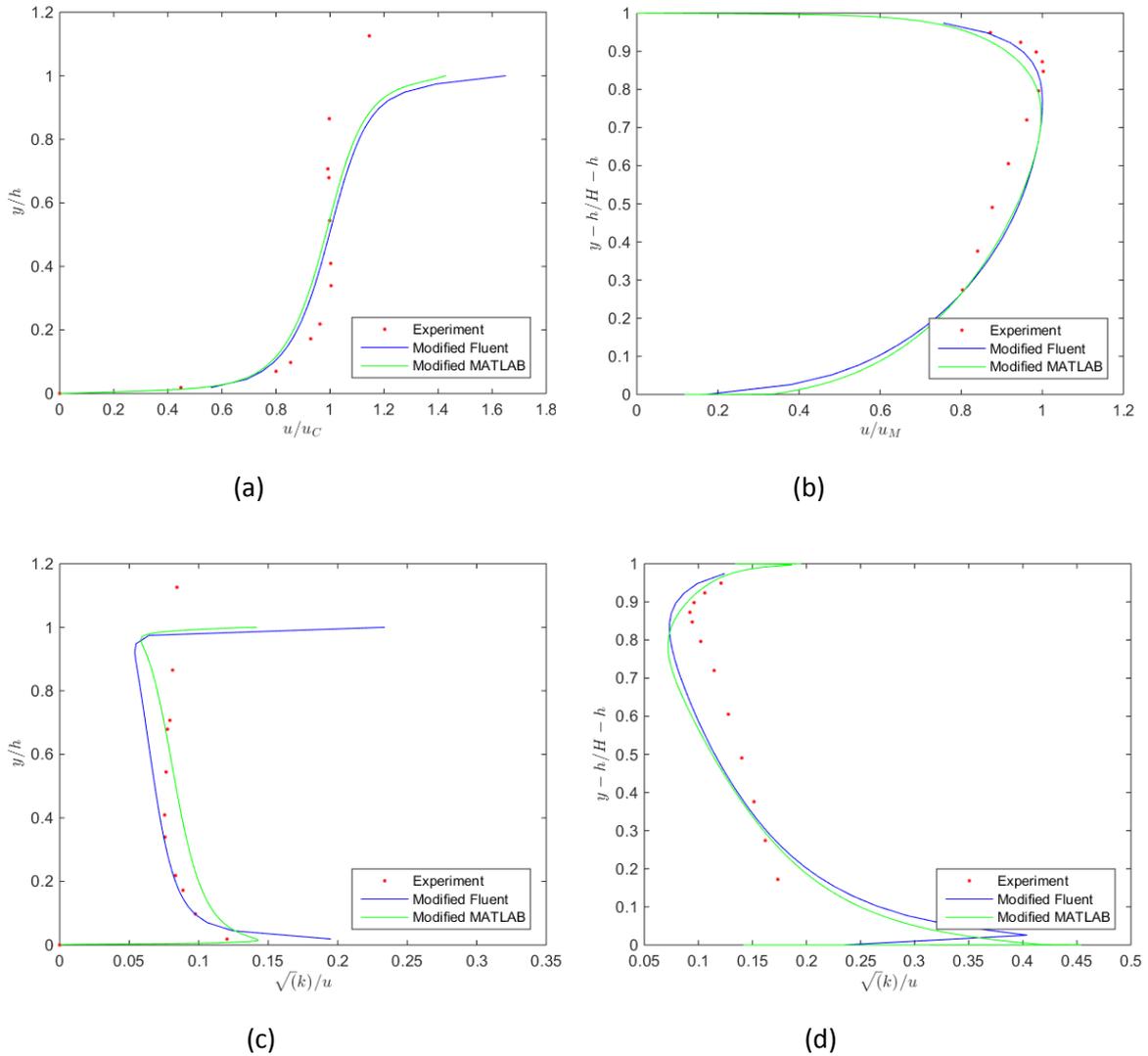


Figure 5.13: Modified SST model applied to Run 400 in the experiments of Fabre et al. (1987). Results for the liquid and gas layers. (a), (b) Velocity profiles (c), (d) Turbulent Kinetic Energy.

It is important to emphasize that the case of Fabre et al. (1987) is slightly downward inclined at an angle of $\theta = -0.0572^\circ$, while that of Akai et al. (1980) is completely horizontal, i.e. $\theta = 0$. The additional effect of the gravity may be a reason for the difference in the accuracy of the predictions for the two experiments. The predictions (at least with the model in MATLAB) are close to the experimental values of Akai et al. (1980), but there is a less good agreement with the experiments of Fabre et al. (1987). The velocity profile in the gas layer is steeper in the case of Fabre et al. (1987) than in the case of Akai et al. (1980).

Table 5.5: Flow rates (all in the unit m^3/s) for the experiments, and as found with MATLAB and Fluent.

Akai						Fabre					
Experiment		MATLAB - Modified SST		Fluent - Modified SST		Experiment		MATLAB - Modified SST		Fluent - Modified SST	
Q_G	Q_L	Q_G	Q_L	Q_G	Q_L	Q_G	Q_L	Q_G	Q_L	Q_G	Q_L
4.9E-3	4.3E-5	3.5E-3	4.8E-5	3.0E-3	6.9E-5	0.1187	0.003	0.1183	0.0031	0.13	0.004

The quantitative comparison between experimental data and simulation results, as given in Table 5.5, shows that the gas flow rate is underpredicted in the case of Akai et al. (1980). In the table Q_G and Q_L both have the unit m^3/s . There is almost a 30% underprediction with the model in MATLAB and about 40% underprediction with the model in Fluent. It can be that the large density difference between the two phases could play an important role here. It is possible that the models/software are not able to simulate flows with high gravity differences.

In regions of large normal strain, the two-equation models produce an artificially high effective viscosity according to Pope (2000). The large density difference between mercury and air also requires a larger damping value at the interface, see Sawko (2012).

