Cranfield University
School of Engineering
Applied Mathematics and Computing Group

Engineering Doctorate
THESIS

Slug initiation and prediction using high accuracy methods - applications with field data

by

Stamatis Kalogerakos

Supervisors:
Prof. Chris Thompson
Dr. Mustapha Gourma

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Engineering.
Cranfield, 2011

Acknowledgements

I would like first of all to thank Prof. Chris Thompson for giving me the opportunity of doing this EngD study and research in Cranfield University. I would like to thank also EPSRC for the funding provided, and BP for the funding as well as the data shared.

Moreover I would like to thank Dr. Mustapha Gourma for his patience, motivation, help and guidance, including many interesting discussions and also arguments! I want also to thank Ninghong Jia and the other colleagues in the Applied Mathematics and Computing Group for making my studies in Cranfield University a great experience. My sincere thanks go also to Kath Tipping for her supporting role in all matters regarding administration of EngD.

Last, but not least, I would like to thank my wife for her understanding, support and love during all these years. My parents receive also my deepest gratitude for the many years of support during my early study years, and my thanks go also to my brother for the good discussions we had.

This is dedicated to my dearest wife and daughter
and also to my parents and my brother.

Executive Summary

The sponsoring company of the project is BP. The framework within which the research is placed is that of the Transient Multiphase Flow Programme (TMF-4), a consortium of companies that are interested in phenomena related to flow of liquids and gases, in particular with relevance to oil, water and air. The deliverables agreed for the project were:

- validating EMAPS through simulations of known problems and experimental and field data concerning slug flow
- introducing numerical enhancements to EMAPS
- decreasing computation times in EMAPS
- using multi-dimensional methods to investigate slug flow

The outcome of the current project has been a combination of new product development (1D multiphase code EMAPS) and a methodological innovation (use of 2D CFD for channel simulations of slugs). These are:

- New computing framework composed of:
  - Upgraded version of 1D code EMAPS
  - Numerical enhancements with velocity profile coefficients

- Validation with wave growth problem
- Parallelisation of all models and sources in EMAPS
- Testing suite for all sequential and parallel cases
- Versioning control (SVN) and automatic testing upon code submission.

- Use of 2D CFD VOF for channel simulation with:
  - Special initialisation techniques to allow transient simulations
  - Validation with wave growth problem
  - Mathematical perturbation analysis
  - Simulations of 92 experimental slug flow cases

The cost of uptake of the above tools is relatively small compared to the benefits that are expected to follow, regarding predictions of hydrodynamic slugging. Depending on the timescales involved, it is also possible to use external consultancies in order to implement the solutions proposed, as these are software based and their uptake could be carried out in a small time-frame. Moreover it may not be necessary to build a parallel hardware infrastructure as it is now possible to have easy access to large parallel clusters and pay rates depending on use.

Contents

List of Figures

LIST OF FIGURES

LIST OF FIGURES

LIST OF FIGURES

List of Tables

Chapter 1

Project Description

1.1 Aim of the Research

The aim of the research is defined as the search for methods that will allow a better understanding of slug initiation and prediction in multiphase flows using high accuracy methods, and the use of these methods in applications with field data. It has been decided that a stronger emphasis should be put also on the slug initiation process, as it forms an integral part in understanding the formation of slugs and the subsequent evolution. The tools to be used will include mathematical modelling, working on a one-dimensional in-house written code (EMAPS), and also using two- and three-dimensional commercial codes.

1.2 Objectives

To achieve the overall aim of the research the following objectives were defined:

1. To produce an updated, robust, flexible and fast version of 1D code EMAPS.

2. To create a computing environment where optimisation and parallelisation of EMAPS can work hand in hand and also be utilised concurrently by multiple users.

3. To include numerical enhancements for EMAPS that will introduce more features of slug flow into the modelling.

4. To validate EMAPS by comparing with analytical and experimental results.

5. To investigate the quality of results obtained by slug flow simulations with 2D Computational Fluid Dynamics (CFD) codes.

6. To conduct technical-economical analyses to determine the advantages for BP of using the tools provided by the current project.

7. To provide practical recommendations.

1.3 Tasks

In order to meet the aim and the objectives outlined above, the research was divided into different tasks. Each of these tasks partially contributed to achieve the objectives and the overall aim of the research. A summary of the tasks that have been undertaken during the thesis are shown below:

1. **Literature Review - Chapter 3.** The review encompasses general concepts of fluid dynamics, including proofs, up to studies regarding slug properties. Equations and models used in one-dimensional multiphase flow are explained, and the equations used in volume of fluid (VOF) modelling for CFD are also included. A background on parallelisation is also included, in preparation for the parallelisation of EMAPS.

2. **EMAPS - Chapter 4.** The EMAPS infrastructure is explained in detail here. EMAPS was also modified in order to incorporate model changes, adaptivity and moreover a new test suite was completed and the whole source code was placed under version control. Details of the OpenMP parallelisation of EMAPS are included (Kalogerakos et al., 2012d). Tests were successful for both uniform and adaptive grids after parallelisation was completed.

   Simulations have been carried out both on cases taken from literature, in particular experiments carried out in Imperial College, London (Manolis, 1995) and also on datasets obtained from BP.

3. **Velocity profiles - Chapter 5.** Velocity profile coefficients have been introduced to the simulations, as deemed to play an important role in the determination of the flow regime maps. Values of the profile coefficients have been first found in literature, and later values obtained from 3D simulations using commercial software FLUENT by Ansys have been used. The correlation giving the best agreement with experimental results (Manolis, 1995) came from pressure-fitted velocity profile coefficients (Kalogerakos et al., 2012b). Simulations using EMAPS with modified velocity profile coefficients have also been carried out on datasets provided by BP.

4. **Wave growth - Chapter 6.** The wave growth problem analysis, consisting of flow determined by the input of an initial sine-wave, has been carried out with EMAPS and compared successfully with results from TRIOMPH, Imperial College. The use of 2D FLUENT to simulate two-phase flow in a channel was initially validated by repeating the wave growth problem but this time using the volume of fluid (VOF) model (Kalogerakos et al., 2010). A full mathematical perturbation analysis was also carried out, in order to

validate the simulation results of 2D FLUENT (Kalogerakos et al., 2012a).

5. **Slug simulations using 2D FLUENT - Chapter 7.** After the wave growth analysis gave support to the use of 2D FLUENT for modelling two-phase flow in channels, it was decided to complete simulations initially of small datasets (Kalogerakos et al., 2011) and then large experimental datasets with slug flow. The results are promising and give further support to the use of 2D FLUENT with VOF model in order to describe slug flow (Kalogerakos et al., 2012c).

6. **Conclusions - Chapter 9 & Chapter 9.5.** Conclusions from the research, including practical recommendations and future work, are included here.

7. **Industrial analysis - Chapter 8.** An overview of BP is given, together with how the slug phenomenon affects its operations. A general analysis of oil economics is also included, in order to contextualise the importance of understanding slug flow. Known effects of slugs in the oil industry will be shown and their cost implications will be given in detail. Advice on approaches to be taken to tackle the slugging phenomenon will also be given, starting from the results of the thesis. The benefits of using the new product (1D EMAPS) and methodological innovation (2D CFD) are explained in the context of a market analysis, together with cost implications and future benefits for BP.

The thesis structure is shown in Fig. 1.1, highlighting the methodological approach in connection with the objectives and tasks.

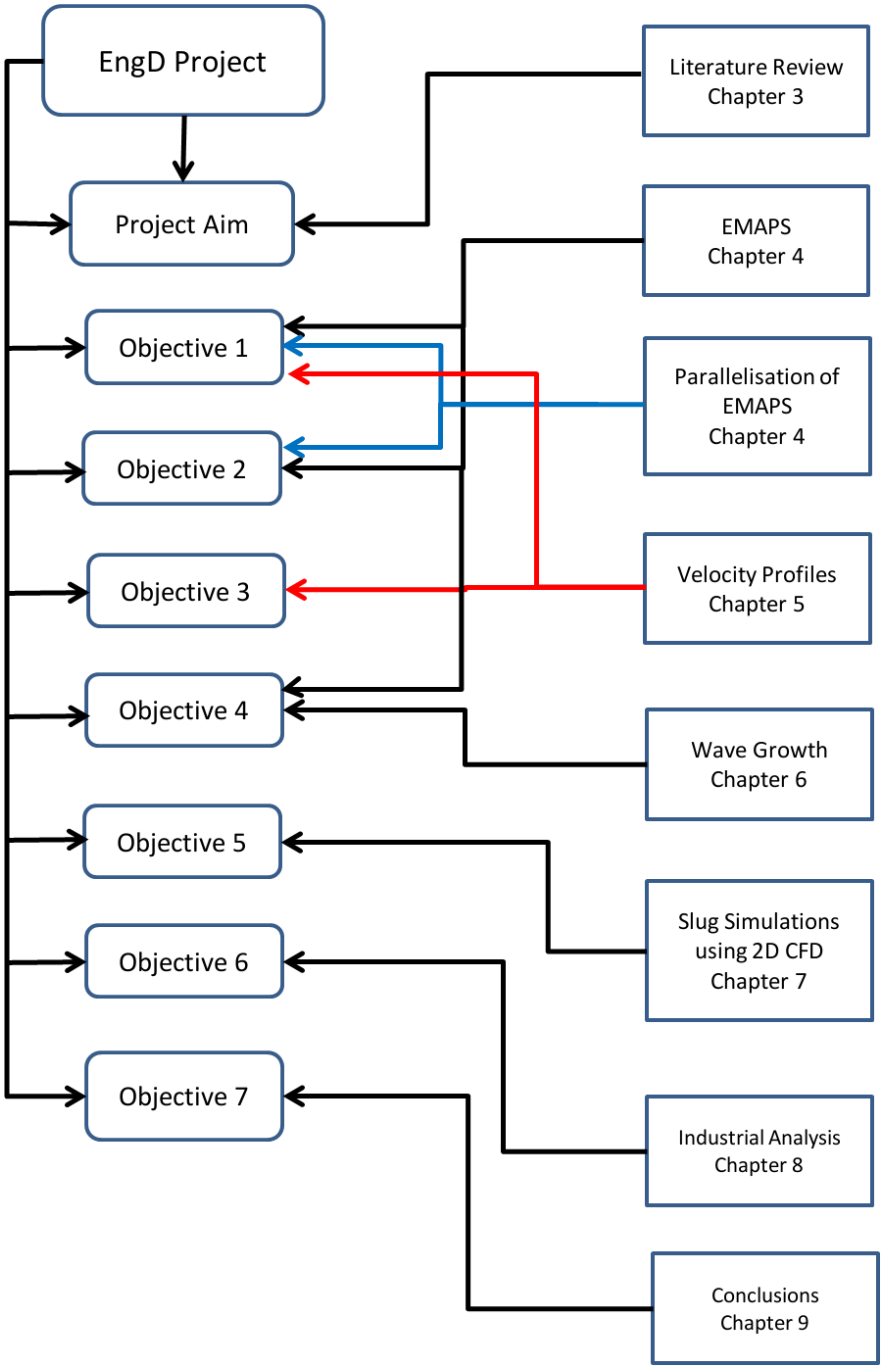


Figure 1.1: Thesis structure, showing relation between objectives and tasks.

1.4 Courses

The following courses have been successfully taken:

- EngD Module: Research planning and report writing (22nd November 2006).
- OLGA Flow Assurance (30th April 2007).
- EngD Module: Technology change and environmental assessment (10th March 2008).
- Star-CD training course (23rd June 2008).
- EngD Module: Systems engineering (8th December 2008).
- OLGA training course (11th May 2009).
- Star-CCM+ training course (2nd February 2010).

1.5 Articles

The following articles have been written and accepted or are in the process of being submitted:

1. S. Kalogerakos, M. Gourma, C. P. Thompson. Comparison between 2-D CFD and 1-D code for Wave Growth Simulations, International Conference on Multiphase Flow, 2010, University of Florida.
2. S. Kalogerakos, M. Gourma and C. P. Thompson. Use of 2-D CFD for simulating two-phase flows in horizontal pipes, International Association of

Science and Technology for Development, Applied Simulation and Modelling, 2011.

3. S. Kalogerakos, M. Gourma and C. P. Thompson. Adjustments of velocity profile coefficients for one-dimensional multiphase flow code, Journal of Computational Physics, 2012. In preparation.

4. S. Kalogerakos, M. Gourma and C. P. Thompson. Y-dependent wave growth analysis for VOF model, Multiphase Science and Technology, 2012. In preparation.

5. S. Kalogerakos, M. Gourma and C. P. Thompson. Comparison between 2D CFD and experiments for slug flow, International Journal of Multiphase Flow, 2012. In preparation.

6. S. Kalogerakos, M. Gourma and C. P. Thompson. Use of OpenMP to parallelise a one-dimensional multiphase code, Computer Physics Communication, 2012. In preparation.

1.6 Relevance of research to industry

The research on slugs is particularly relevant to the oil industry. BP is directly interested because slugging can lead to oscillations in the level of pressure and flow rate, with adverse effects on the production and with increased risks for the machinery involved. Flow rates often have to be reduced in order to avoid the formation of slugs, which results in big losses as the total production will be affected. By studying slug initiation, it will be possible to find alternative solutions that will avoid the occurrence of slugging.

1.7 Relevance of research to doctoral project

Apart from its relevance to the industry, this research is also relevant to a doctoral project. In order to achieve the objectives set, it will be necessary to review the existing literature in the subject and identify the differences between the authors' approaches. It will be desirable to develop new models with a wider range of applicability. Implementation of models will be also part of the research, as it will allow simulation and validation of these models, and give more insight into the processes. Also the use of commercial 2D and 3D codes will be investigated, and results will be compared with 1D simulations and directly with experimental results.

References

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2010), Comparison between 2-D CFD and 1-D code for wave growth simulations, *in* 'International Conference on Multiphase Flow', University of Florida. (cited at page 3)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2011), Use of 2-D CFD for simulating two-phase flows in horizontal pipes, *in* 'International Association of Science and Technology for Development, Applied Simulation and Modelling'. (cited at page 4)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012a), 'Y-dependent wave growth analysis for VOF model', *Multiphase Science and Technology* . In preparation. (cited at page 4)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012b), 'Adjustment of velocity profile coefficients for one-dimensional multiphase flow', *Journal of Computational Physics* . In preparation. (cited at page 3)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012c), 'Comparison between 2D CFD and experiments for slug flow', *International Journal of Multiphase Flow* . In preparation. (cited at page 4)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012d), 'Use of

OpenMP to parallelise a one-dimensional multiphase code', *Computer Physics Communication* . In preparation. (cited at page 3)

Manolis, I. G. (1995), High Pressure Gas-Liquid Slug Flow, PhD thesis, Department of Chemical Engineering and Chemical Technology, Imperial College of Science, Technology and Medicine, UK. (cited at page 3)

Chapter 2

Methodology

2.1 Introduction

The most important, and arguably the most demanding, part of this research is a proper understanding of the relevant subject literature, and the continuous updating and assessment of new publications. The literature review presented here (chapter 3) tries to be exhaustive by giving the reader the ability to quickly grasp the main concepts involved. Proofs of theorems/equations are also given when relevant, or appropriate references are mentioned.

It was also necessary to learn and become an expert user of various software tools in order to carry out the necessary research. The software used include:

- EMAPS(Eulerian Multi-phase Adaptive Pipeline Solver), a software developed in Cranfield University for multiphase flow simulations
- OLGA®, a commercial software originally developed by the Institute for Energy Technology (IFE) in 1983 for Statoil

- FLUENT by ANSYS, a CFD commercial code
- StarCCM+, another CFD commercial code.

A more detailed view of the structure of EMAPS is shown in section 4.1, and the model used in FLUENT is explained in section 3.14.

The environment where the above software tools were operated is Linux. It became also necessary to write scripts in *bash* language in order to facilitate post-processing. Some of such scripts are included in appendix D.

Results using EMAPS and FLUENT will be presented, as these were used to produce a statistically significant number of simulations.

2.2 EMAPS

EMAPS (Eulerian Multiphase Adaptive Pipeline Solver) is the main software with which the one-dimensional simulations shown in this research were conducted. It was written (and rewritten!) in Fortran F90 and it is a one-dimensional fluid code that can simulate single-phase, two-phase and three-phase problems.

EMAPS files are classified into two types: Sources and Models. Ideally the user, when a new model is developed, would write a new model in the Models section and thus update EMAPS without having to change the Sources (which include the solvers). In reality, most new models required some changes to the sources as well, and this situation has created a variety of "branches" of EMAPS running in parallel and diverging.

One of the issues encountered in the early stages of this work, was that the development of EMAPS had been running along diverging branches. Some models were based on version 3.50, where others were based on version 3.60, which

contained also pipe geometry parameters. Moreover adaptivity was also being developed as a separate sub-branch of version 3.60, and some models required changes to be made to the sources rather than just the creation of new models.

A series of tests existed for version 3.50, and a very limited number of tests for version 3.60. The tests run as scripts that would be executed for given input files, and the output generated by the executable being tested would be either compared numerically with the expected solutions or the user would be asked to compare the solutions visually. In the numerical comparison a fail would result if a difference of more than 0.5% were calculated, while tests based on optical comparison are based on user decision.

A new version of EMAPS had to be created, which "ported" older models from EMAPS 3.50 to EMAPS 3.60, and brought some other necessary changes to the source code in line with code optimisation, together with the addition of adaptivity. A whole set of test scripts and test cases was written and the code was set permanently under version control (SVN), in order to avoid repeats of divergent branches. This transformation was very useful when parallelisation was carried out on EMAPS. A full OpenMP parallelisation was then carried out, together with new tests specifically for the parallel version.

2.3 Imperial College Data

The department of Chemical Engineering at Imperial College, London are running a project where they use a high-pressure WASP (Water, Air, Sand and Petroleum) rig with a pipe with a length of 38m and a diameter of 78mm for multiphase experiments. A series of experimental cases with recorded slug flow has been carried out and a detailed catalogue created (Manolis, 1995). Simulations were

also carried out in Imperial College with a code called TRIOMPH where the aim was to reproduce the experimental results.

One the objectives of the research is to test the range of applicability of EMAPS. The applicability was tested by performing two investigations: analysing the results of EMAPS simulations of experimental cases and analysing any differences between EMAPS and TRIOMPH. The results of the comparison with experimental data is reported in section 4.6. Regarding the comparison with TRIOMPH, it was necessary to investigate a relatively simple case and compare the results. A comparison had already been carried out previously (Valluri and Spelt, 2006) and some differences had been shown. The comparison involved running a Watson wave growth test case and analysing the results. New results are shown in section 6.2.

2.4 BP Field Data

Field data were requested and obtained from BP. BP has kindly offered to Cranfield University a plethora of data, comprising the 1987 and 1989 Prudhoe Bay Field trials. These stem from a series of tests carried out by BP Research Centre Sunbury and BP Engineering in conjunction with BP America in August/September in 1987 at Prudhoe Bay, Alaska.

Slug flow was observed in various pads (where a pad is a drill site consisting of a multitude of oil wells), and the level of details provided by the data gives the possibility for simulations to be carried out. In order to avoid complications arising due to elevations, and to limit the occurrence of terrain slugs, it was important to choose a horizontal (as much as possible) section of the pipeline, over which to run the simulation. Two pads in particular were chosen for simulation

with EMAPS: X pad and R pad (sections 4.7.1, 4.7.2). Due to confidentiality, it is not possible to include the full details from the field data. However useful qualitative comparisons can be made with the BP data.

2.5 FLUENT

FLUENT is a commercial computational fluid dynamics (CFD) software that is used in a wide range of applications, including multiphase flow. More details about the approaches used are explained in section 3.14. Two-dimensional simulations of two-phase flow in channels were carried out using FLUENT, both as an initial investigation using the wave growth problem as a validation benchmark (section 6.4), where a full mathematical perturbation analysis was also completed (section 6.5), and also as a series of simulations of experimental slug cases. Results and considerations are explained in chapter 7.

2.6 Industrial analysis

The first year of the Master of Business Administration (MBA) in the School of Management in Cranfield was completed during the first 12 months of the EngD. The courses completed are:

- Accounting
- Economics of Organisations and Strategy
- Financial Management
- Macroeconomics Analysis and Business Environment

- Managing Information Systems
- Operations Management
- Organisational Behaviour/Personal & Professional Development
- People Management
- Project Management Introduction
- Strategic Decision Science
- Strategic Management
- Strategic Marketing
- Supply Chain Management

The course provided a breadth of knowledge that was vital because it allowed the relation between the project and the applications of the results obtained in industry, in particular BP and other oil companies facing similar issues.

In chapter 8, first an overview of BP is given, together with how slug phenomenon affects its operations. A general analysis of oil economics is also included, in order to contextualise the importance of understanding slug flow. Oil companies and/or companies involved in pipe design can decide to undertake case and feasibility studies by choosing one of the various methods discussed in the current thesis. It was important to carry out a market analysis of the products currently used, and also an investigation of the project costs and implementation costs for BP will be completed.

References

Manolis, I. G. (1995), High Pressure Gas-Liquid Slug Flow, PhD thesis, Department of Chemical Engineering and Chemical Technology, Imperial College of Science, Technology and Medicine, UK. (cited at page 13)

Valluri, P. and Spelt, P. (2006), TMF3 Sub-project 1: Multifluid modelling: Comparison between TRIOMPH and EMAPS, *in* 'TMF3', Department of Chemical Engineering, Imperial College, London. (cited at page 14)

REFERENCES

Chapter 3

Literature Review

3.1 Introduction

Before looking at slug flow in more detail, it is important to have an overview of the fields of fluid mechanics and fluid dynamics. Here some theory behind fluid motion will be briefly explained, before moving on to equations for two-phase flow, including the Watson model and the single-pressure model (SPM4s), which will be the most used model in the 1D software EMAPS. The numerical solver used in the SPM4s model will also be explained in detail.

Sections on slugs will follow, including slug characteristics, slug initiation and stability models and discussions regarding one of the most important features of slug flow: slug frequency.

As the commercial code FLUENT with the volume of fluid (VOF) model will be used further on, a section on the VOF model is also included.

3.2 Fluids

A fluid is defined as a substance that will move if a shear force or stress is applied to it. Fluids are normally divided into two groups: liquids and gases. Gases do not have a pre-defined volume or shape, and for a given system of gas, the gas equation of state relates the variables of pressure, temperature and volume. Liquids, on the other hand, possess volume, but normally have very little compressibility and density varies only marginally with temperature or pressure.

There are two main descriptions of fluids: Eulerian and Lagrangian. In the Lagrangian approach an individual particle is being tracked throughout the fluid, while the Eulerian approach tracks the history at a specific point in space, regardless of the specific fluid particle present there at a specific time. In the current project we will only deal with Eulerian descriptions.

3.3 Variables and definitions

An important assumption that will be used is that the fluid can be treated as a continuum, i.e. the mean free path (distance between molecules) is small compared with the physical dimensions of the problem on hand.

Fluid flow can be fully determined by specifying the three-dimensional velocity vector (three components) and two thermodynamic properties, which can include temperature, pressure, density, enthalpy and entropy among others. Normally five independent equations, which comprise of three components of the equation of motion, a continuity equation and an energy equation, should suffice in order to solve the problem being investigated. However in fully turbulent flows, the range of scales makes the system too computationally intensive to solve exactly and the equations must be averaged (RANS) or filtered (LES), thus additional

terms arise and the equations need to be closed using a suitable turbulence model (RANS) or a subgrid model for unresolved scales (LES).

Pressure is one of the variables in fluid flow. In fluid statics, pressure is defined as the normal stress or force per unit area acting on a surface inside the fluid, and it is also known as hydrostatic pressure, and in a certain point it is isotropic in the fluid at rest. The problems investigated in this project deal with fluid dynamics, and in these cases pressure is accompanied by shear forces and stresses. Therefore when defining isotropic pressure it is necessary to take into account extra forces due to viscosity effects.

Viscosity is an indication of a fluid's internal resistance to flow and it is a measure of its resistance to shear stress (Symon, 1971). For Newtonian fluids, which are the ones investigated in the current thesis, the following linear relationship between the shear stress τ and the velocity gradient $\partial u/\partial y$ exists:

$$\tau = \mu \frac{\partial u}{\partial y} \tag{3.1}$$

where μ is the viscosity. Viscosity increases with increasing temperature for gases, while the opposite trend holds for liquids. Viscosity and turbulence can cause shear stresses and therefore friction in a fluid.

When modelling a flow problem, it will be important to consider whether surface tension effects are important. For example in a gas-liquid system, surface tension on the liquid surface is caused by the fact that cohesive forces between liquid molecules near the interface are stronger than the liquid-gas forces. Surface tension is related to the difference in pressure at the interface through the Young-Laplace equation (Young, 1992) $\Delta p = -\gamma \nabla \cdot \hat{n}$ where γ is the surface tension, Δp is the pressure drop across the fluid interface and $\hat{n}$ is the unit normal pointing out of the surface.

3.4 Equations of fluid motion

Most problems can be solved by starting from three fundamental laws:

- Mass conservation
- Newton's second law $F = ma$
- First law of thermodynamics - conservation of energy.

It is assumed that the laws are applied to a fixed control volume, even though the fluid inside is continuously changing.

The methodology will be that of deriving an integral form of the equation, and then finding the corresponding differential equation form.

The Reynolds transport theorem will be used to derive the equations of motion. It states that sum of changes of an intensive property over a control volume has to be equal to any gains or losses through the boundaries of the volume plus the effects from any sources/sinks inside the volume, as shown in Eq. 3.2

$$\frac{d}{dt}\int_{\Omega}\eta dV = -\int_{\partial\Omega}\eta\vec{v}\cdot\vec{n}dA - \int_{\Omega}Q_{\eta}dV \tag{3.2}$$

where Ω is the control volume, $\partial\Omega$ is the bounding surface, η is the intensive property, Q_{η} is the sum of sources and sinks (relevant to η) inside the volume and v is the velocity of the fluid.

Applying the divergence theorem to the surface integral, we obtain:

$$\frac{d}{dt}\int_{\Omega}\eta dV = -\int_{\Omega}\nabla\cdot(\eta\vec{v})\,dV - \int_{\Omega}Q_{\eta}dV \tag{3.3}$$

Further applying Leibniz's rule and combining all the integrals:

$$\int_{\Omega}\left(\frac{\partial\eta}{\partial t} + \nabla\cdot(\eta\vec{v}) + Q_{\eta}\right)dV = 0 \tag{3.4}$$

Because the integral must be zero for any control volume, then also the integrand has to be zero. Therefore:

$$\frac{d\eta}{dt} + \nabla\cdot(\eta\vec{v}) + Q_\eta = 0 \tag{3.5}$$

3.4.1 Mass Equation

Since there is no mass transfer, then there are no sources or sinks of mass inside the fluid, and we can set $Q_\eta = 0$, and $\eta \equiv \rho$ where ρ is the fluid density. The resulting equation is called the continuity equation:

$$\frac{\partial\rho}{\partial t} + \nabla\cdot(\rho\vec{v}) = 0 \tag{3.6}$$

For incompressible fluid, ρ is constant and the continuity equation reduces to:

$$\nabla\cdot\vec{v} = 0 \tag{3.7}$$

3.4.2 Momentum Equation

For the momentum equation, we define $\eta \equiv \rho\vec{v}$, and so the equation becomes:

$$\frac{\partial}{\partial t}(\rho\vec{v}) + \nabla\cdot(\rho\vec{v}\vec{v}) + Q_\eta = 0 \tag{3.8}$$

where $\vec{v}\vec{v}$ is a dyadic tensor product.

3.4.3 Stresses

Body forces are composed of stresses, which can be normal and/or shear stresses, and other body forces, usually gravity and sometimes also superficial tension. Stresses are normally written as σ_{ij} where i is the face on which the stress acts, and j is the direction of the stress, as shown in Fig. 3.1. Thus at each point in a

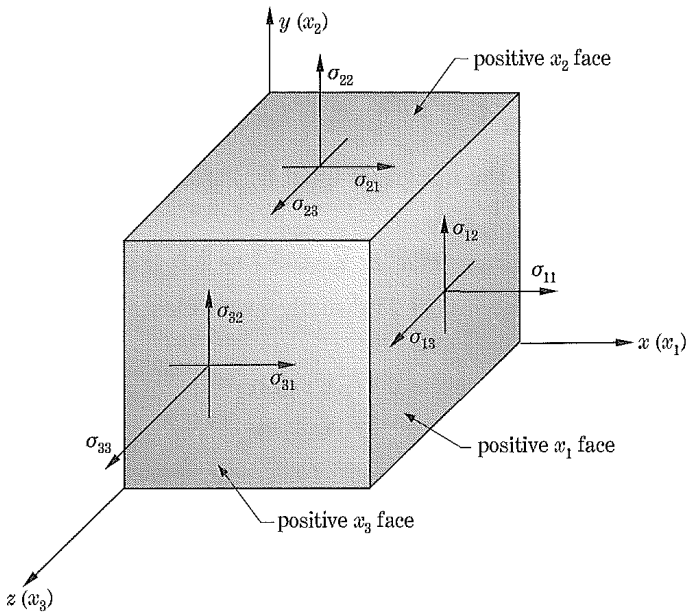


Figure 3.1: Illustration of stresses in space. The stresses on the negative faces (not shown) are equal and opposite to the ones on the positive faces (Hughes and Brighton, 1991).

fluid there is a defined array, or tensor, σ_{ij} with nine components and which is a function of $\vec{r}$ and t, as shown below:

$$\sigma_{ij} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \tag{3.9}$$

Although there are nine total elements, actually there are only six independent components, as the stress tensor has to be symmetric, i.e. $\sigma_{ij} = \sigma_{ji}$. If it was not symmetric, then the infinitesimal elemental volume would have to rotate with infinite angular velocity. The terms σ_{11}, σ_{22} and σ_{33} are the normal stresses, while the other terms are shear stresses. Moreover the stress tensor σ_{ij} can be split into two stress tensors: a mean hydrostatic stress tensor $p\delta_{ij}$ which tries to change

the volume of the stressed body, and a stress deviator tensor T_{ij} which tries to distort it. Thus $\sigma_{ij} = T_{ij} - p\delta_{ij}$ where $p = -(\sigma_{11} + \sigma_{22} + \sigma_{33})/3$ is the mechanical pressure and which in case of frictionless flow is equal to the isotropic pressure.

Thus the momentum equation Eq. 3.8 can be rewritten as:

$$\frac{\partial}{\partial t}(\rho\vec{v}) + \nabla\cdot(\rho\vec{v}\vec{v}) = -\vec{\nabla}p + \vec{\nabla}\cdot T + \vec{f} \tag{3.10}$$

where f are other forces, including gravity and also surface tension force.

3.5 1-D Conservation Equations for two-phase flow

The main subject of study is concentrated around flows in pipelines, where the fluid motion is mostly on one single dimension, so it can be assumed that the velocity is in the x direction, i.e. $\vec{v} = (v(x,t), 0, 0)$. By integrating the equations of motion over a cross section it is possible to obtain a one-dimensional fluid model (Chan and Banerjee, 1981). We want to model one-dimensional flow for

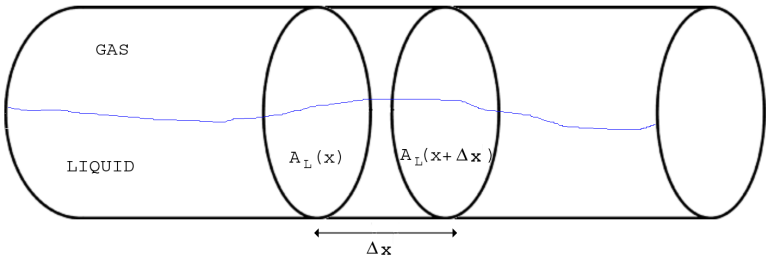


Figure 3.2: Schematic representation of two-phase through pipe.

gas and liquid in a pipe, as shown summarily in Fig. 3.2. We start again from Eq. 3.2, and we choose a control volume bounded by pipe wall section S_w and

two surfaces $A(x)$ and $A(x+\Delta x)$ with normal directions parallel to the pipe axis. By taking Δx sufficiently small, it is possible to approximate

$$\frac{d}{dt}\int_{\Omega}\eta dV \approx \Delta x\frac{\partial}{\partial t}\int_{A(x)}\eta dA \tag{3.11}$$

and

$$\int_{\partial\Omega}\eta\vec{v}\cdot\vec{n}dA = \int_{A(x+\Delta x)}\eta v dA - \int_{A(x)}\eta v dA \tag{3.12}$$

Therefore the conservation equation can be written as:

$$\Delta x\frac{\partial}{\partial t}\int_{A(x)}\eta dA + \int_{A(x+\Delta x)}\eta v dA - \int_{A(x)}\eta v dA = B \tag{3.13}$$

where B are sources/sinks.

3.5.1 Mass Conservation Equation

For the mass conservation equation, $\eta = \rho_k$ where $k = L, G$ depending if it is the density for liquid or gas, and there is no mass transfer, then there are no sinks or sources, and therefore $B = 0$. Thus dividing Eq. 3.13 by Δx and taking the limit $\Delta x \to 0$ we obtain for the liquid phase:

$$\frac{\partial}{\partial t}\left(\rho_L A_L\right) + \frac{\partial}{\partial x}\left(\rho_L v_L A_L\right) = 0 \tag{3.14}$$

where v_L is the mean velocity in the liquid phase obtained from the equation:

$$A_L v_L = \int_{A_L} v dA \tag{3.15}$$

Similarly for the gas phase, the mass conservation equation is:

$$\frac{\partial}{\partial t}\left(\rho_G A_G\right) + \frac{\partial}{\partial x}\left(\rho_G v_G A_G\right) = 0 \tag{3.16}$$

where

$$A_G v_G = \int_{A_G} v dA \tag{3.17}$$

Because the pipe is assumed to be of constant diameter and therefore also of constant cross-section A, then all A_k terms in the mass and momentum equations above can be substituted by corresponding hold-up terms α_k, where α_k=A_k/A. Therefore:

$$\frac{\partial}{\partial t}\left(\rho_k \alpha_k\right) + \frac{\partial}{\partial x}\left(\rho_k v_k \alpha_k\right) = 0 \tag{3.18}$$

3.5.2 Momentum Conservation Equation

For the momentum conservation equation we substitute $\eta = \rho_k v_k$. Starting from the liquid phase, where $k = L$, and defining the velocity profile coefficient as:

$$C_{VL} = \frac{1}{A_L v_L^2} \int_{A_L} v^2 dA \tag{3.19}$$

then for sufficiently small Δx Eq. 3.13 becomes:

$$\Delta x \left[\frac{\partial}{\partial t}\left(\rho_L v_L A_L\right) + \frac{\partial}{\partial x}\left(C_{LV} \rho_L v_L^2 A_L\right)\right] = B \tag{3.20}$$

In the original version of EMAPS it is assumed that $C_{VL} = 1$. In chapter 5 updated values of C_{VL} will be used.

The term B comprises the forces acting on the control volume in the liquid phase, and they include: forces due to wall and interfacial shear stresses, viscous forces and normal forces due to pressure.

$$B = F_{pressure} + F_{viscous} + F_{shear} \tag{3.21}$$

The shear forces are equal to:

$$F_{shear} = \Delta x(-\tau_{Lw} S_W + \tau_{Li} S_I) \tag{3.22}$$

where τ_{Lw} and τ_{Li} are the wall and interfacial shear stresses respectively, and S_W is the wetted perimeted (i.e. the perimeter of the cross-sectional area "touched" by the liquid) and S_I is the interfacial perimeter.

Again, because the pipe is assumed to be of constant diameter and therefore also of constant cross-section A, then all A_k terms in the mass and momentum equations above can be substituted by corresponding hold-up terms α_k, where α_k=A_k/A.

The momentum equation can be rewritten with various extra terms, including gravity in case of pipe inclination (Ishii, 1975):

$$\begin{gathered}\frac{\partial}{\partial t}\left(\rho_k \alpha_k v_k\right) + \frac{\partial}{\partial x}\left(\rho_k \alpha_k {v_k}^2\right) = \\ -\alpha_k \frac{\partial P_k}{\partial x} - \Delta P_{ki} \frac{\partial \alpha_k}{\partial x} + \frac{\partial}{\partial x}\left[\alpha_k \left(\tau_k + {\tau_k}^{Re}\right)\right] + \\ M_{kw} + M_{ki} + \Gamma_k v_{ki} - \rho_k \alpha_k g \sin\beta\end{gathered} \tag{3.23}$$

The variables ρ_k, P_k, α_k and v_k are the fluid density, pressure, volume fraction and velocity of phase k respectively. The term β is the pipe inclination relative to a horizontal level. The parameter ΔP_{ki} is the pressure correction term (arising from the liquid hydrostatic pressure contribution), and τ_k and ${\tau_k}^{Re}$ are the viscous stress and the Reynolds stress. The terms M_{ki} and M_{kw} are the interfacial and wall shear stresses, while Γ_k is the mass transfer term and v_{ki} is the interfacial velocity for each phase.

Since it is known that the total volume fraction of the two phases is equal to one, then the following equation holds:

$$\sum_k \alpha_k = 1 \tag{3.24}$$

The mass and momentum interfacial jump conditions give the following conditions equations:

$$\sum_k \Gamma_k = 0 \tag{3.25}$$

$$\sum_k M_{ki} + \Gamma_k v_{ki} = 0 \tag{3.26}$$

where k takes the value G or L depending on the phase, and i refers to the interfacial term.

Equations 3.18 to 3.26 can model two-flow systems that can be approximated by one-dimensional flow. The term τ_k^{Re} is not used here due to its limited effects in 1D modelling (Park et al., 1998), and moreover the viscous stress τ_k and the mass transfer Γ_k are considered to have a negligible contribution to the total flow and thus will be disregarded in this analysis.

Therefore finally we have a set of 6 equations and 14 unknowns. In order to be able to find some solutions to this set of equations, it is necessary to use constitutive relations. A summary of the forms of constitutive relations that are used in this analysis are shown in the next section.

3.5.3 Closure Laws

The main closure laws used in modelling two-fluid flow are: pressure terms, virtual mass terms and wall shear stress terms.

3.5.3.1 Pressure Terms

The pressure correction term can be defined through the formula $\Delta P_{ki} = P_k - P_{ki}$ and features in the following relation:

$$\alpha_k \frac{\partial P_k}{\partial x} + \Delta P_{ki} \frac{\partial \alpha_k}{\partial x} = \frac{\partial (\alpha_k P_k)}{\partial x} - P_{ki} \frac{\partial \alpha_k}{\partial x} \tag{3.27}$$

As it is necessary to have extra information in order to solve for the four pressure terms, hereby follows a brief summary of different models used.

In case of isothermal flows it can be assumed that the phase pressure depends only on fluid density, i.e. $P_k = P_k(\rho_k)$. If the pressure is equal in both phases, then the

model becomes the single-pressure model (SPM4s). If on the other hand there is a non-negligible pressure difference between the phases, then it is necessary to introduce an extra term representing a local constitutive relation: this model is usually referred as the two-pressure model (TPM)(Ransom and Hicks, 1984).

The interfacial phase pressure (indicated by P_{ki}) is related to the surface tension, and depends on the flow pattern. The following equations describe the interfacial phase pressure in the stratified flow regime (Barnea and Taitel, 1993):

$$P_{Gi} - P_{Li} = \sigma \frac{\partial h_L^2}{\partial x^2} \tag{3.28}$$

where σ is the surface tension and h_L is the height of the liquid in the pipe. For bubbly flow the following equation(Drew and Passman, 1999) was developed:

$$P_{Gi} - P_{Li} = \frac{2\sigma}{r_B} \tag{3.29}$$

where r_B is the curvature radius. In the simple case where the gas and liquid interface pressure are assumed to be equal, then the surface tension can be neglected, and thus the following relation holds:

$$P_{Gi} = P_{Li} = P_i \tag{3.30}$$

The pressure correction term ($\Delta P_{ki} = P_k - P_{ki}$) has various different expressions, depending on the author. For stratified flow, an expression widely used is (Barnea and Taitel, 1993):

$$\frac{\partial(\alpha_k \Delta P_{ki})}{\partial x} = \alpha_k \rho_k g \cos\beta \frac{\partial h_L}{\partial x} \tag{3.31}$$

where β is the angle of the pipe with the horizontal and h_L is the height of the liquid.

3.5.3.2 Interfacial Stress Terms

The interfacial stress term M_{ki} represents stresses on the interface. It is a linear combination of the following physical forces:

$$M_{ki} = M_{ki}^{D} + M_{ki}^{V} + M_{ki}^{B} + M_{ki}^{L} + M_{ki}^{C} \tag{3.32}$$

where D, V, B, L and C refer to steady-state drag, virtual mass, Basset, lift and collision forces respectively. The only forces to be considered usually are the interfacial drag and virtual masses forces, as the effect of the others is small (Chung et al., 1985) or not well known (Ishiima and Mishima, 1984). Moreover in the two-phase models that we will analyse, the virtual mass force is not considered.

The interfacial shear force has a strong dependency on the relevant flow regime. A general expression has been devised by Ishiima and Mishima (1984) who suggested modelling as follows:

$$M_{ki}^{D} = \langle -\tau_{ki} \cdot \nabla \alpha_k \rangle_x + \overline{M}_{ki} \tag{3.33}$$

The first term on the right-hand side represents the effect of the interfacial shear and the volume fraction, and is relevant for separated flow. The second term is the area-averaged particle drag and is important for dispersed flow.

3.5.3.3 Wall Shear Stress Terms

The wall shear stress refers to the stress acting on the phase at the wall. The usual formulation of the wall shear stress term, indicated by M_{kw} is (Levy, 1999):

$$M_{kw} \equiv T_{kw} = -\frac{\tau_k S_k}{A} \tag{3.34}$$

where S_k is the part of the wall in contact with the phase k, over an area A, and τ_k is the wall shear stress of the same phase, defined as

$$\tau_k = \frac{1}{2} f_k \rho_k v_k |v_k| \tag{3.35}$$

and f_k is the wall friction factor.

3.6 Burger Model

The Burger model is a basic model, based on the inviscid Burgers equation (Toro, 1997), which is a single non-linear equation shown below:

$$\frac{\partial v}{\partial t} + \frac{\partial \left(v^2/2\right)}{\partial x} = 0 \tag{3.36}$$

where v is the fluid velocity. It is normally possible to find an analytical solution for the above expression, and in fact the Burger model is used for validation purposes.

3.7 Watson model

Due to the non-conservative nature of the equations that underline most two-fluid models, it is advisable to use a specific two-fluid model. Watson (1990) designed such a model, which greatly reduced the numerical complexity required.

It is assumed that the gas and liquid phases are incompressible, the pipe is circular with a diameter D and inclined at an angle β to the horizontal, and that the fluids flow as a two-phase mixture in a gravitationally separated configuration.