The predictions for the flow rates are in good agreement with the experimental results by Fabre et al. (1987). In fact, we can conclude that for this case there has been a considerable improvement in the robustness and accuracy of prediction models.

The next step now is to assess how Fluent predicts 3D two phase flow. The 3D modelling is discussed in the next section.

5.2 Pipe Flow Results

3D modelling of pipe flow is done in Fluent and the data are taken and compared with the experimental data from Birvalski et al. (2014). Figure 5.13 shows the comparison of the mean scaled streamwise velocity profiles for the laminar case. The velocity predicted by Fluent is in good agreement with the experiment and with the DNS.

For the turbulent flow case, shown in Figure 5.15, the SKW model is used and the velocity profile shows good agreement with the DNS and with experimental data. However, towards the top of the channel the velocity profile is slightly overpredicted by Fluent.

Secondary flow patterns are prevalent in pipes since the depth in the crosswise direction changes. As a result of this, in Figure 5.16 a nonzero spanwise velocity (i.e. vertical velocity component) is found at the vertical centreline. The DNS is in very good agreement with the experiment. The $k - \omega$ model, however, fails to accurately describe this effect of a secondary flow and predicts that the spanwise velocity is 0. It is generally known that two-equation turbulence models like $k - \omega$ and $k - \epsilon$ cannot describe secondary flows, and instead (within the RANS framework) a differential Reynolds stress model should be used.

As a final note, this Fluent program is most faster in simulating this case than the DNS. It took around 10 hours on a single core CPU with a 2.5 GHz clock frequency to obtain the Fluent result, whereas the DNS simulation had taken a month on a computer.

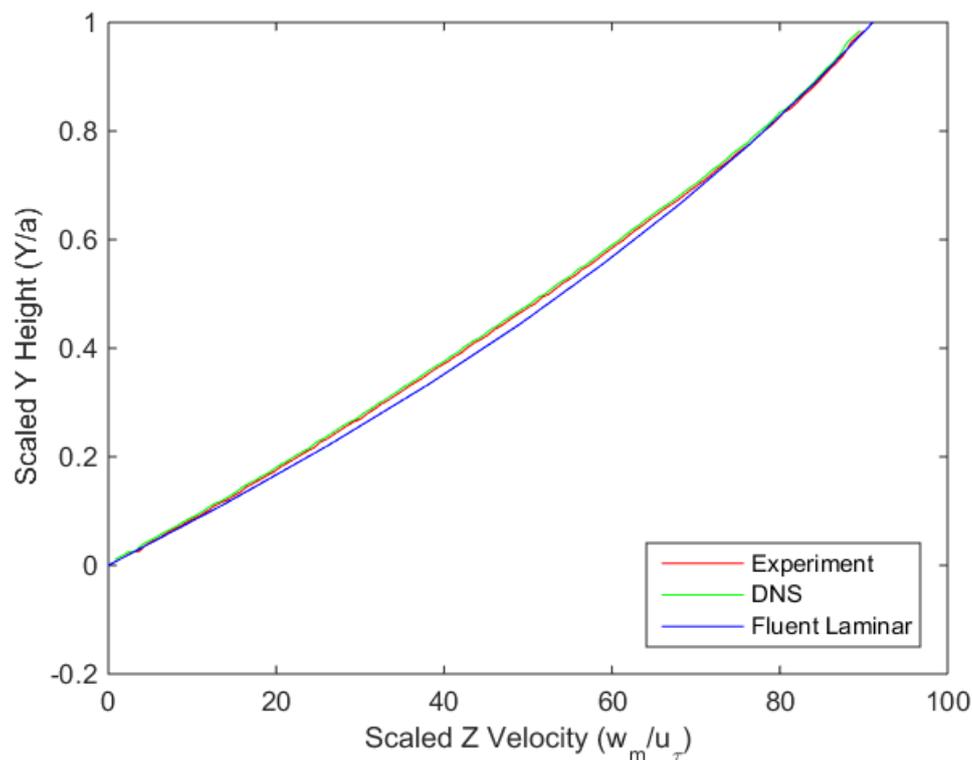


Figure 5.14: Streamwise velocity in the laminar pipe flow case.

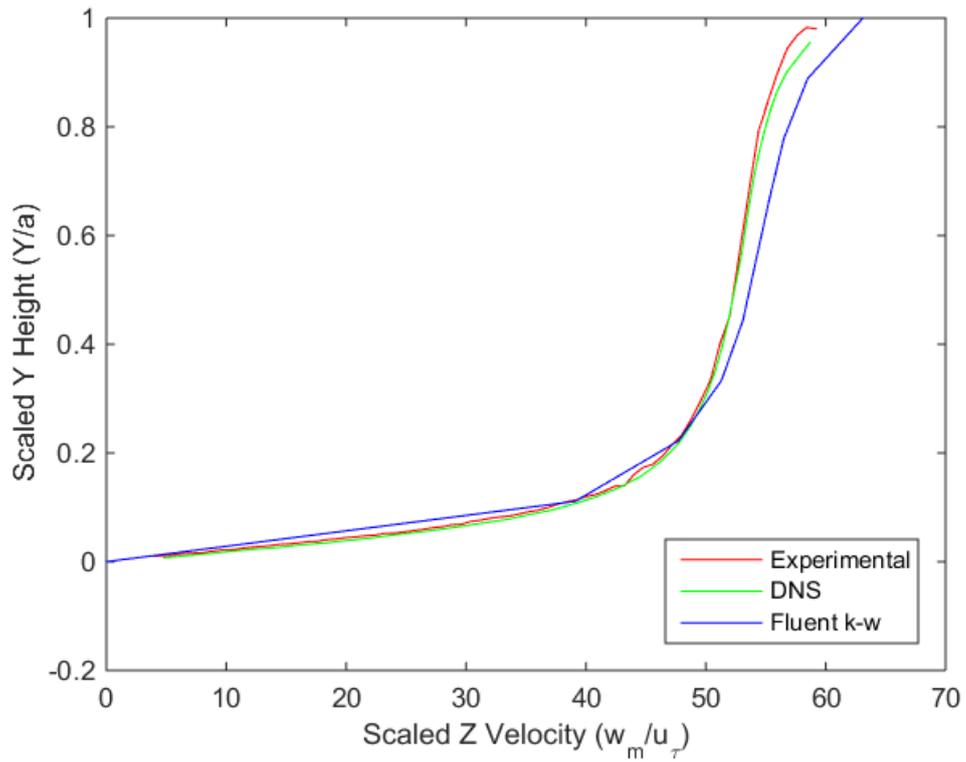


Figure 5.15: Streamwise velocity in the turbulent pipe flow case.

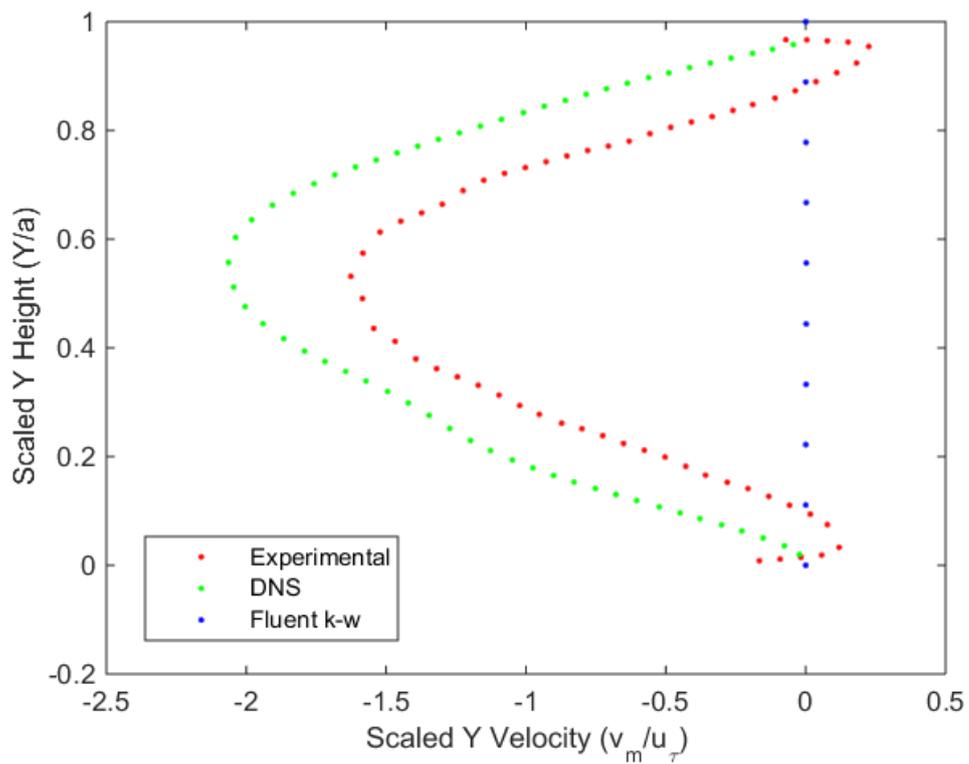


Figure 5.16: Spanwise velocity in the turbulent pipe flow simulation.

CONCLUSION

6.1 Conclusions

Two phase pipe flow has various industrial applications, such as in the oil and gas industry, in water disposal systems and in chemical plants. Simulating such flows with a CFD approach is not a trivial task. Particularly the available RANS models such as $k - \omega$ and $k - \epsilon$ have difficulties in handling the turbulence at the liquid-gas interface. Therefore, the main aim of this thesis was to improve these models, and to demonstrate their performance in predicting two phase flows in channels and pipes using MATLAB and Fluent.

- *Automate the calculation of the value of surface roughness.*
 - In the previous MATLAB model (as built by Chinello, 2015) it was necessary that the user knew beforehand whether the flow at the interface was wavy or smooth. Also a correlation for the wave height at the interface was missing. We have considered various models for this wave height (represented as an effective wall roughness in the turbulence model). Among these models, the one of Oliemans (1987) gives the best agreement with experiments. The accuracy of the prediction improves when the gas flow rate increases. Issa (1998) mentions that a relation between the interface and the gas Reynolds number is geometry dependent. Akai et al. (1980) use empirical correlations to obtain a relationship between the two for their geometry. On the other hand, the Wilcox (2006) condition coupled with the surface roughness models is geometry independent.

In addition to the interface modelling, also the type of turbulence model was investigated.

- *Improve the predictions of either the pressure gradient and liquid level or the flow rates of gas and liquid by changing turbulence models.*

-
- The change from the Modified Standard $k - \omega$ (SKW) to the Modified Shear Stress Transport (SST) model (as implemented in MATLAB) considerably increased the accuracy of the results. For Run 600, the values for the gas and liquid flow rates, as compared to the experimental values of Fabre et al. (1987), has improved from a deviation of 12.1% and 62.5% with the modified SKW model to a deviation of 0.3% and 50% with the modified SST model.
 - *Comparison between MATLAB and Fluent predictions*
 - MATLAB results are compared with predictions by the commercial CFD software package ANSYS Fluent. Fluent accurately predicts single phase flows with or without turbulence in complex geometries with relative ease. However, when a two phase stratified flow is considered the results can be very inaccurate. This is because the turbulence at the interface is not damped in the available turbulence models. This results in a shift of the location of the maximum gas velocity in channel flow towards the top wall. Fluent also has issues when simulating two phase flows, with regards to convergence and oscillating residuals.
 - There is a discrepancy between the results produced by Fluent with the SKW and SST models. There is significant evidence to believe that the SST model in Fluent applies a Pseudo Damping Mechanism (PDM) to dampen the turbulence at the liquid-gas interface, but this is not described in the Fluent user manual. The simulations with the NM SST and NM SKW models, as implemented in MATLAB, show inaccurate velocity profiles.
 - The Non-Modified SKW model in Fluent is comparable with the Non-Modified SKW model in MATLAB. However, to our surprise, the results with the Non-Modified SST model in Fluent are comparable with the results of the Modified SST model MATLAB. It thus seems that Fluent is indeed dampening the turbulence at the interface for the SST model (but not for the SKW model). However, this is not mentioned or explained in the Fluent manual.
 - A next step in this research was to compare MATLAB and Fluent for the Modified SKW and Modified SST models. Thereto the User Defined Function (UDF) in Fluent was used. However, Fluent gives as output for the interface value ω_i that is different from the imposed value. The only way out is to iteratively impose a value at the interface (and manually start a new simulation after each update) until the desired value appears in the output.