The gas and liquid mass balance equations are summed to give a total mass conservation equation:

$$\frac{\partial}{\partial t}\left(\rho_L \alpha_L + \rho_G \alpha_G\right) + \frac{\partial}{\partial x}\left(\rho_L \alpha_L v_L + \rho_G \alpha_G v_G\right) = 0 \tag{3.37}$$

By combining the gas and liquid momentum the following equation results:

$$\frac{\partial}{\partial t}\left(\rho_L v_L - \rho_G v_G\right) + \frac{\partial}{\partial x}\left(\frac{1}{2}\rho_L v_L^2 - \frac{1}{2}\rho_G v_G^2 + \left(\rho_L - \rho_G\right) g \cos\beta h_L\right) = H \quad (3.38)$$

where

$$H = -\left(\rho_L - \rho_G\right) g \sin\beta + \left(\frac{1}{A_L} + \frac{1}{A_G}\right)\tau_I S_I + \frac{\tau_G S_G}{A_G} - \frac{\tau_L S_L}{A_L} \quad (3.39)$$

where S_I is the interfacial wetted perimeter while S_G and S_L are the gas and liquid wetted perimeters. The parameters τ_G and τ_L are the gas and liquid wall shear stresses respectively, while τ_I is the interfacial shear stress.

Two more conditions hold for this model. The first one is the geometric constraint that the total area occupied by the two phases must be equal to the total cross-sectional area of the pipe, i.e.

$$A_L + A_G = A \text{ or } \alpha_L + \alpha_G = 1 \quad (3.40)$$

The second condition is derived from the fact that the phases are assumed to be incompressible, and so by dividing the mass conservation equations by the appropriate density we obtain:

$$\frac{\partial}{\partial x}\left(\alpha_L v_L + \alpha_G v_G\right) = 0 \quad (3.41)$$

which can be expressed as:

$$\alpha_L v_L + \alpha_G v_G = Q(t) \quad (3.42)$$

where $Q(t)$ is a known function of time, dependent upon the inlet flow rates.

Thus eventually we need to solve only two differential conservative equations and two algebraic equations. This system is suitable for stratified flows and the computational times are acceptable.

3.8 Single Pressure Model

The single pressure model (SPM4s) treats the liquid phase as being incompressible, and it treats the gas phase as possessing a certain degree of compressibility, according to the thermodynamic equation of state. This model can be described by the following conservation equations:

- Mass Conservation

$$\begin{aligned} \frac{\partial(\rho_G \alpha_G)}{\partial t} + \frac{\partial(\rho_G \alpha_G v_G)}{\partial x} &= 0 \\ \frac{\partial(\rho_L \alpha_L)}{\partial t} + \frac{\partial(\rho_L \alpha_L v_L)}{\partial x} &= 0 \end{aligned} \tag{3.43}$$

- Momentum Conservation

$$\begin{aligned} \frac{\partial(\rho_G \alpha_G v_G)}{\partial t} + \frac{\partial(\rho_G \alpha_G v_G^2)}{\partial x} &= -\alpha_G \frac{\partial P}{\partial x} + B_{fG} + T_I + T_{Gw} \\ \frac{\partial(\rho_L \alpha_L v_L)}{\partial t} + \frac{\partial(\rho_L \alpha_L v_L^2)}{\partial x} &= -\alpha_L \frac{\partial P}{\partial x} - P_c \frac{\partial \alpha_L}{\partial x} + B_{fL} - T_I + T_{Lw} \end{aligned} \tag{3.44}$$

T_{kw} represents the wall shear stress for a phase k, T_I is the interfacial shear stress, and B_{fk} is the corresponding gravity force given by $-\rho_k \alpha_k g sin\beta$. The gas pressure correction term has been dropped, and P_c is the liquid pressure correction term, given by

$$P_c = \rho_L \alpha_L g cos\beta \frac{dh_L}{d\alpha_L} \tag{3.45}$$

and h_L is the height of the liquid in case of stratified flow.

3.9 Numerical solver for one-dimensional two-phase flow model

The numerical solver used in most of the one-dimensional simulations carried out with EMAPS is called $AUSMDV^*$. $AUSMDV^*$ evolved from a new class of upwind schemes that are known to be remarkably robust and stable for a variety of multiphase flow fields. These Advection Upstream Splitting Methods ($AUSM$) were first developed by Liou and Steffen (1993) and Wada and Liou (1994). Later on, in order to solve viscous flows at all speeds, improved versions of $AUSM$-family schemes were introduced in Liou (1996) and Liou (2006) and tested in different situations in Tiselj and Petelin (1997), Mary et al. (2000), Evje and Fjelde (2003), Evje and Flatten (2003) and Garcia-Cascales and Paillere (2006), amongst others. These schemes were called $AUSM+$, $AUSM+-up$ and $AUSMDV$.

In Evje and Flatten (2003), a hybrid version named $AUSMDV^*$ was tested. The authors showed that although this numerical scheme does not offer a high level of robustness at quick transitions, it is efficient and accurate with reduced computational cost, and is suitable to simulate slow transitions occurring in multi-phase flows in pipes. Therefore it was decided that the $AUSMDV^*$ scheme would be ideally suited to the requirements of the current thesis.

It is normal practice to express the continuity and momentum equations (Eq. 3.44 and 3.43) in compact form.

Let U be the vector of unknown fields, $F(U)$ the flux vector, $H(P)$ the non-conservative coupling matrix and source vector $S(U)$ be the inter-phase and wall friction term:

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = H(P) \cdot \frac{\partial U}{\partial x} + S(U, Q) \tag{3.46}$$

The common step to $AUSM$-family schemes is to decompose the flux vector into convective and pressure components: $F(U) = F^c(U) + F^P(U)$ or

$$F^c(U) = \begin{pmatrix} \alpha_G\rho_G \cdot u_G \\ \alpha_G\rho_G \cdot u_G^2 \\ \\ \alpha_L\rho_L \cdot u_L \\ \\ \alpha_L\rho_L \cdot u_L^2 \end{pmatrix} \quad \text{and} \quad F^P(U) = \begin{pmatrix} 0 \\ \\ \alpha_G \cdot P \\ \\ 0 \\ \\ \alpha_L \cdot P \end{pmatrix}$$

One starts to construct $AUSMDV^*$ scheme by using the generic form of the convective and pressure flux for each phase k, as follows

$$F_k^c(U) = \begin{pmatrix} \alpha_k\rho_k \cdot u_k \\ \\ \alpha_k\rho_k \cdot u_k^2 \end{pmatrix} \quad \text{and} \quad F_k^P(U) = \begin{pmatrix} 0 \\ \\ \alpha_k \cdot P \end{pmatrix}$$

The index k is then dropped for simplicity, and the discretisation of the system (3.46) is considered at the cell interfaces $j + 1/2$, which results in the following relations:

$$F_{j+1/2}^c = \begin{pmatrix} (\alpha\rho)_j \cdot \widetilde{u}_j^+ + (\alpha\rho)_{j+1} \cdot \widetilde{u}_{j+1}^- \\ \\ s_f \cdot (\alpha\rho \cdot u^2)_V + (1 - s_f) \cdot (\alpha\rho \cdot u^2)_D \end{pmatrix}$$

$$F^P_{j+1/2} = \begin{pmatrix} 0 \\ \\ (\alpha P)_j \cdot P^+_j + (\alpha P)_{j+1} \cdot P^-_{j+1} \\ \\ \end{pmatrix}$$

Indices $(\alpha\rho\cdot u^2)_V$ and $(\alpha\rho\cdot u^2)_D$ stand for the discretisation of the term $\alpha\rho\cdot u^2$ with $AUSMV$ and $AUSMD$ schemes respectively (Trepanier et al., 1991), and these are given in detail in the next section.

3.9.1 Convective flux discretisation

$$\begin{cases} (\alpha\rho\cdot u)_{1/2} &= (\alpha\rho)_j \cdot \widetilde{u}^+_j + (\alpha\rho)_{j+1} \cdot \widetilde{u}^-_{j+1} \\ \\ (\alpha\rho\cdot u^2)_V &= (\alpha\rho u)_j \cdot \widetilde{u}^+_j + (\alpha\rho u)_{j+1} \cdot \widetilde{u}^-_{j+1} \\ \\ (\alpha\rho\cdot u^2)_D &= \dfrac{1}{2}\Big[(\alpha\rho u)_{1/2} \cdot (u_{j+1} + u_j) - |\,(\alpha\rho u)_{1/2}\,| \cdot (u_{j+1} - u_j)\Big] \end{cases} \tag{3.47}$$

The velocities splitting needed in $AUSMDV^*$ is

$$\widetilde{u}^+_j = \begin{cases} \xi_j \cdot \dfrac{\left(u_j + \widetilde{c}_{1/2}\right)^2}{4} + (1-\xi_j) \cdot \dfrac{(u_j + |u_j|)}{2} & \text{if } u_j \leq \widetilde{c}_{1/2} \\ \\ \dfrac{(u_j + |u_j|)}{2} & \text{if } u_j > \widetilde{c}_{1/2} \end{cases}$$

and

$$\widetilde{u}_{j+1}^{-} = \begin{cases} -\xi_{j+1} \cdot \dfrac{\left(u_{j+1} - \widetilde{c}_{1/2}\right)^2}{4} + (1 - \xi_{j+1}) \cdot \dfrac{(u_{j+1} - |u_{j+1}|)}{2} & \text{if } u_{j+1} \leq \widetilde{c}_{1/2} \\ \dfrac{(u_j - |u_j|)}{2} & \text{if } u_{j+1} > \widetilde{c}_{1/2} \end{cases}$$

while the speed of sound is $\widetilde{c}_{1/2} = max(c_j, c_{j+1})$ and s_f is a switch function depending on the local volume fraction. The parameters ξ_j and ξ_{j+1} are problem density dependent, and they are weighted coefficients in order to ensure stability. They are given by:

$$\begin{aligned} \xi_j &= (1 - \phi_j) \cdot \frac{(\rho/\alpha)_j}{(\rho/\alpha)_j + (\rho/\alpha)_{j+1}} + \phi_j \\ \xi_{j+1} &= (1 - \phi_{j+1}) \cdot \frac{(\rho/\alpha)_{j+1}}{(\rho/\alpha)_j + (\rho/\alpha)_{j+1}} + \phi_{j+1} \end{aligned} \tag{3.48}$$

with $\phi_j = \left[e^{-\kappa_1 \cdot \alpha_g} + e^{-\kappa_2 \cdot \alpha_l}\right]_j$ and $\kappa_1 = 50, \quad \kappa_2 = 500$
and $s_f = max(\phi_j, \phi_{j+1})$. Parameters κ_1 and κ_2 are problem dependent (Liou and Wada, 1997). ϕ_j is a smooth function which is close to unity in single phase regions. The quantity s_f appears only in the convective flux and ensures a combination between AUSMV and AUSMD fluxes (stability and accuracy).

3.9.2 Pressure flux discretisation

A similar approach to the one above is performed for the pressure splitting that is based on the common speed of sound and is expressed by

$$P_j^+ = \begin{cases} \dfrac{1}{\widetilde{c}_{1/2}} \cdot \left(2 - \dfrac{u_j}{\widetilde{c}_{1/2}}\right) & \text{if } |u_j| \leq \widetilde{c}_{1/2} \\ \dfrac{1}{u_j} & \text{if } |u_j| > \widetilde{c}_{1/2} \end{cases}$$

and

$$P_{j+1}^- = \begin{cases} \dfrac{-1}{\widetilde{c}_{1/2}} \cdot \left(2 + \dfrac{u_{j+1}}{\widetilde{c}_{1/2}}\right) & \text{if } |u_{j+1}| \leq \widetilde{c}_{1/2} \\ \dfrac{1}{u_{j+1}} & \text{if } |u_{j+1}| > \widetilde{c}_{1/2} \end{cases}$$

3.9.2.1 Source terms discretisation

The index k representing the phase is restored, and source terms on the right hand side of the system 3.46 are discretised using the following scheme:

$$\Sigma_k \left(U, Q, P\right)_j = H_k(\widetilde{P}_j, \widetilde{\rho}_j) \cdot \left(\frac{\widehat{Q}_j - \widehat{Q}_{j-1}}{\Delta x}\right)_k + S_k(U_j, Q_j) \tag{3.49}$$

where Δx is the cell size and averaged fields $\widetilde{U}_j$ and $\widehat{Q}_j$ are expressed as

$$\begin{aligned} \widetilde{U}_j &= \frac{U_{j-1} + 2 \cdot U_j - U_{j+1}}{4} \\ \text{and} \qquad \widehat{Q_j} &= \frac{Q_j + Q_{j+1}}{2} \end{aligned} \tag{3.50}$$

$$Q_k = \begin{pmatrix} 1 \\ \\ \\ \alpha_k \end{pmatrix}, \qquad S_k(U, Q) = \begin{pmatrix} 0 \\ \\ \\ \pm \tau_i \cdot S_i - \tau_{kw} \cdot S_{kw} - \rho_k \cdot g sin\beta \end{pmatrix}$$

and the coupling sub-matrix is $H_k(P,\rho) = \begin{pmatrix} 0 & 0 \\ 0 & P_k - \rho_k \cdot g cos\beta \end{pmatrix}$

3.9.3 $AUSMDV^*$ numerical scheme

Collecting the different discretised components in Eq. 3.46, the solution is advanced explicitly in time intervals $t \in [t_n, t_{n+1}]$ with a time step Δt in the following manner:

$$\begin{aligned} U_{kj}^{n+1} &= U_{kj}^{n} - \frac{\Delta t}{\Delta x}\left[(F_{j+1/2}^{c} - F_{j-1/2}^{c}) + (F_{j+1/2}^{P} - F_{j-1/2}^{P})\right]_k \\ &+ \Delta t \cdot \Sigma_k\left(U_j, Q_j, P_j\right) \end{aligned} \tag{3.51}$$

Hence most steps in the $AUSMDV^*$ scheme used in EMAPS have been explained in detail, as this forms the basis of the one-dimensional simulation results carried out in this thesis.

3.10 Summary of slug characteristics

Slug flow is composed of a sequence of liquid slugs and large stratified gas/liquid zones. In the regime flow map it is located between stratified flow and dispersed bubbly flow. As flow rates increase, the liquid level increases and becomes wavier until most of the cross-section of the pipe is blocked by a wave. This accumulation of liquid is called slug, and it moves along the pipe, pushed by the gas flow. Behind the slug moves an elongated gas bubble over a thin liquid film. A schematic representation of slug is shown in Fig. 3.3. A slug unit is defined as

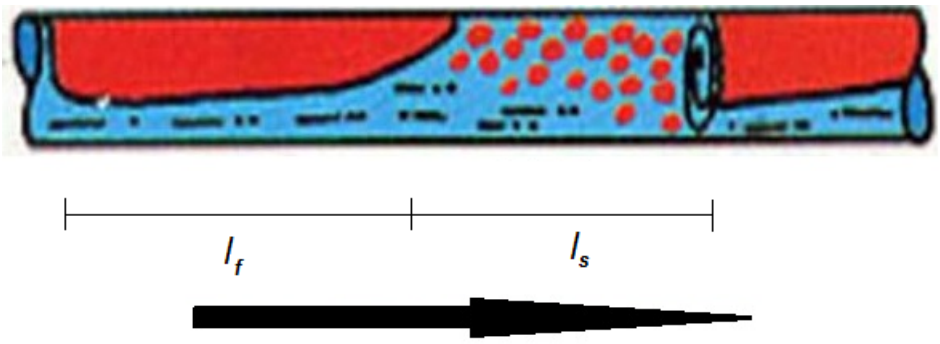


Figure 3.3: Schematic representation of a slug, moving from left to right, with gas in red and liquid in blue. Slug body length l_s and liquid film length l_f are also shown.

the combination of liquid film region and slug body region:

$$l_U = l_f + l_s \tag{3.52}$$

where l_U is the slug unit length, l_f is the length of the liquid film region, and l_s is the length of the slug body region.

In a circular pipe of constant diameter, the volumetric phase fraction α is defined as the ratio of the area occupied by the phase over the total area of the internal pipe:

$$\alpha_k = \frac{A_k}{A} \tag{3.53}$$

where k is the phase referred to.

3.10.1 Slug Translational Velocity

Normally in steady flow the slug translational velocity is equal to the velocity of the large Taylor bubble nose, and this also represents the slug tail velocity. Thus when the flow is fully developed, the slug front velocity is the same as the translational slug velocity.

In the work by Nicklin et al. (1962) the translational velocity is shown as a function of the mixture superficial velocity U_M (where mixture superficial velocity is the sum of the liquid and gas superficial velocities, and superficial velocity is defined as the product of the physical velocity and the phase volume fraction) and the drift velocity u_d, which is the velocity of a large bubble propagating in an inert liquid.

$$u_t = C_0 U_M + u_d \tag{3.54}$$

The dimensionless coefficient C_0 is determined experimentally and in horizontal flows it ranges from 0.95 (Singh and Griffith (1976)) to 2.0 (Odozi (2000)).

In the work by Dukler and Hubbard (1975), where it is assumed that there is no gas entrainment in the liquid film and no liquid droplets in the liquid bubble, the following expression for the translation slug velocity was obtained:

$$u_t = U_M + \frac{\dot{M}_{L-P}}{\rho_L \alpha_{L_s} A} \tag{3.55}$$

where α_{L_s} is the slug body liquid holdup, ρ_L is the liquid density and $\dot{M}_{L-P}$ is the rate of mass picked up at the slug front from the liquid film.

On the other hand Bendiksen (1984) suggested that the slug translational velocity should be a function of the mixture Froude number Fr and the critical Froude number Fr_{cr}, defined as follows:

$$\begin{aligned} Fr &= \frac{U_M}{\sqrt{\frac{\rho_L - \rho_G}{\rho_L} gD}} \\ Fr_{Cr} &= 3.5 \end{aligned} \tag{3.56}$$

where g is the gravitational acceleration, ρ_G is the gas density and D is the diameter. Thus the following values of C_0 and u_d in Eq. 3.54 were suggested:

$$\begin{aligned} C_0 = 1.0 \quad u_d = 0.542\sqrt{\left(\frac{\rho_L - \rho_G}{\rho_L}\right) gD} \quad \text{if} \quad Fr < Fr_{cr} \\ C_0 = 1.2 - 1.3 \quad u_d = 0 \quad \text{if} \quad Fr > Fr_{cr} \end{aligned} \tag{3.57}$$

3.11 Slug initiation models

Hydrodynamic slugs can be formed through different mechanisms. Moreover upstream and downstream boundary conditions can play an important role in the formation of flow instabilities (Bendiksen and Malnes, 1987). Therefore it is important to consider not only the processes of slug initiation, but also the conditions that allow slugs to reach stability.

A summary of the past literature on slug initiation will be presented in this section. The initiation of a wavy flow regime in gas-liquid pipeline flows is frequently modelled using long-wave Kelvin-Helmholtz analysis. The classical Kelvin-Helmholtz (KH) instability is observed in stratified flow of two incompressible inviscid fluids, of different velocities, in horizontal layers. The instability will appear in the form of waves generated on the liquid interface. Surface tension stabilises the short wavelength instability, up to a certain velocity threshold. Wallis and Dobson (1973) applied the Kelvin-Helmholtz mechanism and arrived at the conclusion that for long waves in a square channel, under low-pressure conditions, and with gas pressure much less than liquid pressure, the following theoretical stability criterion holds:

$$\rho_G u_G^2 < \Delta\rho g h_G \tag{3.58}$$

where h_G is the height of the gas layer and $\Delta\rho = \rho_L - \rho_G$. On the other hand, Wallis and Dobson (1973) performed also experiments in square channels with heights varying between 2.54 and 30.5cm, and they found the following stability criterion:

$$\rho_G u_G^2 < 0.25\Delta\rho g h_G \tag{3.59}$$

Thus the inviscid KH analysis used in Eq. 3.58 over-predicts the region where flow can remain stable in a low pressure system.

Taitel and Dukler (1976) used the KH instability and the Bernoulli equation and they derived the following stability criterion for circular channels:

$$\rho_G u_G^2 < C_2 H_G \Delta\rho g cos\beta \tag{3.60}$$

where $H_G = A_G/A_L^{'}$ and $A_L^{'} = dA_L/dh$ and β is the angle of the pipe relative to the horizontal. Moreover $C_2 = (1 - h/D)^2$ where h is the height of the liquid in the pipe. Two important assumptions in the derivation of this stability criterion are that the gas density is small compared with the liquid density and the liquid velocity is small compared with the gas velocity.

Mishima and Ishii (1980) analysed the relationship between wave amplitude and wavelength, and introduced the concept of the most dangerous waves, which occur at a wavelength where interfacial instabilities grow fastest. They also proposed that this will be the dominant wavelength of disturbances immediately before slug initiation.

Wu et al. (1987) developed a more general stability criterion for stratified gas-liquid flow compared with the one derived by Taitel and Dukler (1976). By applying linear stability analysis to the stratified flow equations in the inviscid limit, Wu et al. (1987) obtained:

$$(U_G - U_L)^2 < \left(\frac{H_G}{\rho_G} + \frac{H_L}{\rho_L}\right) \Delta\rho g cos\beta \tag{3.61}$$

where the notation is equivalent to the one in Eq. 3.60 with $H_L = A_L/A_L^{'}$. With increasing pressure the gas density increases as well, and therefore the term H_L/ρ_L becomes more influential compared with the H_G/ρ_G term. Thus a larger region of stratified flow can exist under high-pressure conditions compared to the result given by Taitel and Dukler (1976). On the other hand, Wu et al. (1987) also found that their expression overpredicted the area of the stratified flow region. Thus they combined the previous expression with the full viscous theory as given

by Wallis and Dobson (1973), and they obtained the following modified stability criterion:

$$(U_G - U_L)^2 < \left(\frac{H_G}{\rho_G} + \frac{H_L}{\rho_L}\right) \Delta\rho g cos\beta - (C_V - C_{IV})^2 \frac{(A_G\rho_L - A_L\rho_G)^2}{\rho_L\rho_G A_L A_G} \quad (3.62)$$

where C_V and C_{IV} are the critical viscous and inviscid wave velocities (Barnea and Taitel, 1993).

A comparison of the stability criterion in Eq. 3.62 with experimental data sets has been carried out by Crawley et al. (1992). They found that for data sets with pipe diameters in the range 2.5-30cm, pressures in the range 1-30bar and inclinations between -2 and +2 degrees, the viscous KH stability criterion appears to give a good prediction, and in particular, at low gas velocities, it shows a better agreement than the Taitel and Dukler (1976) criterion.

In the study by Mata et al. (2002) it is duly noted that the viscous KH analysis (as carried out by Taitel and Dukler (1976) and Barnea and Taitel (1993)) is based on the assumption that instability occurs at long wavelengths. This is due to the fact that the stratified flow equations are based on the plug-flow assumption, where fluid motion in assumed to happen only in the direction of the pipe axis. These equations predict that the most unstable wave has a finite wavelength. On the other hand, in rectangular geometry where equations can be solved exactly and it is not necessary to make the plug-flow assumption, it has been shown that most unstable waves have a finite wavelength (see studies by Wallis and Dobson (1973) and Mishima and Ishii (1980)). Moreover in experiments conducted by Fan et al. (1993) in circular channels it has been found that disturbances with well-defined wavelengths (resonance waves) are present immediately prior to the formation of slugs.

A variety of articles have been written regarding the issue of well-posedness for the two-fluid model. A recent one (Liao et al., 2008) summarises well the

currently accepted analysis: when the relative velocity between the liquid and gas exceed a critical value, the governing equations of the two-fluid model do not possess real characteristics. The critical value coincides with the inviscid Kelvin-Helmholtz (IKH) stability criterion. Since the viscous Kelvin-Helmholtz instability is triggered at a lower slip velocity than that of the IKH instability, the system can still be well-posed while a small disturbance grows in time for a certain range of slip velocities. Within this range, it is possible to simulate transition from stratified to slug flow using a viscous two-fluid model.

3.12 Slug Stability Models

Once a slug has been formed, it is important to be able to predict under what conditions it will be stable. Although the presence of interfacial instabilities is required in order for slugs to be formed, this does not guarantee that the slugs formed will remain stable.

Slug stability will occur if the volume of liquid entering the slug is equal to or larger than the liquid volume shed at the back of the slug. This can be expressed in the following form:

$$Q_{out} < (V_s - u_f)A_f \tag{3.63}$$

where V_s is the velocity of the slug front, u_f is the velocity of the liquid film, A_f is the area of the liquid film and Q_{out} is the volumetric flow rate out of the slug.

In particular, Bendiksen and Espedal (1992) found that in order for stable slug flow to be sustained, it is necessary that the bubble front velocity U_B is less than the slug front velocity V_S

$$U_B < V_S \tag{3.64}$$

It should be noted that Eq. 3.64 is equivalent to Eq. 3.63 when $Q_{out} = (U_B - u_f)A_f$. They proposed that for unstable surface waves and under high-pressure conditions, it is important that Eq. 3.64 is satisfied. A series of experiments carried out at Sintef are in agreement with their findings.

A further necessary criterion is the slug growth criterion. Although many different kinds of waves may form, some even leading to liquid bridging of the pipe, the importance lies in the way that resulting slugs decay over a wide range of flow rates. The criterion for sustaining slug growth can be expressed as:

$$U_B < U_{GD} \frac{\alpha_D - \alpha_S \frac{U_{GS}}{U_{GD}}}{\alpha_D - \alpha_S} \tag{3.65}$$

where U_{GD} is the gas velocity in the stratified flow prior to the transition to slug flow, U_{GS} is the average gas velocity in the slug region, α_S is the volumetric gas fraction in the slug, and α_D is the volumetric gas fraction in the downstream gas.

3.13 Slug Frequency

The definition of slug frequency (indicated by ϕ) is that of the mean number of slug units passing a stationary observer in unit time (Hubbard, 1965):

$$\phi = \frac{N}{t} \tag{3.66}$$

where N is the number of slugs passing in time t. The slug frequency increases with decreasing pipe diameter and increasing liquid flow rate (Taitel and Dukler, 1977). An interesting feature is that when the slug frequency is plotted versus the mixture velocity then the curve shows a minimum (Hill and Wood, 1990), and when analysing flows with low gas flow rates the presence of minima is associated with regime transitions from laminar to turbulent in the inlet region (Tronconi,

1990). Also if there is a positive inclination from the horizontal then the slug frequency increases with increasing pipe inclination (Hill and Wood, 1990).

A number of small sets of experiments have been performed from which correlations for slug frequencies have been developed. These correlations can be useful for validation of slug flow calculations, although only for the narrow range of physical properties in which the correlations are valid. The most frequently used correlations are:

- Gregory and Scott (1969) used a carbon dioxide-water system in a 19mm diameter horizontal pipe, and combined the results with those obtained by Hubbard (1965). The correlation obtained is:

$$\phi = 0.0226 \left[\frac{U_L}{gD} \left(\frac{19.75}{U_m} + U_m \right) \right]^{1.2} \tag{3.67}$$

  where U_m is the mixture superficial velocity, U_L is the liquid superficial velocity and D is the pipe diameter. This correlation is only valid for small (19mm) range diameter, horizontal pipes.

- Greskovich and Srier (1972) rearranged the previous correlation and obtained the following equation:

$$\phi = 0.0226 \left[\lambda \left(\frac{2.02}{D} + Fr_{mix} \right) \right]^{1.2} \tag{3.68}$$

  where Fr_{mix} is the mixture Froude number, defined as:

$$Fr_{mix} = \frac{U_m^2}{gD} \tag{3.69}$$

  and λ is the input liquid quality, defined as:

$$\lambda = \frac{U_L}{U_L + U_G} \tag{3.70}$$

  This correlation has the same limitations as Gregory and Scott (1969).

- Taitel and Dukler (1977) used a mechanistic model to predict the slug frequency for horizontal and near-horizontal pipes. It is assumed that at the inlet of the pipe there is gas-stratified flow, and solitary waves form and grow on the unstable surface, having the effect of bridging of the pipe and blocking of the gas passage. Consequently the liquid level downstream of the formation point decreases below the equilibrium level, and the film will rebuild to its equilibrium level before the whole cycle is repeated. Thus Taitel and Dukler (1977) defined the slug frequency to be equal to the inverse of the time interval required by the film to rebuild to its equilibrium level. The time interval was calculated using one-dimensional mass and momentum balances for each phase. Disagreements with the model (Davies, 1992) were based on the observation that slug initiation would occur before the level built up to the equilibrium level. The main limitation of this model stems from the assumption that at the inlet there is gas-stratified flow.

- Tronconi (1990) assumed (neglecting surface tension effects) that the slug frequency is inversely proportional to half the period of finite amplitude waves, formed at the gas-liquid interface. After estimating the wave properties using the theory of finite amplitude waves, the final equation showed the dimensionless slug frequency Φ to be related to the dimensionless actual gas velocity $\tilde{u}_G$ (defined as $\tilde{u}_G \equiv u_G/\sqrt{gD}$) and dimensionless equilibrium gas height $\tilde{h}_G$ (defined as $\tilde{h}_G \equiv h_G/D$):

$$\Phi = \frac{\phi D}{u_G}\frac{\rho_L}{\rho_G} = 0.61\frac{\tilde{u}_G}{\tilde{h}_G} \tag{3.71}$$

- Hill and Wood (1990) analysed experimental data collected from BP's research centre and found a correlation between the dimensionless frequency and the equilibrium stratified holdup ε_{Le}, calculated using methods introduced

in Taitel and Dukler (1976). The equation obtained is:

$$\Phi = \frac{\phi D}{(u_G - u_L)} = 2.74 \frac{\varepsilon_{Le}}{(1 - \varepsilon_{Le})} \tag{3.72}$$

This correlation is the one with the widest range of applicability due to the variety of experimental data it is based on, but it also inherits the limitations that exist in Taitel and Dukler (1976).

3.14 VOF Model in Ansys FLUENT

When using FLUENT, it is important to realise that the underlying model is different from the one used in EMAPS. In the simulations carried out with FLUENT for wavegrowth and Manolis cases, it was decided to use a channel to simulate the pipe, i.e. two-dimensional simulations.

The VOF model was chosen due to its speed and its proven record in tracking interfaces. The VOF model solves a single set of momentum equations and tracks the volume fraction of each of the fluids throughout the domain. The main limitations of the VOF model that could affect our simulations involve two features, the first one being that it is necessary to use the pressure-based solver, and the second one that only one of the phases can be modelled as a compressible ideal gas.

The tracking of the interface between the two phases (in our simulations) is accomplished by the solution of a continuity equation for the volume fraction of the two phases (FLUENT, 2006). For the i^{th} phase, the equation has the following form:

$$\frac{1}{\rho_i} \left(\frac{\partial}{\partial t}(\alpha_i \rho_i) + \nabla \cdot (\alpha_i \rho_i \vec{v_i}) = \sum_{j=1}^{n} (\dot{m}_{ji} - \dot{m}_{ij}) \right) \tag{3.73}$$

where $\dot{m}_{ij}$ is the mass transfer from phase i to phase j and $\dot{m}_{ji}$ is the mass transfer from phase j to phase i.

In the simulations for the work in this thesis, the volume fraction equation was solved through implicit discretisation. In the implicit scheme, the choice is among standard finite-difference interpolation schemes, QUICK, second order Upwind and first order Upwind, and the Modified HRIC schemes, in order to obtain the face fluxes for all cells, including those near the interface. For the current simulations preference was given to Modified HRIC schemes, as they are more appropriate for VOF simulations, as explained in the next section.

$$\frac{\alpha_i^{n+1}\rho_i^{n+1} - \alpha_i^n \rho_i^n}{\Delta t} V + \sum_f (\rho_i^{n+1} U_f^{n+1} \alpha_{i,f}^{n+1}) = \left[S_{\alpha_i} + \sum_{j=1}^{n} (\dot{m}_{ji} - \dot{m}_{ij}) \right] V \quad (3.74)$$

The volume fraction values are required at the current time step, and therefore a standard scalar transport equation (3.74) is solved iteratively for each of the secondary-phase volume fractions at each time step. The implicit scheme has been used for both transient and steady-state calculations.

In the geometric reconstruction approach, the standard interpolation schemes that are used in FLUENT are used to obtain the face fluxes whenever a cell is completely filled with one phase or another. When the cell is near the interface between two phases, the geometric reconstruction scheme is used.

The geometric reconstruction scheme represents the interface between fluids using a piecewise-linear approach. In FLUENT this scheme is the most accurate and is applicable for general unstructured meshes. It assumes that the interface between two fluids has a linear slope within each cell, and uses this linear shape for calculation of the advection of fluid through the cell faces.

The first step in this reconstruction scheme is calculating the position of the linear interface relative to the centre of each partially-filled cell, based on information

about the volume fraction and its derivatives in the cell. The second step is calculating the advecting amount of fluid through each face using the computed linear interface representation and information about the normal and tangential velocity distribution on the face. The third step is calculating the volume fraction in each cell using the balance of fluxes calculated during the previous step.

3.14.1 Modified HRIC Scheme

For simulations using the VOF multiphase model, upwind schemes (which normally look at values "upstream") are generally unsuitable for interface tracking because of their diffusive nature. Central differencing schemes, while generally able to retain the sharpness of the interface, are unbounded and often give unphysical results. In order to overcome these deficiencies, FLUENT uses a modified version of the High Resolution Interface Capturing (HRIC) scheme, and this scheme was used in most of the 2D simulations carried out. The modified HRIC scheme is a composite NVD (normalised variable diagram) scheme that consists of a non-linear blend of upwind and downwind differencing (Muzaferija et al., 1998).

The donor-acceptor approach is used near the interface (FLUENT, 2006). The scheme identifies one cell as a donor of an amount of fluid from one phase and another (neighbour) cell as the acceptor of that same amount of fluid, and is used to prevent numerical diffusion near the interface.

First, the normalised cell value of volume fraction $\widetilde{\phi}_c$, is computed and is used to find the normalised face value $\widetilde{\phi}_f$, as follows:

$$\widetilde{\phi}_c = \frac{\phi_D - \phi_U}{\phi_A - \phi_U} \tag{3.75}$$

where A is the acceptor cell, D is the donor cell, and U is the upwind cell, and

$$\widetilde{\phi}_f = \begin{cases} \widetilde{\phi}_c & \widetilde{\phi}_c < 0 \text{ or } \widetilde{\phi}_c > 1 \\ 2\widetilde{\phi}_c & 0 \leq \widetilde{\phi}_c \leq 0.5 \\ 1 & 0.5 \leq \widetilde{\phi}_c \leq 1 \end{cases} \tag{3.76}$$

Here, if the upwind cell is not available (e.g. unstructured mesh), an extrapolated value is used for ϕ_U. Directly using this value of $\widetilde{\phi}_f$ causes wrinkles in the interface, if the flow is parallel to the interface. Therefore FLUENT switches to ULTIMATE QUICKEST scheme (the one-dimensional bounded version of the QUICK scheme as per Leonard (1991)) based on the angle between the face normal and interface normal:

$$\phi_f^{\widetilde{UQ}} \equiv \begin{cases} \widetilde{\phi}_c & \widetilde{\phi}_c < 0 \text{ or } \widetilde{\phi}_c > 1 \\ MIN\left(\widetilde{\phi}_f, \frac{6\widetilde{\phi}_c+3}{8}\right) & 0.5 \leq \widetilde{\phi}_c \leq 1 \end{cases} \tag{3.77}$$

This leads to a corrected version of the face volume fraction $\widetilde{\phi}_f^*$:

$$\widetilde{\phi}_f^* = \widetilde{\phi}_f\sqrt{\cos\theta} + (1 - \sqrt{\cos\theta})\phi_f^{\widetilde{UQ}} \tag{3.78}$$

where

$$\cos\theta = \frac{\nabla\phi \cdot \vec{d}}{|\nabla\phi||\vec{d}|} \tag{3.79}$$

and $\vec{d}$ is a vector connecting cell centres adjacent to the face f.

The face volume fraction is now obtained from the normalised value computed above as follows:

$$\phi_f = \widetilde{\phi}_f^*(\phi_A - \phi_U) + \phi_U \tag{3.80}$$

The modified HRIC scheme provides improved accuracy for VOF calculations when compared to QUICK and second order schemes, and is less computationally expensive than the Geo-Reconstruct scheme. It has to be remembered though that first order upwind schemes are known to introduce numerical diffusion when large gradients exist (eg. at the interface), and numerical diffusion on the momentum equation introduces numerical viscosity. On the other hand, discrete equations are normally more diffusive than the original differential equations.

3.14.2 Density and other material properties

In a two-phase system (as per our simulations), the density of each cell is calculated as:

$$\rho = \alpha_2\rho_2 + (1 - \alpha_2)\rho_1 \tag{3.81}$$

where it is assumed that 1 and 2 are the subscripts representing the two phases, and that the volume fraction of the second phase is tracked. All other material properties are calculated in the same way.

3.14.3 Momentum Equation

A standard single momentum equation is solved for the whole domain, and the velocity field is the same for all phases in each cell (and varies from cell to cell). The momentum is shown below:

$$\frac{\partial}{\partial t}(\rho\vec{v}) + \nabla\cdot(\rho\vec{v}\vec{v}) = -\nabla p + \nabla\cdot\left[\mu\left(\nabla\vec{v} + \nabla\vec{v}_T\right)\right] + \rho\vec{g} + \vec{F} \tag{3.82}$$

where ρ is the density, v is the velocity, μ is the viscosity, p is the pressure, g is the gravitational acceleration and F is the source term. Also it should be noted that if there are large velocity differences between the phases, then the accuracy of the velocities computed near the interface may be reduced.

3.14.4 Energy Equation

The energy equation refers to the mixture and is shown below:

$$\frac{\partial}{\partial t}(\rho E) + \nabla\cdot(\vec{v}(\rho E + p)) = \nabla\cdot(k_{eff}\nabla T) + S_h \tag{3.83}$$

where density (ρ) and effective thermal conductivity (k_{eff}) are shared by the phases. The source term S_h contains contributions from radiation and any other volumetric heat sources. It should be kept in mind that in the VOF model, energy E and temperature T are mass-averaged variables:

$$E = \frac{\sum_{i=1}^{n} \alpha_i \rho_i E_i}{\sum_{i=1}^{n} \alpha_i \rho_i} \tag{3.84}$$

where E_i for each phase i is based on the specific heat of that phase and the shared temperature.

3.14.5 Surface Tension

Surface Tension effects in the interface between the two phases can be included in the VOF model, and simulations were run both with surface tension turned on and off.

The surface tension model used in FLUENT is the continuum force model proposed by Brackbill et al. (1992), and it results in a source term in the momentum equation. The source term is a volume force which can be expressed in the following form:

$$F_{vol} = \sum_{pairs\ ij, i<j} \sigma_{ij} \frac{\alpha_i \rho_i \kappa_j \nabla \alpha_j + \alpha_j \rho_j \kappa_i \nabla \alpha_i}{\frac{1}{2}(\rho_i + \rho_j)} \tag{3.85}$$

where σ is the surface tension coefficient, κ is the surface curvature.

If there are only two phases present, $\kappa_i = -\kappa_j$ and $\nabla\alpha_i = -\nabla\alpha_j$, and so we have

$$F_{vol} = \sigma_{ij}\frac{\rho\kappa_i\nabla\alpha_i}{\frac{1}{2}(\rho_i + \rho_j)} \tag{3.86}$$

where $\rho = \sum_q \alpha_q\rho_q$ is the volume-averaged density.

In order to determine if the effects of surface tension are important it is necessary to check the values for the combination of Reynolds number Re and either capillary number Ca, or Weber number We. The capillary number is defined as $Ca = \frac{\mu U}{\sigma}$, and the Weber number is defined as $We = \frac{\rho L U^2}{\sigma}$, where μ is the viscosity, U is the free-stream velocity (i.e. the velocity away from any object or boundaries in the part of the flow not disturbed by any object or boundaries), σ is the surface tension coefficient and L is the characteristic length. If $Re \ll 1$ and $Ca \gg 1$ or if $Re \gg 1$ and $We \gg 1$, then surface tension effects can be neglected. Including surface tension effects when not needed will increase the computation time but will not actually produce wrong results.

3.14.6 Turbulence

Turbulent flows are characterised by fluctuating velocity fields. These fluctuations mix transported quantities such as momentum, energy, and species concentration, and cause the transported quantities to fluctuate as well. Since these fluctuations are often of small scale and high frequency, it is computationally intractable to simulate them directly. The instantaneous, exact governing equations can be time- or space-averaged to filter out the small scales, and thereby resulting in a modified set of equations that are computationally less demanding to solve. However, the modified equations contain additional unknown variables, and models are needed to determine these variables in terms of known quantities.

The turbulence models that have been used in the current thesis are $k-\epsilon$ and Reynolds stress model (FLUENT, 2006).

3.14.6.1 The standard $k-\epsilon$ model

The standard $k-\epsilon$ model (Launder and Spalding, 1972) in FLUENT is a two-equation model in which the solution of two separate transport equations allows the turbulent velocity and length scales to be independently determined. It is a semi-empirical model, and the derivation of the model equations relies on phenomenological considerations and empiricism. It is a very popular model in industrial flow and heat transfer simulations, due to its robustness, economy, and reasonable accuracy for a wide range of turbulent flows.

The standard $k-\epsilon$ model is based on model transport equations for the turbulence kinetic energy k and its dissipation rate ϵ. The model transport equation for k is derived from the exact equation, while the model transport equation for ϵ was obtained using physical reasoning and bears little resemblance to its mathematically exact counterpart (FLUENT, 2006). In the derivation of the $k-\epsilon$ model, it was assumed that the flow is fully turbulent, and the effects of molecular viscosity are negligible. The standard $k-\epsilon$ model is therefore valid only for fully turbulent flows.

3.14.6.2 The Reynolds stress model

The Reynolds stress model is the most elaborate turbulence model that FLUENT provides. It closes the Reynolds-averaged Navier-Stokes equations by solving transport equations for the Reynolds stresses, together with an equation for the dissipation rate. Therefore five additional transport equations are required in 2D flows and seven additional transport equations must be solved in 3D.

Since the Reynolds stress model accounts for the effects of streamline curvature, swirl, rotation, and rapid changes in strain rate in a more rigorous manner than one-equation and two-equation models, it has greater potential to give accurate predictions for complex flows. However, the fidelity of Reynolds stress model predictions is still limited by the closure assumptions employed to model various terms in the exact transport equations for the Reynolds stresses. The modelling of the pressure-strain and dissipation-rate terms is particularly challenging, and often considered to be responsible for compromising the accuracy of Reynolds stress model predictions.

The Reynolds stress model involves calculation of the individual Reynolds stresses $\overline{u_i' u_j'}$, using differential transport equations (Launder, 1989). The individual Reynolds stresses are then used to obtain closure of the Reynolds-averaged momentum equation.

The exact form of the Reynolds stress transport equations may be derived by taking moments of the exact momentum equation. This is a process wherein the exact momentum equations are multiplied by a fluctuating property, the product then being Reynolds-averaged. Unfortunately, several of the terms in the exact equation are unknown and modelling assumptions are required in order to close the equations (FLUENT, 2006).

3.15 Parallelisation concepts

It was decided to carry out a full parallelisation of the 1D code EMAPS, which will explained in detail in chapter 4. Parts of this work were done in collaboration with D. Marski and A. Taillandier (Taillandier, 2011). Before starting any parallelisation task, it is important to consider the theoretical limits of such an operation. Fig.