- For the Modified SST model, however, the output value of ω_i increases for a certain input value and decreases for some other input values. The trend with the Modified SKW model is always decreasing, while that in the Modified SST model is irregular. This might be related to the finding that Fluent seems to internally dampen the turbulence at the interface for the SST model.
- Another limitation of Fluent is that the user cannot impose a flow rate as an input when there is more than one phase involved.
- *Comparison of model predictions with experimental data from Akai et al. (1980) and Fabre et al. (1987).*
 - Using the suggested improvements for the model in MATLAB and Fluent, the channel flow experiments were simulated and discussed. The modified SST model was used. The slightly wavy case of Akai et al. (1980) is not accurately predicted, neither by the model in MATLAB nor by the model in Fluent. For the gas flow rate there is a 30% underprediction with the model in MATLAB and a 40% underprediction in the model in Fluent. The liquid flow rate is overpredicted by 14% with the model in MATLAB and by 37% with the model in Fluent.
 - The large density difference between the two phases in the Akai et al. (1980) experiment may have caused the simulations to be inaccurate. A large density difference requires a large damping value imposed at the interface according to Sawko (2012). This is to compensate for the overprediction of the interface turbulence by the two-equation models.
 - The experiment by Fabre et al. (1987) for the wavy interface is predicted very accurately by the Modified SST model, both with MATLAB and with Fluent. The gas flow rate is overpredicted by 0.3% with MATLAB and by 9% with Fluent. The liquid flow rate is overpredicted by 3% by MATLAB and by 33% with Fluent. Although the case of Fabre et al. is affected by gravity (as the pipe is slightly downward inclined) and that of Akai et al. is not (their pipe is fully horizontal), the predictions for the case of Fabre et al. are far better than for the case of Akai et al.
- *Model predictions for the Segregated Liquid Phase (SLP) in the two phase pipe flow as measured by Birvalski et al. (2014)*
 - The 2D two phase stratified flow case can be considered as a stepping stone to the 3D modelling. The $k - \omega$ RANS simulations, as carried out with Fluent for the segregated liquid phase (SLP) are compared

with the experimental data of Birvalski et al. (2014) and with the DNS results of Mehdi Niazi (2014). The velocity profile in the streamwise direction as predicted by Fluent is almost identical to the experimental and DNS results. However, the velocity profile with Fluent in the spanwise direction does not show the existence of secondary flows and therefore this velocity component is zero. This is a shortcoming of the model. The RANS simulation requires far less computer time than the DNS.

6.2 Recommendations

The surface roughness predictions as carried out with MATLAB are less accurate for lower flow rates. It will be valuable to further investigate different surface roughness and wave amplitude models for channel flows. A starting point for this can be the thesis of Espedal (1998).

It is recommended to further investigate the encountered turbulence damping phenomenon at the liquid-gas interface for the SST model in Fluent. Trying to discuss this directly with ANSYS Fluent may help here. Another approach can be to attach a DEFINE_PROFILE UDF instead of the DEFINE_ADJUST UDF for the ω_i . However, this does only work when imposing regular inlet and outlet boundary conditions, and it does not work with periodic boundary conditions.

The simulation results are not in agreement with the experimental results of Akai et al. (1980). There might also be something wrong with the experiment, which can only be verified by carrying out a similar experiment.

The 3D modelling is only done for the liquid phase. Developing a 3D RANS model for the entire pipe flow (i.e. liquid layer and gas layer) would be a possibility for future research.

Fluent has shortcomings in the flexibility to impose modified interface conditions. Therefore it is recommended to do the same comparison study with other CFD packages, such as OpenFoam or STAR CCM+.

6.3 Final Remark

The objectives of this thesis have been met. This research has shed light on various aspects of RANS models applied to two phase flows. Also the possibilities of using third-party CFD tools have been evaluated. It looks like there is plenty of room for further investigation before it can be concluded that a reliable interface handling in for channel and pipe flow in the context of $k - \omega$ and $k - \epsilon$ models exists.