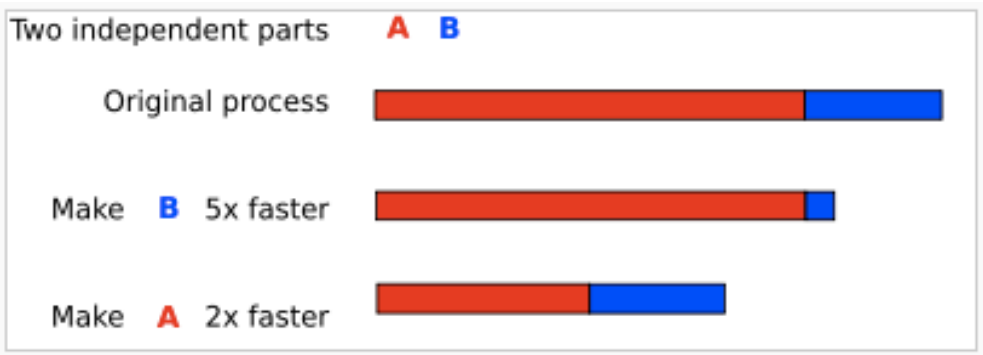


Figure 3.4: Prioritisation of optimisation

3.4 shows the expected speed-up as established by Amdahl, who expressed it as a simple mathematical equation, called Amdahl's law (Chandra et al., 2000):

$$S = \frac{1}{(1-F) + \frac{F}{S_p}}$$

where S is the speed-up, F the fraction of the code that is parallelised and S_p the speed-up achieved in the parallel section. When considering the best speed-up possible in the parallel section, then (in simple cases) $S_p = p$, with p being the number of processors used. So by altering the formula, we obtain:

$$S < \frac{p}{p(1-F) + F}$$

Therefore, when F equals one, the best speed-up possible will be p. However, if only 80% of the code is parallelised, then the best speed-up expected on four cores is:

$$S < \frac{4}{4(1-0.8) + 0.8} = \frac{4}{1.6} = 2.5$$

Thus the binding conclusion, as stated in Chandra et al. (2000), is that eventually the performance of the application will be limited by F, the proportion of the

code that was parallelised, regardless of the efficiency of the parallel code and the number of processors used.

Before going into the details of the two tools, OpenMP and MPI, the basic concepts will be explained.

3.15.1 Execution model

The way those two options are executed is likely to be one of the main differences, along with the memory architecture. In fact OpenMP only uses one process and creates new threads whenever needed, whereas MPI requires multiple processes from the beginning.

In the Encyclopedia of Computer Science (Ralston et al., 2000) a process is defined as a program in execution on a machine, consisting of one or more threads, an address space, and communication ports. On the other hand, a thread is a primitive process in one of four states: running, ready, waiting, or suspended. A thread is first created in its suspended state and will not become ready until a signal is received. Thus the threads within the same process form a team that must cooperate towards a common computational goal; they share the same address space and cannot be protected from one another. Furthermore, when it is created, a process has one thread, and it can create and control additional threads.

An important difference between processes and threads is that processes cannot share data with each other through memory access. Therefore in order to communicate, they need to do so explicitly, which means that both processes need to be aware of the communication taking place and allocate execution time to deal with it. For this communication to occur, a special layer called inter-process communication (IPC) is used in order to exchange messages among threads in different address

spaces on the same or different machines. A multi-threaded program however can use the same memory space, which means that the threads do not need to be both active when the communication takes place.

Both ways of communication have advantages and disadvantages. With the explicit communication, a lot of time can be spent waiting, and the risk of dead-locks happening is higher. However with the shared memory the data can be corrupted if accessed simultaneously.

3.15.2 Memory architecture

In one case the memory is private to each process, but when staying within the same process, some memory can be shared between the different threads. The two architectures are illustrated in Fig. 3.5 and 3.6. For both architectures the more independent the data are, the easier the introduction of concurrency is.

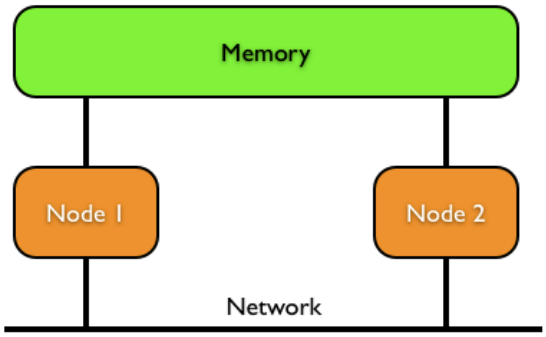


Figure 3.5: Shared memory architecture.

3.15.2.1 Shared memory

Shared memory architecture can be extremely fast, as memory sits on the same chip, or machine, and the connectivity is generally fast. Another advantage is

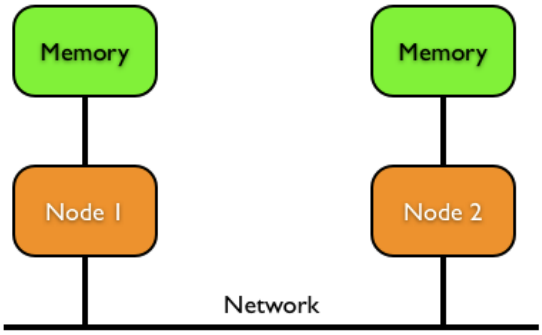


Figure 3.6: Distributed memory architecture.

that the need for explicit synchronisation is minimal. However the access to shared memory has to be regulated, in order to avoid data corruption.

Another limitation is scalability. Shared memory machines are difficult to build when high performance is targeted, or rather parallelism with a high number of executing nodes. The reason this is difficult is the shared memory part. All processing nodes need to have direct access to the same bit of memory. Often this is done through some kind of RAM. The processing nodes in the shared memory machine need to be connected to each other and able to share the same address space. Usually this is done by sharing the same motherboard, but scaling a motherboard to share more than two or four processors is difficult. Hence shared memory machines with a higher number of processors get very expensive.

3.15.2.2 Distributed memory

As opposed to shared memory machines, distributed ones can be built easily. They only need a set of standard PCs connected to each other through a network. The rest is done on the software side, and on this part there are some open source and free software products that can help setting up a distributed memory machine.

Of course, in order to get a high performance distributed memory machine, further requirements are necessary. For instance the computers themselves should be of good quality, and in most cases, the network should be very fast, as usually the communication is the weakest link. As far as scalability is concerned, where in the shared memory paradigm the entire machine needs to be rebuilt with more slots for processors on a large motherboard, in the distributed memory paradigm, eg. a cluster composed by standard PCs sharing the same network, the only thing that will be required is to add another PC with the necessary software and connect it to the network. Thus a distributed system is far more scalable compared to a shared memory system.

3.15.3 MPI

MPI stands for Message Passing Interface. It is a language independent communications protocol used to create parallel applications. As mentioned before, it uses a distributed memory architecture, as the communication is between different processes, but it can also run on shared memory systems. In most languages MPI consists of an application programming interface (API), which is generated through a set of functions and routines that can be called directly from the chosen programming language.

Nodes are addressed via a number (positive integer), and the communication is done through simple methods where the user specifies the data that have to be sent or received, and locations for reading, writing and so on.

Some of the advantages of MPI are (Gropp et al., 1999):

- Universality: MPI can run on nearly any type of hardware connected by a network (fast or slow).

- Expressivity: it is a useful and complete model to express parallel algorithms and provides the controls to deal with data locality (i.e. making sure that most the data needed by one processor is available in the memory physically attached to it), missing from other models such as OpenMP.

- Ease of debugging: due to the memory model used (distributed), the errors due to unexpected overwriting of memory are easier to spot due to more explicit references.

- Performance: due to a more direct handling of the memory, and the fact that it is split over several processes, the performance in terms of memory and cache is often better compared to shared memory parallelism.

3.15.3.1 Single Program, Multiple Data

The main difference, as stated before, is in the execution model and memory architecture. MPI would require major changes in the whole program, and the same version could not be used to run on a machine not supporting it. This is different to the OpenMP version, as OpenMP is used within the comment section of the code and so, when using a compiler without OpenMP support, the code can still be compiled. This helps in the maintenance of the software and future developments as everything is in the same place. The changes that have to be made for MPI affect the whole program. Unlike OpenMP where the parallelism can be invoked locally, MPI requires it to work from the start until the end. Once it is set up and working however, the impact of MPI on the code is not important and can be compared to the OpenMP code. These differences stem from the fact that MPI uses multiple processes to introduce parallelism.

3.15.4 OpenMP

OpenMP stands for Open Multi-Processing and is an application programming interface, consisting of compiler directives, library routines and environment variables. Due to compiler directives the implementation of OpenMP can be done in the main branch of the software, because when the compiler does not support OpenMP, the software will still compile, as all the OpenMP code can be hidden in comments.

Data can easily be shared, which makes the process of trial and error easy. Parallelism can be implemented starting at the end of the execution chain, whereas MPI would have to be implemented at the very beginning.

A particular aspect of OpenMP, compared to other methods like MPI, is that it works with shared memory. This creates different possibilities regarding dealing with data.

3.15.4.1 Shared & private memory

OpenMP is working with threads and for simplicity one thread per node is being counted. Each thread has different types of memory which it can access, and the two parts of interest are shared memory and private memory. Shared memory is available to all threads at all times, whereas private memory is accessible only by the one thread to which it is private. This is illustrated in Fig. 3.7

Unless some special action is taken, all the information stored in the private memory is lost when the thread is removed.

Shared variables have to be declared when the parallel region is invoked, and therefore they have to be created before the parallel region is started. Furthermore,

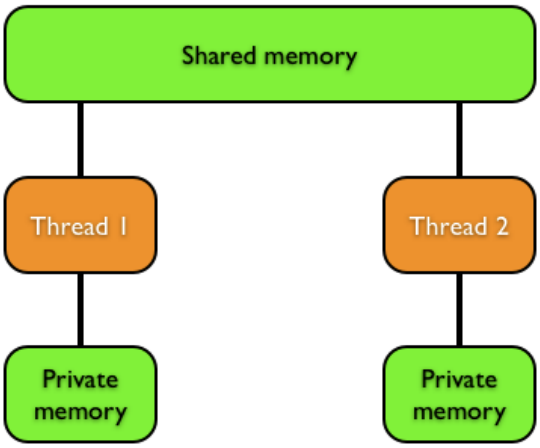


Figure 3.7: Shared and private memory for threads in a multi-threaded process.

it is not possible to create shared variables while in the parallel region. However it is possible to create private variables in each thread.

3.15.4.2 Communication between threads

Communication as such, like thread 0 sending a string to thread 1, is not available in OpenMP. It could be done through some shared variables, but it would be very cumbersome, as one thread would have to write to the shared variable, instruct the second thread that it can read the variable, and then the second thread would read the variable, and then signal that the shared one is free to be used again by other threads.

3.16 Conclusions

As explained in the introduction, it is important that, before looking at slug flow in more detail, a basic knowledge of the fields of fluid mechanics and fluid dynamics is obtained. Here the theory behind fluid motion has been briefly

explained, before moving on to equations for two-phase flow, including the Watson model and the single-pressure model SPM4s (which will be the most used model in the 1D software EMAPS later on) . The numerical solver used in the SPM4s model has also been explained in detail.

There are various sections on slugs, including slug characteristics, slug initiation and stability models and discussions regarding one of the most important features of slug flow: slug frequency. Some of the slug features mentioned here may also be of use to future users of EMAPS.

As the commercial code FLUENT with the volume of fluid (VOF) model will be used further on, a detailed section on the VOF model was also included.

Basic parallelisation concepts have been explained in order to give a relevant background to the OpenMP parallelisation carried out on the 1D code EMAPS.

References

Barnea, D. and Taitel, Y. (1993), 'Kelvin-Helmholtz stability criteria for stratified flow: viscous versus non-viscous (inviscid) approaches', *International Journal of Multiphase Flow* **19**(4), 639–649. (cited at page 30, 45)

Bendiksen, K. H. (1984), 'An experimental investigation of the motion of long bubbles in inclined tubes', *International Journal of Multiphase Flow* **10**(4), 467–483. (cited at page 42)

Bendiksen, K. H. and Espedal, M. (1992), 'Onset of slugging in horizontal gas-liquid pipe flow', *International Journal of Multiphase Flow* **18**(2), 237–247. (cited at page 46)

Bendiksen, K. H. and Malnes, D. (1987), 'Experimental data on inlet and outlet effects on the transition from stratified to slug flow in horizontal tubes', *International Journal of Multiphase Flow* **13**(1), 131–135. (cited at page 43)

Brackbill, J. U., Kothe, D. B. and Zemach, C. (1992), 'A continuum method for modeling surface tension.', *J. Comput. Phys* **100**, 335–354. (cited at page 55)

Chan, A. M. C. and Banerjee, S. (1981), 'Refilling and rewetting of a hot horizontal tube. Part II: Structure of a two-fluid model', *Journal of Heat Transfer* **103**, 287–292. (cited at page 25)

REFERENCES

Chandra, R., Menon, R., Dagum, L., Kohr, D., Maydan, D. and McDonald, J. (2000), *Parallel Programming in OpenMP*, Morgan Kaufmann. (cited at page 59)

Chung, L. Y., Drew, D. A. and Lahey, J. R. T. (1985), 'An analysis of wave propagation in bubbly two component two-phase flow', *Journal of Heat Transfer* **107**, 402–408. (cited at page 31)

Crawley, C. J., Wallis, G. B. and Barry, J. J. (1992), 'Validation of a one-dimensional wave model for the stratified-to-slug flow regime transition, with consequence for wave growth and slug frequency', *International Journal of Multiphase Flow* **18**(2), 249–271. (cited at page 45)

Davies, S. R. (1992), Studies of two-phase intermittent flow in pipelines, PhD thesis, Imperial College, University of London, U.K. (cited at page 49)

Drew, D. A. and Passman, S. L. (1999), *Theory of multicomponent fluids*, Vol. Applied Mathematical sciences of *135*, Springer. (cited at page 30)

Dukler, A. E. and Hubbard, M. G. (1975), 'Model for gas-liquid slug flow in horizontal and near horizontal tubes', *Industrial and Engineering Chemistry Fundamentals* **14**(4), 337–347. (cited at page 42)

Evje, S. and Fjelde, K. K. (2003), 'On a rough AUSM scheme for a one-dimensional two-phase model', *Computers & Fluids* **32**(10), 1497 – 1530. (cited at page 35)

Evje, S. and Flatten, T. (2003), 'Hybrid flux-splitting schemes for a common two-fluid model', *Journal of Computational Physics* **192**(1), 175 – 210. (cited at page 35)

Fan, Z., Lusseyran, F. and Hanratty, T. J. (1993), 'Initiation of slugs in horizontal gas-liquid flows', *AIChE Journal* **39**(11), 1741–1753. (cited at page 45)

FLUENT (2006), *FLUENT 6.3 User Guide*, FLUENT Inc. (cited at page 50, 52, 57, 58)

Garcia-Cascales, J. R. and Paillere, H. (2006), 'Application of AUSM schemes to multi-dimensional compressible two-phase flow problems', *Nuclear Engineering and Design* **236**(12), 1225 – 1239. (cited at page 35)

Gregory, G. A. and Scott, D. S. (1969), 'Correlation of liquid slug velocity and frequency in horizontal co-current gas-liquid slug flow', *AIChE Journal* **15**(6), 933–935. (cited at page 48)

Greskovich, E. J. and Srier, A. L. (1972), 'Slug frequency in horizontal gas-liquid flow', *Ind. Eng. Process Des. Develop.* **11**(2), 317–318. (cited at page 48)

Gropp, W., Lusk, E. and Skjellum, A. (1999), *Using MPI, 2nd Edition. Portable Parallel Programming with the Message Passing Interface*, The MIT Press. (cited at page 63)

Hill, T. J. and Wood, D. G. (1990), A new approach to the prediction slug frequency, *in* '65th Annual Technical Conference of the Society of Petroleum Engin.', New Orleans, LA. (cited at page 47, 48, 49)

Hubbard, M. (1965), An analysis of horizontal gas-liquid slug flow, PhD thesis, University of Houston. (cited at page 47, 48)

Hughes, W. F. and Brighton, J. A. (1991), *Fluid Dynamics*, McGraw-Hill Book Company. (cited at page xiv, 24)

Ishii, M. (1975), *Thermo-Fluid Dynamic Theory of Two-Phase Flow*, Eyrolles, Paris. (cited at page 28)

Ishiima, M. and Mishima, K. (1984), 'Two-fluid model and hydrodynamic constitutive relations', *Nuclear Engineering and Design* **82**, 107–126. (cited at page 31)

Launder, B. E. (1989), 'Second-moment closure: Present... and future?', *Inter. J. Heat Fluid Flow* **10(4)**, 282–300. (cited at page 58)

Launder, B. E. and Spalding, D. B. (1972), *Lectures in Mathematical Models of Turbulence*, Academic Press, London, England. (cited at page 57)

Leonard, B. P. (1991), 'The ultimate conservative difference scheme applied to unsteady one-dimensional advection', *Comp. Methods Appl. Mech. Eng.* **88**, 17–74. (cited at page 53)

Levy, S. (1999), *Two-Phase flow in complex systems*, John Wiley and Sons, Inc. (cited at page 31)

Liao, J., Mei, R. and Klausner, J. F. (2008), 'A study on the numerical stability of the two-fluid model near ill-posedness', *International Journal of Multiphase Flow* **34**, 1067–1087. (cited at page 45)

Liou, M. (1996), 'A sequel to AUSM: AUSM+', *Journal of Computational Physics* **129**(2), 364 – 382. (cited at page 35)

Liou, M. (2006), 'A sequel to AUSM, Part II: AUSM+-up for all speeds', *Journal of Computational Physics* **214**(1), 137 – 170. (cited at page 35)

Liou, M. and Steffen, C. J. (1993), 'A new flux splitting scheme', *Journal of Computational Physics* **107**(1), 23 – 39. (cited at page 35)

Liou, M. and Wada, Y. (1997), 'An Accurate and Robust Flux Splitting Scheme for Shock and Contact Discontinuities', *SIAM J. Scientific Computing* **18**, 633–657. (cited at page 38)

Mary, I., Sagaut, P. and Deville, M. (2000), 'An algorithm for unsteady viscous flows at all speeds', *International Journal for Numerical Methods in Fluids* **34**(5), 371–401. (cited at page 35)

Mata, C., Pereyra, E., Trallero, J. L. and Joseph, D. D. (2002), 'Stability of stratified gas-liquid flows', *International Journal of Multiphase Flow* **27**, 599–616. (cited at page 45)

Mishima, K. and Ishii, M. (1980), 'Theoretical prediction of onset of horizontal slug flow', *ASME J. Fluids Engng.* **102**, 441–445. (cited at page 44, 45)

Muzaferija, S., Peric, M., Sames, P. and Schellin, T. (1998), A two-fluid Navier-Stokes solver to simulate water entry, *in* '22nd Symposium on Naval Hydrodynamics', Washington, DC, pp. 277–289. (cited at page 52)

Nicklin, D. J., Wilkes, J. O. and Davidson, J. F. (1962), 'Two-phase flow in vertical tubes', *Trans. Int. Chem. Engs.* **40**, 61–68. (cited at page 42)

Odozi, U. A. (2000), Three-phase gas/liquid/liquid slug flow, PhD thesis, Imperial College, University of London, U.K. (cited at page 42)

Park, J. W., Drew, D. A. and Lahey, J. R. T. (1998), 'The analysis of void wave propagation in adiabatic monodispersed bubbly two-phase flows using an ensemble-averaged two-fluid mode', *International Journal of Multiphase Flow* **24**, 1205–1244. (cited at page 29)

Ralston, A., Reilly, E. D. and Hemmendinge, D. (2000), *Encyclopedia of Computer Science*, Wiley-Blackwell; 4th Edition. (cited at page 60)

Ransom, V. H. and Hicks, D. L. (1984), 'Hyperbolic two-pressure models for two-phase flow', *Journal of Computational Physics* **53**, 124–151. (cited at page 30)

Singh, G. and Griffith, P. (1976), Down sloping inclined pipe pressure drop and holdup, *in* 'Petroleum Mechanical Engineering and pressure Vessels and Piping Conference, Mexico City', ASME New York City. (cited at page 42)

Symon, K. (1971), *Mechanics*, Addison-Wesley. (cited at page 21)

Taillandier, A. (2011), M.P.I. Implementation of an existing Computational Fluid Dynamics research code, Master's thesis, Cranfield University. (cited at page 58)

Taitel, Y. and Dukler, A. E. (1976), 'A model for predicting flow regime transitions in horizontal and near horizontal gas-liquid flow', *AIChE.J.* **22**, 47–55. (cited at page 43, 44, 45, 50)

Taitel, Y. and Dukler, A. E. (1977), 'A model for predicting slug frequency during gas-liquid flow in horizontal and near horizontal pipes', *International Journal of Multiphase Flow* **3**, 585–596. (cited at page 47, 49)

Tiselj, I. and Petelin, S. (1997), 'Modelling of two-phase flow with second-order accurate scheme', *Journal of Computational Physics* **136**(2), 503 – 521. (cited at page 35)

Toro, E. F. (1997), *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer. (cited at page 32)

Trepanier, J. Y., Reggio, M., Zhang, H. and Camarero, R. (1991), 'A finite-volume method for the Euler equations on arbitrary Lagrangian-Eulerian grids', *Computers & Fluids* **20**(4), 399 – 409. (cited at page 37)

Tronconi, E. (1990), 'Prediction of slug frequency in horizontal two-phase slug flow', *AIChE Journal* **36**, 701–709. (cited at page 47, 49)

Wada, Y. and Liou, M.-S. (1994), A flux splitting scheme with high-resolution and robustness for discontinuities, *in* D. G. Fletcher, ed., 'Presented at the 32nd Aerospace Sciences Meeting and Exhibit, Reno, NV, 10-13 Jan. 1994; sponsored by AIAA', pp. 10–13. (cited at page 35)

Wallis, G. and Dobson, J. (1973), 'The onset of slugging in horizontal stratified air-water flow', *International Journal of Multiphase Flow* **1**, 173–193. (cited at page 43, 45)

Watson, M. (1990), *Non Linear Waves in Pipeline Two-Phase Flows*, Proceedings of the 3rd International Conference on Hyperbolic Problems, Uppsala, Sweden. (cited at page 32)

Wu, H. L., Pots, B. F. M., Hollenberg, J. F. and Meerhoff, R. (1987), Flow pattern transitions in two-phase gas/condensate flow at high pressure in an 8-inch horizontal pipe, *in* 'Proc. 3rd Inter. Conf. Multiphase Flow', The Hague, The Netherland. (cited at page 44)

Young, H. D. (1992), *Physics*, Addison-Wesley. (cited at page 21)

Chapter 4

EMAPS

4.1 Introduction

The acronym for EMAPS is Eulerian Multiphase Adaptive Pipeline Solver. It is a 1D code written in Fortran. Its design is mainly modular, using an iterative scheme structure enhanced by the possibility of applying different configuration methods, combined with various physical models.

EMAPS is composed of Sources and Models. Ideally the user, when a new model is developed, would write a new model in the Models section and thus update EMAPS without having to change the Sources (which include the solvers). In reality, most new models required some changes to the sources as well, and this situation has created a variety of "branches" of EMAPS running in parallel and diverging.

A simplified view of the architecture of EMAPS is shown in Fig. 4.1.

The three main parts are:

1. **Pre-EMAPS** The pre-processor for input text files

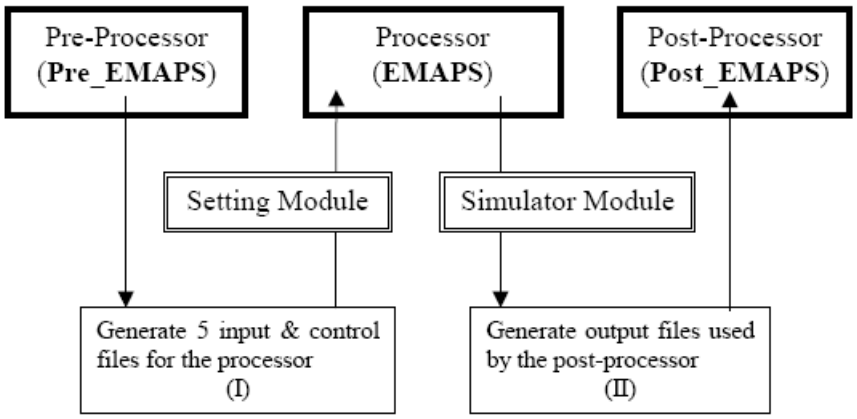


Figure 4.1: EMAPS architecture

2. **EMAPS** The actual solver or processor

3. **Post-EMAPS** The post-processor for analysing results of the simulation.

The input files used by the pre-processor are simple text files and need to be completed by the user according to a precise format. The files are:

1. **Pipe.txt** Pipe topography: length, diameter, inclination and mesh size

2. **Problem.txt** Test case name, initial and boundary condition data

3. **Control.txt** Time step information, numerical schemes

4. **Model.txt** Mathematical model, phase friction, interfacial pressure.

5. **Fluids.txt** Physical properties of the fluids.

The processor reads the input files generated by the pre-processor, and initialises the global variables that will be used for the simulations. The solver chosen by the user in the input files will be taken from the module *simulator*. An overview of the main EMAPS modules and their connections is shown in Fig. 4.2. The

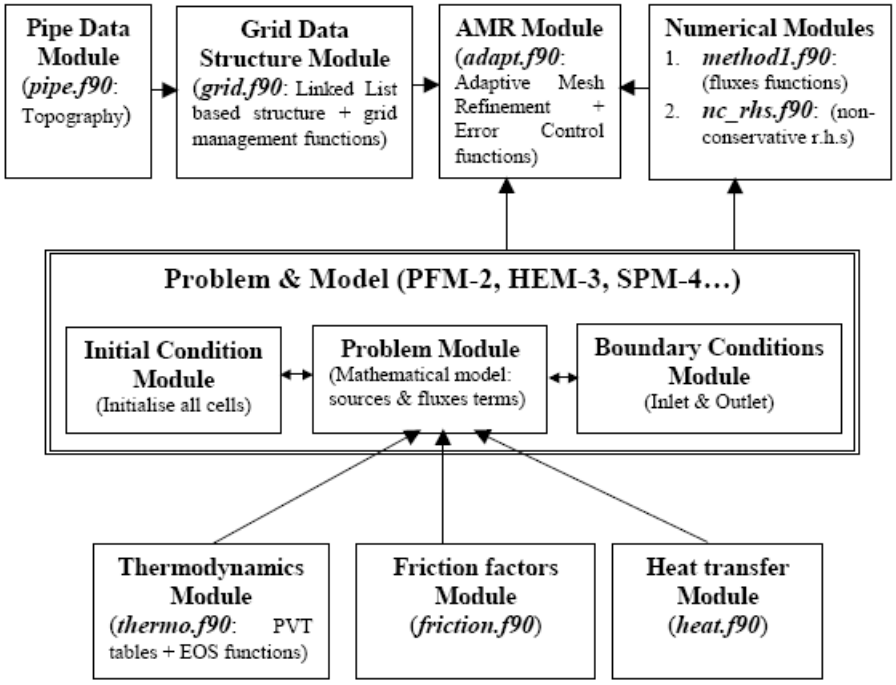


Figure 4.2: EMAPS main modules

post-processor is user-based, as a command-line program is provided which splits the results file into smaller files for plotting purposes. As part of the objective of increasing the accuracy of slug initiation, the choice of input files is essential for the correct running of the simulation, even more so as often the problems investigated are not well-defined. The difficulty of choosing input data refers not only to initial data that may not have been directly measurable experimentally in order to allow direct comparisons, but also regarding the choice of solver, time-step, mathematical model and other properties.

4.2 New Version

One of the issues encountered was that the development of EMAPS had been running along diverging branches. Some mathematical models were based on version 3.50, where others were based on version 3.60, which contained also pipe geometry parameters and enabled the simulation of pipes with inclined sections. Moreover adaptivity was also being developed as a separate sub-branch of 3.60, and some models had required modifications of the sources rather than just the creation of new models themselves.

A series of tests existed for version 3.50, and a very limited number of tests for version 3.60. The tests run as scripts that would be executed for given input files, and the output generated by the executable being tested would be either compared numerically with the expected solutions or the user would be asked to compare visually. In the numerical comparison a fail would result if a difference of more than 0.5% were calculated, while tests based on optical comparison are obviously based on user decision.

A new version of EMAPS had to be created, which "ported" models written in the old format from EMAPS 3.50 to EMAPS 3.60, and brought some other necessary changes to the source code in line with code optimisation, together with the addition of adaptive mesh refinement. This section of work was necessary but quite time-consuming, as development on EMAPS has previously been carried out on divergent code branches.

4.3 Adaptive Mesh Refinement (AMR)

One of the branches of EMAPS that was ported into the new version was one with Adaptive Mesh Refinement (AMR) enabled.

When complex flows are involved, one would be tempted to use a very fine grid over the whole problem in order to overcome numerical problems. But in the case of long pipes, such a method would prove very time-consuming and most likely also counterproductive, as such detail may turn out to be unnecessary. When AMR is enabled, the local resolution of the computational grid is matched to the requirements of the local flow solution, by locally and automatically modifying the computational grid, both in time and in space. Thus, very fine mesh cells are precisely concentrated and restricted only to regions where needed, and elsewhere the computational grid may be coarse. The criterion for refinement used is based on the value of the velocity gradient combined with the Kelvin-Helmholtz criterion (Jia, 2012). This method can dramatically reduce the computational effort required to perform simulations of multiphase flow problems.

4.4 Restarting EMAPS simulations

EMAPS had the issue that restarting could not be properly completed, i.e. if a simulation is stopped then normally it is not possible to restart it from the last simulation point. The restarting option has now been fully implemented. A script has been written that reads the latest result output file and substitutes the values into the input files, thus allowing seamless restarting with no loss of data.

4.5 Parallelisation

4.5.1 Profiling

In order to determine which areas of the software it is most crucial to speed-up, profiling of EMAPS was carried out. Profiling is a dynamic program analysis and

it was used in order to highlight areas of high memory usage and frequency and duration of function calls, in order to determine areas in which parallelisation would offer the greatest benefit. There are many profiling applications available, and for the current project the profiling application used was *gprof* (Darmawan et al., 2003). Because of the presence of many functions and modules in EMAPS, the output of *gprof* is quite complex. Fig. 4.3 and 4.4 show the average values of execution time in percentage and number of calls inside of each module, while Fig. 4.5 and 4.6 show in more detail the percentage of execution time taken in a series of simulations and the number of calls to each function.

% of total execution time	**Average**
▼ % time	
► Adapt	16.12
► Flux ausm	2.51
► Flux basic	28.51
► Friction drag	1.62
► Friction wall	2.39
► Grid	3.41
► Hydraulic	0.86
► NCT scheme	6.31
► Output	0.01
► Problem	14.41
► PV relax	1.09
► Solvers	6.69
► System	12.89
► Thermodyn	1.03

Figure 4.3: Average execution time used per module in percentages.

Although the figures only give a partial glimpse of the behaviour of the simulations, three points can be observed:

- some functions are called in almost all simulations
- some functions are only present in one or two simulations, but make up a large part of the total execution time
- execution time of a function is not always directly proportional to the number of times it is called in a simulation.

Total number of function calls	Average
▶ Adapt	1,596,524
▶ Basic	145,528,184
▶ Boundary1	250,115
▶ Flux ausm	2,006,423,259
▶ Flux basic	406,114
▶ Friction drag	1,933,426,606
▶ Friction Wall	8,213,720,562
▶ Grid	1,043,203
▶ Hydraulic	1,426,999,437
▶ NCT Scheme	453,172
▶ Output	28
▶ Problem	11,234,820,395
▶ PV Relax	1,500,743,727
▶ Solvers	453,172
▶ Thermodyn	2,698,679,321

Figure 4.4: Average function calls per module.

% of total execution time	burger_aea_square_sim1	drift_riemann_sim1	euler_rp_123_sim1	shallow_dam_break_sim1	spm4s_shaha_sim1	watson_fauc et_sim1	Average
▼ % time							
▼ Adapt	25.44	17.58	23.37	23.45	6.23	9.57	16.12
adapt_mp_patch_corrector_	9.15	8.79	11.51	11.73	0.00	3.59	6.60
adapt_mp_patch_predictor_	13.06	8.51	10.53	10.61	4.83	5.03	8.14
▼ Flux ausm	0.00	0.00	0.00	0.00	27.61	0.00	2.51
flux_ausm_mp_ausmdv_star_	0.00	0.00	0.00	0.00	19.10	0.00	1.74
▼ Flux basic	28.43	46.40	42.25	42.78	0.00	10.93	28.51
flux_basic_mp_fct_	12.90	22.63	26.11	26.60	0.00	5.01	14.76
flux_basic_mp_richtmyer_	15.53	23.77	16.14	16.18	0.00	5.92	13.75
▶ Friction drag	1.40	0.00	0.00	0.00	3.58	2.55	1.62
▶ Friction wall	0.00	1.32	0.00	0.00	4.77	7.46	2.39
▼ Grid	6.56	2.57	4.02	3.64	2.74	2.84	3.41
grid_mp_update_buffer_	6.56	2.57	4.02	3.64	2.74	2.84	3.39
▶ Hydraulic	0.00	0.00	0.00	0.00	1.74	7.74	0.86
▼ NCT scheme	5.90	7.04	7.71	9.74	9.80	2.24	6.31
nct_scheme_mp_nc_rhs_scheme_	5.90	7.04	7.71	9.74	0.00	2.24	5.42
▶ Output	0.09	0.03	0.00	0.00	0.00	0.00	0.01
▼ Problem	12.45	6.66	7.25	5.73	14.75	21.84	14.41
problem_mp_eigenvalues_	0.00	0.00	0.03	0.00	0.23	2.96	0.30
problem_mp_physical_flux_	4.02	0.43	1.13	1.65	0.00	1.38	1.55
problem_mp_physical_rhs_	5.27	0.66	1.46	1.25	2.51	0.65	2.60
problem_mp_primitive_data_	0.00	1.91	2.91	0.00	2.41	8.92	2.24
▼ PV relax	2.67	0.76	1.00	1.07	0.50	0.72	1.09
pv_relax_mp_relax_pressure_	2.67	0.76	1.00	1.07	0.50	0.72	1.09
▼ Solvers	6.22	12.58	8.34	9.30	2.38	2.25	6.69
solvers_mp_compute_residual_	6.22	12.58	8.34	9.30	2.38	2.25	6.69
▼ System	12.02	3.79	2.90	4.17	19.37	17.64	12.89
_intel_fast_memcmp	0.59	0.23	0.20	0.25	4.01	0.97	1.64
for_cpstr	10.68	3.40	2.31	3.48	8.56	15.77	7.43
▶ Thermodyn	0.00	0.78	3.04	0.00	4.25	0.00	1.03

Figure 4.5: Execution times used per function in percentages.

These points are important when considering which sections to parallelise.

Total number of function calls	burger_aea_square_sim1	drift_riemann_sim1	euler_rp_123_sim1	shallow_dam_break_sim1	spm4s_shaha_sim1	watson_faucet_sim1	Average
▼ Adapt	337,783	200,000	882,906	274,680	6,606,415	2,012,253	1,596,524
adapt_mp_patch_corrector_	84,445	40,000	147,151	91,560	0	670,751	206,952
adapt_mp_patch_predictor_	84,445	40,000	147,151	91,560	2,751,266	670,751	457,067
▶ Basic	0	1,600,810,025	0	0	0	0	145,528,184
▶ Boundary1	0	0	0	0	2,751,266	0	250,115
▼ Flux ausm	0	0	0	0	22,070,655,852	0	2,006,423,259
flux_ausm_mp_ausmdv_star_	0	0	0	0	2,751,266	0	250,115
▼ Flux basic	160,854	80,000	294,302	183,120	0	1,341,502	406,114
flux_basic_mp_fct_	80,427	40,000	147,151	91,560	0	670,751	203,057
flux_basic_mp_richtmyer_	80,427	40,000	147,151	91,560	0	670,751	203,057
▶ Friction drag	0	0	0	0	6,878,165,000	7,243,664,921	1,933,426,606
▶ Friction Wall	0	1,600,240,000	0	0	11,005,064,000	1,068,285,236	8,213,720,562
▼ Grid	92,541	80,000	294,302	183,120	5,502,532	1,341,502	1,043,203
grid_mp_update_buffer_	92,541	80,000	294,302	183,120	5,502,532	1,341,502	872,558
▶ Hydraulic	0	0	0	0	5,502,619,105	10,194,374,699	1,426,999,437
▼ NCT Scheme	80,427	40,000	147,151	91,560	2,751,266	670,751	453,172
nct_scheme_mp_nc_rhs_scheme_	80,427	40,000	147,151	91,560	2,751,266	670,751	453,172
▶ Output	215	44	0	0	0	0	28
▼ Problem	3,259,760,572	6,002,680,063	9,071,864,184	7,326,430,301	26,026,793,862	21,199,975,041	11,234,820,395
problem_mp_eigenvalues_	0	0	0	0	1,375,633,000	3,353,755,000	429,944,364
problem_mp_physical_flux_	1,287,282,562	800,240,000	2,943,902,906	0	0	2,416,567,210	1,414,713,321
problem_mp_physical_rhs_	1,287,041,281	800,120,000	2,943,461,453	0	1,375,633,000	2,414,554,957	1,539,186,665
problem_mp_primitive_data_	0	1,600,800,021	120,445,610	0	4,126,928,035	3,890,129,892	1,340,189,909
▼ PV Relax	643,400,000	400,000,000	1,471,510,000	915,600,000	1,375,633,000	3,353,755,000	1,500,743,727
pv_relax_mp_relax_pressure_	643,400,000	400,000,000	1,471,510,000	915,600,000	1,375,633,000	3,353,755,000	1,500,743,727
▼ Solvers	80,427	40,000	147,151	91,560	2,751,266	670,751	453,172
solvers_mp_compute_residual_	80,427	40,000	147,151	91,560	2,751,266	670,751	453,172
▶ Thermodyn	0	1,600,800,021	4,536,745,698	0	16,400,172,200	0	2,698,679,321

Figure 4.6: Function calls per function.

4.5.2 Analysis

Strictly from a software point of view, EMAPS is based on an iterative numerical method in order to calculate the solution, which depends on the mesh and grid size, the number of iterations done and the physical model used.

The main loop in EMAPS in an iterative one, and therefore it is not possible to parallelise it, unless a complete re-write of the software is carried out. The execution flow and data structure are crucial for the parallelisation task, as the first one will show where to start the parallel regions in order to minimise the number of times that new threads are created, while the second one will show how to split the tasks across the threads.

The blocks present in the execution flow are the initialisation, the solving and the clean-up at the end, which includes outputting data into files. The solving block is very large and contains two core elements, the skeleton for the iterative

process and the math models, as shown in Fig. 4.7.

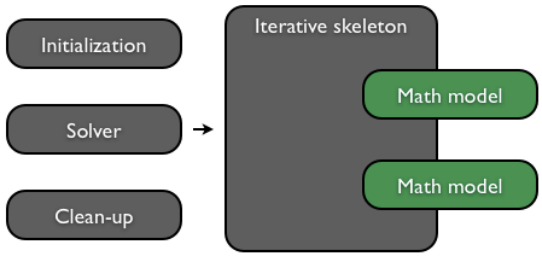


Figure 4.7: Overview of the execution flow.

In Fig. 4.4 it is possible to observe that on average the functions and subroutines in the *PROBLEM* module, which is the actual math model, are called eleven billion times, which is the highest number in the figure. Therefore the actual calculations are mainly done at the outer end of the call tree, and this indicates that a parallelisation at this end would be very costly in terms of synchronisation and other overheads.

In the data structure it is possible to categorise data into two main types, permanent and non-permanent ones. Pipe characteristics defined in the input files are examples of permanent data, while cell values in the grid are examples of non-permanent data. Grids are stored in a structure data type, and they are global variables in the sequential version.

The grids themselves are stored in a structure data type, but those are global variables in the sequential version. Multiple pipe elements can be defined in a single simulation and in the sequential version each pipe component is represented by a single grid. Fig. 4.8 shows how the grids are handled in the program. The variable *Grid handle* is global and it contains pointers to all the grids available.

The grid itself is declared as *TYPE* variable, which in Fortran is used as a wrapper.

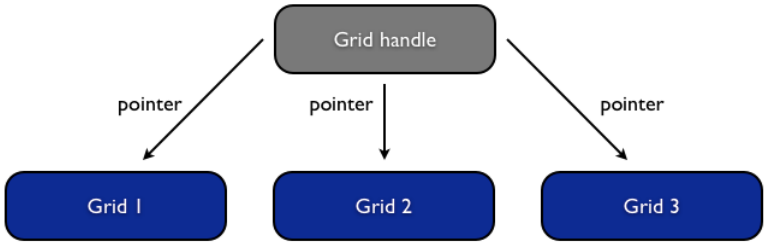


Figure 4.8: Grid management.

The grid layout is shown in Fig. 4.9. The diagram shows the presence of core cells as well as ghost cells. Those ghost cells are used to pass values at the borders of the grid to other grids and store the values from other grids. Moreover data arrays contain a second dimension, which is used to calculate different values for each cell. As an example, the first row of the matrix stores the velocities of the fluid at the different points, while the second row stores pressure values.

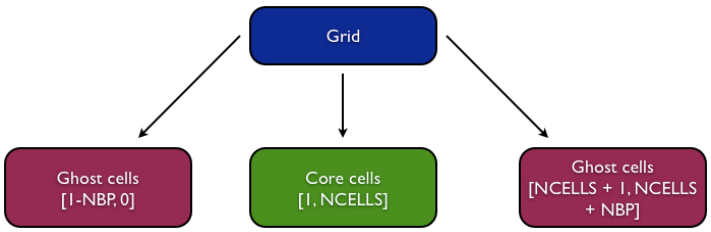


Figure 4.9: Grid cell structure.

Adaptive mesh refinement is used in EMAPS with the aid of levels (Fig. 4.10). When the error in a given grid is higher than a specified tolerance, two new grids will be created on the next level. Those two grids will have the same number of cells as the original grid, but each of them will only cover half the length of the original one (Δx is halved).