Bibliography

- Akai, M., Inoue, A., Aoki, S. & Endo, K., 1980. A Co-Current Stratified Air-Mercury Flow with Wavy Interface. *International Journal of Multiphase Flow*, Volume 6, pp. 173-190.
- Baker, A., Nielsen, K. & Gabb, A., 1988. Pressure Loss, Liquid Holdup Calculations Developed. *Oil & Gas Journal*, Volume 86:11, pp. 55-59.
- Berthelsen, P. A. & Ytrehus, T., 2005. Calculation of stratified wavy two-phase flows in pipes. *International Journal of Multiphase Flow*, Volume 31, pp. 571-592.
- Birvalski, M., Tummers, M. J., Delfos, R. & Henkes, R. A. W., 2014. PIV Measurements of Waves and Turbulence in Stratified Horizontal Two-phase Flow. *International Journal of Multiphase Flows*, Volume 62, pp. 161-173.
- Charnock, H., 1955. Wind Stress on a Water Surface. *Quarterly Journal of the Royal Meteorological Society*, 81(350), pp. 639-640.
- Chinello, G., 2015. *CFD Modelling of Two Phase Stratified Flow in Oil and Gas Transportation*, Pisa.
- Cohen, L. S. & Hanratty, T. J., 1968. Effect of Waves at a Gas-Liquid Interface on a Turbulent Air Flow. *Journal of Fluid Mechanics*, 31(3), pp. 467-479.
- Daniloff, C., 2013. *Cigarette Smoke*. [Art].
- Espedal, M., 1998. *An Experimental Investigation of Stratified Two-Phase Pipe Flow at Small Inclinations*, s.l.: PhD thesis, Norwegian University of Science and Technology.
- Fabre, J., Masbernat, L. & Suzanne, C., 1987. Stratified Flow, Part I: Local Structure. *Multiphase Science and Technology*, Volume 3, pp. 285-301.
- Fernandez-Flores, R., 1984. *Etude des Interactions Dynamiques en Ecoulement Diphasique Stratifie*, s.l.: Institut National Polytechnique de Toulouse.
- Fitremann, J. M. & Rosant, J. M., 1981. A Model for Stratified Gas-Liquid Turbulent Flow in Ducts of Arbitrary Cross-Section. *Journal de Physique*, 16(3), pp. 93-103.
- Hanjalic, K. & Launder, B. E., 1972. A Reynolds Stress Model of Turbulence and its Application to Thin Shear Flows. *Journal of Fluid Mechanics*, Volume 52, pp. 609-638.
- Holmas, K. et al., 2005. *Simulation of Wavy Stratified Two-Phase Flow using Computational Fluid Dynamics (CFD)*, BHR Group.
- Issa, R., 1988. Prediction of Turbulent, Stratified, Two-Phase Flow in Inclined Pipes and Channels. *International Journal of Multiphase Flow*, 14(2), pp. 141-154.
- Jones, W. & Launder, B., 1972. The Prediction of Laminarization with a Two-equation Model of Turbulence. *International Journal of Heat and Mass Transfer*, 15(2), pp. 301-314.
- Jones, W. & Launder, B., 1973. The Calculation of Low-Reynolds Number Phenomena using Two-equation Model of Turbulence. 16(6), pp. 1119-1130.

- Kolmogorov, A. N., 1942. Equations of Turbulent Motion of an Incompressible Fluid. *Izvestia Academy of Sciences, USSR; Physics*, 6(1,2), pp. 56-58.
- Kordyban, E., 1974. Interfacial Shear in Two-Phase Wavy Flow in Closed Horizontal Channels. *Journal Fluids engineering*, Volume 96, pp. 97-102.
- Line, A., Masbernat, L. & Soualmia, A., 1996. Interfacial Interactions and Secondary Flows in Stratified Two-Phase Flow. *Chemical Engineering Communications*, 141(1), pp. 303-329.
- Lockhart, P. & Martinelli, R., 1949. Proposed Correlation of Data for Isothermal Two-Phase, Two-Component Flow in Pipes. *Chem. Engg. Prog*, 45(1), pp. 39-48.
- Lorencez, C., Nasr-Esfahany, M., Kawaji, M. & Ojha, M., 1997. Liquid Turbulence Structure at a Sheared and Wavy Gas-Liquid Interface. *International Journal of Multiphase Flow*, 2(23), pp. 205-226.
- Menter, F. R., 1993. *Zonal Two Equation k-w Turbulence Models for Aerodynamic Flows*. s.l., American Institute of Aeronautics and Astronautics.
- Menter, F. R., Kuntz, M. & Langtry, R., 2003. Ten Years of Industrial Experience with the SST Turbulence Model. *Turbulence, Heat and Mass Transfer*, pp. 625-632.
- Morris, S., 2007. *The Inevitability of Bumps*. [Art] (The New York Times).
- Niazi, M., 2014. *DNS of Stratified Partially Filled Pipes*, Delft.
- Nichols, R. H. & Nelson, C. C., 2004. Wall Function Boundary Conditions Including Heat Transfer and Compressibility. *AIAA Journal*, 42(6), pp. 1107-1114.
- Oliemans, R. V. A., 1987. *Modelling of Gas-Condensate Flow in Horizontal and Inclined Pipes*. s.l., ASME Pipeline Engineering Symposium.
- Pope, S. B., 2000. *Turbulent Flows*. s.l.:Cambridge University Press.
- Pots, B. F. M. & Oliemans, R. V. A. T. N., 1988. The KSLA Method for Gas/Liquid Two Phase Pipe Flow Calculations. pp. 88-315.
- Reichhardt, T., 2009. *Air Space Mag*. [Online]
Available at: <http://www.airspacemag.com/daily-planet/how-is-a-volcano-like-a-jet-engine-57116783/?no-ist>
[Accessed April 2016].
- Sawko, R., 2012. *Mathematical and Computational Methods of Non-Newtonian, Multiphase Flows*, s.l.: PhD Thesis, Cranfield University.
- Shih, T. H. et al., 1999. *Turbulence Surface Flow and Wall Function*. s.l., American Institute of Aeronautics and Astronautics.
- Spalding, D. B., 1961. A Single Formula for the Law of the Wall. *Journal of Applied Mechanics*, 28(3), pp. 444-458.
- Srichai, S., 1994. *High Pressure Separated Two-Phase Flow*, s.l.: PhD. Thesis, Imperial College of Science, Technology and Medicine.
- Weisman, J., 1983. Two Phase Flow Patterns. *Chapter 15 of Handbook of Fluids in Motion*, pp. 409-425.

Wilcox, D. C., 1998. *Turbulence Modeling for CFD*. 2nd ed. s.l.:DCW Industries.

Wilcox, D. C., 2006. *Turbulence Modeling for CFD*. 3rd ed. s.l.:DCW Industries.

Appendices

A

MATLAB Code

Below the MATLAB code is shown. This is for the Run 600 of Fabre. The two parts of the program are

- Intuitive
- Solver

The basic structure of the two parts is mentioned here and some pieces of the code are listed as well.

The Intuitive contains all commands related with (in chronological order) the following aspects:

- Inputs – Q_G and Q_L , H and the guessed values of dp/dx and h
 - Open Loop
 - 1st call to the Solver function
 - Convergence check (If converged move to GOTO)
 - Change h
 - 2nd call to the Solver function (with new h value)
 - Store the value of the partial derivatives of Q_G and Q_L w.r.t h
 - Change dp/dx
 - 3rd call to the Solver function
 - Store the value of the partial derivatives of Q_G and Q_L w.r.t dp/dx
 - Solve the equations of partial derivatives to obtain a new value of dp/dx and h for a new iteration.
 - GOTO
 - Print results

The Solver contains all commands related with (In chronological order)

- Grid Generation
- Initialise Turbulence Closure Coefficients
 - Open Outer Loop

- Inner Loop 1: Define turbulence viscosity and other model parameters
- u – equation
 - RHS_j definition
 - Creating a_j, b_j & c_j arrays
 - Solve with Tridiagonal Matrix Algorithm
- k – equation
 - RHS_j definition (Extra Production Limiter Term)
 - Creating a_j, b_j and c_j arrays
 - Solve with Tridiagonal Matrix Algorithm
- ω – equation
 - Boundary values in first and last 5 nodes
 - RHS_j definition (Production Limiter Term + Cross diffusion term)
 - Creating a_j, b_j and c_j arrays
 - Surface Roughness Models – Oliemans (1987) etc.
 - Impose ω at the interface
 - Solve with Tridiagonal Matrix Algorithm
- Close Outer Loop
- Calculate values of Flow Rates and Velocities

INTUITIVE

```

clc, clear, close all;

%% INPUT Fabre et al. 1987

% Volumetric flow rates
% Qg_input=0.227;           %RUN 250 smooth case
% Qg_input=0.377;           %RUN 400 wavy case
% Qg_input=0.5935;          %RUN 600 wavy case

% Ql_input=0.015;

% Channel height
H=0.1;

% Inclination channel
teta=0.0572;

% Density of the fluids
rho_G=1.18; % Air
rho_LI=998; % Water

% Viscosity of the fluids
mu_G=1.85e-5;
mu_LI=1e-3;

```

```

% Initial guess liquid height
h=0.0215;
% Initial guess pressure drop
dpdx=14.8;

% Mesh input
nj_G=123;      %internal nodes gas side
nj_LI=83;      %internal nodes liquid side
xmesh1=1.01;   % refinement liquid wall
xmesh2=-0.01; % refinement liquid interface
xmesh3=1.01;   % ref gas interface
xmesh4=-0.01;

[OPEN LOOP]

%% [1st Solver Call];

ks_plusiter(LOOP)=ks_plus;
%break      % Removing comment here will exchange the inputs and outputs

%% [CONVERGENCE CHECK]

if abs(Qg_input-Qg)/Qg_input*100<0.05 && abs(Ql_input-Ql)/Ql_input*100<0.05

    break

else

    Qg_old=Qg;
    Ql_old=Ql;
    h_old=h;
    dpdx_old=dpdx;

    %% [CHANGE h]
    eps=0.001;
    h=h+eps;

    %% [2nd Solver Call];

    %% [STORE PARTIAL DERIVATIVES]

    dQg_h=(Qg-Qg_old)/eps;
    dQl_h=(Ql-Ql_old)/eps;

    h=h_old;
    eps=0.001;

    %% [CHANGE dp/dx]
    dpdx=dpdx_old+eps;

    %% [3rd Solver Call];

    %% [STORE PARTIAL DERIVATIVES]
    dQg_dp=(Qg-Qg_old)/eps;
    dQl_dp=(Ql-Ql_old)/eps;

```

```

        %% [SOLVE PARTIAL DERIVATIVES
        A=[dQg_h dQg_dp; dQl_h dQl_dp];

        B=[Qg_input-Qg_old; Ql_input-Ql_old];

        x=A\B;

    end

    h=h_old+ x(1);
    dpdx=dpdx_old+ x(2);

    if h<0
        h=h_old;
    else
    end

end

%% [GOTO]
%% [PRINT OUTPUTS]

SOLVER

function [Inputs]=Solver(Ouputs)

%% Input

%%Fluid kinematic viscosities [m^2/s]
ni_G=mu_G/rho_G;
ni_LI=mu_LI/rho_LI;

% Inclination
g=9.81;

% Number of iterations
niter_max=350;
%%Tolerance for residuals
TOL = 10e-15;

% Under-relaxation factors
relax_u= 0.3;
relax_k= 0.3;
relax_w= 0.6;

%% Biexponent grid with refinement at the interface

%% [GRID GENERATION]
%%Calculation of the cell nodes

        nj=nj_LI+nj_G; % Total number of internal nodes
        j(1:nj+3)=0; % Y-coordinate of the nodes
        dj(1:nj+2)=0; % Distance between nodes

%Liquid domain first half
ninter=(nj_LI+1)/2; % Distance between two nodes

```

```

xmesh=xmesh1;          % x-parameter
Ratio=exp((10/niter)*(xmesh-0.5)); % Ratio
dj_ini=(h/2)*((1-Ratio)/(1-(Ratio^niter)));

for i=1:(nj_LI+1)/2
    dj(i)=dj_ini*(Ratio^(i-1));
    j(i+1)=j(i)+dj(i);
    dist(i)=j(i);
end

%Liquid domain second half
%Similar to above

%Gas domain first half
%Similar to above

%Gas domain second half
%Similar to above

%% Boundary conditions and initialisation

% Initialise arrays
u(1:ny+2)=1;
u(1)=0;
u(ny+2)=0;

w = ones (1,ny+2);
k = ones (1,ny+2);
.
.
.

% Turbulence model constants and Initialisation

% log- law
kappa = 0.41;
Cplus = 5;

%% [INITIALISE TURBULENCE CLOSURE COEFFICIENTS]
% k-w SST closure coefficients
R_k=6;
R_w=2.95;
.
.
.

%% [OPEN OUTER LOOP]

%% Solution

for niter=1:niter_max % Outer For Loop

    %% [OPEN INNER LOOP 1]
    for i=2:ny+1 % Definition of turbulent viscosity and clousure coefficient
        according to k and w previously calculated

            %% [MODEL CONSTANTS  $\alpha_i^*$ ,  $\beta_i$ ,  $\sigma_k$ , etc]
            .
            .