Figure 4.11 gives a brief summary of how the levels are handled. The *LEVEL_HANDLE* points always to the first level, and from there it is possible to move through the levels by using the pointers inside each level. The level is a custom type in Fortran,

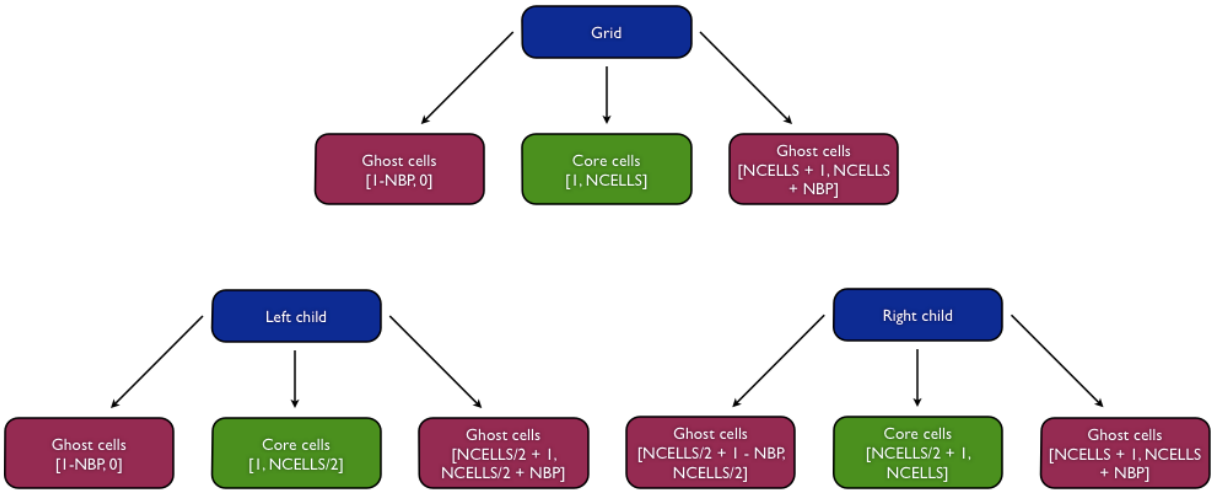


Figure 4.10: Adaptive Mesh Refinement.

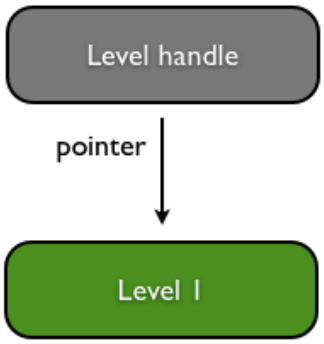


Figure 4.11: Level management.

just like Grid (or *TPATCH*). Figures 4.12 and 4.13 show the grid structure from two different viewpoints. The first one is seen from a "parent grid" while the second one is seen from a "child grid" viewpoint. The names of parent and child are related to the adaptive mesh refinement.

4.5.3 Testing

In order to ascertain that the parallel version of EMAPS was consistent with the original, sequential version, it was necessary to enhance the existing test scripts.

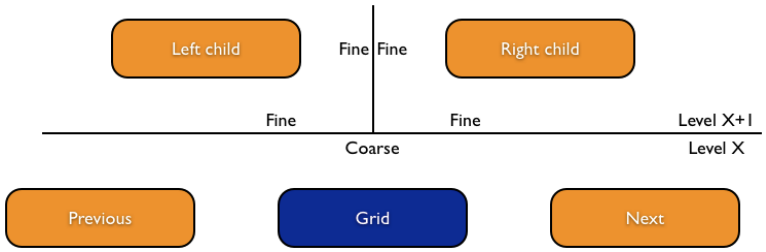


Figure 4.12: Grid structure, seen from a parent point of view.

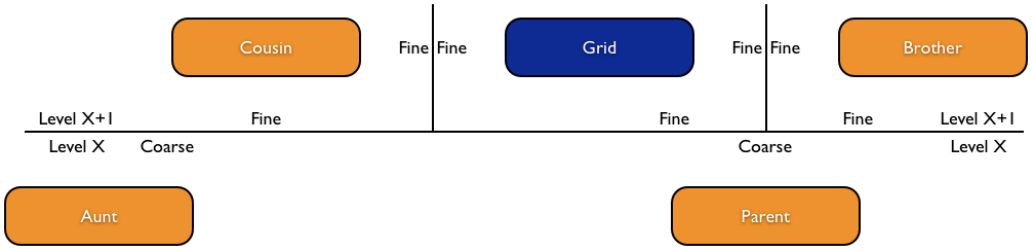


Figure 4.13: Grid structure, seen from a child point of view.

A complete validation test suite was written, allowing running of tests for all different models and comparing new output files with original ones. Test reports include passes and/or fails for each model, and in case of failure, the discrepancy is shown. Moreover parallelisation brings important changes to the source code, and for debugging and checking the quality of the software it was necessary to write a complete testing framework.

The main idea behind the testing consists of running predefined tests and comparing specific output files to pre-existing output files. The pre-existing output files and the corresponding input files are located in folders named after each test case. Sequential test cases are all under a folder called *examples*, while all parallel test cases are under a folder called *examples_parallel*. In the case of parallel tests there are different directories for outputs resulting from 1, 2, 3 and 4 processors used: this check is necessary to make sure that no major differences occur due to different grid partitioning. Most test cases have a uniform and also

an adaptive grid option, in order to test that adaptivity still works correctly after any modification.

The actual test scripts are in folders called *Test_Scripts_Serial* and *Tests_Scripts_Parallel*. All executables are created, new folders are copied from the *examples* directory containing the input files, and then each individual test is run. For each mathematical model there is a series of tests, eg. for Watson model there are 8 test cases:

- Faucet
- Faucet_AMR
- Geometry
- Geometry_AMR
- Shaha
- Shaha_AMR
- Wave_Growth
- Wave_Growth_AMR

The output files will then be compared with the ones from *examples* folder and if there is a discrepancy of more than 0.5% for any line of data, then the test will fail for that case and all lines that caused the failure will be saved in a file for later use.

The whole test suite can be run manually by running a bash script, called *run_all_tests*. It will also run automatically whenever someones makes a change to the EMAPS code (see next section for versioning control).

4.5.4 Continuous integration

Due to the presence of various colleagues working simultaneously (and also using more than one machine each) on the source code and adding new models, it became necessary to establish a continuous integration environment. Originally EMAPS was on CVS (Concurrent Version System), but it was migrated to Sub-version (SVN), a newer software versioning and revision control system. The availability of many graphical user interfaces for interacting with SVN repositories helped in deciding in favour of SVN. After users submit their changes (integration) back to the server, an automated build of the software and a complete run of the test suite is completed. In case of failure, the output will include time of last submission and name of user, plus details of the failure. In order for continuous integration to work, it is important that users submit their changes frequently, in order to catch errors early in the stage of development. Currently all users have switched to SVN and are happy with the benefits that it has brought. Fig. 4.14 shows a schematic representation of the automated continuous integration environment. A general assessment on a future MPI implementation was also

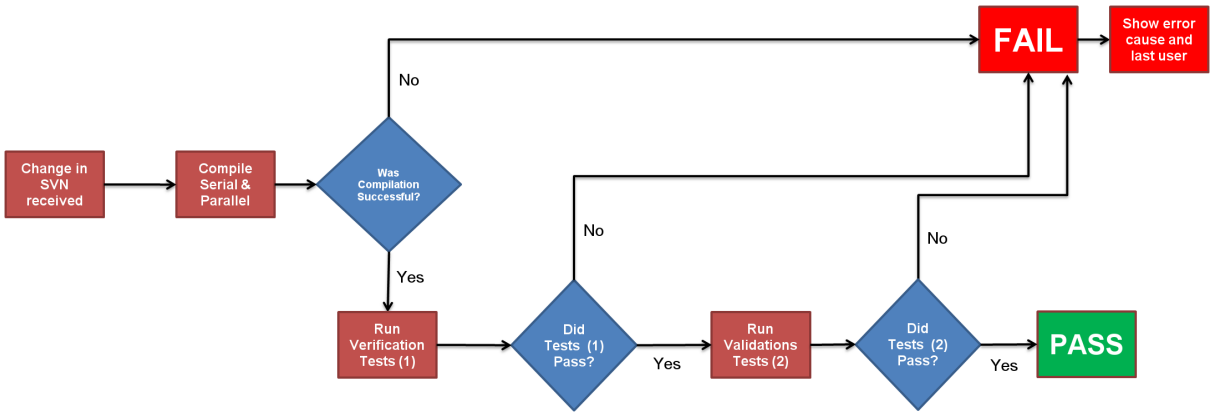


Figure 4.14: Continuous integration environment with automated testing.

carried out, although the MPI implementation was deferred for future work.

4.5.5 MPI

MPI represents a different challenge in its implementation, and though it has clearly a drawback in the maintainability of the application, the scalability is much higher compared to OpenMP. In fact, for OpenMP to work, the machine needs to work with shared memory, so basically what is needed is a single PC with a lot of cores. The scalability of the application still needs to be assessed, and as the quantity of data seems to be limited due to the one dimensional character of the application, the full potential of MPI might not be of use. Furthermore, an important feature of the EMAPS software, the adaptive mesh refinement, might kill the efficiency of an MPI application due to the very high number of communications required.

4.5.6 OpenMP

OpenMP was used to implement parallelism, because it is the quickest method in the available time-frame for this particular case, as the parallel regions can be invoked locally.

From the literature review on OpenMP it can be concluded that in order for the shared memory architecture to work efficiently, there should be no communication between the threads at any time. This requires that data read in different threads must be shared, and therefore shared data have to be defined before creating the parallel region. This should not be an issue when dealing with small parallel regions, going over a single OpenMP construct or just a few of them. However when trying to implement a rather large parallel region, it could become rather cumbersome to determine all the required variables and make sure they do exist and are allocated in memory before starting the parallel region.

A particular difficulty is added by the Adaptive Mesh Refinement used in EMAPS, as grids cannot be created "on the fly" when required. Because it is impossible to create these grids in the shared memory, they have to be taken into account as well. Moreover the actual data in the grids are handled with allocatable arrays and, because threads cannot allocate memory in the shared memory space, this has to be done before starting the parallel region. Hence an important alteration has to be made to the existing code to ensure a correct parallelisation.

4.5.6.1 New grid structure

In order to handle the grids in this shared memory environment, it is necessary to create new global variables. This has to be done with two objectives in mind: keeping the current idea about grid creation on demand and enabling iteration over the grids on a same level.

Thus a new set of arrays will need to be introduced. One array will contain the actual grid objects, and a set of arrays will contain the pointers to those objects in the given context. This is illustrated in Fig. 4.15. In this example there are two levels: x and $x + 1$. The pointers in the levels don't need to be in a specific order, as their only purpose is to make sure the work can easily be split.

As level x contains the parent grids to those in level $x + 1$, or put in another way, the grids in level $x + 1$ are the child grids of those in level x, the maximum number of grids in level $x + 1$ is exactly double the maximum number of grids in level x. As the size of the arrays cannot be changed during runtime in the threads in the parallel region, they will have the size of the maximum number of grids for the given level. Hence, when there is no grid for some of the pointers, they will be *NULL* pointers.

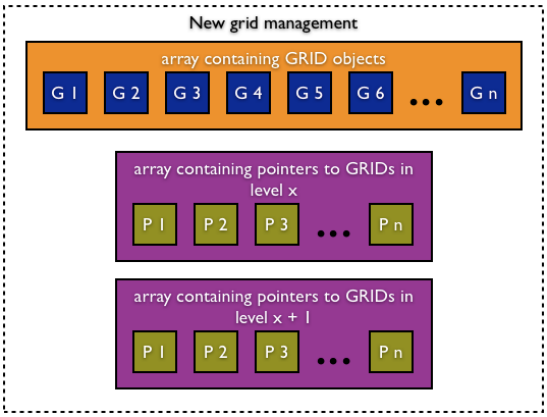


Figure 4.15: Illustration of the new grid management separating the actual objects from their use in the different levels.

As all the grids will be created at the start of the application, an alteration will have to be made to the grid type, as it will need a flag to store whether it is in use or not. By doing this, the process of creating grids when required can be maintained, but instead of actually creating a new grid, an existing one from the objects array (with flag set to "not in use") will be taken and filled with the required data.

Moreover as the grids are created in memory before parallelism starts, and also before they are actually fully initialised, they all have a fixed size. Their size is equal to the maximum size that will be required. As some grids will not have this exact size, further amendments have to be made to the grid type, enabling the grids to be used with a smaller size than they actually are in memory.

As stated earlier, a level hierarchy is used in EMAPS. On each level there is a given number of grids, related to grids on the neighbouring levels. Grids on the same level are fairly independent, except for a couple of ghost cells, and during the calculations made in each iteration, there are only very few cross-references. However references to related grids on other levels occur more often.

Therefore the loops working on the grids can be parallelised using the fork-join model (Sun Studio 11, 2005). The basic of a fork-join implementation is to invoke a parallel region, and then split the work equally over the processing nodes. But loops that are iterating through the levels should not be parallelised using this model, as they are not very independent, due to the relations between the grids on the different levels.

Here is an example of the application of parallelisation using the fork-join:

```
! INVOKE PARALLELISM HERE
! SPLIT WORK
DO WHILE (GRID in LEVEL)
  ! DO SOMETHING WITH THE GRID
ENDDO

! DO SOME OTHER THINGS

DO WHILE (LEVEL)
  ! DO SOMETHING WITH THE LEVEL
  ! SPLIT WORK
  DO WHILE (GRID in LEVEL)
    ! DO SOMETHING WITH THE GRID
  ENDDO
  LEVEL = LEVEL%NEXT
ENDDO
```

So as can be observed in this example, a new issue appears: some of the code in the parallel region might need to only be executed once, as, unless work is split, it will run on all processors. There are constructs in the OpenMP specification that enable this kind of behaviour, but these restrictions force the application to

be sequential in this part. Because all benefits of parallelism are lost there, it is crucial to limit these constructs to a bare minimum.

4.5.7 Simulations with parallel version of EMAPS

Tests were carried out on EMAPS using three different mathematical models: Burger, Watson and Single Pressure Model (SPM4s). The details of these models are included in the literature review, in sections 3.6, 3.7 and 3.8.

The model that showed the best speed-up was the Single Pressure Model. Speed-ups of simulations run in parallel compared to sequential simulations became significant when longer pipes and/or time scales were involved. There was an even higher speed-up when comparing parallel AMR to sequential AMR runs, due to the distribution of the grids over different processors. Successful simulations of pipes up to 100 km long were carried out. Inclined pipes were also successfully simulated in parallel.

Graphs showing speed-ups for the different models are shown in Fig. 4.16 and 4.17. The best results are obtained for simulations with SPM4s model, as expected due to its structure.

The Burger model is clearly not suited for parallelisation due to its simplicity, which allows it to be efficient even in sequential simulations.

Thus in summary, the following features (among others) of EMAPS were successfully parallelised:

- All mathematical models
- Any new models: they can be parallelised by following a simple set of instructions, without the need of a deep understanding of OpenMP

- Adaptive mesh refinement
- Geometry: inclined/vertical/horizontal pipes
- Test suite with cruise control: parallel runs are now part of the testing.

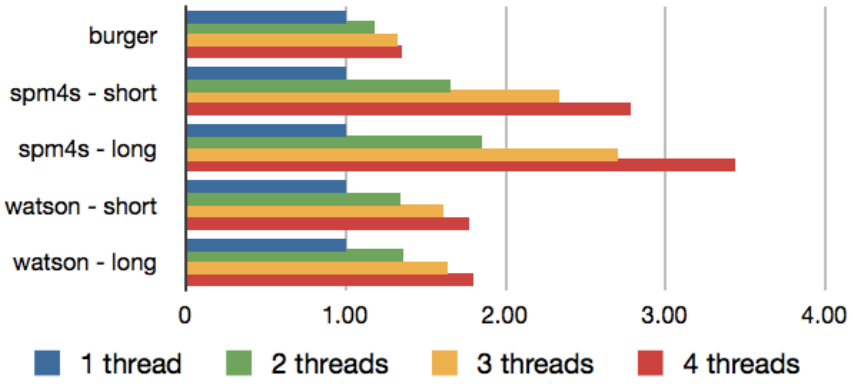


Figure 4.16: Speed-ups for simulations using Burger, Watson and Spm4s models. The baseline is the parallel version with only one thread. *Long* stands for long pipe.

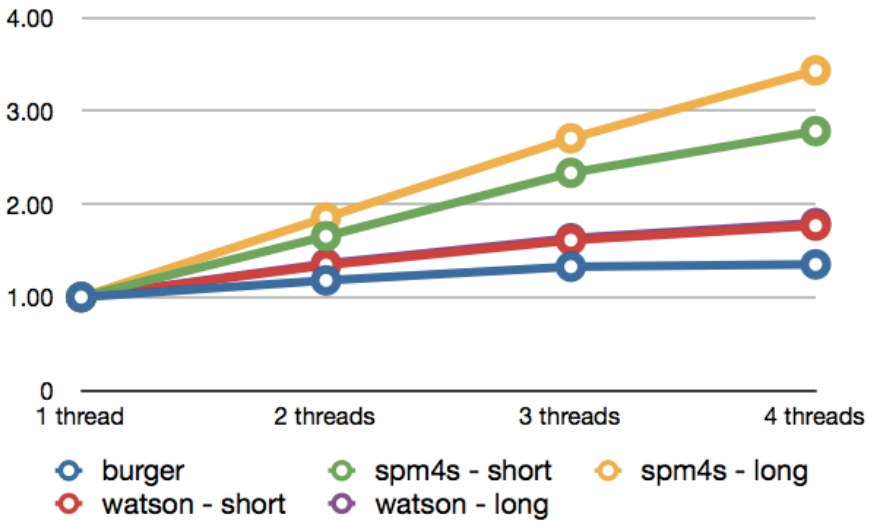


Figure 4.17: A different view of Fig. 4.16. The baseline is the parallel version with only one thread.

4.5.8 Summary of Parallelisation work

Some parts of the parallel part of the code are highly dependent upon the grid size and other parameters determining the time the simulation runs, and it is possible that at a certain stage the efficiency of a parallel run will have reached its maximum with the chosen number of cores. All present models have been parallelised, and instructions on how to parallelise any future models are also provided, with minimal knowledge of parallelisation required for any future user. The efficiency of the parallel version may however vary from one model to another.

The testing suite and the continuous integration environment introduced will allow developers of EMAPS to work more efficiently by enabling them to find early any errors and also if their changes affect other models, without having to manually check through all of them.

The speed-ups have been best for single pressure model, and even more so for long pipes and/or long simulation times. Further speed-ups were observed when adaptive mesh refinement was used. The parallelisation of EMAPS has enabled it to be used on long pipes, which normally would have required an inordinate amount of time to run.

The journal publication of the above procedures (Kalogerakos et al., 2012d) can also be useful to anybody who is working on a code using iterative schemes and grid that wishes to parallelise it. The main issues faced and the reasons for the various approaches are given and it is quite likely that another project will face similar issues when parallelising.

4.6 SPM4s for hydrodynamic slug flow cases

The SPM4s model (explained in section 3.8) was used in order to simulate experimental results obtained in Imperial College (Manolis, 1995). These results are known to exhibit slug flow.

It was assumed that at the start of the simulation the initial conditions included a uniform stratified flow. The boundary conditions are fixed flow rates and liquid holdup at the inlet, and fixed pressure at the outlet. Moreover it was assumed that outlet pressure was equal to atmospheric pressure, and temperature was equal to 21C at inlet. In order to compare the results of EMAPS with the results obtained by WASP at Imperial College, the length of the pipe (horizontal) was set to 38m, the diameter equal to 0.078m, and the step was chosen to be 0.036m. This resolution was found to satisfy mesh independence.

Case	V_{SG} (m/s)	V_{SL} (m/s)	α_L (s^{-1})	$\phi_{Manolis}$ (s^{-1})	ϕ_{EMAPS} (s^{-1})	$\phi_{Gregory}$ (s^{-1})
22	4.016	0.519	0.670	0.133	0.184	0.136
36	1.548	0.519	0.808	0.244	0.234	0.178
37	3.135	0.534	0.715	0.194	0.129	0.143
38	2.058	0.498	0.766	0.217	0.158	0.151

Table 4.1: Cases used for hydrodynamic slug flow simulation

Table 4.1 shows the initial conditions for the simulated cases, and it also allows a direct comparison between the experimental frequencies as given by Manolis $\phi_{Manolis}$ and the frequencies calculated with EMAPS simulations ϕ_{EMAPS}. V_{SG} is

the gas superficial gas velocity, V_{SL} is the liquid superficial velocity and α_L is the initial liquid holdup. The frequencies calculated using the correlation given by Gregory and Scott (Gregory and Scott, 1969) are also included, but it has to be kept in mind that this correlation was deduced from a set of experimental data from a pipe of a much smaller diameter.

All plots of holdup vs. time were obtained with data at a distance of 20m from the inlet, and it was assumed that the velocity profile coefficient C_V is equal to 1 (see chapter 5 for more details). Simulations of experimental case 22 can be seen in Fig. 4.18, where it can be appreciated that the gas velocity increases when the liquid holdup is high, and thereby it pushes the liquid with a resulting increase in the liquid velocity as well. A history of the slugs passing a specific point (in this case, 20m from the inlet) can be seen in Fig. 4.19: this graph can be used to calculate the slug frequency. Simulations of more cases and also snapshots at different times can be seen in appendix A.1.

In the simulations it is observed that a large wave starts from the inlet, and later it changes into a slug. Sometimes it can also occur that a large slug merges with a smaller slug, and this phenomenon will have a direct impact on the measured frequency. Thus is it advisable to wait for the simulation to reach at least 20 seconds, in order for slugs to stabilise and for the slug frequency to be measured in a reliable and consistent manner.

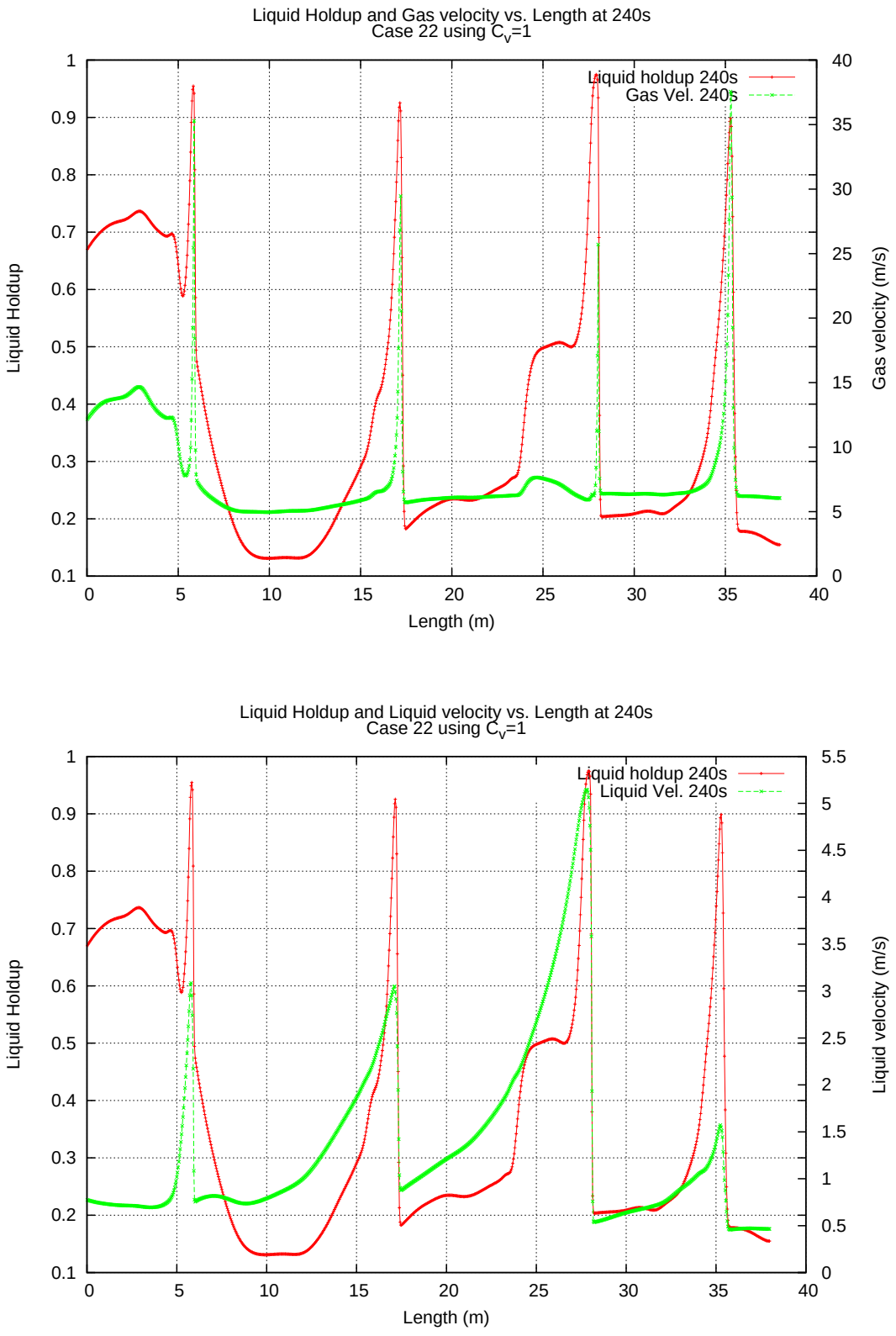


Figure 4.18: Case 22 with C_V=1. Top: liquid holdup and gas velocity vs. pipe length at 240s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 240s.

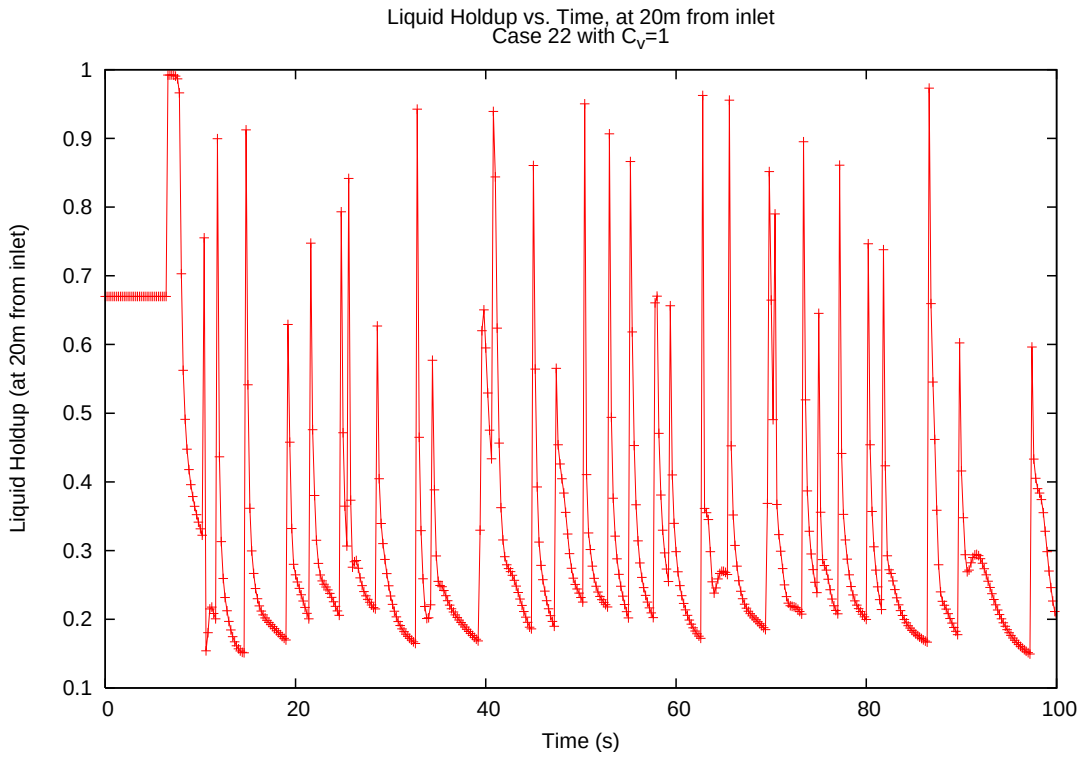
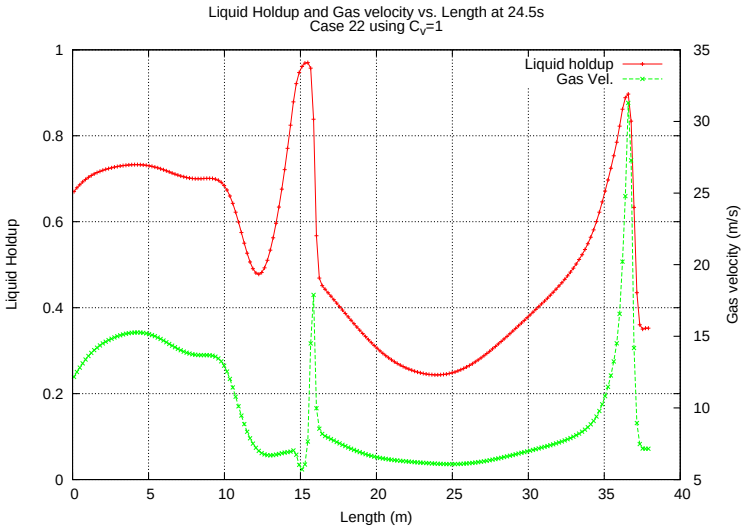
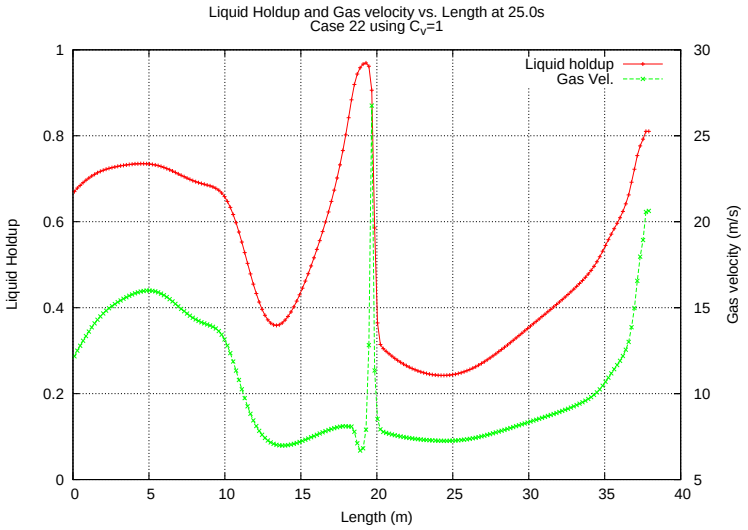


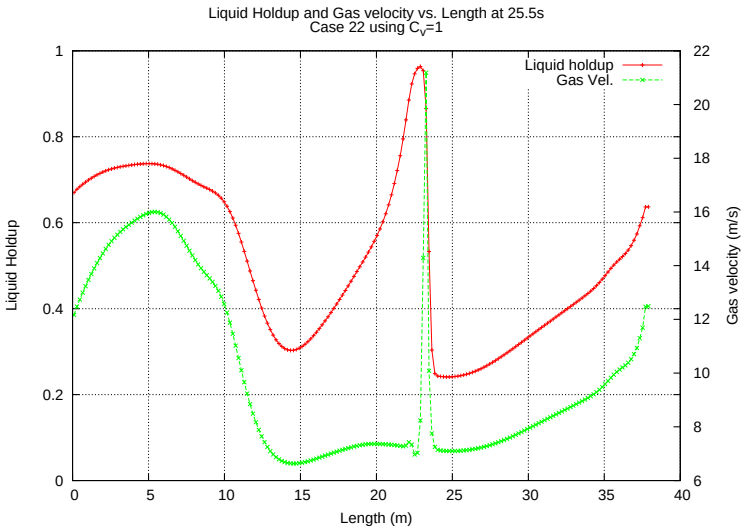
Figure 4.19: Case 22 with C_V=1. Liquid holdup vs. time, measured at 20m from the inlet. Slug frequency is deduced from this graph.

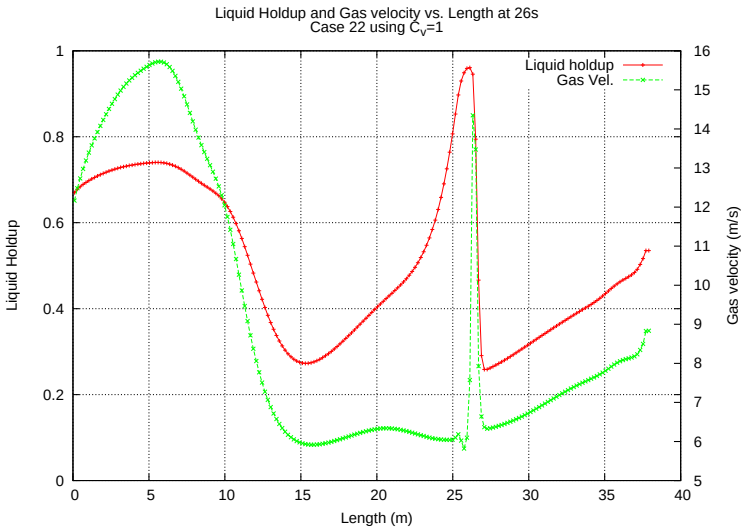
In Fig. 4.20 it is possible to observe an individual slug as it moves along the pipe. This is an important feature of EMAPS, to be able to track individual slugs. Gas velocities follow the shape of the slug as expected. At 24.5 seconds it can be observed the gas velocities appear to be higher than in the other graphs: this could be due to the fact that the liquid holdup becomes very close to 1 and therefore the gas velocity increases rapidly due the very small gas holdup. In cases where the liquid holdup becomes extremely close to 1 (less than a set tolerance limit), then a filtering correction is implemented, whereby a maximum liquid holdup is set and gas velocity is set equal to mixture velocity.

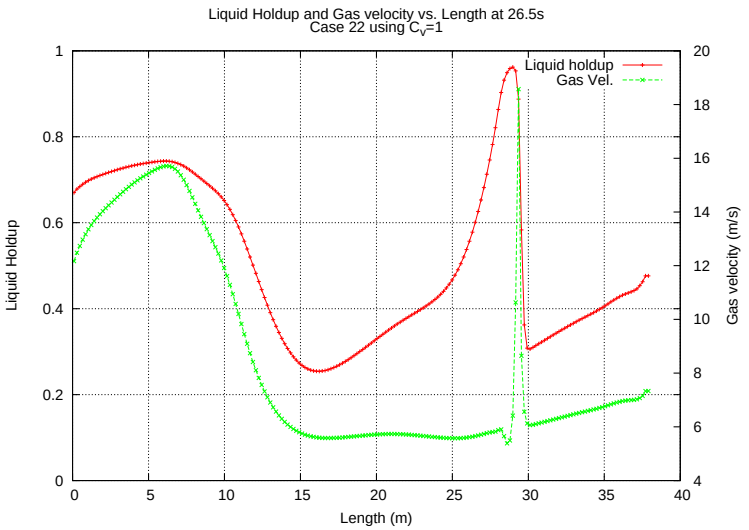
Comparisons were carried out between the outputs, in particular regarding evolution of liquid holdup vs. time and length and slug frequency. The original figure of











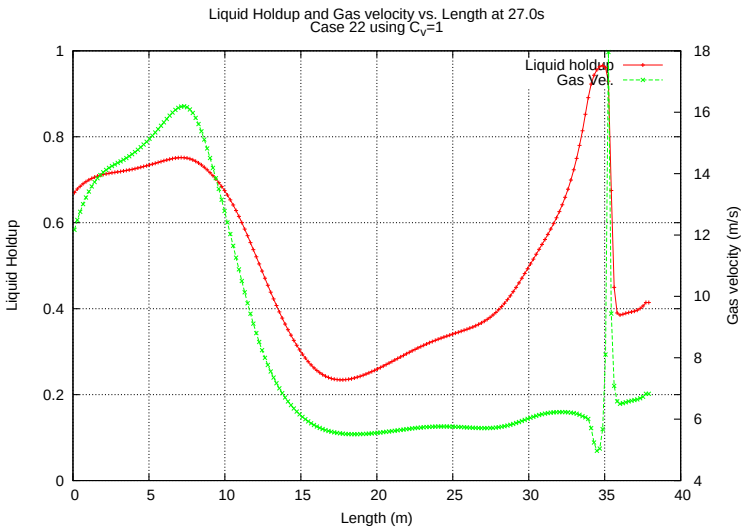


Figure 4.20: Case 22 with C_V=1. The slug can be tracked moving along the pipe.

frequency results can be seen in Fig. 4.21.

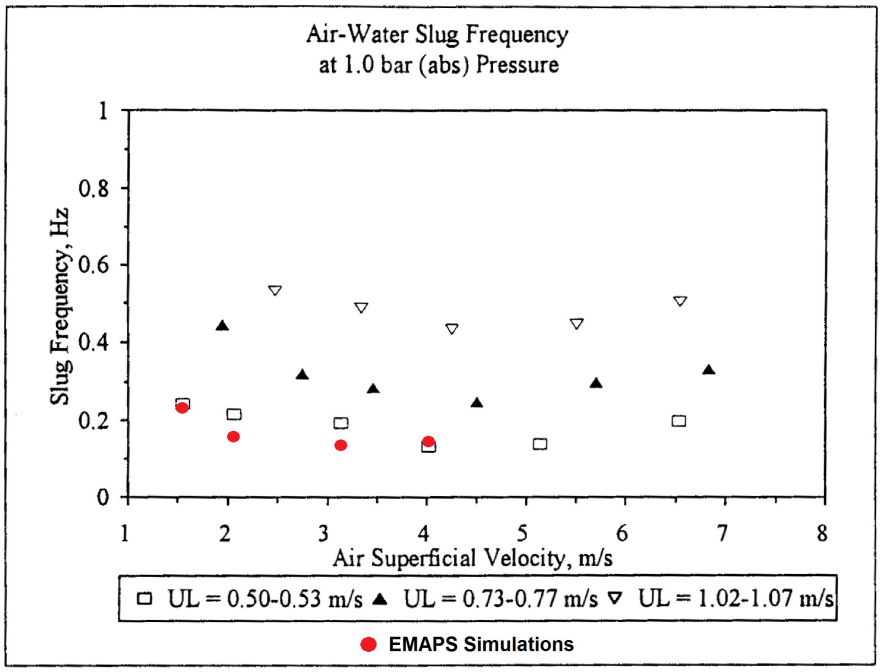


Figure 4.21: Air-Water Frequency vs Air Superficial Velocity at 1.0 bar pressure (Manolis, 1995). EMAPS results are shown in red.

The liquid superficial velocities that have been chosen are in the range 0.519-0.534 m/s, therefore the slug frequency expected according to Fig. 4.21 should be in the range 0.1-0.3s^{-1}, and the results obtained with EMAPS simulations are indeed in the range, as shown in red in the same graph. This is very encouraging, as slug frequencies are notoriously difficult to predict correctly.

4.7 BP Field Data

The data offered by BP to Cranfield University include the 1987 and 1989 Prudhoe Bay Field trials. These stem from a series of tests carried out by BP Research

Centre Sunbury and BP Engineering in conjunction with BP America in August and September in 1987 at Prudhoe Bay. Data were collected from many pads, but for the purpose of the current research the pads chosen were the X-Pad and the R-Pad (Hill and Turner, 1988), where a pad is a drill site consisting of a multitude of oil wells. The tests originally carried out along the X pad flowline were done at 6 different locations. At the start of the flowline (800 ft from the start) stratified-wavy flow was observed. By 1500 ft the predominant flow regime was still stratified-wavy with some short slugs, and at 2100 ft from the start rapid slug growth occurred. The reason that specifically the X-Pad and the R-Pad were chosen is that they provide a situation where slugs were identified and measured, and also they offered the longest section of completely horizontal pipeline, ideal for simulations without the involvement of extra geometry. Data provide details of the initial flow rates (which can be converted to superficial velocities), the pipe properties, the fluid properties, temperature and pressure, and the type of flow that was observed. More details have not been included here due to confidentiality issues.

4.7.1 Simulation of X-Pad

The perfectly horizontal section of X-Pad measures 1425ft, from 11786ft to 13211ft from the start of the pad, with a constant elevation of 43ft. The model used to simulate was the single pressure model (SPM4s). The same resolution was used as with the Manolis cases, since it was found to be mesh independent at that stage.

From the graphs of liquid holdup and gas velocity (m/s) versus length (m) of the pipe (Fig. 4.22 to Fig. 4.25), it can be seen that there are slugs moving from left to right, in the direction of the flow. At t=5.0s, a series of slugs of smaller length

appear to be travelling, again in the direction of the flow. The behaviour of the gas velocity behind the slug appears to confirm that they are slugs, by pushing the liquid upwards. After the distinctly shaped slugs have disappeared from the first 40m section of the pipe, there are some formations that may require further investigation (see section 5.7).

This was an interesting case as it showed that slugs can be predicted with the single pressure model, as long as some specific initial conditions are met, in this case more extreme than those used so far in the wave propagation experiments and in TRIOMPH code simulations. What can be successfully predicted here is the onset of slugging in this pipe with the parameters given, and will allow countermeasures to be taken in order to limit the effects of slugging.

The BP data showed that slugs formed in the X-Pad configuration. The simulations of this drill site using EMAPS with SPM4s showed that slugs were simulated in the calculations. Whilst the quantitative comparison cannot be presented here, this qualitative comparison is very encouraging and provides evidence that EMAPS has the capability to simulate slugs in flows of engineering interest.