```

```

        .

        %% [RANS Terms]
        %k Dissipation
            Y_k(i)=beta_star_i(i)*w(i);

        %Omega Dissipation
            Y_w(i)=beta_i(i)*w(i);

        .
        .
        .

    % Turbulent Viscosity Definition
        ni_t(i)=a_1*k(i)*1/(max(a_1*w(i),F_3(i)*Rot_T(i)*F_2(i)));

    end
    ni_t(1) = 0;% Turbulent viscosity at walls
    ni_t(ny+2) = 0;

%% [u LOOP]

%% ----- u eqn ----- %%

% Source term and RHS definition
    for i=2:ny+1
        S(i) = (1/rho(i))*dpdx+g*sind(teta) ;
    end

%% [RHS j DEFINITION]

    for i=2:ny+1
        RHS(i)= -S(i);
    end

%% [INTERIOR ARRAY DEFINITION]

    for i=2:ny+1
        ni_L = 0.5/rho(i)*(ni_t(i-1)*rho(i-1)+ni_t(i)*rho(i)) + 0.5*(mu(i-1)+mu(i))/rho(i); %Sum of kinematic viscosity & kinematic dynamic viscosity
        ni_R = 0.5/rho(i)*(ni_t(i+1)*rho(i+1)+ni_t(i)*rho(i)) + 0.5*(mu(i+1)+mu(i))/rho(i);

        %% [CREATE a_j, b_j & c_j arrays]

        a(i) = ni_L/((y(i)-y(i-1))*((j(i)-j(i-1)))));
        c(i) = ni_R/((y(i+1)-y(i))*((j(i)-j(i-1)))));
        b(i) = -a(i)-c(i);
    end

%Calculation to get ustar
    stress_L = mu(2)*(u(2)-u(1))/(y(2)-y(1));
    stress_R = -mu(ny+1)*(u(ny+2)-u(ny+1))/(y(ny+2)-y(ny+1));
    u_tau_L=sqrt(stress_L/rho(2));
    u_tau_R=sqrt(stress_R/rho(ny+1));
    yplus_L = u_tau_L*y(2)/ni(2);
    yplus_R = u_tau_R*(H-y(ny+1))/ni(ny+1);

```

```

% Storing of the old velocity for relaxation purpose
uold=u;

% Storing coefficient for residuals calculation purpose
aold=a;
bold=b;
cold=c;
RHSold=RHS;

%% [SOLVE THE TDMA]

for i=2:ny
    b(i+1)=b(i+1)-(a(i+1)/b(i))*c(i) ;
    RHS(i+1)=RHS(i+1)-(a(i+1)/b(i))*RHS(i) ;
end

u(ny+1)=RHS(ny+1)/b(ny+1) ;

for i=ny:-1:2
    RHS(i)=RHS(i)-c(i)*u(i+1) ;
    u(i)=RHS(i)/b(i) ;
end

% Under-relaxation
u = relax_u*u + (1-relax_u)*uold ;

%Residuals
for i=2:ny+1
    residual_u(i)=(aold(i)*u(i-1)+bold(i)*u(i)+cold(i)*u(i+1)-RHSold(i));
end
residual_umax=max(abs(residual_u));

residual_umax_a(niter)=residual_umax;

%% [k LOOP]

%% ----- k eqn -----%

%Exact same steps as the u equation, replace u with k. Additionally, Source term S(i)
= 0 and Production term addition is shown below

%% [RHS j DEFINITION EXTRA PRODUCTION TERM]

for i=2:ny+1
    RHS(i)= -S(i)- min((ni_t(i))*((u(i+1)-u(i-1))/(y(i+1)-y(i-1)))^2,10*beta_star(i)*k(i)*w(i)) ;
end

.
.
.

%% [INTERIOR ARRAY DEFINITION]

.
.

```

```

.

%% [CREATE  $a_j, b_j$  &  $c_j$  arrays]
.
.
.

%% [SOLVE THE TDMA]
.
.
.

%% [ $\omega$  LOOP]
%% ----- w eqn ----- %

%% [BOUNDARY VALUES IN FIRST & LAST 5 NODES NEAR WALL]
% w value at the first 5 nodes near the wall

for i=2:6
    if i==2
        w(i) = 6*ni(i)/(beta*abs((dist(i))*(dist(i))));
    else
        w(i) = 10*6*ni(i)/(beta*abs((dist(i+1)-dist(i))*(dist(i+1)-dist(i))));
    end
end

for i=(ny-3):(ny+1)
    if i==ny+1
        w(i) = 10*6*ni(i)/(beta*abs((dist(i-1)-dist(i))*(dist(i-1)-dist(i))));
    else
        w(i) = 6*ni(i)/(beta*abs((dist(i))*(dist(i))));
    end
end

%% [RHS j DEFINITION EXTRA PRODUCTION TERM + CROSS DIFFUSION TERM]

% Source term and RHS definition
for i=6:ny-3
    S(i) =D_w(i) ;
end

for i=6:ny-3
    RHS(i) = -S(i) - min(alpha(i)*((u(i+1)-u(i-1))/(y(i+1)-y(i-1)))^2,10*beta_star(i)*w(i)*k(i));
end

% Coefficients redefinition at the interface gas side
stress_i_G= (dpx+g*rho_G*sind(teta))*(H-h)-stress_R;
u_tau_i_G=sqrt(stress_i_G/rho_G);

% Pots et. al(1988)
delta_h=0.00506*(1+tanh((abs(u_G_avg-u(86))-4 .572)/1.2192))/2;

%% [SURFACE ROUGHNESS MODELS OLIEMANS (1987) ETC.]