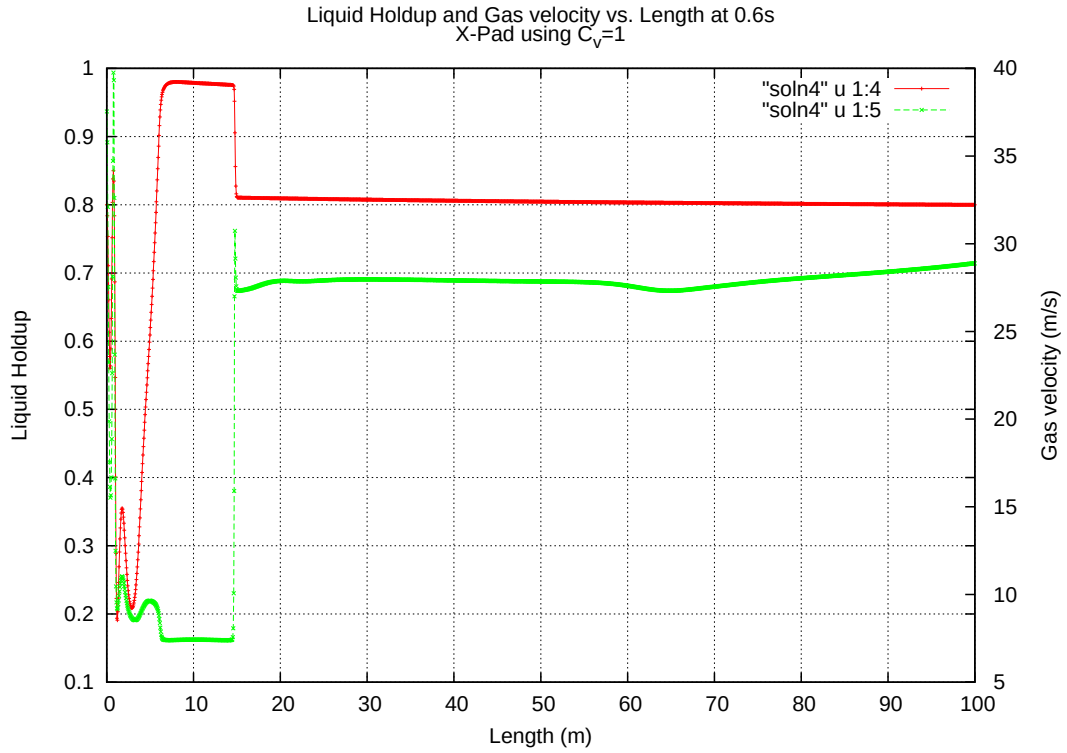


Figure 4.22: X-Pad: Liquid holdup/Gas velocity vs. Length for t=0.6sec

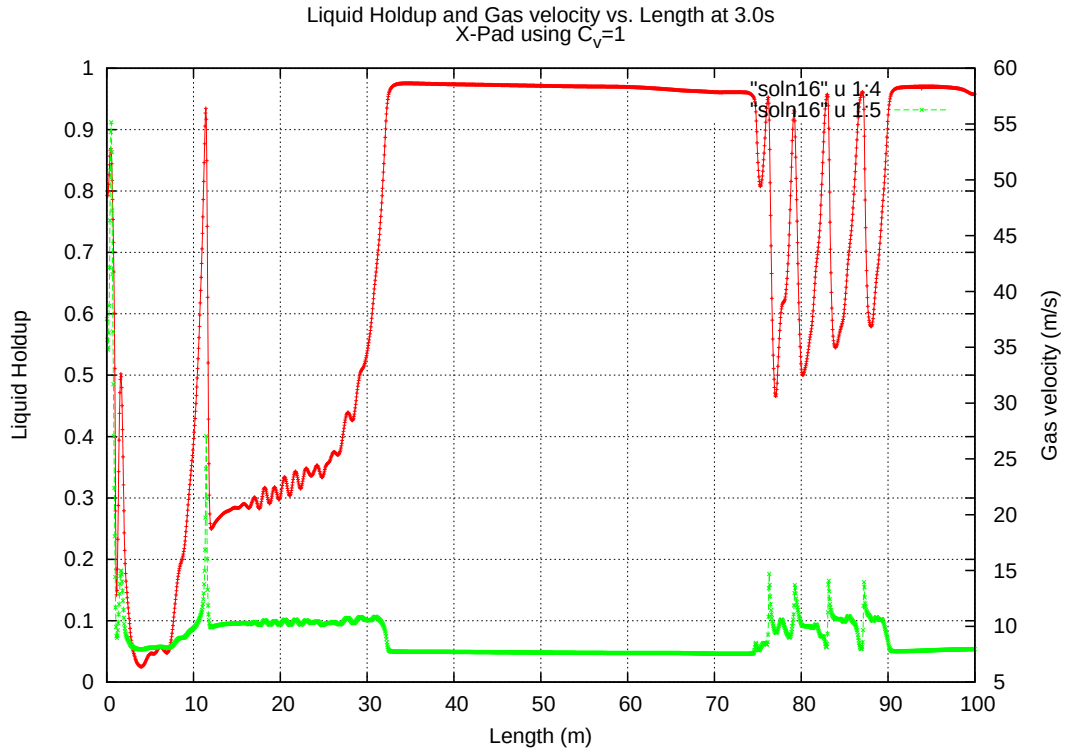


Figure 4.23: X-Pad: Liquid holdup/Gas velocity vs. Length for t=3.0sec

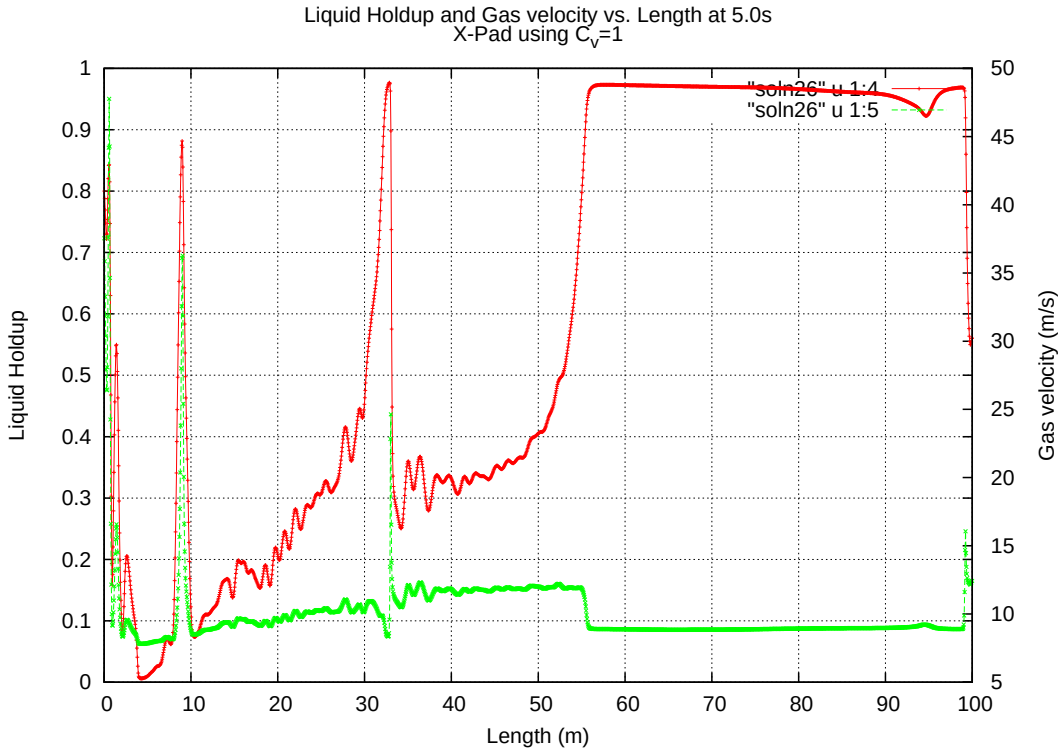


Figure 4.24: X-Pad: Liquid holdup/Gas velocity vs. Length for t=5.0sec

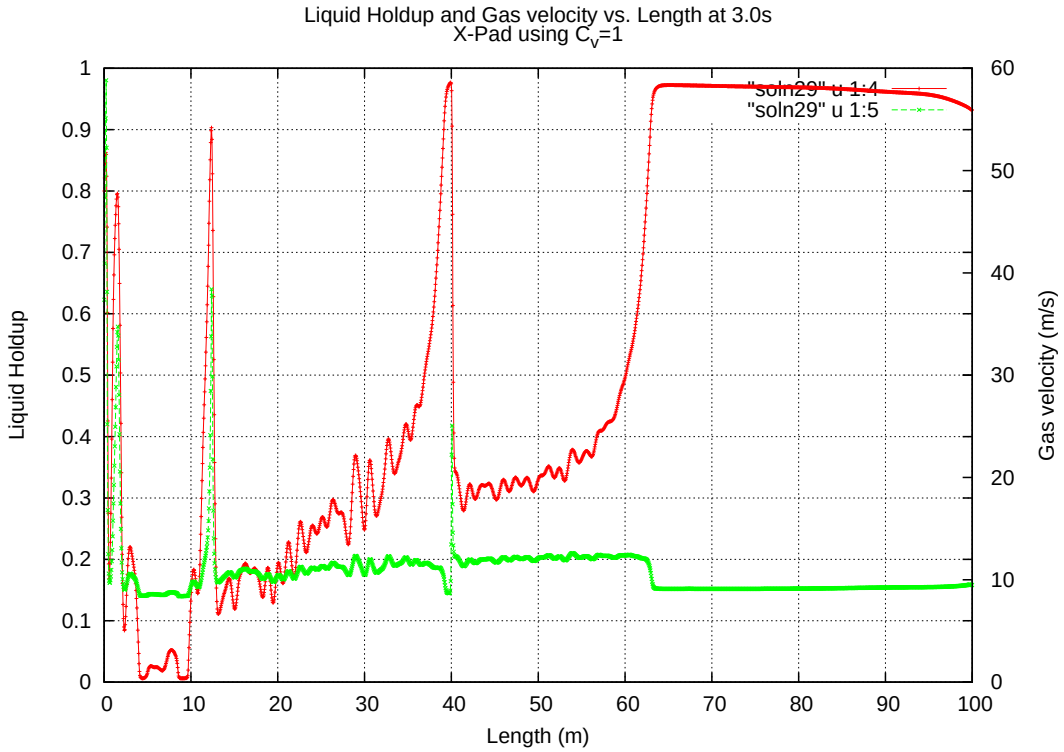


Figure 4.25: X-Pad: Liquid holdup/Gas velocity vs. Length for t=5.6sec

4.7.2 Simulation of R-Pad

A straight section was chosen with a length of 100m, and inlet gas superficial velocity 8.412m/s, liquid superficial velocity 0.427 and initial liquid holdup 0.800. These values were taken again from the data kindly provided by BP. For consistency, the SPM4s model in EMAPS was used. In Fig. 4.26 there is clear slug formation, but at 30m there appears to be a perturbation that is not straightforward to explain. It could be due to the numerical scheme rather than the physical processes. It can be also observed in the X-Pad. At 6s (Fig. 4.27) slugs are still propagating even though the previously mentioned perturbation has now increased in amplitude. At 9s (Fig. 4.28) and even more at 20s (Fig. 4.29) it appears that this perturbation has merged and there is a succession of high peaks of liquid holdup, which could be slugs but do not present some of the usual characteristics.

In observing the graph of liquid holdup vs. time for a point at a distance of 20m from the inlet (Fig. 4.31), the slugs appear to form a stable pattern, with a brief break (i.e. non-continuation of slugs) at a time of around 86s. Also the graph of frequency (Fig. 4.30) shows a frequency which decreases almost linearly from $0.57s^{-1}$ to $0.51s^{-1}$ with increasing distance from the inlet.

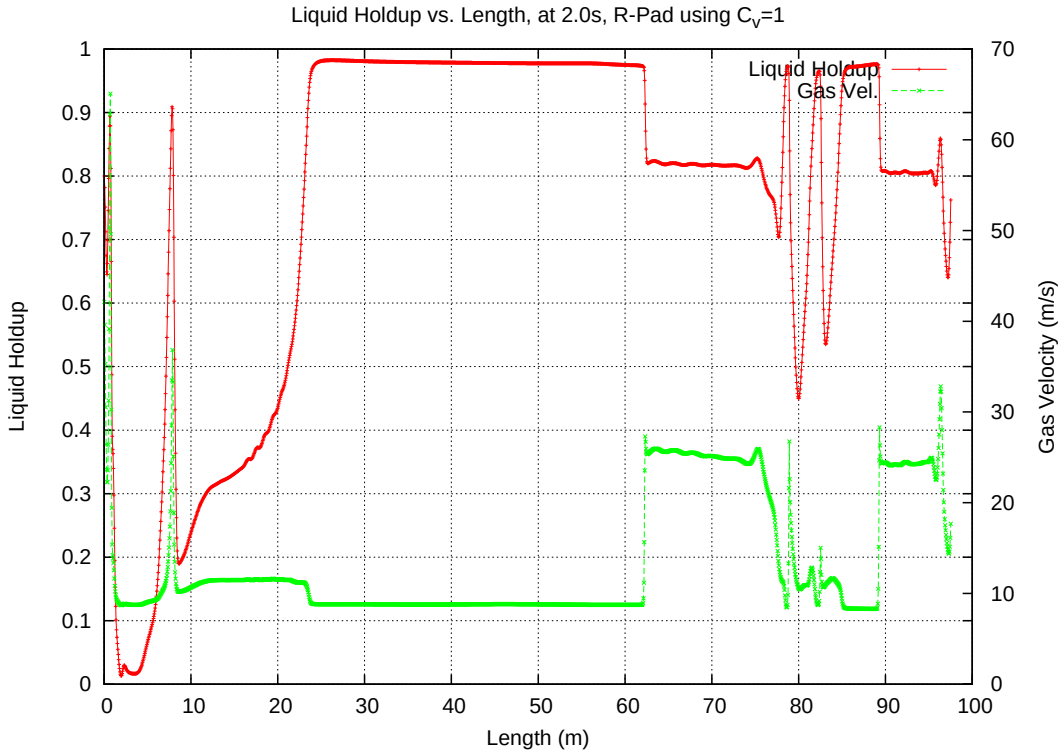


Figure 4.26: R-Pad: Liquid Holdup and Gas Velocity vs. Length at 2.0s

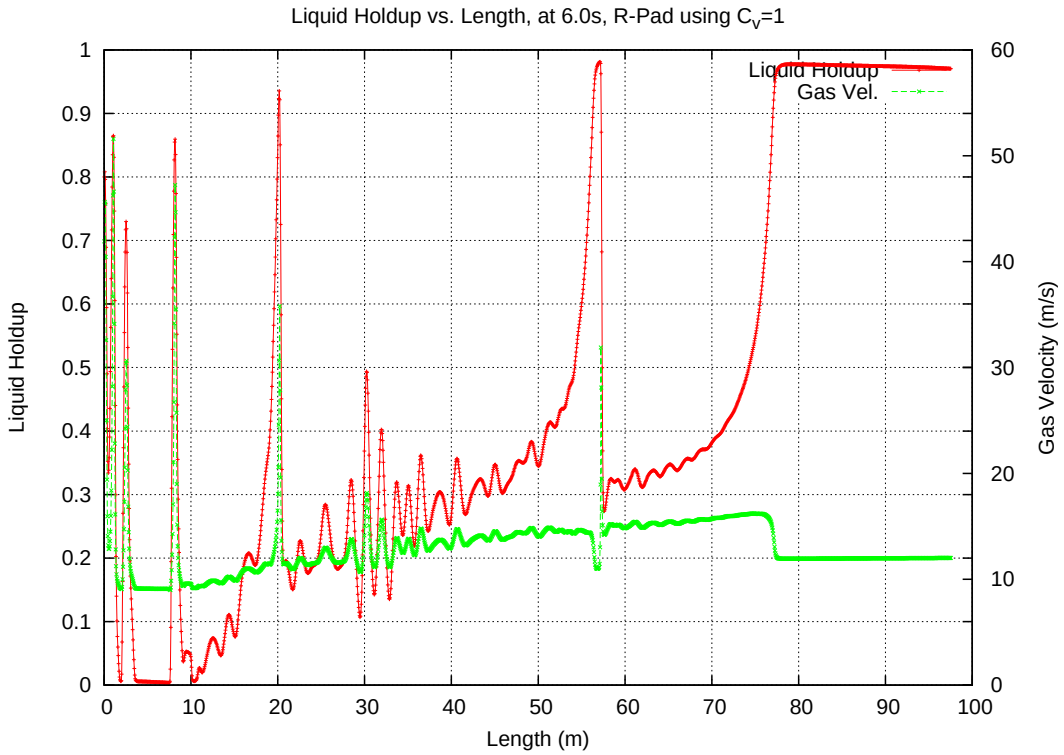


Figure 4.27: R-Pad: Liquid Holdup and Gas Velocity vs. Length at 6.0s

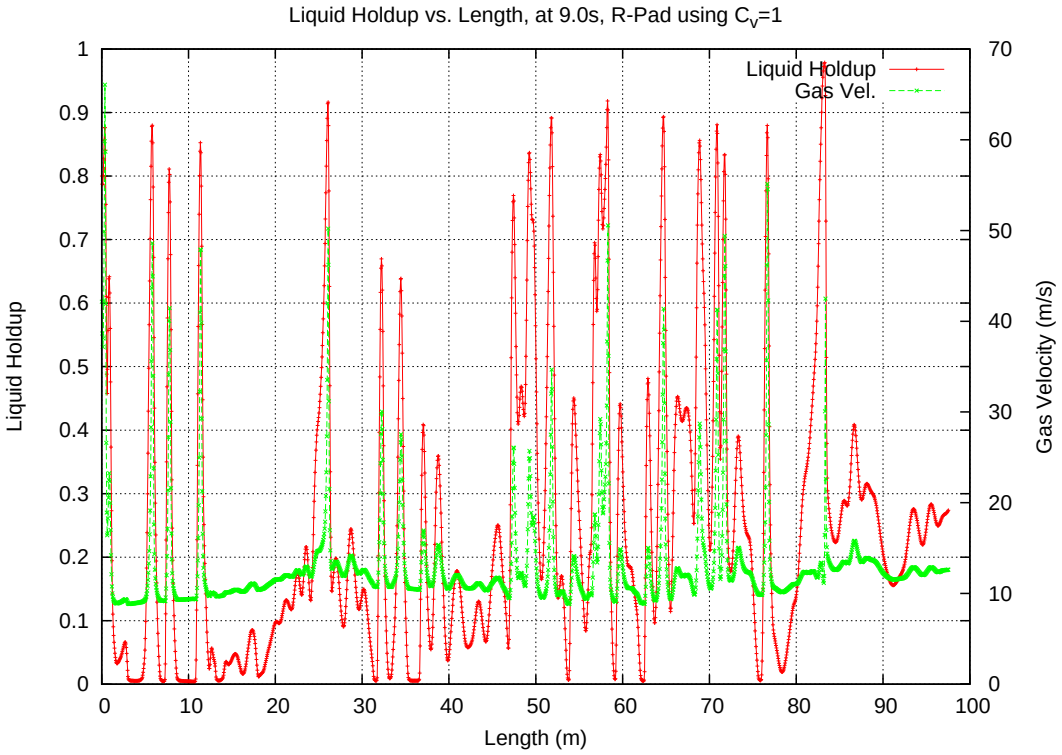


Figure 4.28: R-Pad: Liquid Holdup and Gas Velocity vs. Length at 9.0s

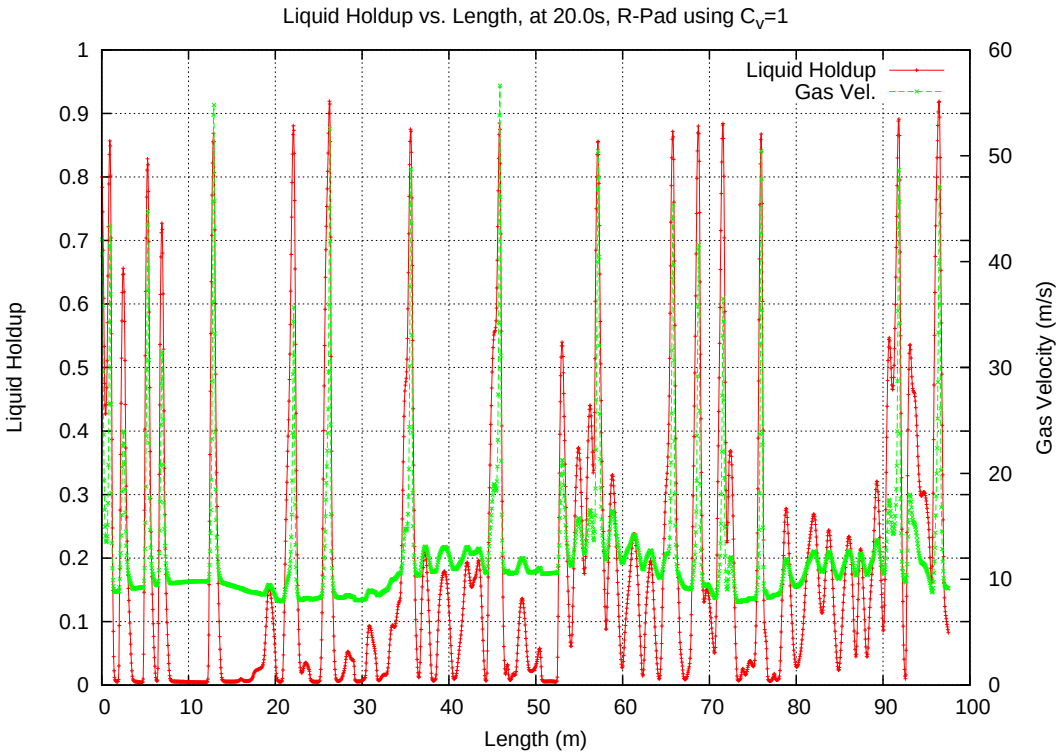


Figure 4.29: R-Pad: Liquid Holdup and Gas Velocity vs. Length at 20.0s

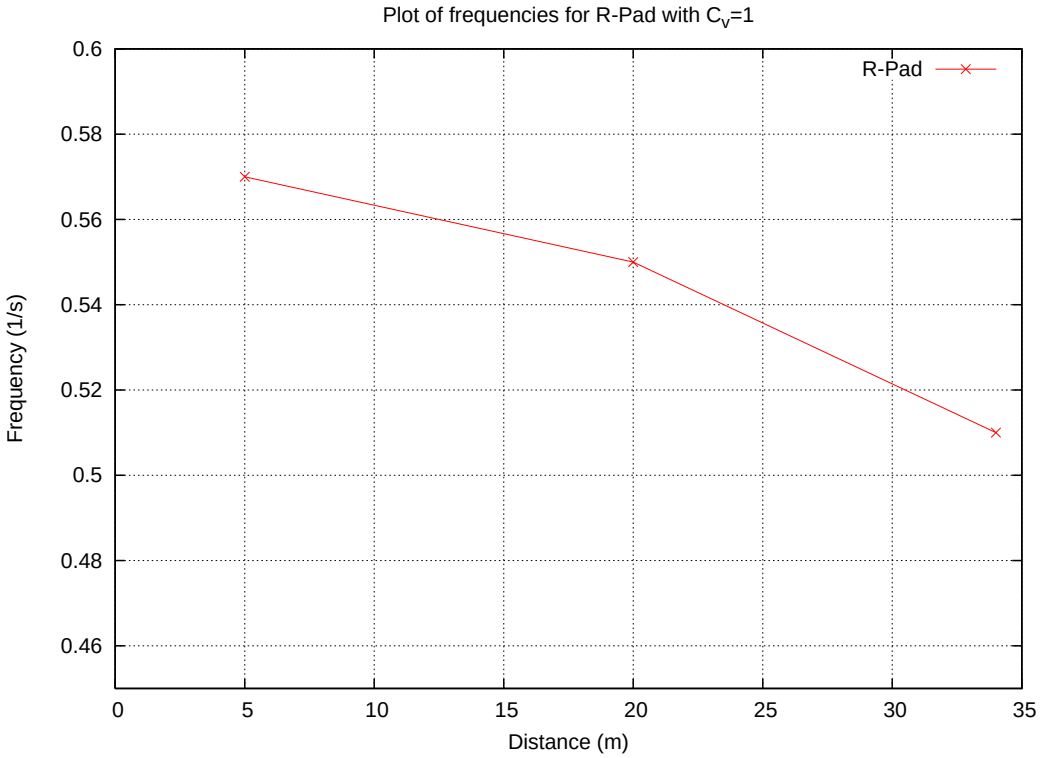


Figure 4.30: R-Pad: Frequency vs. Length

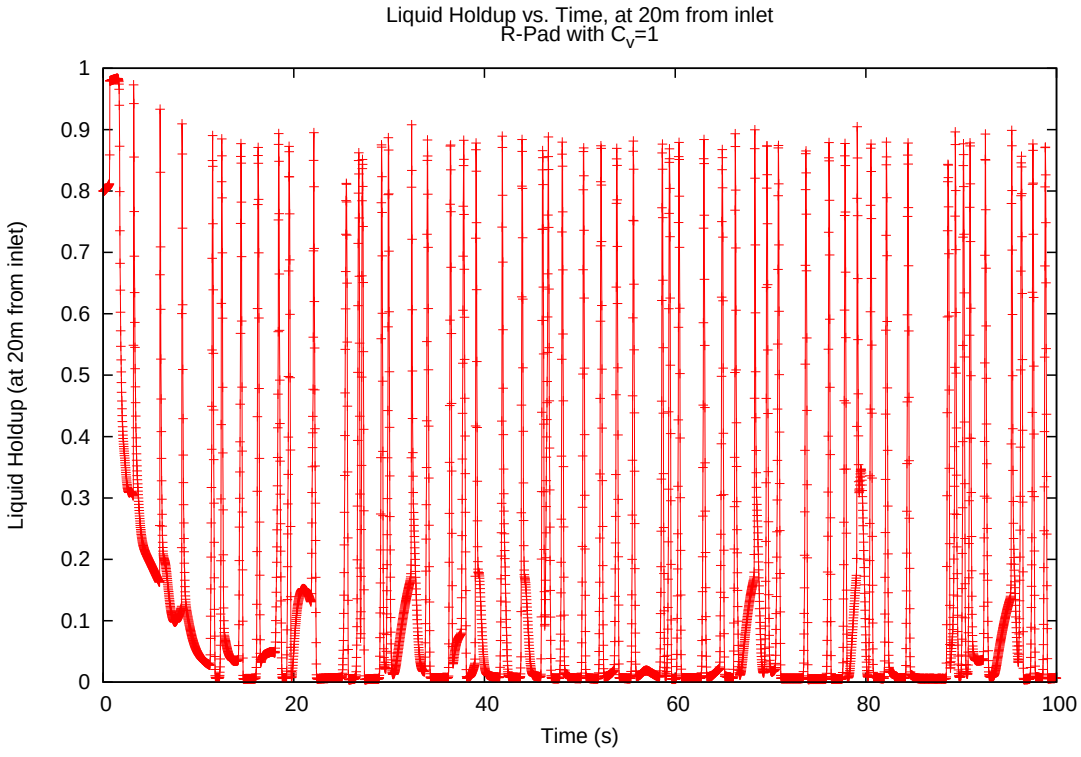


Figure 4.31: R-Pad: Liquid holdup vs. Time at 20m from inlet

4.8 Conclusions

A new version of EMAPS was successfully created by porting versions from divergent branches into a single one. All models, together with geometry capability (inclined/horizontal/vertical pipes) and adaptive mesh refinement were migrated to Sub-version (SVN), a newer software versioning and revision control system, combined with a continuous integration environment (cruise control). After users submit their changes (integration) back to the server, an automated build of the software and a complete run of the test suite is completed.

A new, parallel version of EMAPS was also completed. Parallel simulations on shared memory architecture can now be carried out and good speed-ups have been achieved.

Slug flow simulations were carried out using the single pressure model on data from Imperial College (Manolis, 1995) and good agreement was shown. Simulations on field data (X and R pad) from BP (Hill and Turner, 1988) were also carried out and the onset of slug flow was correctly predicted. It has to be kept in mind that EMAPS may not work outside the region of well-posedness of the two-fluid model, as the equations will become ill-posed. This is an important limitation which affects any one-dimensional code based on the two-fluid model.

References

Darmawan, B., Kamers, C., Pienaar, H. and Shiu, J. (2003), *AIX 5L Performance Tools Handbook*, IBM Redbooks. (cited at page 82)

Gregory, G. A. and Scott, D. S. (1969), 'Correlation of liquid slug velocity and frequency in horizontal co-current gas-liquid slug flow', *AIChE Journal* **15**(6), 933–935. (cited at page 99)

Hill, T. and Turner, P. (1988), Prudhoe Bay Western operating area multiphase flow data August - September 1987, Technical report, BP International, Research Centre. (cited at page 104, 112)

Jia, N. (2012), Numerical simulation of two-phase flow in gas and non-Newtonian shear-thinning fluid flows in pipelines, PhD thesis, School Of Engineering, Cranfield University. (cited at page 81)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012d), 'Use of OpenMP to parallelise a one-dimensional multiphase code', *Computer Physics Communication* . In preparation. (cited at page 97)

Manolis, I. G. (1995), High Pressure Gas-Liquid Slug Flow, PhD thesis, Department of Chemical Engineering and Chemical Technology, Imperial College of Science, Technology and Medicine, UK. (cited at page xv, 98, 103, 112)

REFERENCES

Sun Studio 11 (2005), *OpenMP API User's Guide*, Sun Microsystems, Inc. (cited at page 94)

Chapter 5

Velocity Profile Coefficients

5.1 Introduction

When the one-dimensional system of equations is formulated, one method that can be used is that of the control volume method. The momentum conservation can then be written as:

$$\frac{\partial}{\partial t}\rho \int_{A(x)} u dA + \rho \frac{\partial}{\partial x} \int_{A(x)} u^2 dA = B \tag{5.1}$$

As the equation is averaged over the pipe cross-section, it is natural to define a velocity profile coefficient, defined as C_{Vk}:

$$C_{Vk} = \frac{< u_k^2 >}{< u_k >^2} \tag{5.2}$$

where $< u_k^2 >$ is the average velocity squared for phase k, defined as:

$$< u_k^2 >= \frac{1}{A_k} \int_{A_k} u_k^2 dA \tag{5.3}$$

and $< u_k >$ is the average velocity for phase k, defined as:

$$u_k = \frac{1}{A_k} \int_{A_k} u_k dA \tag{5.4}$$

Mathematically it can observed that normally the average of the square of the velocities is not equal to the square of the averages of the velocities, and therefore C_{Vk} is not normally equal to 1.

5.2 Velocity Profile Coefficients - EMAPS

All models used in EMAPS are based on the simplified assumption that C_V is equal to 1. However, various literature sources report varying values of velocity profiles for different flows, and it has been shown that even small variations of the value of C_V from 1 can change the flow regime map significantly, specifically regarding transition to slug flow (Vielliard (2003)). In Table 5.1 there is a list

Velocity Profile	C_{vk}
Flat velocity profile	1.00
Turbulent flow	1.10
Laminar flow	1.33

Table 5.1: Velocity Profile coefficients (Schulkes, 1994)

of values that have been used. The single pressure model SPM4s in EMAPS has been modified in such a way as to check for the Reynolds number for each phase, and then decide whether it is a laminar flow or turbulent flow, thereby assigning it a predetermined value of C_V. The phase velocity flux function, responsible for computing the physical convective flux terms (in our case, from the AUSMDV* scheme, see section 3.9), was modified in such a way as to obtain a new momentum flux, equal to the original momentum flux multiplied by the velocity profile coefficient. Table 5.2 shows in detail the features of all Manolis

experimental cases that have been used, where V_{SG} is the gas superficial velocity, V_{SL} is the liquid superficial velocity, and α_L is the initial liquid holdup.

Case	V_{SG} (m/s)	V_{SL} (m/s)	α_L
22	4.016	0.519	0.670
36	1.548	0.519	0.808
37	3.135	0.534	0.715
38	2.058	0.498	0.766

Table 5.2: Cases used for hydrodynamic slug flow simulation

The graphs of the four cases, assuming C_V=1, are shown in section 4.6 and they include liquid holdup, liquid superficial velocity and gas superficial velocity.

Following the recommendations by Schulkes (Schulkes, 1994), it was decided to assign to the velocity profile coefficient values of C_V=1.1 for Reynolds number larger than critical, and C_V=1.33 for Reynolds number smaller than critical. The graphs showing liquid holdup vs. time, at a distance of 20m from inlet, do not show any appreciable difference, when changing from critical Reynolds number 2100 (Fig. 5.1) to 2000 (Fig. 5.2). The issue encountered here is the validity of the C_V values used, as they have been taken from literature and may not be directly applicable to our case.

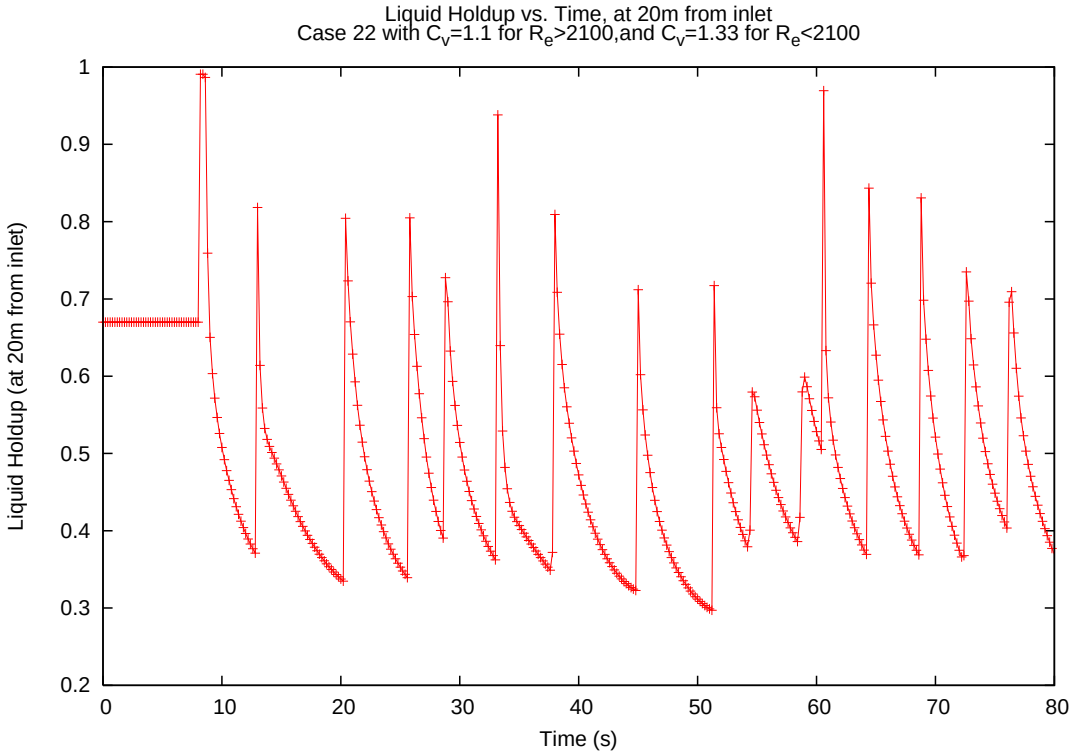


Figure 5.1: Case 22: Liquid Holdup vs. Time, at 20m from inlet

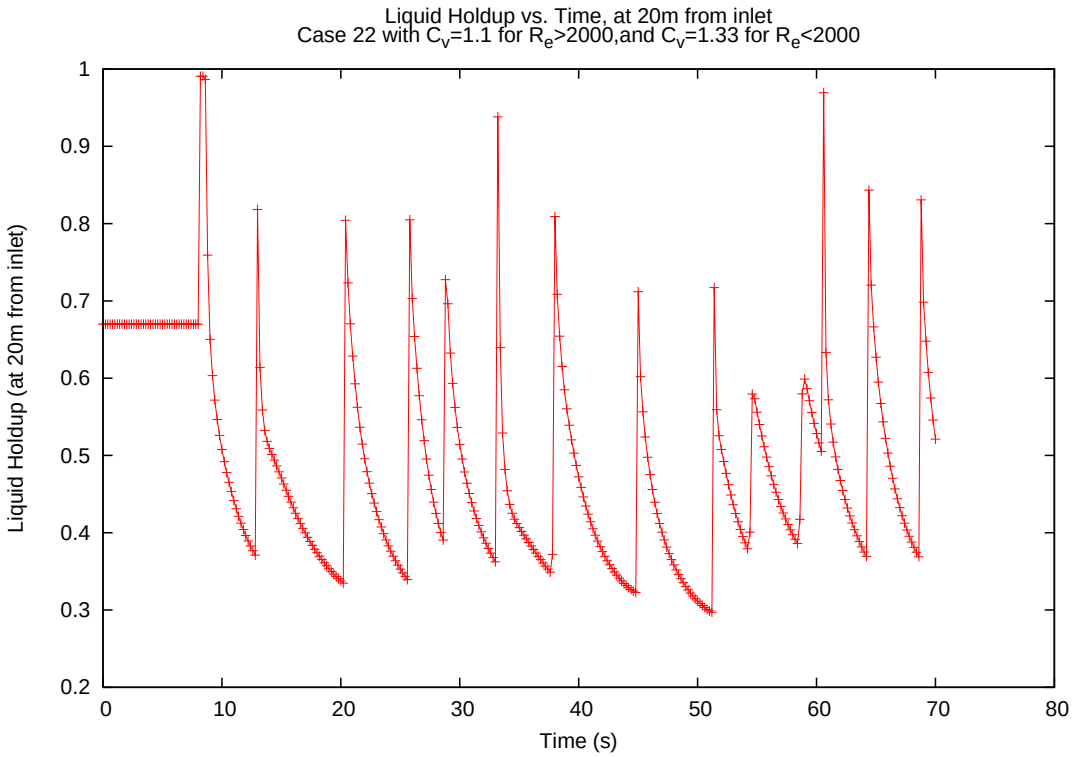


Figure 5.2: Case 22: Liquid Holdup vs. Time, at 20m from inlet

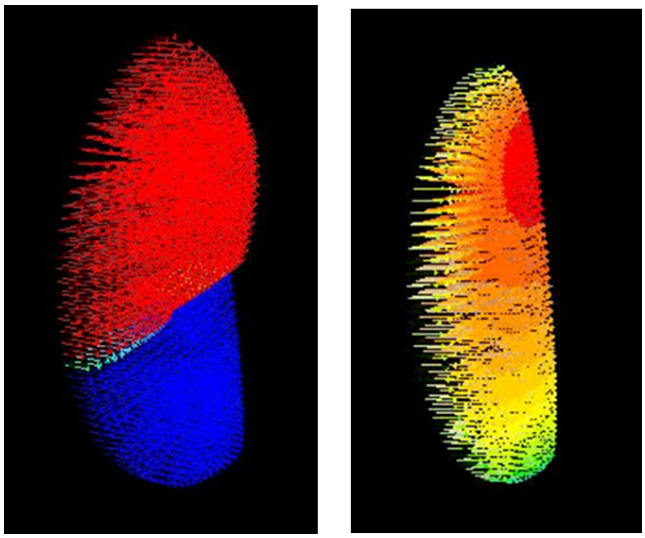


Figure 5.3: 3D FLUENT simulation of Case 36: Velocity vectors. Left figure is stratified flow, while right figure is slug flow.

To obtain more relevant values of velocity profile coefficients, it was decided to use the results obtained by running a 3D simulation using the commercial software FLUENT. Using FLUENT (Volume of Fluid model and $k-\epsilon$ turbulence model), a full simulation of a 13m pipe was carried out with the same initial and boundary conditions as in Manolis case 36. The 3D simulation was completed after three weeks of computations, and from the results it was possible to extract the velocities and therefore calculate average values for velocity profiles. The use of the $k-\epsilon$ turbulence model was deemed appropriate because the 3D simulation results showed good agreement with experimental results. The actual directional velocity vectors can be seen in Fig. 5.3, where the left figure is in stratified flow and the right figure is in slug flow. Fig. 5.4 shows the change of velocity profile when going from 1m from the inlet to 6m from the inlet and it allows a direct comparison.

After averaging, the following values of C_V were obtained:

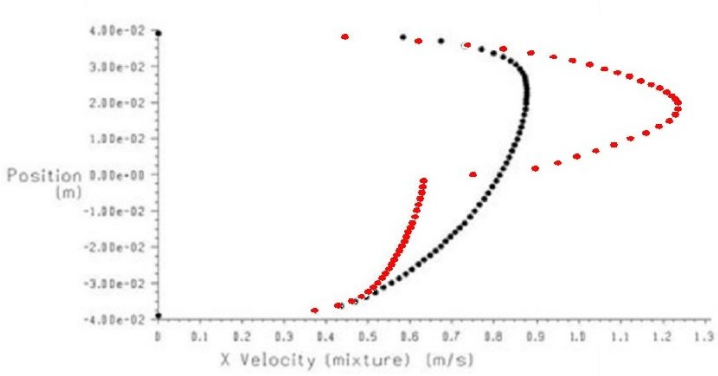


Figure 5.4: Figure showing the change in velocity profile from 1m (red) to 6m (black) from the inlet

$C_V = 1.107$ for stratified flow

$C_V = 1.043$ for slug flow

EMAPS simulations of Manolis cases were then repeated using these newly modified values of C_V, where C_V was set equal to 1.107 if the Reynolds number was less than 2300, and equal to 1.043 otherwise. The graphs that were directly compared with the previous results include:

- Liquid holdup and Gas Superficial velocity vs. Distance from Inlet
- Liquid holdup and Liquid Superficial velocity vs. Distance from Inlet
- Liquid Holdup vs. Time at 20m from inlet

Here the graphs for snapshots at 210s, 240s and time graphs are shown for case 22 (Fig. 5.5 5.6, 5.7) and compared with earlier results in chapter 4 obtained using C_V=1, but graphs for all cases both at 210s and at 240s from inlet are included in appendix A.2. Specific comparisons regarding slug frequency and slug arrival times have been considered in later sections, section 5.3 and section 5.4. Gas

and liquid velocities are also used in order to ascertain that the perturbations observed were indeed slugs. These are the differences that were observed:

- In cases 22 and 36, it can be observed that with the modified C_V there is one more slug at 210s.
- In cases 22 and 36, it can be observed that with the modified C_V there is one fewer slug at 240s.
- In case 37, there is a distinct series of slugs observable at 240s with modified C_V, while originally there is no clear feature.
- For case 38, with modified C_V the slug which is normally present at 210s for C_V=1, has disappeared.
- For case 38, at 240s both simulations show a slug, although at different points along the pipe.

Whilst there is no clear trend in the change in behaviour, there are some significant differences in the observed behaviour, and it appears that depending on initial velocities, there will be a change (increase or decrease or later initiation) of slug formation.

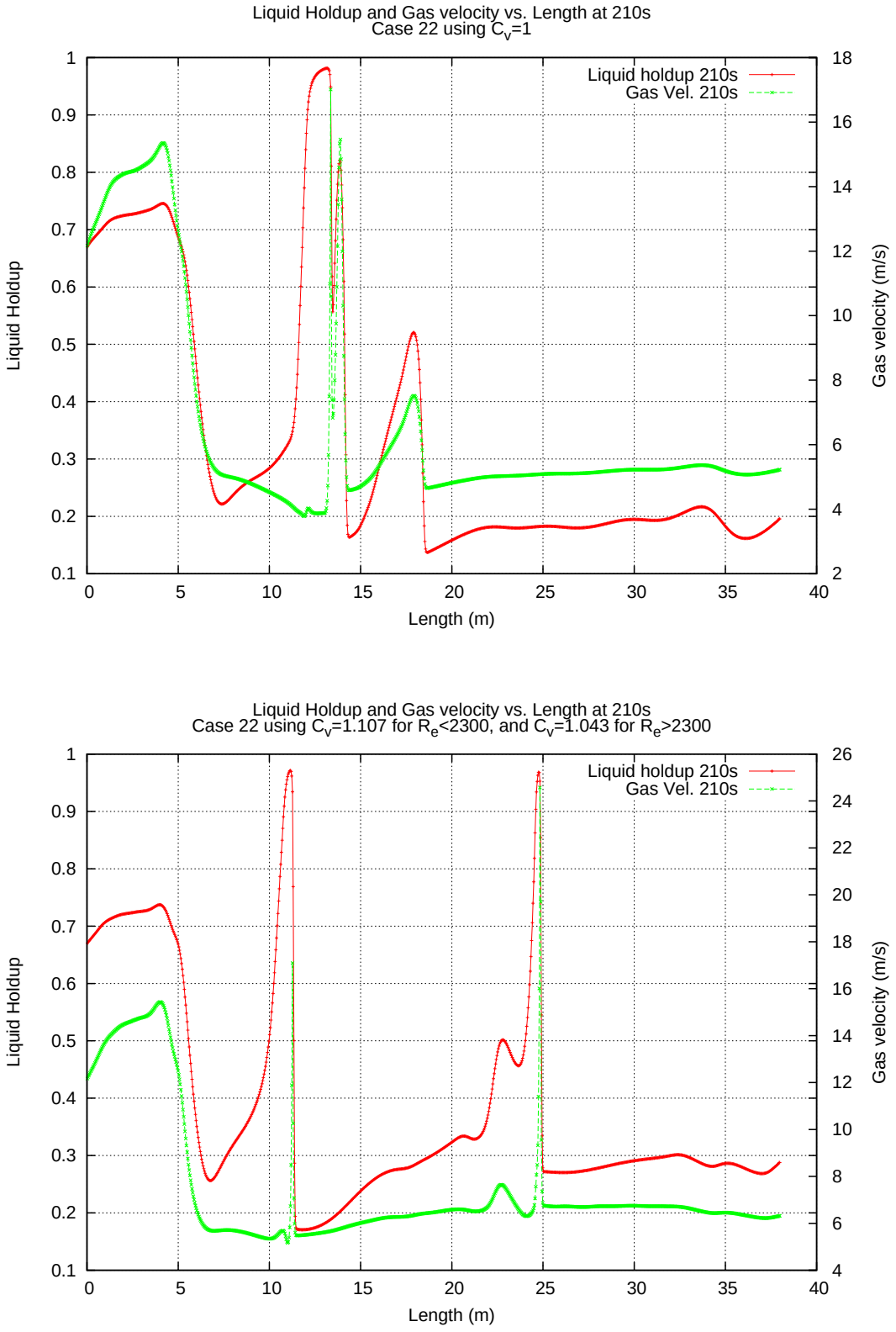


Figure 5.5: Case 22. Liquid holdup and gas velocity vs. pipe length at 210s. Top: with C_V=1. Bottom: with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300.

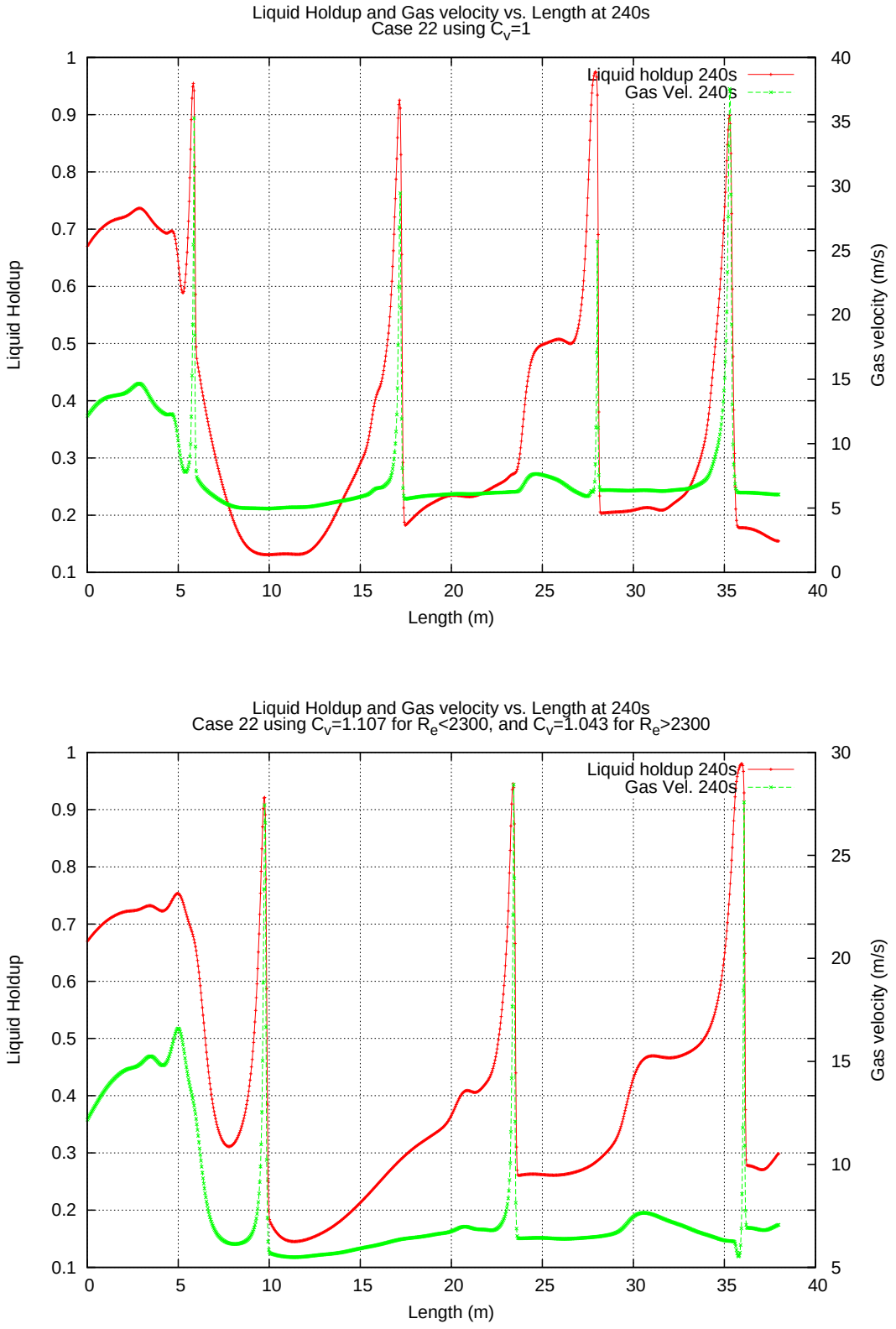


Figure 5.6: Case 22. Liquid holdup and gas velocity vs. pipe length at 240s. Top: with C_V=1. Bottom: with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300.

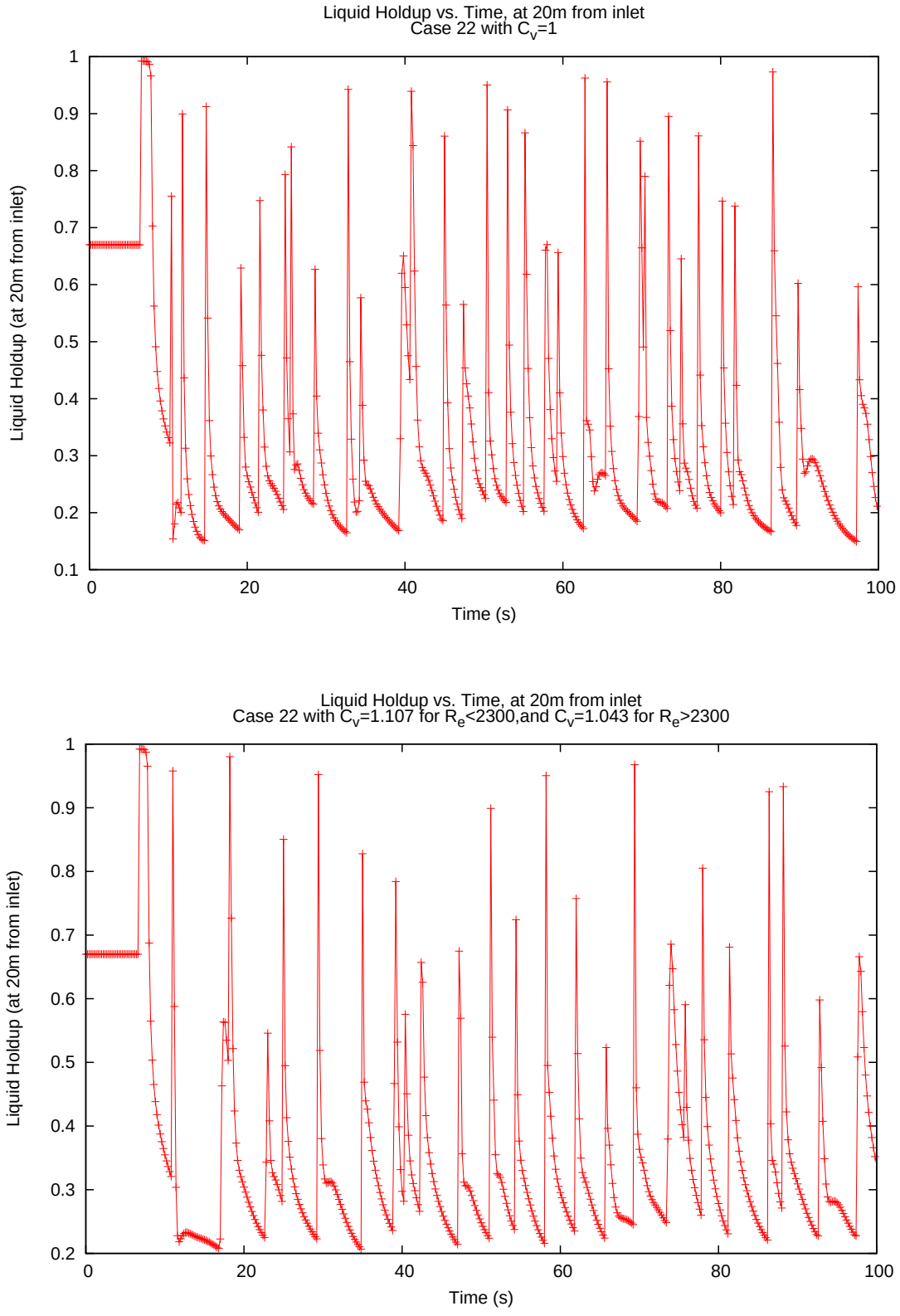


Figure 5.7: Case 22. Liquid holdup vs. time, measured at 20m from the inlet. Top: with C_V=1. Bottom: with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300.

5.3 Evolution of slug frequency

During EMAPS simulation of Manolis cases (Manolis (1995)) slugs have been observed at five different locations along the pipe, and it has been possible to follow the evolution of slug frequencies along the pipe, for different superficial velocities. Using C_V=1, the frequencies calculated are shown in Fig. 5.8, while for modified C_V (i.e. C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300) the frequencies calculated are shown in Fig. 5.9. Manolis found the following results for the frequencies:

- Case 22: 0.1333 Hz
- Case 36: 0.2444 Hz
- Case 37: 0.1944 Hz
- Case 38: 0.2167 Hz

Unfortunately, using the modified C_V leads to an increase to all the frequencies, and therefore the results obtained are not in agreement with the experimental results, and the original results obtained with C_V=1 showed a better agreement. Although the mathematical basis of the implementation of a C_V different from 1 in the momentum equation is a valid one due to the integration over the pipe cross-section when deriving the 1D equations, there are many ways of assigning values to C_V, and what has been shown so far is that an assignment strictly based on the Reynolds number is not an appropriate one.

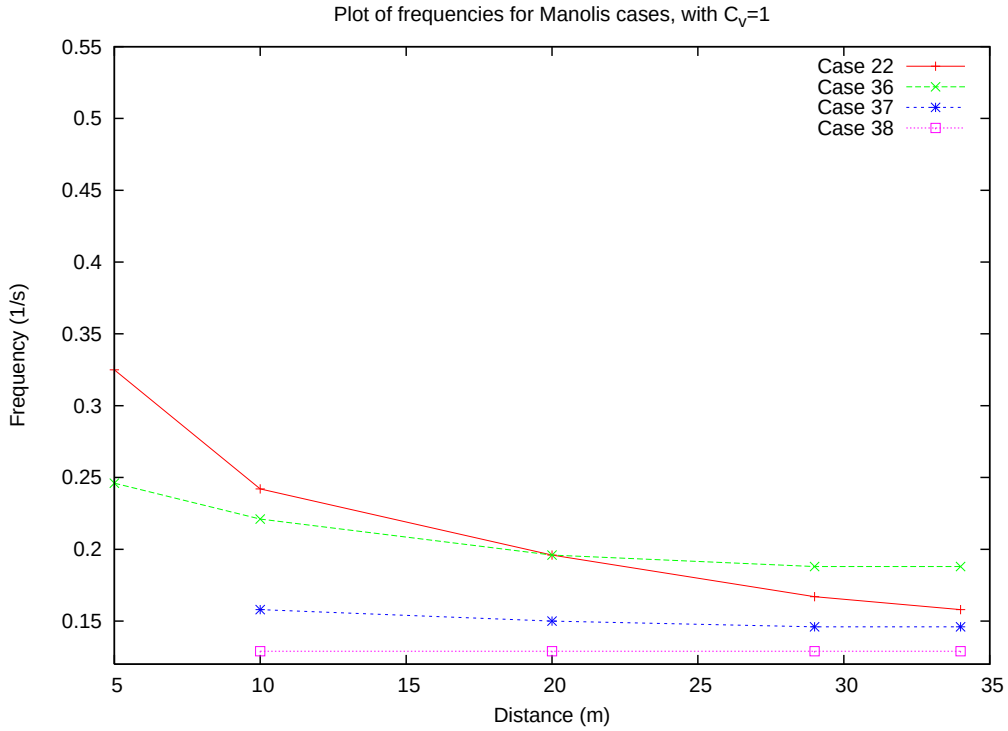


Figure 5.8: Range of frequencies for C_V=1

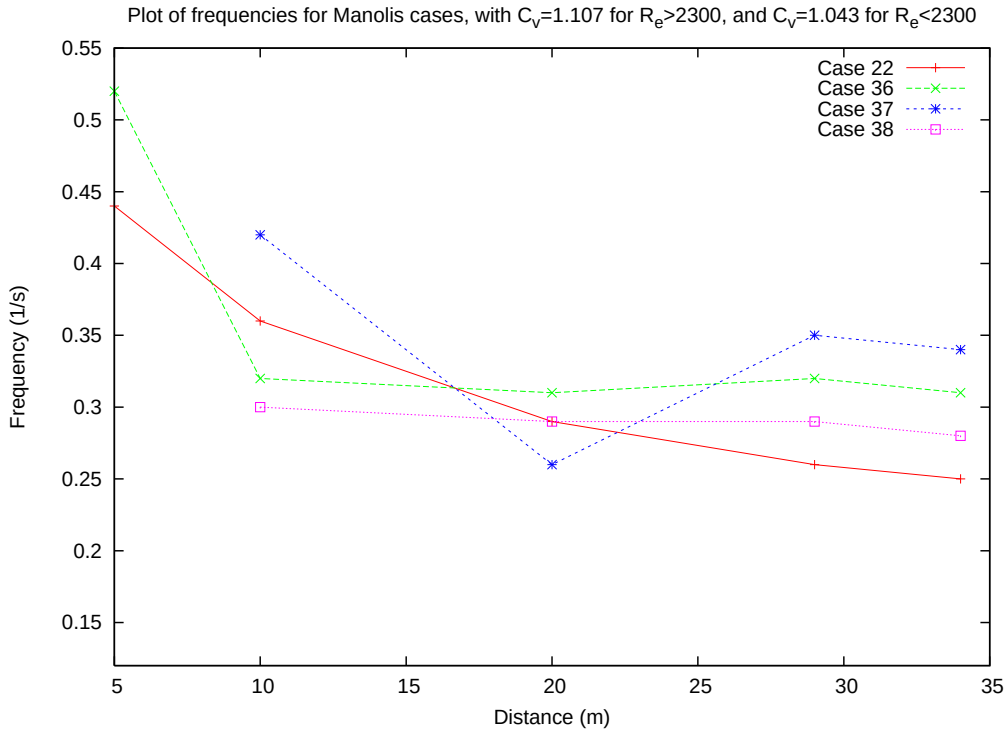
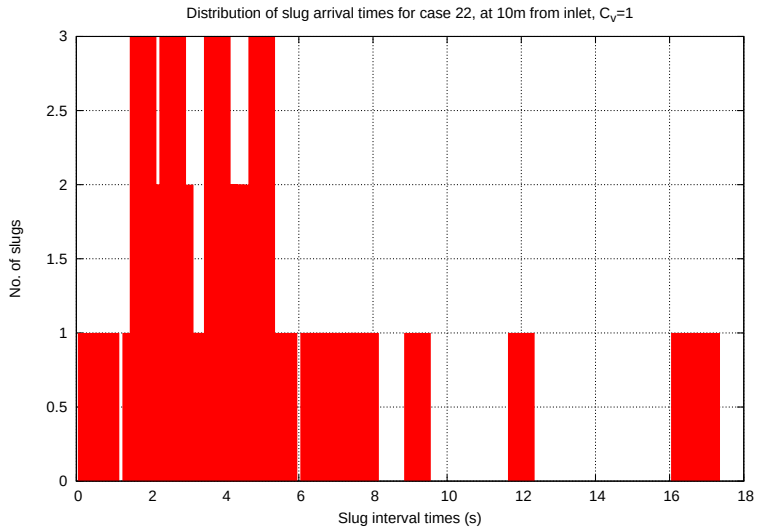


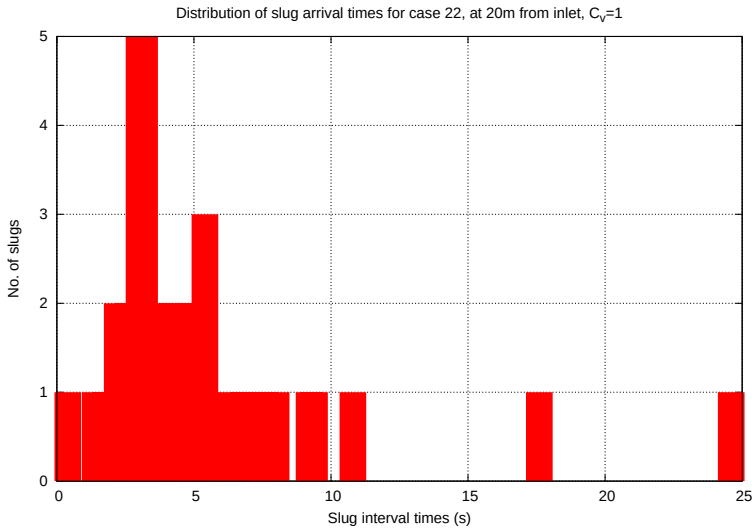
Figure 5.9: Range of frequencies for C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300

5.4 Distribution of slug interval times

It was decided to investigate the evolution along the pipeline of the distribution of time intervals between slugs, and find any changes when using the modified velocity profile coefficients. Here Manolis cases 22 and 36 are investigated. In comparing Manolis case 22 when using C_V=1 (Fig. 5.10) and when using the modified C_V (Fig. 5.12), it can be seen that there is almost no change at all in the distribution at a distance of 29m from the inlet, while at 10m there is a much higher concentration of slug times at around 7s when using the modified C_V. On the other hand, at a distance of 20m, there is the opposite effect, i.e. a broadening of the peak and of the distribution.

For Manolis case 36 the slug interval times are shown in Fig. 5.11 for C_V=1 and in Fig. 5.13 for modified C_V. Again there appears to be very little change between the two graphs at a distance of 29m from the inlet. On the other hand, at a distance of 20m with the modified C_V there is a tightening of the distribution, and at a distance of 10m there is a much higher peak. The values of this simulation appear to be in better agreement with the ones by Ujang et al. (2006) than when carried out with C_V=1. The results overall are inconclusive, although they do show a better agreement than with slug frequencies.





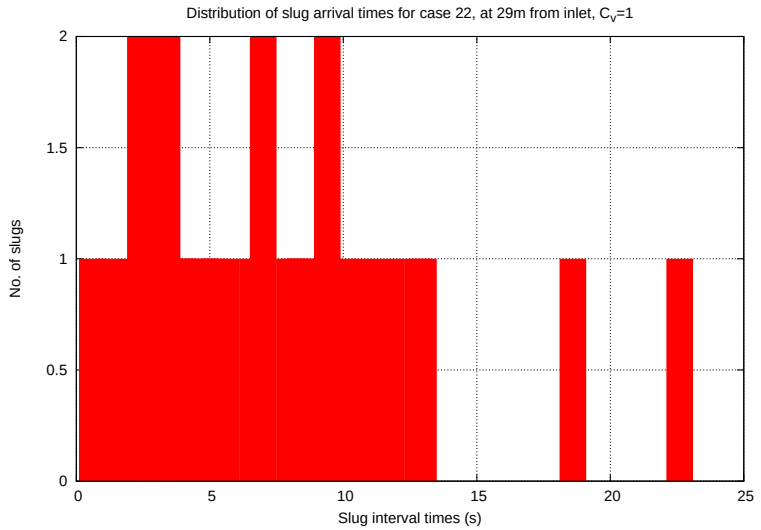
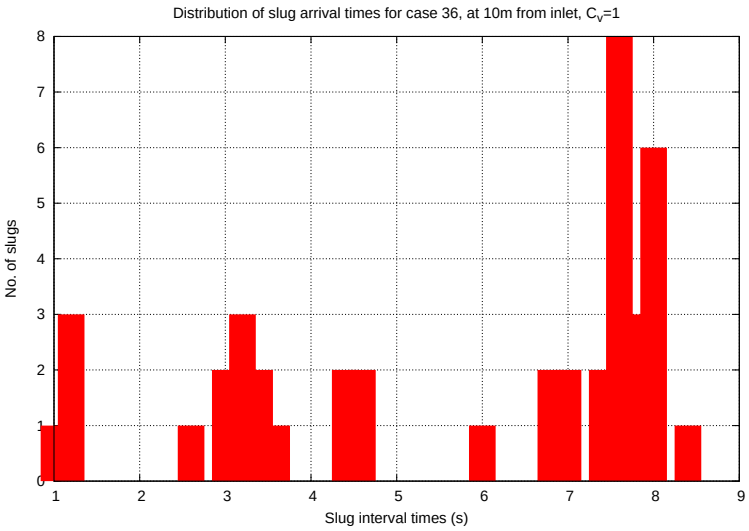
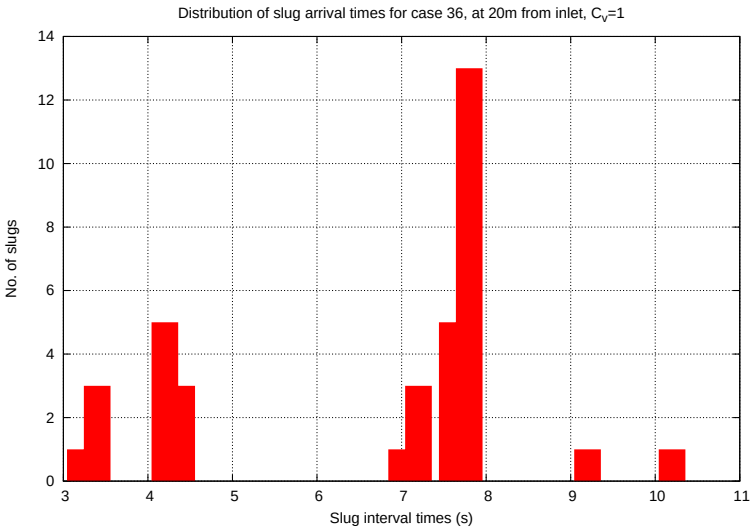


Figure 5.10: Slug interval times at Case 22, at 10m, 20m, and 29m from inlet, with C_V=1





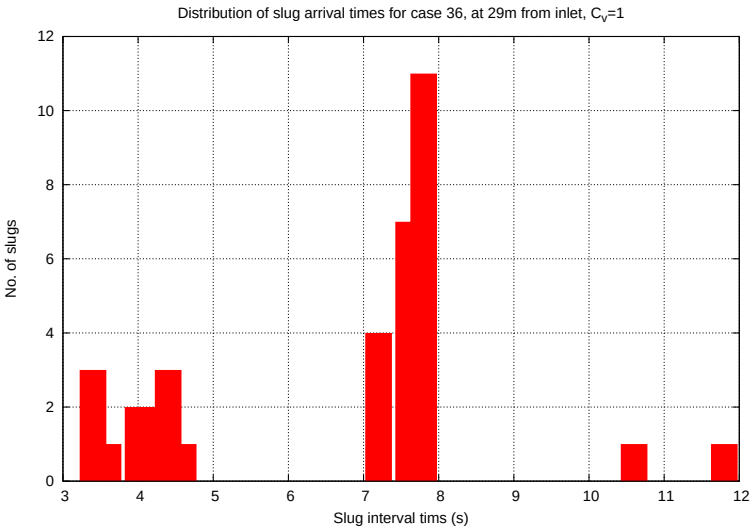
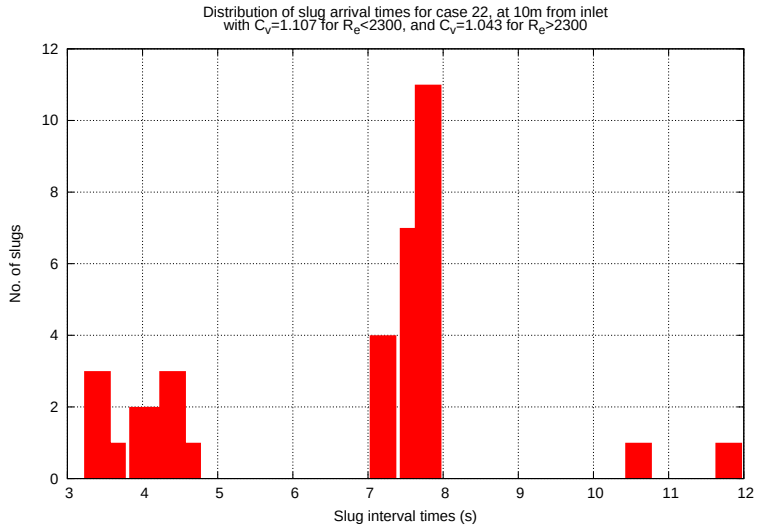
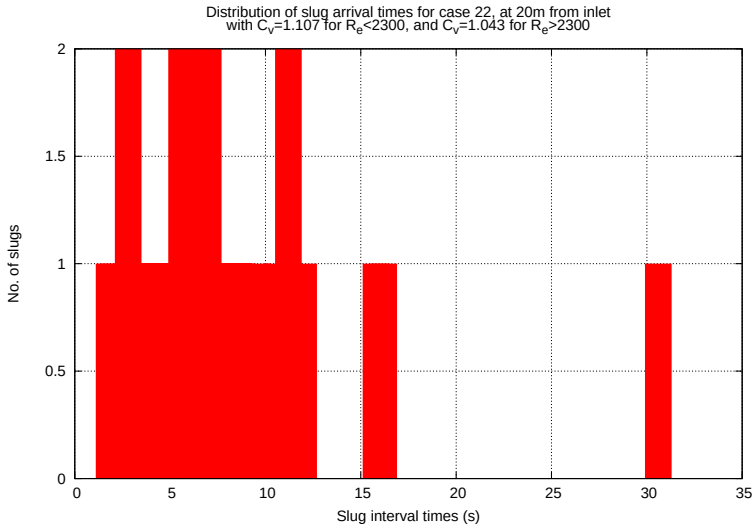


Figure 5.11: Slug interval times at Case 36, at 10m, 20m, and 29m from inlet, with C_V=1





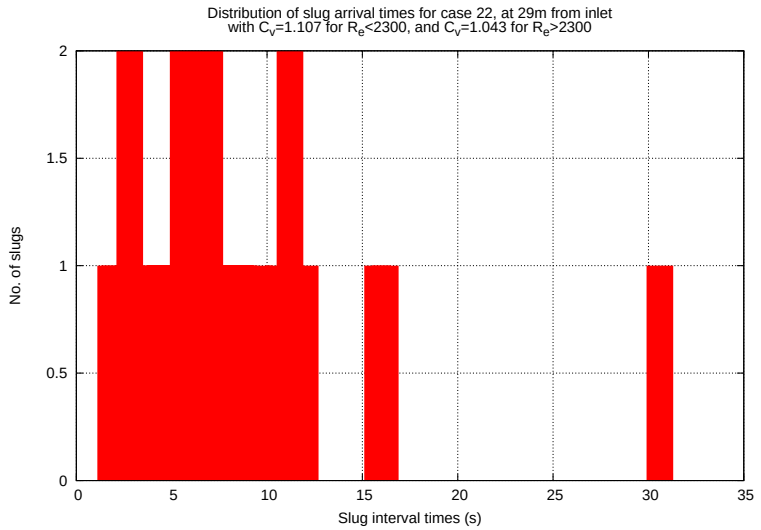


Figure 5.12: Slug interval times at Case 22, at 10m, 20m, and 29m from inlet, with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300

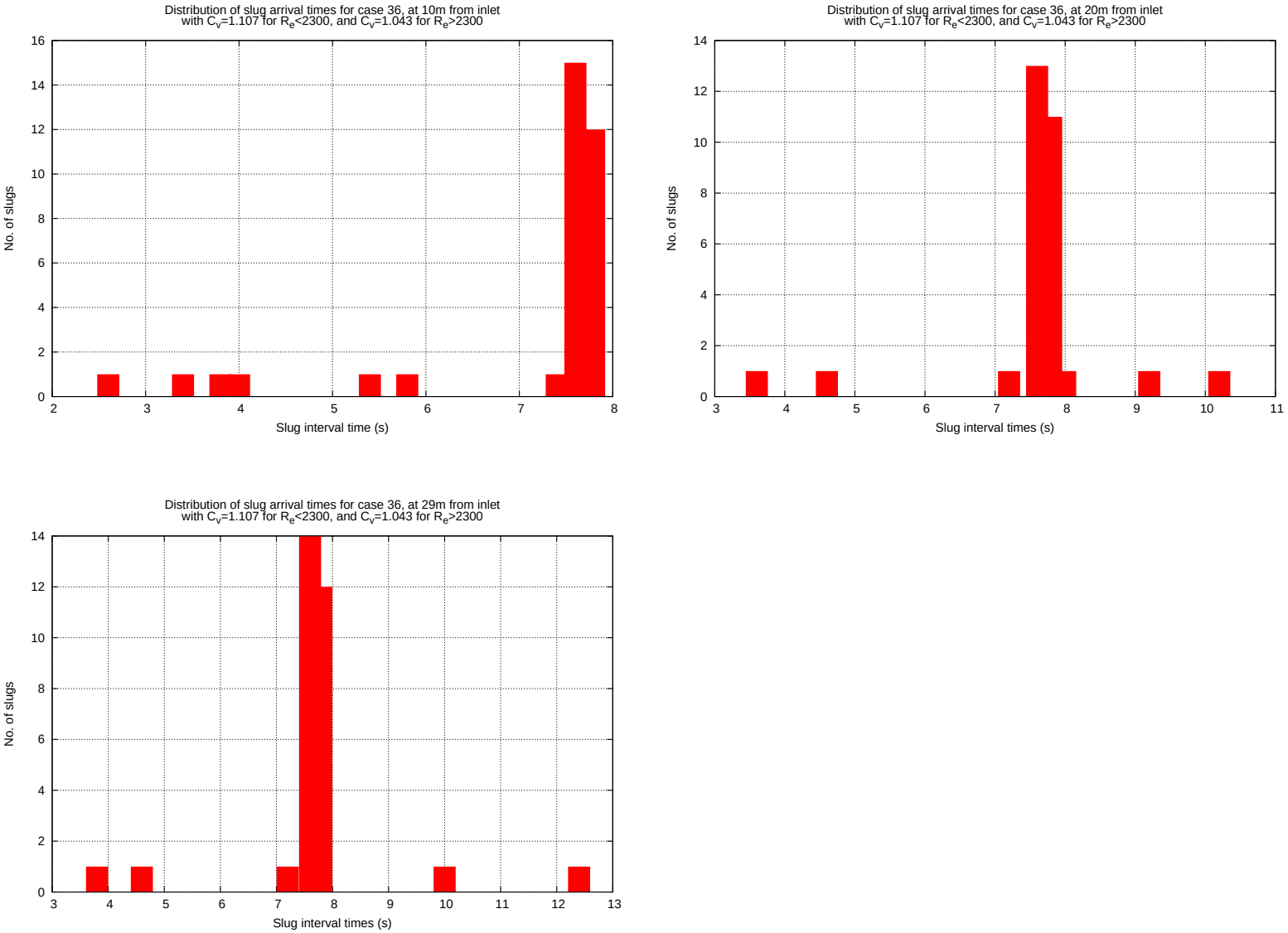


Figure 5.13: Slug interval times at Case 36, at 10m, 20m, and 29m from inlet, with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300

5.5 Analysis with flow variables

A more detailed analysis of correlations of velocity profile coefficients versus flow variables from the FLUENT simulations was carried out. Correlations against the following variables were carried out:

- Pressure
- Phase Height
- Pressure gradient
- Combination of Pressure and Phase Height

Correlations between each flow variable shown above and velocity profile coefficient of the corresponding phase were investigated using second order polynomial fitting. Although in theory it could be possible to find better fittings with higher order polynomials, it was decided to use only second order fitting because in the momentum equation the highest order is indeed second order and therefore it would be arbitrary to use a higher order. Nevertheless it has to be kept in mind that such correlations are derived from a statistical analysis and do not have a direct relation to the flow equations. The Bernoulli equation does give a quadratic relationship between pressure and speed for single phase flow, but the flow analysed here is multiphase flow and therefore more complex. Assuming y is the velocity profile coefficient, and x is the flow variable being investigated, then the polynomials are of the form

$$y = ax^2 + bx + c \tag{5.5}$$

The norm of the residual (N) from the fitting will give an indication of the appropriateness of the fitting, with the smaller norm giving normally a better fitting. Graphs of the various correlations are shown in Fig. 5.14.

The values from the fitting of correlation between pressure and C_V were found to be as follows:

- Liquid: a=1.0508E-06, b=-0.23347, c=12968, N=0.062549
- Gas: a=8.0379E-07, b=-0.17857, c=9919, N=0.1184

The values from the fitting of correlation between phase height and C_V were found to be as follows:

- Liquid: a=10.761, b=-0.4584, c=1.0293, N=0.09393
- Gas: a=-8.174, b=1.3655, c=1.0276, N=0.14064

Although pressure gradient is a quantity of importance in slug formation, the correlation between pressure gradient and C_V did not report any meaningful residual, therefore it was not pursued further.

The flow variable that appears to produce a correlation with velocity profile coefficient with the smallest norm of residual is pressure (Kalogerakos et al., 2012b). It is not immediately obvious why pressure should correlate with the velocity profile coefficient, but the results give a strong indication of this relation and any future studies should investigate a larger number of flow variables.

5.6 Simulation of Manolis cases with flow variable dependent C_V

Simulations of Manolis experimental cases were repeated using EMAPS and the velocity profile coefficient C_V expressed each time as function of pressure, height, and combination of pressure-height. The slug frequencies of these simulations are shown in Table 5.3, together with slug frequencies found during simulations using different C_V correlations. It appears that again the pressure-fitted C_V gives the best agreement between experimental and EMAPS slug frequencies (Kalogerakos et al., 2012b). This is to be expected as the only variable that showed a quadratic correlation was indeed the pressure. The slug frequency was predicted with an average discrepancy of 7.1% (and average standard deviation 2.9%).

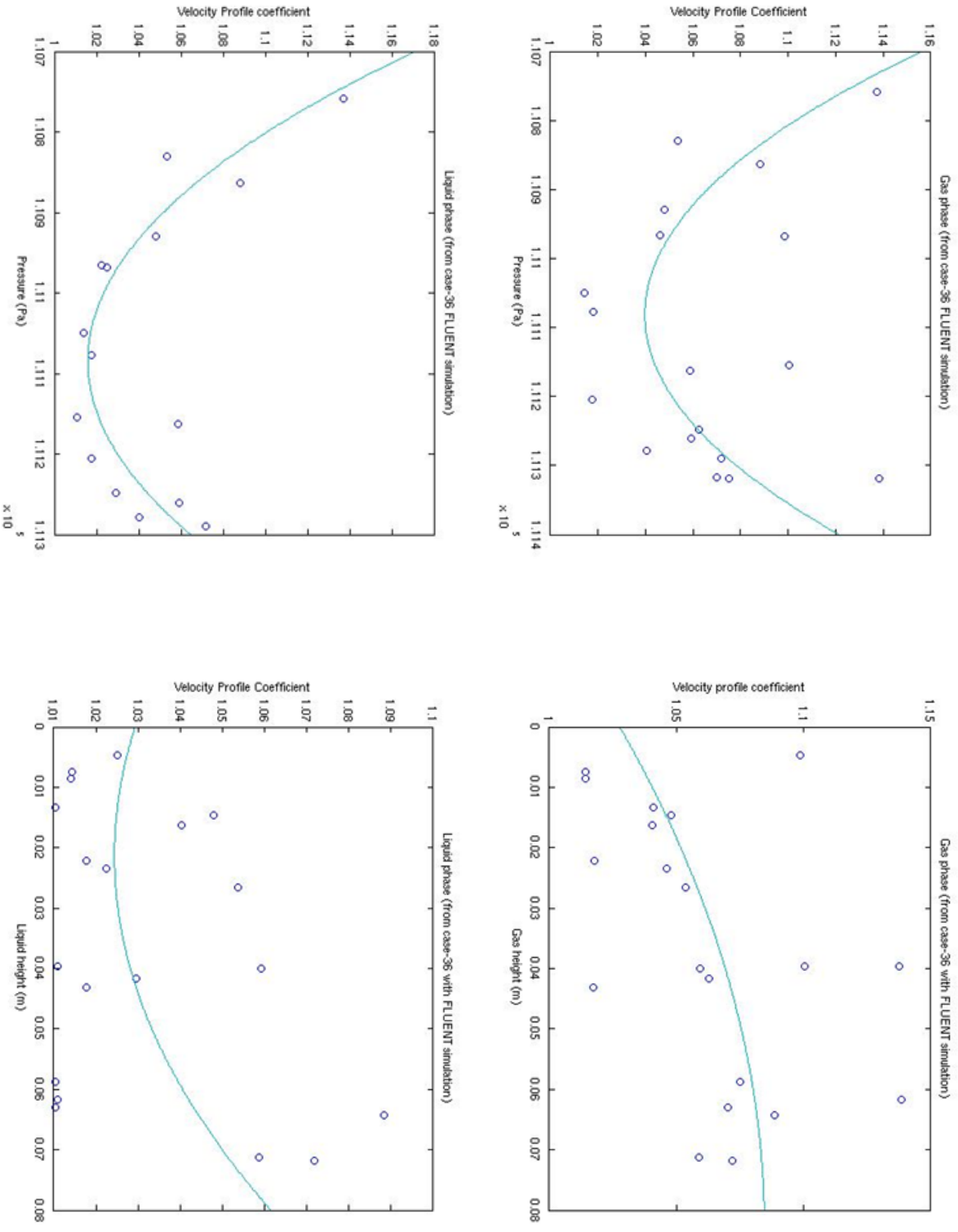


Figure 5.14: Fitting of Velocity profile Coefficient vs. Pressure and Phase Height

	Case 22 Slug Frequency	Case 36 Slug Frequency	Case 37 Slug Frequency	Case 38 Slug Frequency
Manolis thesis	0.133	0.244	0.194	0.217
C_V=1	0.184	0.234	0.129	0.158
C_V with Reynolds number	0.400	0.320	0.300	0.470
C_V with pressure fitting	0.120	0.220	0.205	0.200
C_V with height fitting	0.263	0.131	0.170	0.210
C_V with pressure-height fitting	0.277	0.223	0.213	0.210

Table 5.3: Table of frequencies (Hz) with modified C_V

5.7 X-Pad Simulation with modified C_V

Simulations were carried out on the X-Pad using the Reynolds number based C_V, in order to compare with the values obtained in section 4.7.1, which where carried out assuming that C_V=1. The perturbation which was observed already in the simulation of the R-Pad was seen in simulations on the X-Pad as well. It is barely visible at 6s in Fig. 5.15, but it increases in amplitude in Fig. 5.16 and even more so in Fig. 5.17. It is already merged with the main slugs in Fig. 5.18, as it can be seen in more detail in the magnification in Fig. 5.19. The frequency has increased compared with the one calculated with the simulation done using C_V=1, as it can be seen in Fig. 5.20.