```

```

% %Ways to write value of ks

% % Fernandez-Florez (1984)
% ks=delta_h*sqrt(2*(1-(56*ni_G/(u_tau_i_G*DhG))^2))/sqrt(2);

% %Cohen and Hanratty Model (1968)
% ks=3*sqrt(2)*delta_h;

% Oliemans Model (1987)
if 2*delta_h/sqrt(2)<h
    ks=3*sqrt(2)*2*delta_h/sqrt(2);
else ks=3*sqrt(2)*h;
end

% Wilcox simplified Surface Roughness (2006)
w_w = (40000*ni_G/ks^2);

%% [IMPOSE  $\omega$  AT THE INTERFACE]
a(ny_LI+2)=0;
b(ny_LI+2)=1;
c(ny_LI+2)=0;
RHS(ny_LI+2)=w_w;

% Coefficient definition at the boundary

RHS(7) = RHS(7) - a(7)*w(6) ;
RHS(ny-4) = RHS(ny-4) - c(ny-4)*w(ny-3) ;

% Storing of the old w for relaxation purpose
wold=w;

% Storing coefficient for residuals calculation purpose
aold=a;
bold=b;
cold=c;
RHSold=RHS;

%% [SOLVE THE TDMA]
% Gauss elimination of a tridiagonal system
for i=7:ny-5
    b(i+1)=b(i+1)-(a(i+1)/b(i))*c(i) ;
    RHS(i+1)=RHS(i+1)-(a(i+1)/b(i))*RHS(i) ;
end

w(ny-4)=RHS(ny-4)/b(ny-4);

for i=ny-5:-1:7
    RHS(i)=RHS(i)-c(i)*w(i+1) ;
    w(i)=RHS(i)/b(i) ;
end

% Under-relaxation
w = relax_w*w + (1-relax_w)*wold ;

%Residuals
for i=8:ny-5
    residual_w(i)=(aold(i)*w(i-1)+bold(i)*w(i)+cold(i)*w(i+1)-RHSold(i));
end
residual_wmax=max(abs(residual_w));

```

```

residual_wmax_a(niter)=residual_wmax;

% Print on monitor number of iteration and store array iteration to plot the
residuals
niter;
niter_a(niter)=niter;

% covergence stop
residual_all_max =[residual_umax; residual_kmax ;residual_wmax];
residual_all_max=max(residual_all_max);
if residual_all_max<TOL
break
end

end

%% Flow rates calculation, mean velocities, Re

for i=1:length(y)
if y(i)<h
u_L(i)=u(i);
y_L(i)=y(i);
else
u_G(i)=u(i);
y_G(i)=y(i);
end
end
end
%% [CLOSE OUTER LOOP]

%% [CALCULATE VALUES OF FLOW RATES AND VELOCITIES]
u_G=u_G(length(u_L+2):length(u_G)) ;
y_G=y_G(length(y_L+2):length(y_G)) ;

xx = linspace(h,H,1000);
yy = spline(y_G,u_G,xx);

pp=spline(y_G,u_G);
Qg= quad(@(x)ppval(pp,x),h,H); %Simpson quadrature.

Ubg= Qg/((H-h));

xx = linspace(0,h,1000);
yy = spline(y_L,u_L,xx);

pp=spline(y_L,u_L);
Ql= quad(@(x)ppval(pp,x),0,h); %Simpson quadrature.

Ubl= Ql/(h);

Qg;
Ql;

Ubl;
Ubg;

```

Fluent DEFINE_ADJUST UDF

```

#include "udf.h"
DEFINE_ADJUST(adjust_omega, mixture_domain)
{
    int phase_domain_index;
    cell_t cell, c0;
    face_t f;
    Thread *interface_thread;
    Thread *t0;
    int zoneid = 2;
    interface_thread = Lookup_Thread(mixture_domain, zoneid);
    begin_f_loop(f, interface_thread)
    {
        /* c0 and t0 identify the adjacent cell */
        c0 = F_C0(f, interface_thread);
        t0 = THREAD_T0(interface_thread);
        /* this loops over all cells adjacent to wall and lets the UDM = 2.0 */
        C_O(c0, t0) = 2200;
    }
    end_f_loop(f, thread)
}

```

For the SKW case of Fluent, Fabre Run 400 (Figures 5.7, 5.8). The DEFINE_ADJUST UDF code is written in a C program, it is called 'adjust_omega'. The 'mixture_domain' pointer is passed as an argument. It contains the pointers to every location on the domain. Faces, Cells and Threads are initialised. Lookup_Thread retrieves the thread pointer for a location specified in the geometry. Here, it is '2' which is the location of the interface and 'interface_thread' contains this location.

A face loop is opened for the interface location and the connectivity macros F_C0 & THREAD_T0 contain all the face centroids and cell thread pointers respectively.

Finally, the specific dissipation rate macro – C_O at location (c0, t0) is updated with the constant value of $2200s^{-1}$. The loop and the program are closed shortly after.

Fluent DEFINE_PROFILE UDF

```

#include "udf.h"
DEFINE_PROFILE(stresswall, t, i)
{
    face_t f;
    cell_t c;
    Thread *tc;
    real x[ND_ND];
    real tau = -0.0107;
    real y;
    real abcd;
    begin_f_loop(f, t)
    {
        F_CENTROID(x, f, t);
        c = F_C0(f, t);
        tc = THREAD_T0(t);
        if (-0.024 < x[0] && x[0] < -0.0225)
        {
            F_PROFILE(f, t, i) = tau*(-100 * pow((x[0]) / 0.025, 2) - 180 *
((x[0]) / 0.025) - 80);
        }
        else if (0.0225 < x[0] && x[0] < 0.024)
        {
            F_PROFILE(f, t, i) = tau*(-100 * pow((x[0]) / 0.025, 2) + 180 *
((x[0]) / 0.025) - 80);
        }
        else
        {
            F_PROFILE(f, t, i) = tau;
        }
    }
    end_f_loop(f, t)
}

```

The wall shear stress function for the 3D SLP case is imposed along the streamwise ‘Z’ component. ‘t’ & ‘i’ are the variables passed from Fluent to the UDF, where the former is a thread pointer and the latter is hooked onto the variable in Fluent. Here, this variable is the ‘Z’ component of the shear stress.

A face loop is opened where all the values of the shear stress along the face centroid are modified accordingly using ‘if – else’ statements.