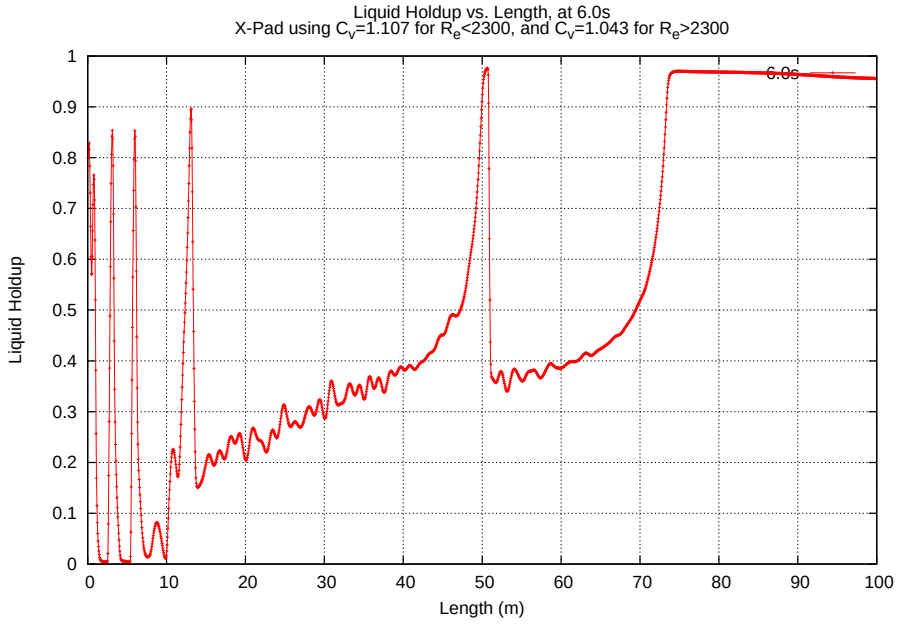


Figure 5.15: X-Pad: Liquid Holdup vs. Length at 6.0s

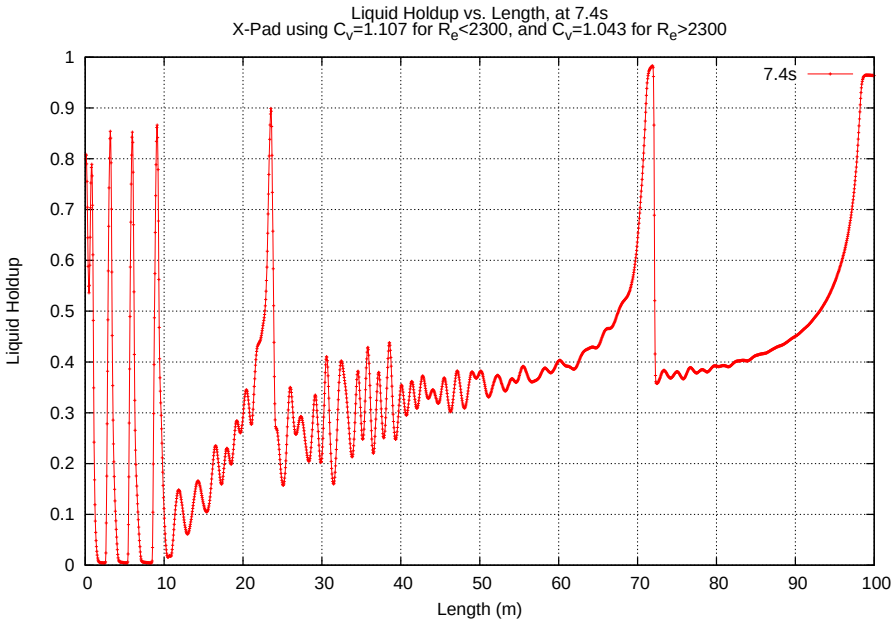


Figure 5.16: X-Pad: Liquid Holdup vs. Length at 7.4s

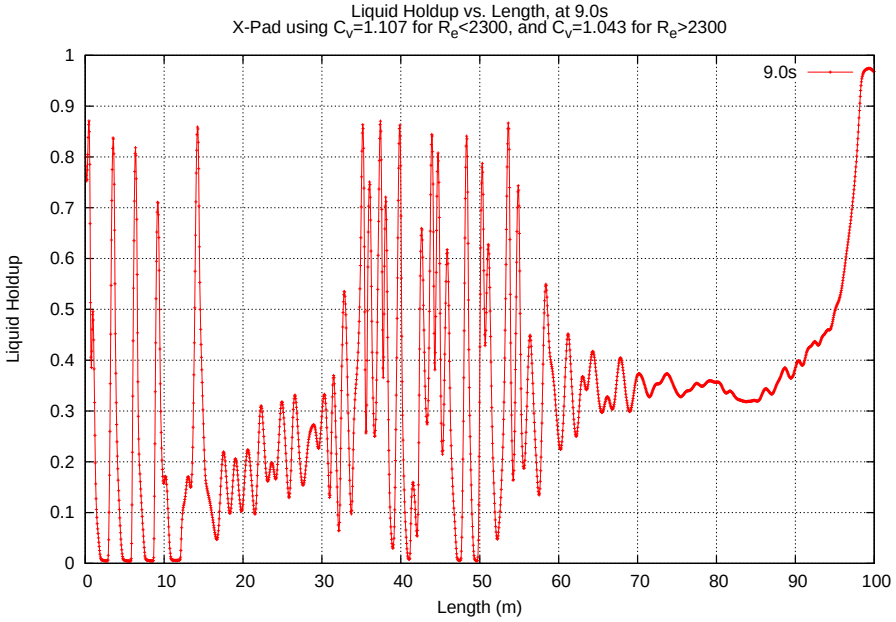


Figure 5.17: X-Pad: Liquid Holdup vs. Length at 9.0s

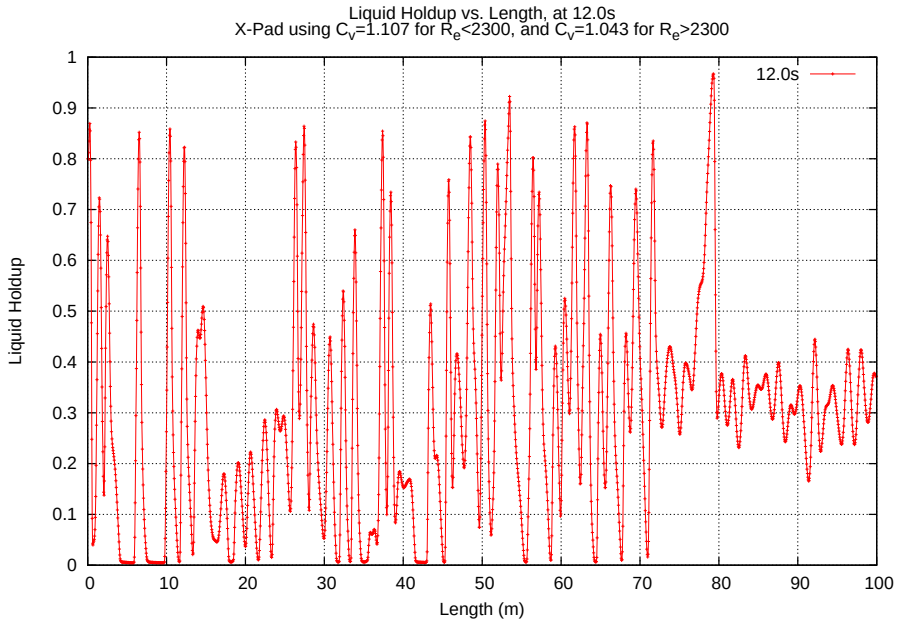


Figure 5.18: X-Pad: Liquid Holdup vs. Length at 12.0s

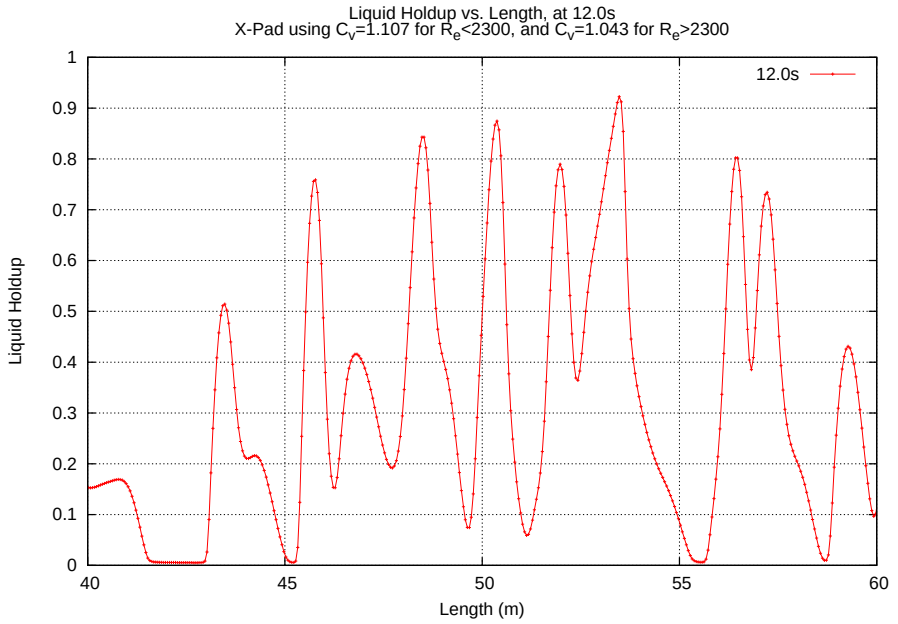


Figure 5.19: X-Pad: Liquid Holdup vs. Length at 12.0s, zoomed

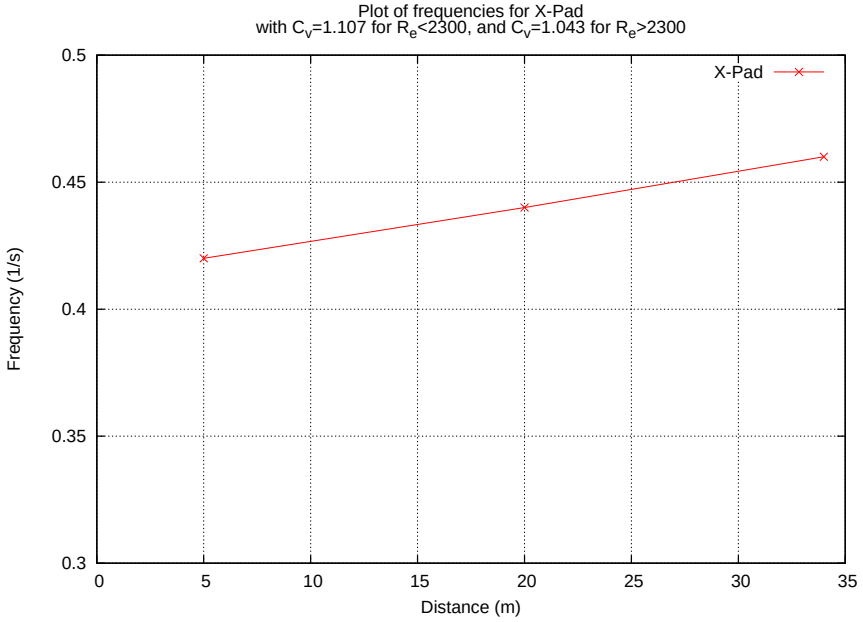


Figure 5.20: X-Pad: Frequency vs. Length

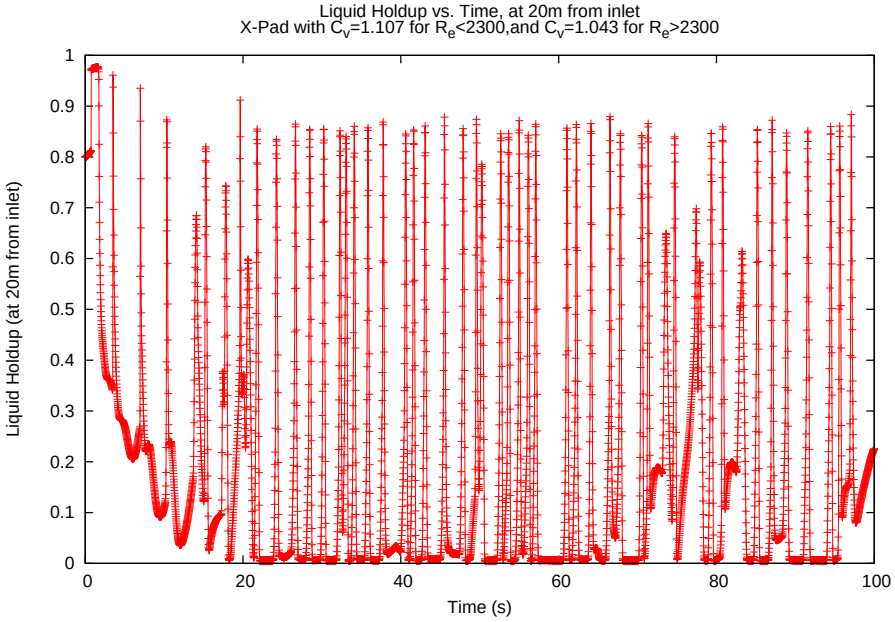


Figure 5.21: X-Pad: Liquid holdup vs. Time at 20m from inlet

Thus simulations of X-Pad using Reynolds based, modified C_V produced different results regarding slug frequency than when taken with C_V=1, due to the numerical disturbance appearing at 20m and 50m from the inlet.

When using pressure-fitted results, normalised with pipe diameter (i.e. pressure normalised as $P/\rho gd$), this perturbation appears to have disappeared, as it can be seen comparing Fig. 5.22 and Fig. 5.23. The same phenomenon was observed also on the R-Pad section, previously simulated with EMAPS assuming C_V=1. Thus the use of the pressure-fitted C_V improves the results of simulations of X-Pad and R-Pad obtained with EMAPS, by removing the numerically created perturbation. Specific values from BP data are not shown again due to confidentiality issues.

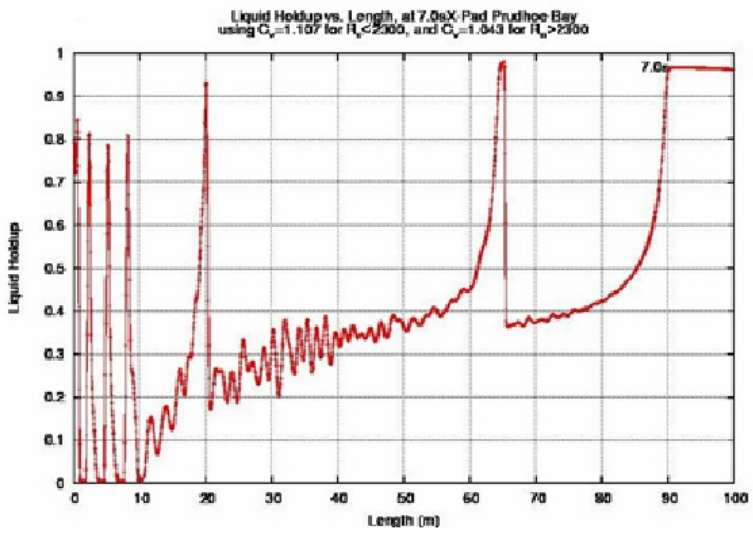


Figure 5.22: X-Pad simulation in EMAPS, using Reynolds number based C_V. Perturbation visible

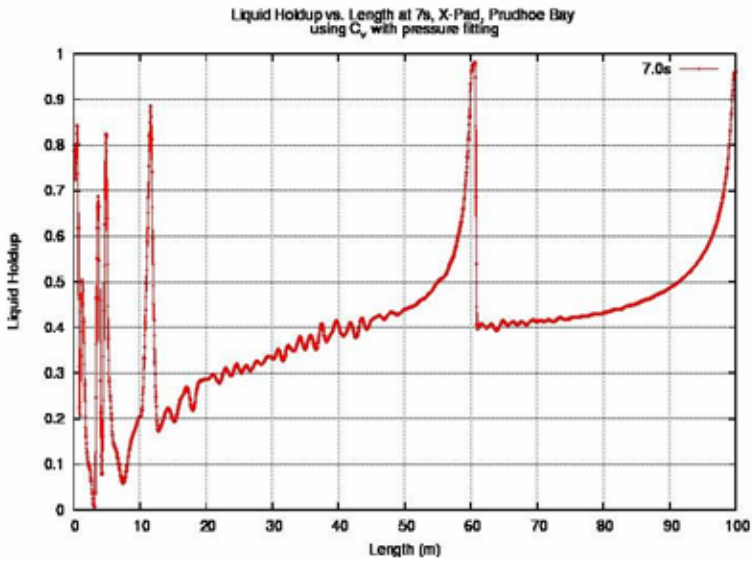


Figure 5.23: X-Pad simulation in EMAPS, using pressure-fitted C_V.

5.8 Conclusions

After analysing the mathematical models behind EMAPS, it was observed that when averaging the momentum equations over the cross-sectional area of the pipe, the term of velocity profile coefficient arises. This coefficient normally has a value different from 1 but in the original version of EMAPS it was assumed equal to 1. Therefore it was decided to enhance the momentum equations by introducing varying velocity profile coefficients. Simulations were carried out with C_V taken from literature (Schulkes, 1994) and applicable to relevant Reynolds number ranges. Moreover, a 3D CFD simulation was carried out for the same problem and various flow parameters were fitted against C_V in order to find a correlation. Comparisons were carried out with experimental results from WASP facility in Imperial College, London (Ujang et al., 2006). Using the fitted C_V, EMAPS simulations were repeated on the same case studies, and on pipe sections from BP's Prudhoe Bay. Fittings of C_V against pressure give better results for slug frequency and slug lengths (Kalogerakos et al., 2012b). Although it is not straightforward to generalise these correlations to other ranges of flow parameters, nevertheless the use of a modified velocity profile coefficient appears to be a necessary step when using 1D multiphase codes.

References

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012b), 'Adjustment of velocity profile coefficients for one-dimensional multiphase flow', *Journal of Computational Physics* . In preparation. (cited at page 133, 134, 143)

Manolis, I. G. (1995), High Pressure Gas-Liquid Slug Flow, PhD thesis, Department of Chemical Engineering and Chemical Technology, Imperial College of Science, Technology and Medicine, UK. (cited at page 125)

Schulkes, R. M. S. M. (1994), 'The evolution and bifurcation of a pendant drop', *Journal of Fluid Mechanics Digital Archive* **278**, 83–100. (cited at page xxiv, 116, 117, 143)

Ujang, P. M., Lawrence, C. J., Hale, C. P. and Hewitt, G. F. (2006), 'Slug initiation and evolution in two-phase horizontal flow', *International Journal of Multiphase Flow* **32**, 527–552. (cited at page 127, 143)

Vielliard, N. M. L. (2003), Development of a new model for the prediction of the stratified-to-slug flow transition, Technical report, Norsk Hydro ASA. (cited at page 116)

Chapter 6

Wave Growth and Perturbation Analysis

6.1 Wave growth simulation with EMAPS

For validation purposes, the wave growth problem was chosen as it is a problem with a known analytical solution, and moreover it had been previously simulated in Imperial College using their software TRIOMPH (Valluri and Spelt, 2006). It is an important benchmark both for EMAPS and also 2D CFD.

Using EMAPS 3.60, it was decided to carry out a simulation of the numerical wave growth of the incompressible Watson model, in order to investigate how an initial perturbation will evolve in the conditions set, and to compare both with the results obtained by Omgba-Essama (2004) and with the ones obtained in Imperial College with TRIOMPH. The problem consists of an air/water mixture velocity of 2.4m/s, with the gas and liquid superficial velocities being 2.0m/s and 0.4m/s respectively. The length of the pipe is set to 38m, to correspond with the length of the pipe used in Imperial College, with a diameter of 78mm. The step

was chosen, after a few trials, to be 36cm. The properties of the fluids are as shown below:

- Air
  - Density: 1.21 kg/m^3
  - Viscosity: 1.77E-05 kg/m.s
- Water
  - Density: 998 kg/m^3
  - Viscosity: 1.14E-05 kg/m.s

The friction factors used are the Blasius-Hand-Taitel [BHT] (Blasius (1911), Spedding and Hand (1997) and Taitel and Dukler (1976)), and an equilibrium liquid holdup was obtained for this case as α=0.758. To obtain the numerical growth rate, it is necessary to slightly perturb with a sine wave this initial equilibrium holdup and look at the evolution of the peak values as a function of time. This can be summarised by the following equation:

$$\begin{aligned} \alpha_{L0} &= \alpha_L^{eq}\left[1 + A_0 sin\left(\tfrac{\pi}{4}(x - x_s)\right)\right] \quad \text{if } x[x_s, x_f] \\ \alpha_{L0} &= \alpha_L^{eq} \qquad\qquad\qquad\qquad\quad \text{otherwise} \end{aligned} \tag{6.1}$$

where α_{L0} is the initial liquid holdup, α_L^{eq} is the liquid holdup at equilibrium, the amplitude of the perturbation A_0 is set to 1%, and the start and final locations of the perturbation are defined as $x_s = 15m$ and $x_f = x_s + 8m$. This gives a series of initial values which were fed manually into EMAPS by providing an appropriate input file *Problem.txt*. The number of cells used in the pipe was 2000 for all simulations. A plot of the initial holdup at t=0s is shown in Fig. 6.1.

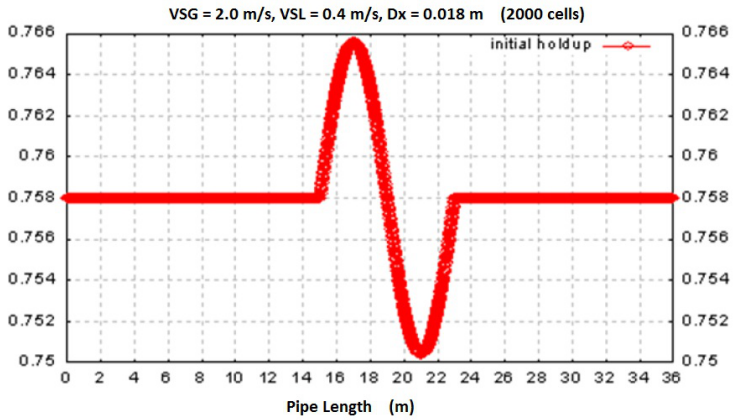


Figure 6.1: Pipe profile of the initial perturbed liquid holdup (time = 0.0s)

A plot using the following factors:

- Gas/Wall friction factor: Taitel and Dukler (1976) - turbulent flow
- Liquid/Wall friction factor: Spedding and Hand (1997)
- Gas/Liquid interface friction factor: Taitel and Dukler (1976)

produced the wave growth shown in Fig. 6.2.

In order to investigate the effect of using different factors, the following set was used:

- Gas/Wall friction factor: Taitel and Dukler (1976)
- Liquid/Wall friction factor: Taitel and Dukler (1976)
- Gas/Liquid interfacial friction factor: Taitel and Dukler (1976)

The plot resulting from using these factors is shown in Fig. 6.3.

A further simulation was carried out with the following set of factors:

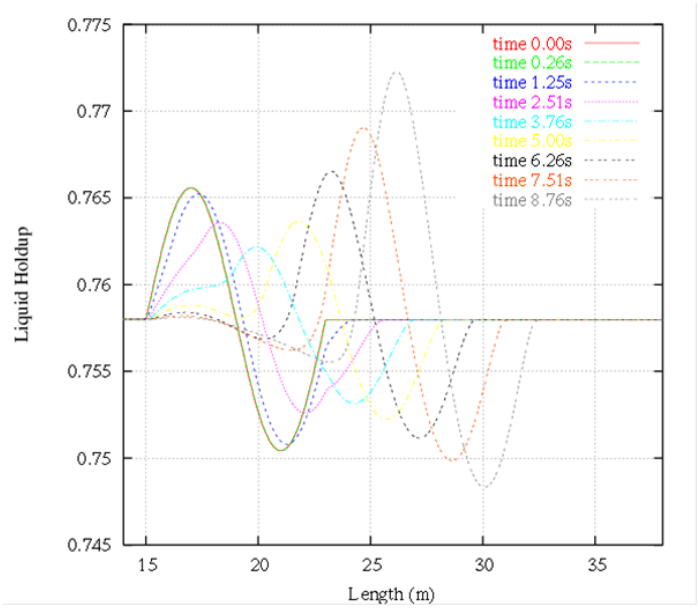


Figure 6.2: Liquid Holdup vs. Distance for BHT simulation

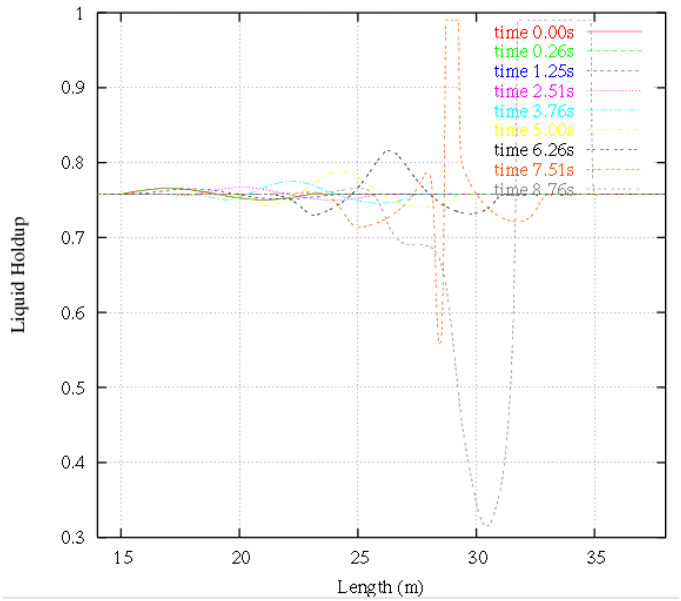


Figure 6.3: Liquid Holdup vs. Distance for TTT simulation

- Gas/Wall friction factor: Taitel and Dukler (1976)
- Liquid/Wall friction factor: Taitel and Dukler (1976)
- Gas/Liquid interfacial friction factor: Kowalski (1987)

This time the plot resulting is shown in Fig. 6.4.

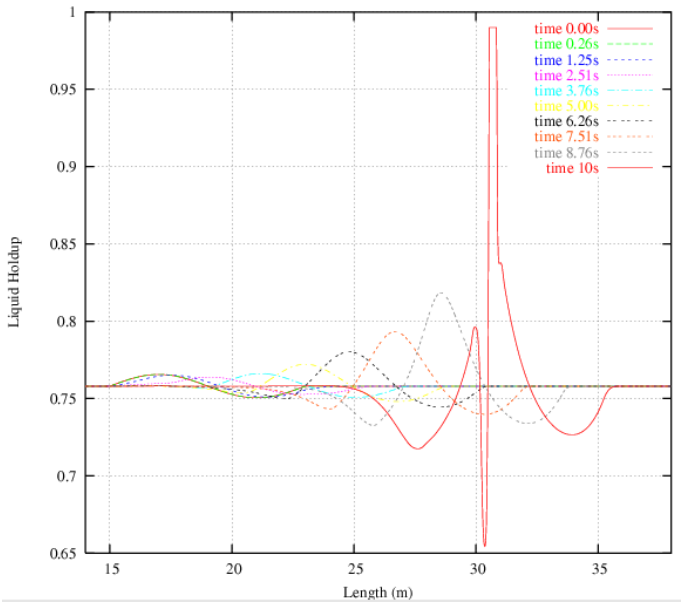


Figure 6.4: Liquid Holdup vs. Distance for TTK simulation

The best agreement with TRIOMPH results (including wave evolution) was obtained using BHT friction factors, and therefore this model will be used for more detailed calculation in the next section.

6.2 Comparison with TRIOMPH simulations

Because of consistency, it is important to investigate any discrepancies between results of codes using the same models and input data. Regarding the wave

growth simulation in section 6.1, at the beginning of the simulation, the initial wave decreases in amplitude, while later it grows again. This behaviour is common to both plots. But there is a distinct divergence of the plot of maximum holdup vs. time, after 15s. Maximum holdup is calculated as the maximum measured holdup for a specific time over the whole length of the pipe. In Fig. 6.5, on the left is the result with 1500 mesh points, while on the right after increasing the number of mesh points used in EMAPS 3.60 to 2500, there is a distinct improvement in agreement between the results of EMAPS and TRIOMPH (Valluri and Spelt, 2006). A further increase to 3000 and then 3500 in total mesh points did not produce a significant change in the graph, therefore it was deduced that mesh independence was reached at 2500 mesh points for EMAPS 3.60.

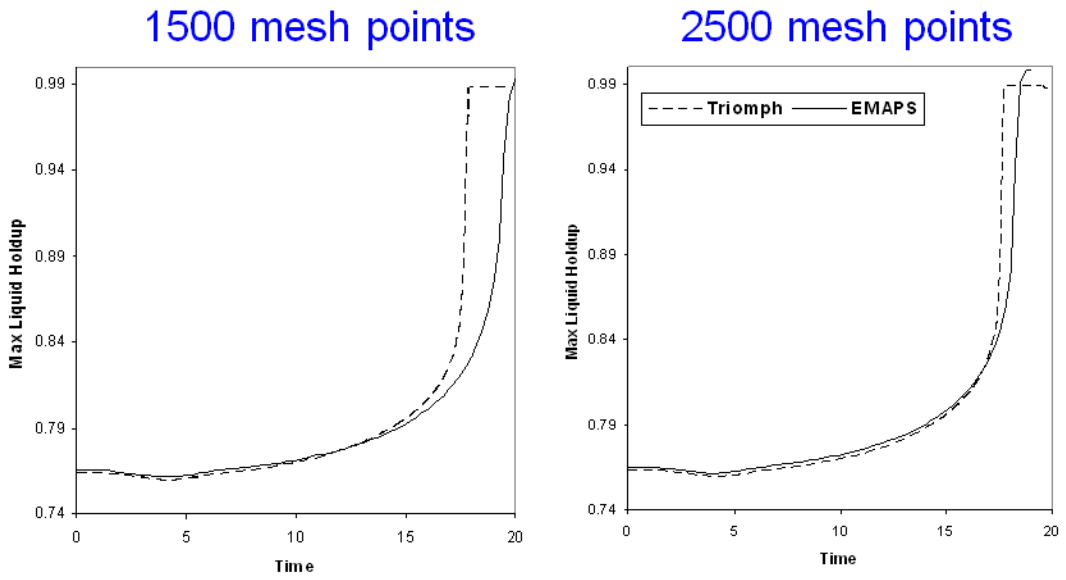


Figure 6.5: Comparison between TRIOMPH and EMAPS 3.60

New simulations were carried out using the new version EMAPS 3.70, which uses Reynolds number calculated using the superficial velocity. This time it was

necessary to use 3000 cells in order to obtain mesh independence. An almost equivalent result to the one obtained with the TRIOMPH code was obtained using 3000 cells, as shown in Fig. 6.6. A more detailed plot of the standalone EMAPS

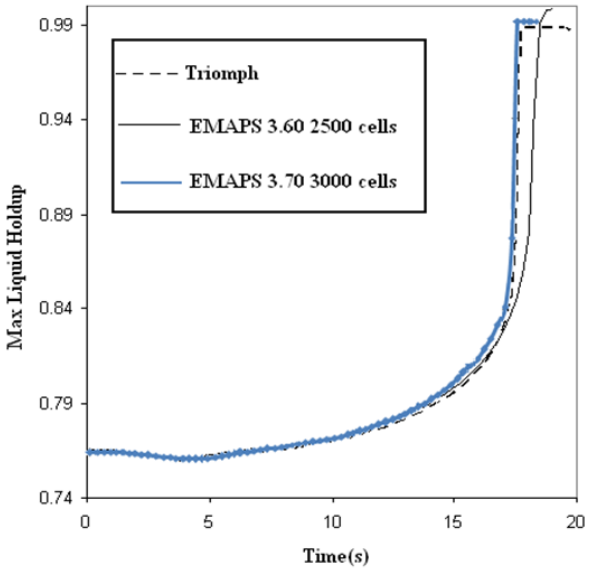


Figure 6.6: Comparison between Triomph and EMAPS 3.70 with 3000 cells

3.70 result is shown in Fig. 6.7. Thus the change in the calculation of Reynolds number in EMAPS has shown that in the case of incompressible flow and with the same input, the two codes EMAPS 3.70 and TRIOMPH do give equivalent results. This was very important in order to show that there are not major differences between the way the solvers are implemented (even though it does not necessarily follow logically that they are both correct, although the chances that they are so are higher!). Custom-made scripts involved in post-processing can be seen in appendix D.

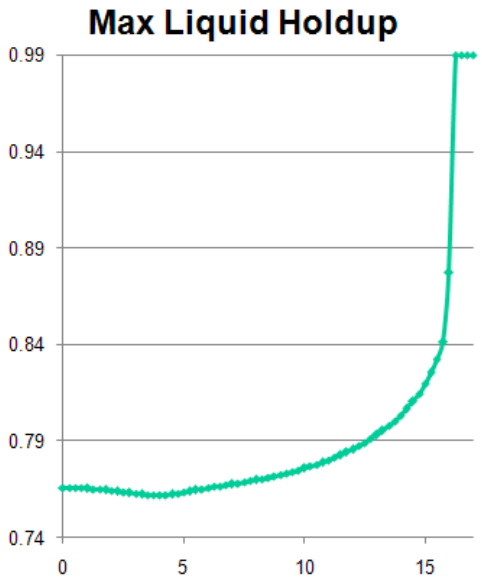


Figure 6.7: EMAPS 3.70 with 3000 cells. Reynolds number calculated using superficial velocity.

6.3 Adaptivity

A recent research carried out by a Cranfield student, Jia (2007), has shown the advantages of the implementation of adaptivity in the EMAPS code. By increasing the number of cells only in the proximity of "interesting" regions, i.e. regions where perturbations are localised at a certain time, the total computational time is noticeably reduced. In the diagram 6.8 the total times in running a Watson model wave growth case using just uniform grid (no adaptivity) and times using adaptive grid are compared, with the relative speed-up shown on the right.

Moreover, in Fig. 6.9 the time evolution of the liquid holdup with adaptive grid is shown as an example of the application of adaptivity. The level of refinement changes according to the precision required, as shown on the right vertical scale of each plot. For example, in the plot at t=0.26s, the refinement level is 4 (the

Levels of refinement	Timings (s)		Speed-up
	Uniform	Adaptive	
1	18.861 s (2000 cells)	-	-
2	70.250 s (4000 cells)	42.544 s	1.65
3	283.224 s (8000 cells)	123.254 s	2.298
4	18min 16s (16000 cells)	281.087 s	3.899

Figure 6.8: Comparison between Uniform grid and Adaptive grid (Jia, 2007)

highest) between 15 and 24m, as that is where the maximum change in liquid holdup occurs. A slight smaller refinement level of 3 is between 10 and 15m and between 24 and 28m, as that is the area of transition between steady flow and the change in liquid holdup. In all other areas, a grid level of 2 is sufficient. If we apply adaptivity to the problem investigated in section 6.2, and we repeat the simulation of the Watson wave growth with BHT friction factors, the result is practically overlapping between the uniform and adaptive. It is possible that due to its simplicity no new features were introduced by the use of adaptivity, and increasing the cell number did not bring a particular advantage in the final result, apart from a decrease in computation time. The two plots are shown in Fig. 6.10.

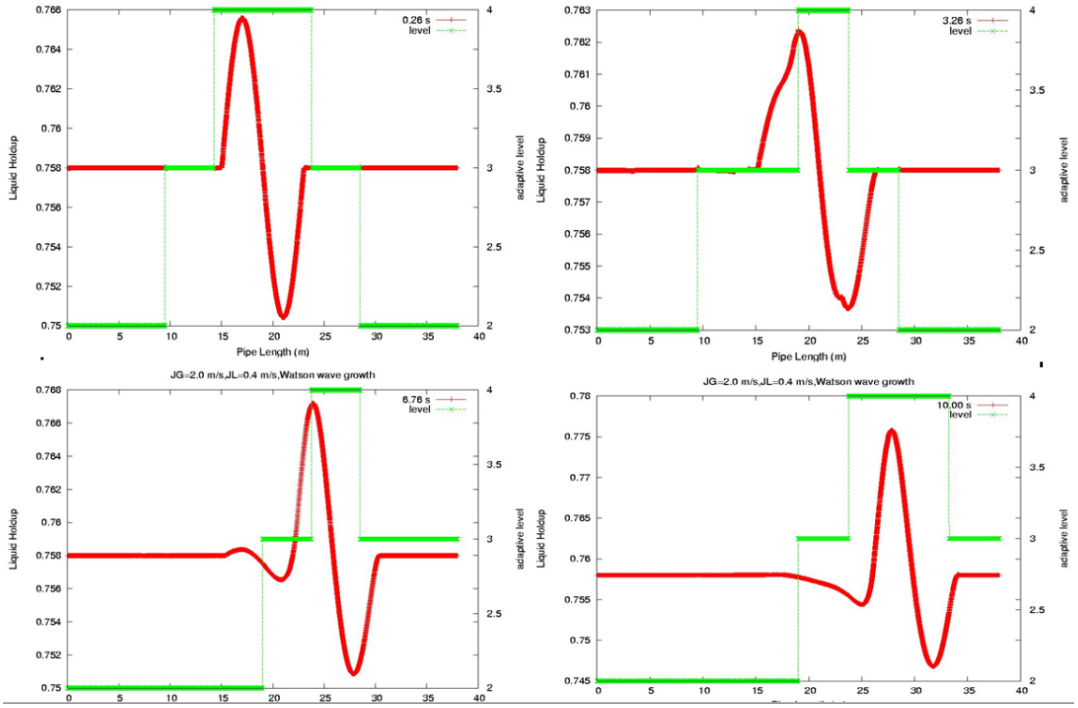


Figure 6.9: Time evolution of liquid holdup with adaptive grid (Jia, 2007)

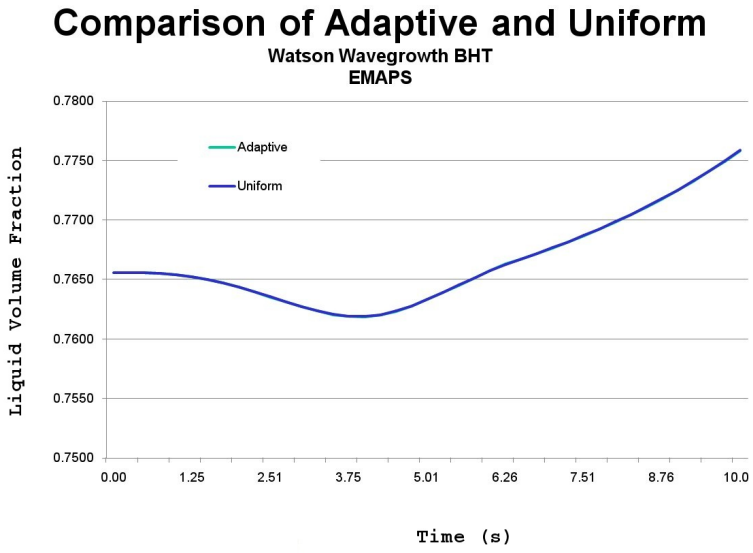


Figure 6.10: Uniform grid and Adaptive grid for BHT wavegrowth

6.4 Simulation of Wave growth with FLUENT 2D

A 2D simulation of the previously investigated wave growth problem for validation purposes. A sine wave was introduced at 15m, with length of 8m, and constant superficial velocities were imposed for liquid 0.4 m/s and gas 2.0m/s (water and air); these conditions were implemented by writing custom user code (UDFs) in C++ for FLUENT (see appendix B.1). The initial condition is shown in Fig. 6.11. Simulations were initially carried out using a $k - \epsilon$ turbulence model. Slugs appeared to form, but there is an issue with the fact that a perturbation appears to stem from the inlet (in the FLUENT simulation) and propagate along the channel. Snapshots of volume fraction vs. pipe length have been taken at times t=0.495s (Fig. 6.15), 0.6s (Fig. 6.16) and 1.165s (Fig. 6.17). Moreover, on a time-scale of less than 1 second, many little perturbations appear on the sine-wave, and these eventually increase and affect the whole perturbation. These can be better seen in the snapshots of liquid height vs. pipe length at times t=0.495s (Fig. 6.12), 0.6s (Fig. 6.13) and 1.165s (Fig. 6.14). In the EMAPS simulation on the other hand, the sine-wave is propagating but keeps its general sine shape well beyond 10 seconds.

Simulations were repeated using the Reynolds Stress turbulence model, as implemented as standard in FLUENT. There appeared to be a very quick transition from original flow to churn flow, with little evidence of slug flow, in a timescale of less than 1 second, as shown in Fig. 6.19 and also Fig. 6.18. It appears that using the Reynolds Stress model speeds up the whole process as compared to the results using $k - \epsilon$, and most phenomena observed are unstable, including irregular gas slugs/bubbles with elongated shapes.

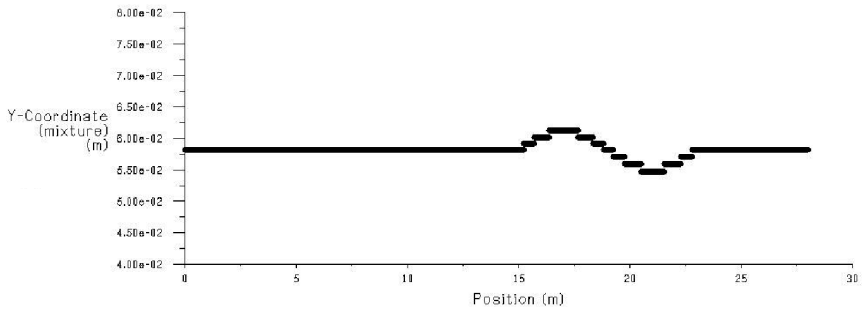


Figure 6.11: Liquid height vs. Distance, showing sine-wave before start of simulation, for 38m pipe.

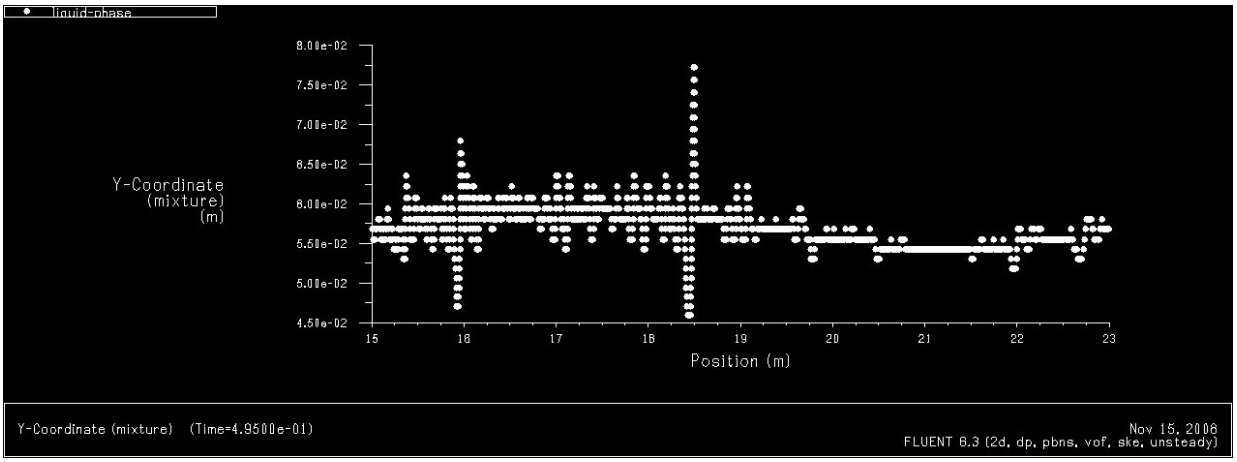


Figure 6.12: Liquid height vs. Distance at 0.495s, for section of 38m pipe ($k - \epsilon$)

In order to gain a further insight into the issues of the perturbation at the inlet, a simulation on a smaller pipe of 6m, and with a larger amplitude of initial sine-wave perturbation (as shown in Fig. 6.20) was carried out. The same problem was observed here as well.

Thus initial trials of simulations of wave growth in FLUENT 2D lead to the conclusion that it is not possible to observe the wave growth that is visible in EMAPS simulations of the same problem, due to the fact that a perturbation appears to stem from the inlet (in the FLUENT simulation) and propagate along

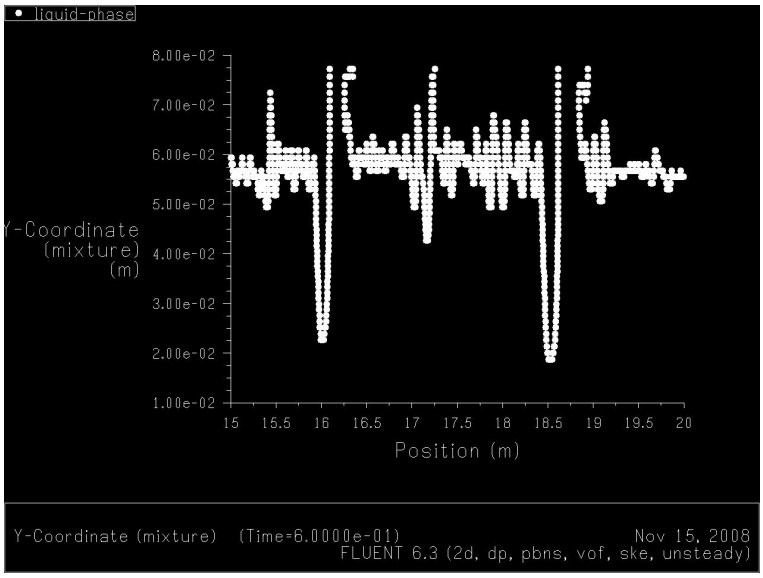


Figure 6.13: Liquid height vs. Distance at 0.6s, for section of 38m pipe ($k - \epsilon$)

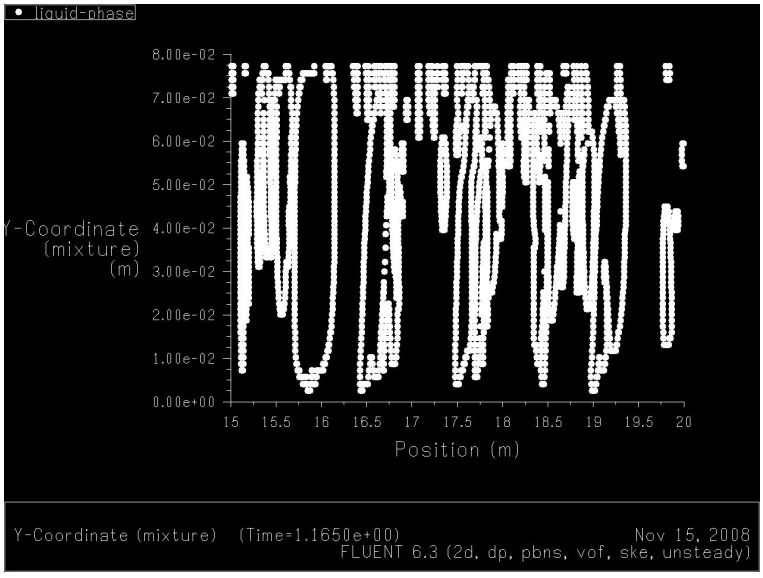


Figure 6.14: Liquid height vs. Distance at 1.165s, for section of 38m pipe ($k - \epsilon$)

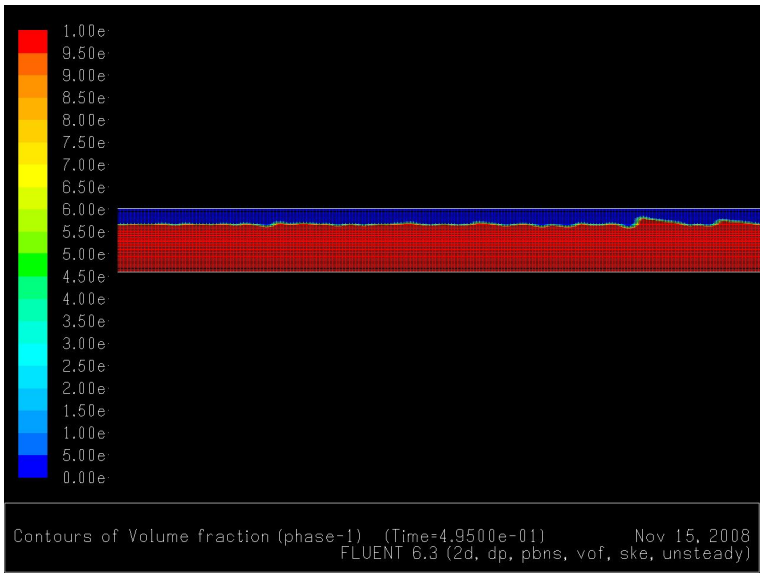


Figure 6.15: Contours of Volume fraction at 0.495s, for a section of 38m pipe ($k - \epsilon$)

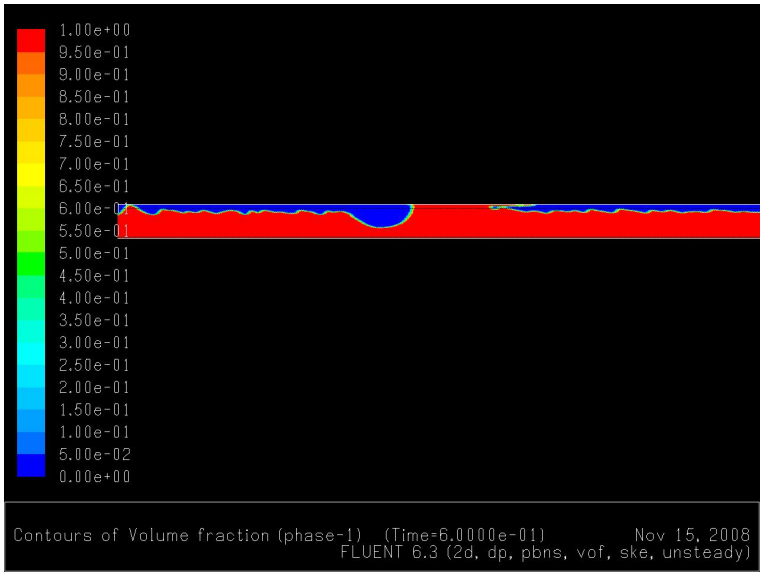


Figure 6.16: Contours of Volume fraction at 0.6s, for a section of 38m pipe ($k - \epsilon$)

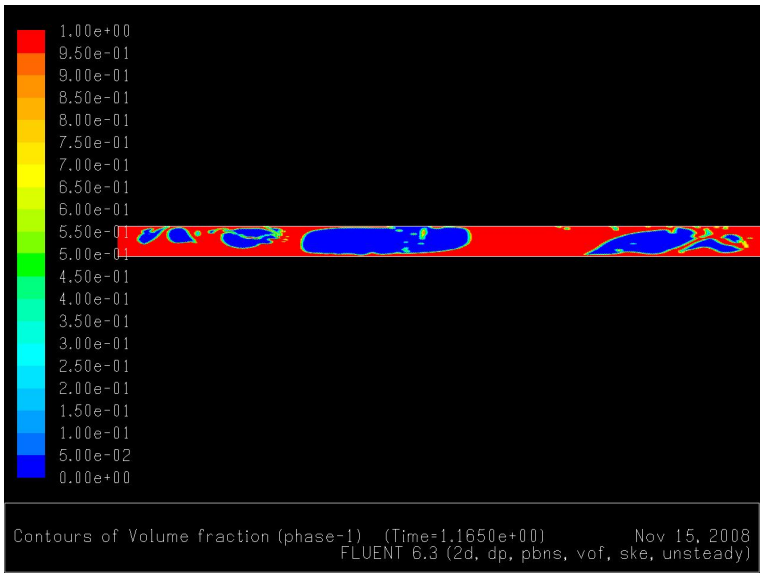


Figure 6.17: Contours of Volume fraction at 1.165s, for a section of 38m pipe ($k-\epsilon$)

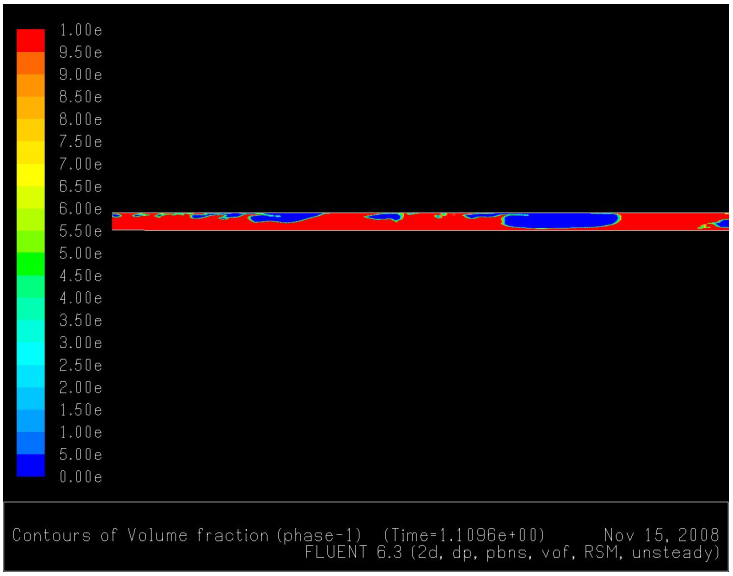


Figure 6.18: Contours of Volume fraction at 1.11s, for a section of 38m pipe (Reynolds stress)

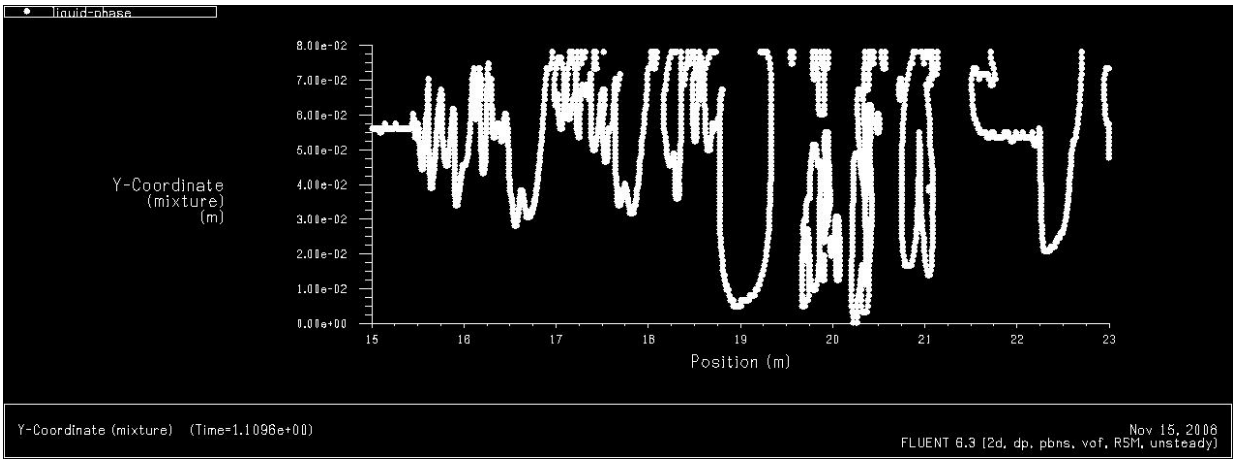


Figure 6.19: Liquid height vs. Distance at 1.11s, for section of 38m pipe (Reynolds stress)

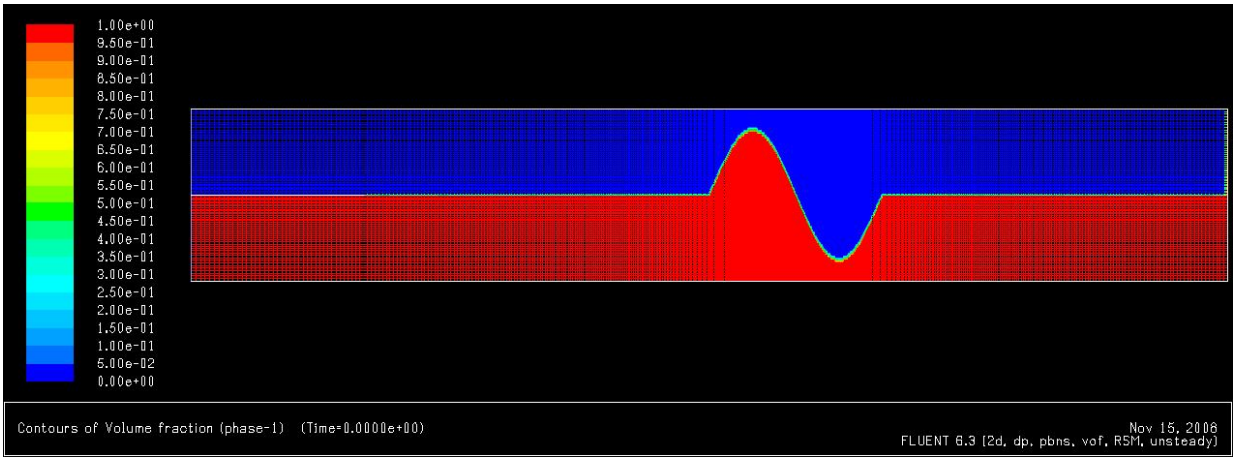


Figure 6.20: Contours of Volume fraction, showing sine-wave before start of simulation, for 6m pipe

the channel, quickly neutralising the wave. In the EMAPS simulation on the other hand, the sine-wave is propagating but keeps its general sine shape well beyond 10 seconds.

6.4.1 Wave growth: Incompressible Flow

It was thus decided to attempt to counteract the formation of the perturbation at the inlet by first carrying out a steady state simulation in FLUENT, and then applying the sine-wave to the steady state solution and starting the transient simulation from that point. Gas was set to be incompressible in the FLUENT simulation, inline with EMAPS settings. Custom-made scripts involved in processing of the results can be seen in appendix D. The steady state simulation converged, and then the transient simulation was started from the steady state simulation result. Again both $k-\epsilon$ and Reynolds stress turbulence models were used, and this time the best results were obtained using the Reynolds stress model. This is perhaps expected since the secondary gradients appear exactly in the Reynolds stress model, and the stresses are not assumed to be isotropic unlike in the $k-\epsilon$ mode. The Reynolds stress model is able to predict swirl and rapid changes in the strain rate, and when simulating wave growth in 2D the evolution is much more complex compared to 1D and may contain such features. Graphs of the wave growth at different times are shown in Fig. 6.23 and Fig. 6.24 (contours of volume fraction). Using Matlab, it was possible to estimate the rate of growth. The rate of growth in EMAPS simulation of the wave growth problem was estimated to be 0.31 with residual 0.013, while in the FLUENT 2D simulation it was estimated to be 0.34 with residual 0.089. These values were calculated by estimating the slope of log(max liquid holdup) vs log(time). As shown in Fig. 6.22, the time after which the wave starts to grow is around 1s and is vastly different compared to

EMAPS (around 11s, as shown in Fig. 6.21), but the wave growth rate is similar as it only concerns the rate of growth rather than its position.

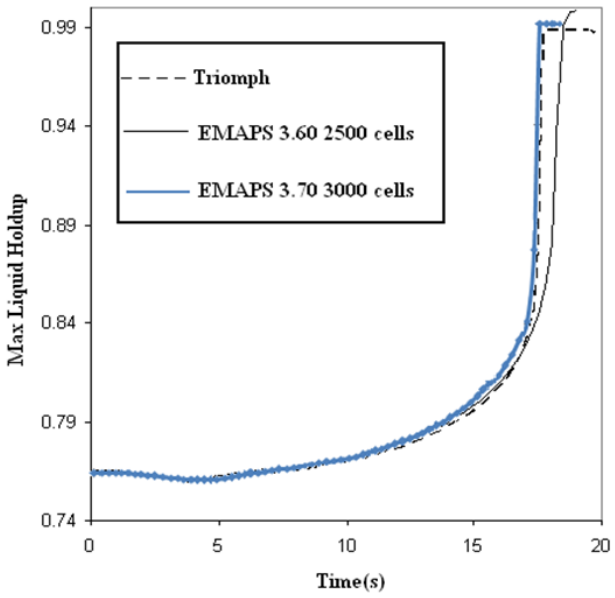


Figure 6.21: Wave growth as calculated by EMAPS, compared with TRIOMPH results EMAPS 3.70 uses Reynolds number defined using superficial velocities.

It was observed that mesh refinement is essential: with coarse grid, wave growth rate was 0.137 with 0.064 residual, therefore a much worse result. Mesh independence was established (results shown in section 7.3). The details of the meshes used are shown in Table 6.1:

	Max face area (m^2)	Cells	Faces	Nodes
Coarse Mesh	3.608137e-02	186,760	377,626	190,867
Refined Mesh	2.070366e-02	336,076	679,504	343,429

Table 6.1: Comparison of meshes used for wave growth in FLUENT 2D

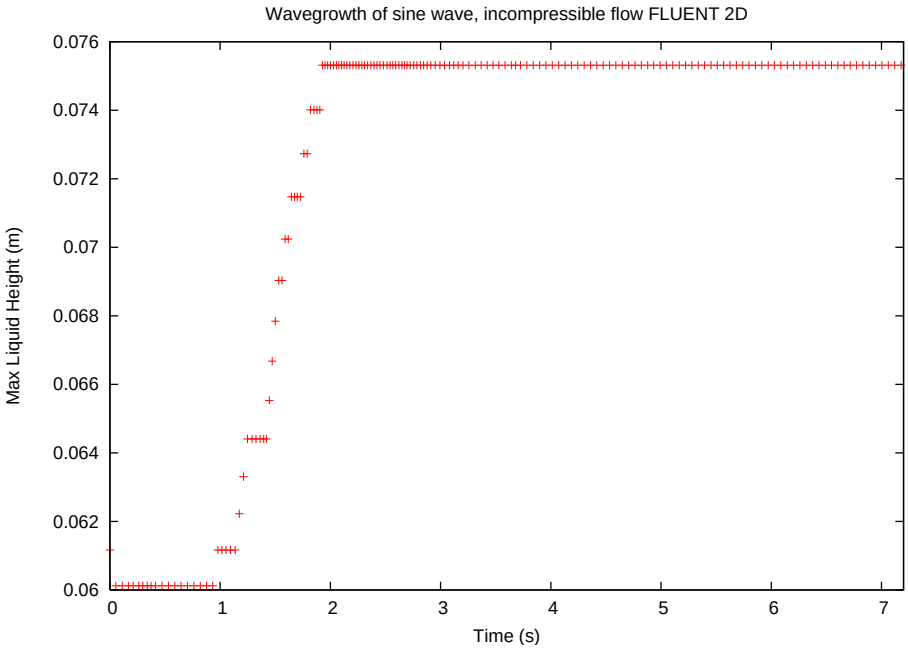


Figure 6.22: Wave growth as simulated using incompressible FLUENT 2D

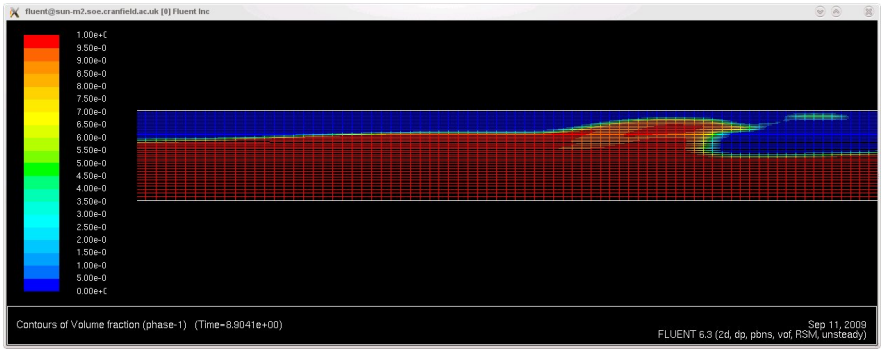


Figure 6.23: Contour of liquid volume fraction using Fluent 2D simulation of wavegrowth after 8.9s

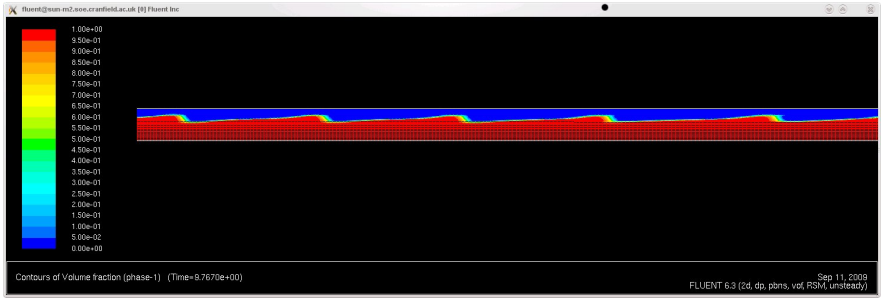


Figure 6.24: Contour of liquid volume fraction using Fluent 2D simulation of wavegrowth after 9.8s

6.4.2 Wave growth: Compressible Flow

As TRIOMPH from Imperial College, London have repeated the wave growth problem using compressible flow, it was decided to carry out a simulation also using compressible flows in FLUENT. The settings are shown in appendix B.2. The graph of maximum liquid heights vs. time is shown in Fig. 6.25 and it starts occurring at around 3s, thus marginally closer to EMAPS as compared with the incompressible flow results. Using Matlab, wave growth rate was found to be 0.32 with residual 0.062, thus again a better result. Therefore it appears that using compressible flow in FLUENT 2D for wave growth analysis does indeed give results that are very close to the ones obtained in 1D code, even though the model is different.

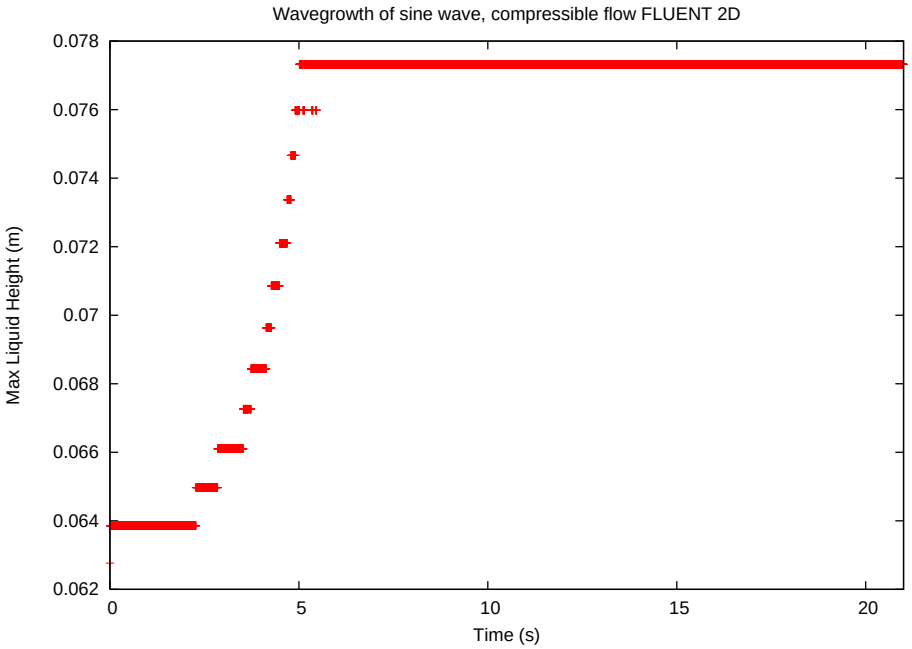


Figure 6.25: Wave growth as simulated using compressible FLUENT 2D

6.5 VOF model: Perturbation Analysis

In order to carry out a validation on the number obtained in the previous section for the rate of growth, a perturbation analysis of the VOF model was completed. A perturbation was introduced in the continuity and momentum equations, and the dispersion frequency was calculated. Initially any second order terms are neglected, and therefore as shown later surface tension is neglected in this scenario.

- Continuity eqn. : $\partial_t \alpha_L + \partial_x(\alpha_L u) = 0$
- Momentum eqn. : $\partial_t(\rho\vec{u}) + \vec{\nabla}(\rho\vec{u}\cdot\vec{u} + p) = \partial_x(\mu\partial_x\vec{u}) + \partial_y(\mu\partial_y\vec{u}) + \rho\vec{g} + \vec{F}$

We will assume that all time-dependent functions can be expressed as $\psi = \psi_0 + \bar{\psi}e^{j(\omega t - kx)}$. Moreover (FLUENT, 2006) the source term F can be expressed as:

$$\vec{F} = \sigma_{ab}\frac{\rho\kappa_a\vec{\nabla}\alpha_a}{\frac{1}{2}(\rho_a + \rho_b)} \tag{6.2}$$

where the indices a and b are the two phases, σ_{ab} is the surface tension coefficient, and κ_a is the curvature at the surface where the surface tension is calculated. Therefore we have the following:

- Volume fraction $\alpha = \alpha_0 + \bar{\alpha}e^{j(\omega t - kx)}$
- Velocity $u = u_0 + \bar{u}e^{j(\omega t - kx)}$
- Viscosity $\mu = \mu_0 + \bar{\mu}e^{j(\omega t - kx)}$
- Density (mixture) $\rho = \rho_0 + \bar{\rho}e^{j(\omega t - kx)}$

It is also assumed that we can express the pressure as $p = (\gamma - 1)c_V T\rho \equiv \Gamma\rho$, but also at equilibrium $\vec{\nabla}p_0 = \rho_0\vec{g}$. This equality has to be used later when working

on the pressure term. T is the temperature, γ is the ratio of specific heats and c_V is the specific heat coefficient at constant volume. Also $\Delta\rho \equiv \rho_a - \rho_b$. Our final aim is to find a good approximation for the dispersion frequency ω and compare with the results obtained using the 2D wave growth in FLUENT.

Normally $\bar{\psi}$ should be function of y, assuming the pipe is in the x direction. Initially it is assumed that $\bar{\psi}$ is constant, in order to simplify the calculations. Moreover we also neglect initially any second order terms or higher in $\bar{\psi}_1\bar{\psi}_2$, although these will have to be checked later in order to confirm that indeed these terms have little impact on the final value.

From the continuity equation (with the substitution $\phi = \omega t - kx$):

$$\begin{aligned} & j\omega\bar{\alpha}e^{j\phi} + \partial_x(\alpha_0 u_0 + \alpha_0\bar{u}e^{j\phi} + u_0\bar{\alpha}e^{j\phi} + \bar{u}\bar{u}e^{2j\phi}) = 0 \\ & \Rightarrow j\omega\bar{\alpha}e^{j\phi} - kj\alpha_0\bar{u}e^{j\phi} - kju_0\bar{\alpha}e^{j\phi} = 0 \\ & \Rightarrow \omega\bar{\alpha} - k\alpha_0\bar{u} - ku_0\bar{\alpha} = 0 \\ & \Rightarrow \bar{\alpha} = \frac{k\alpha_0\bar{u}}{\omega - ku_0} \end{aligned} \tag{6.3}$$

From the momentum equation, the following can be derived:

$$\partial_t(\rho u_0 + u_0\bar{\rho}e^{j\phi} + \bar{u}\rho_0 e^{j\phi} + \bar{\rho}\bar{u}e^{2j\phi})$$

$$+\nabla((\rho_0 + \bar{\rho}e^{2j\phi})(u_0^2 + 2u_0\bar{u}e^{j\phi} + \bar{u}^2e^{2j\phi}) + p_0 + \Gamma\bar{\rho}e^{j\phi})$$

$$= \partial_x((\mu_0 + \bar{\mu}e^{j\phi})(-jk\bar{u}e^{j\phi})) + 0 + (\rho_0 + \bar{\rho}e^{j\phi})g + F$$

$$\Rightarrow j\omega u_0\bar{\rho}e^{j\phi} + \bar{u}\rho_0 j\omega e^{j\phi} + \nabla((\rho_0 u_0^2 + 2u_0\bar{u}\rho_0 e^{j\phi} + \rho_0\bar{u}^2e^{2j\phi} + u_0^2\bar{\rho}e^{j\phi} \tag{6.4}$$

$$+2\bar{\rho}u_0\bar{u}e^{2j\phi} + \bar{\rho}\bar{u}^2e^{3j\phi}) + 0 + \Gamma\bar{\rho}e^{j\phi}) = -k^2\mu_0 e^{j\phi} + (0 + \bar{\rho}e^{j\phi})g + F$$

$$\Rightarrow j\omega(u_0\bar{\rho} + \bar{u}\rho_0)e^{j\phi} - jk(2u_0\bar{u}\rho_0 + u^2\bar{\rho})e^{j\phi} - jk\Gamma\bar{\rho}e^{j\phi}$$

$$= -k^2\mu_0\bar{u}e^{j\phi} + \bar{\rho}e^{j\phi}g + F$$

We need to evaluate F. Looking again at Eq. 6.2, the curvature can be defined as $\kappa_a = \vec{\nabla}\cdot\hat{\vec{n}}$, where the normal vector $\vec{n}$ is defined as $\vec{n} = \vec{\nabla}\alpha$. Because we know that $\alpha = \alpha_0 + \bar{\alpha}e^{j(\omega t - kx)}$, then the normal vector is $\vec{n} = -jke^{j(\omega t - kx)}$, and the modulus is $||\vec{n}|| = ke^{-\omega_i t}|\bar{\alpha}|$, bearing in mind that $\omega = \omega_r + j\omega_i$ and so $e^{j\phi} = e^{j(\omega_r t - kx)}e^{-\omega_i t}$. Therefore the unit normal vector is:

$$\hat{\vec{n}} = \frac{\vec{n}}{||\vec{n}||} = -j\frac{e^{j\phi}e^{\omega_i t}}{|\bar{\alpha}|}\begin{pmatrix} \bar{\alpha} \\ \\ 0 \end{pmatrix} \tag{6.5}$$

It follows that the curvature is given by:

$$\kappa_a = -ke^{j\phi}e^{\omega_i t}\frac{\bar{\alpha}}{|\bar{\alpha}|} \tag{6.6}$$

And the surface tension source term F is given by:

$$\begin{aligned} F &= \sigma(\rho_0 + \bar{\rho}e^{j\phi})(-k)e^{j\phi}e^{\omega_i t}\frac{\bar{\alpha}\vec{\nabla}\alpha}{\frac{1}{2}|\bar{\alpha}|(\rho_a+\rho_b)} \\ &\approx \sigma\rho_0(-k)e^{j\phi}e^{\omega_i t}\frac{\bar{\alpha}\vec{\nabla}\alpha}{\frac{1}{2}|\bar{\alpha}|(\rho_a+\rho_b)} \\ &= \sigma\rho_0(-k)e^{j\phi}e^{\omega_i t}\frac{\bar{\alpha}(-jke^{j\phi}\bar{\alpha})}{\frac{1}{2}|\bar{\alpha}|(\rho_a+\rho_b)} \end{aligned} \tag{6.7}$$

Thus the surface tension term is second order and in the first approximation it will be neglected.

In order to proceed further, we need to relate the mixture density ρ with the volume fraction α. We know from section 3.14.2 that:

$$\begin{aligned} \rho &= \alpha_L\rho_L + \alpha_G\rho_G = \rho_L + \alpha_G(\rho_G - \rho_L) \\ &= \rho_L + (\alpha_0 + \bar{\alpha}e^{j\phi})(\rho_G - \rho_L) \\ &= \rho_L + \alpha_0(\rho_G - \rho_L) + \bar{\alpha}(\rho_G - \rho_L)e^{j\phi} \\ &\equiv \rho_0 + \bar{\rho}e^{j\phi} \end{aligned} \tag{6.8}$$

where $\rho_0 = \rho_L + \alpha_0\Delta\rho$ and $\bar{\rho} = \bar{\alpha}\Delta\rho$.

Carrying on from equation 6.4 and keeping only first order terms,

$$\begin{aligned} &j\omega(u_0\bar{\rho} + \bar{u}\rho_0) - jk(2u_0\bar{u}\rho_0 + u_0^2\bar{\rho} + \Gamma\bar{\rho}) = -k^2\mu_0\bar{u} + \bar{\rho}g \\ &\Rightarrow ju_0\bar{\rho}(\omega - ku_0) + j\bar{u}\rho_0(\omega - 2ku_0 - \Gamma\bar{\rho}) = -k^2\mu_0\bar{u} + \bar{\rho}g \end{aligned} \tag{6.9}$$

Substituting $\bar{\rho} = \bar{\alpha}\Delta\rho = \frac{k\alpha_0\bar{u}}{\omega - ku_0}\Delta\rho$, we obtain:

$$ju_0k\alpha_0\bar{u}\Delta\rho + j\bar{u}\rho_0(\omega - 2ku_0) = -k^2\mu_0\bar{u} + \frac{k\alpha_0\bar{u}}{\omega - ku_0}\Delta\rho(g + jk\Gamma) \tag{6.10}$$

Normally we should substitute $\omega = \omega_r + j\omega_i$, however we will initially assume (just as a preliminary check of magnitude) that the main component of the frequency is the real part. Thus, assuming $\omega \approx \omega_r$, we have:

Real Part

$$\begin{aligned} k^2\mu_0\bar{u}(\omega - ku_0) &= k\alpha_0\bar{u}\Delta\rho g \\ \Rightarrow \omega &= \frac{\alpha_0\Delta\rho g}{k\mu_0} + ku_0 \end{aligned} \tag{6.11}$$

Imaginary Part

$$\begin{aligned} &u_0k\alpha_0\bar{u}\Delta\rho + \bar{u}\rho_0(\omega - 2ku_0) = \frac{k\alpha_0\bar{u}\Delta\rho}{\omega - ku_0}k\Gamma \\ &\Rightarrow \omega(u_0k\alpha_0\Delta\rho) - k^2u_0^2\alpha_0\Delta\rho + \rho_0(\omega^2 - 3\omega ku_0 + 2k^2u_0^2) = k^2\alpha_0\Delta\rho\Gamma \\ &\Rightarrow \rho_0\omega^2 + \omega(u_0k\alpha_0\Delta\rho - 3\rho_0ku_0) + k^2(2\rho_0u_0^2 - u_0^2\alpha_0\Delta\rho - \alpha_0\Delta\rho\Gamma) = 0 \end{aligned} \tag{6.12}$$

This is just a second order equation of the form $ax^2 + bx + c$, where the solution is given by $\dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a}$. In the previous equation, the terms are:

$$b^2 = u_0^2k^2\alpha_0^2\Delta\rho^2 + 9\rho_0^2k^2u_0^2 - 6\rho_0\Delta\rho u_0^2k^2\alpha_0$$

$$4ac = 8k^2\rho_0^2u_0^2 - 4k^2u_0^2\alpha_0\rho_0\Delta\rho - 4k^2\alpha_0\rho_0\Delta\rho\Gamma$$

$$\begin{aligned} \sqrt{b^2 - 4ac} &= k(u_0^2\alpha_0^2\Delta\rho^2 + u_0^2\rho_0^2 - 2u_0^2\alpha_0\rho_0\Delta\rho + 4\alpha_0\rho_0\Delta\rho\Gamma)^{1/2} \\ &= ku_0\rho_0\left(1 + \alpha_0^2\left(\tfrac{\Delta\rho}{\rho_0}\right)^2 - 2\alpha_0\tfrac{\Delta\rho}{\rho_0} + 4\tfrac{\alpha_0}{u_0}\Gamma\tfrac{\Delta\rho}{\rho_0}\right)^{1/2} \end{aligned}$$

When using the real part solution for the dispersion frequency as per eqn. 6.11, and substituting the variables used for FLUENT simulation of wave growth, the number obtained is 0.31, which is remarkably close to the value 0.34 obtained with the simulation. This would indicate that as a magnitude check the procedure is

consistent, however we should keep in mind that although the motion is assumed to be only in the x-direction, the perturbations depend on the position on the y-axis and therefore we need to assume that $\bar{\psi} \equiv \bar{\psi}(y)$ and is not constant. This task is currently being completed in preparation for publication (Kalogerakos et al., 2012a). The full analysis has not been carried out at this stage because it would involve making a choice between two different domains, which would affect the final results.

6.6 Conclusions

The wave growth problem analysis, a problem with known analytical solution, consisting of flow determined by the input of an initial sine-wave, has been carried out with EMAPS and the wave growth rate obtained compared successfully with results from TRIOMPH, Imperial College. The use of 2D FLUENT to simulate two-phase flow in a channel was initially validated by repeating the wave growth problem but this time using the volume of fluid (VOF) model, both by using incompressible and compressible gas flow (Kalogerakos et al., 2010). A full mathematical perturbation analysis on the VOF model was also carried out, in order to validate the simulation results of 2D FLUENT (Kalogerakos et al., 2012a). The wave growth rate calculated numerically as a first approximation was close to the one measured from the FLUENT simulation. Thus the theoretical validation gave a good preliminary agreement. A more detailed analysis is now being carried out where a y-dependency of the perturbation is assumed, but this is future work.

References

Blasius, H. (1911), 'Das Ahnlichkeitsgesetz bei Reibungsvorgangen', *Physik. Zeitschr.* **XII**(2), 11751177. (cited at page 148)

FLUENT (2006), *FLUENT 6.3 User Guide*, FLUENT Inc. (cited at page 168)

Jia, N. (2007), Adaptive mesh refinement in solution of two-phase flow problems, Master's thesis, School Of Engineering, Cranfield University. (cited at page xviii, 154, 155, 156)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2010), Comparison between 2-D CFD and 1-D code for wave growth simulations, *in* 'International Conference on Multiphase Flow', University of Florida. (cited at page 173)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012a), 'Y-dependent wave growth analysis for VOF model', *Multiphase Science and Technology* . In preparation. (cited at page 173)

Kowalski, J. E. (1987), 'Wall and interfacial shear stress in stratified flow in a horizontal pipe', *AIChE Journal* **33**(2), 274–281. (cited at page 151)

Omgba-Essama, C. (2004), Numerical Modelling of Transient Gas-Liquid Flows (Application to Stratified & Slug Flow Regimes), PhD thesis, School of Engineering, Cranfield University. (cited at page 147)

Spedding, P. L. and Hand, N. P. (1997), 'Prediction in stratified gas-liquid co-current flow in horizontal pipelines', *Int. J. Heat Mass Transfer* **40**(8), 1923–1935. (cited at page 148, 149)

Taitel, Y. and Dukler, A. E. (1976), 'A model for predicting flow regime transitions in horizontal and near horizontal gas-liquid flow', *AIChE.J.* **22**, 47–55. (cited at page 148, 149, 151)

Valluri, P. and Spelt, P. (2006), TMF3 Sub-project 1: Multifluid modelling: Comparison between TRIOMPH and EMAPS, *in* 'TMF3', Department of Chemical Engineering, Imperial College, London. (cited at page 147, 152)

Chapter 7

2D CFD simulation of slugs

7.1 Initial Investigation

After having shown that the wave growth problem can be simulated in a satisfactory manner using 2D Fluent, thus resulting in a simulation time that is approximately 10 times smaller compared with 3D Fluent, it was important to investigate and compare results of using 2D Fluent for slug simulations. The gas phase was treated first as incompressible and then as compressible.

Fluent simulations were carried out mostly in parallel environments, and often clusters of local machines with a high number of cores were used, shared with other colleagues. A Sun Grid Engine (SGE) environment was created, which was quite challenging at the beginning but once it was completed it allowed a much better coordination. Details of the setup can be seen in appendix E.

Incompressible flow simulations were the most straightforward to set up, although the running time was longer as velocities were generally higher than with compressible flow and therefore the time step (which had been selected to be adaptive) stayed lower than in compressible flow. In compressible flows it was necessary to reduce

the Courant number in order to avoid divergence. Moreover reverse flow was observed to happen much more frequently in compressible flows, as intuitively expected.

Tests were carried out where surface tension was set on and off. When activated, it was set equal to 0.073, as per EMAPS calculations. Surface tension contributed only on second order terms in wave growth analysis, therefore there was a presumption that it may be neglected in simulations, but it has been observed from simulations that the absence of surface tension greatly reduces the formation of slugs. It was thus decided to carry out further simulations by treating the gas phase as compressible and with the presence of surface tension.

7.2 Validation with small set of cases

It has been observed that with the VOF model Fluent will often introduce unnatural numerical perturbations, particularly when there are large differences in velocities between the phases at the interface (Kalogerakos et al., 2010). A solution to this problem has been found to be to run each simulation initially as a steady one, and then switch to transient once convergence or a temporary equilibrium has been reached. This method will allow a smoother interaction at the interface, and it avoids the formation of perturbations stemming from numerical effects. If the slug simulations had been carried out without the initial procedure of steady state simulation, followed by the transient, then the discrepancies between experiments and simulations would have been far higher. Therefore all cases were simulated with the initial procedure above.

Experiments carried out by Manolis (Manolis, 1995) have been used in order to compare results with known experimental data. The values of the initial

conditions can be seen in Table 7.1, where V_{SG} is the gas superficial velocity, V_{SL} is the liquid superficial velocity, and α_L is the initial liquid holdup.

The flow regime map used can be seen in Fig. 7.1. The regions that have been tested are in the stratified and in the slug flow regime.

Table 7.1: Manolis Cases used for hydrodynamic slug flow simulation

Case	V_{SG} (m/s)	V_{SL} (m/s)	α_L
22	4.016	0.519	0.670
36	1.548	0.519	0.808
38	2.058	0.498	0.766

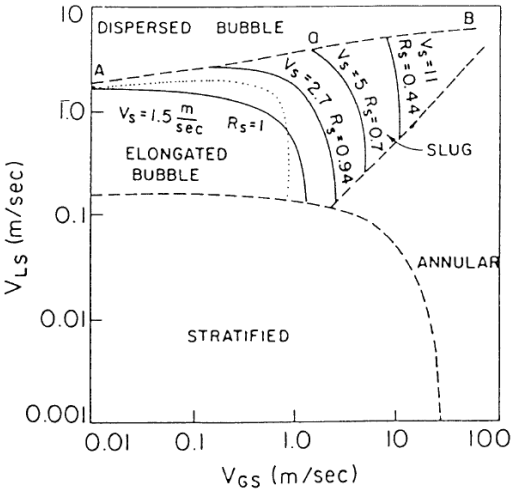


Figure 7.1: Flow regime map showing flow transition boundaries (Barnea and Brauner, 1985).

It is important that solutions obtained satisfy mesh independence, therefore for each simulation four different meshes were used, each with a higher number of

cells as shown in Table 7.2. The mesh refers to a horizontal channel of length 30m and diameter 78mm. The term cell configuration refers to number of cells in the y direction and number of cells in the x direction, so for example 20-3000 refers to 20 cells in the diameter direction and 3000 cells in the pipe length direction. An example of a graph of liquid holdup vs. time for case 36 is shown in Fig. 7.2.

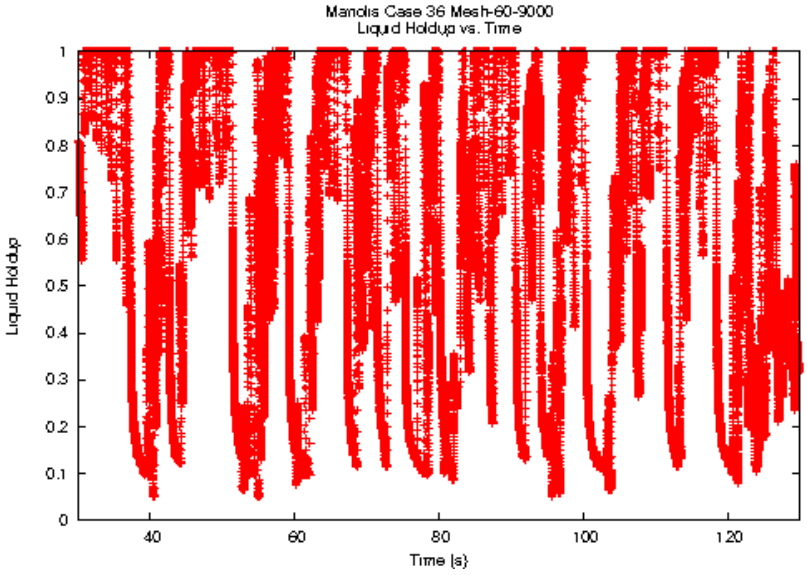


Figure 7.2: Liquid holdup vs. time for Manolis case 36

As shown in Fig. 7.3 all three simulations of Manolis cases seem to converge to stable frequency results (change of less than 0.5%) after a certain refinement level. It can also be observed that the frequency results are within 4% of the experimental results. This agreement gives more support to the possibility of using two-dimensional Fluent in order to carry out pipe simulations for two-phase flows (Kalogerakos et al., 2010). Extrapolations to more phases and/or more complicated shapes are not straightforward, and more tests will have to be carried out in order to confirm that. In more complicated shapes with less symmetry compared with a pipe, the approximation of a pipe using a two-dimensional channel will probably not be appropriate.

Table 7.2: Slug frequencies in FLUENT 2D simulations of Manolis cases 22, 36, 38

Case	Cells	2D Fluent Frequency (Hz)	Experimental Manolis Frequency (Hz)
22	20-3000	0.0255	0.1333
22	40-6000	0.1286	0.1333
22	60-9000	0.1300	0.1333
22	80-12000	0.1304	0.1333
36	20-3000	0.1970	0.2444
36	40-6000	0.2360	0.2444
36	60-9000	0.2380	0.2444
36	80-12000	0.2383	0.2444
38	20-3000	0.1467	0.2167
38	40-6000	0.2000	0.2167
38	60-9000	0.2250	0.2167
38	80-12000	0.2265	0.2167

7.3 Further validation

When looking at a typical flow regime map (Fig 7.1) for a pipe similar to the one used in our simulations, then a slug regime is expected in the following range of superficial velocities:

- Liquid superficial velocity: 0.3 m/s - 3.0 m/s
- Gas superficial velocity: 0.9 m/s - 3.3 m/s

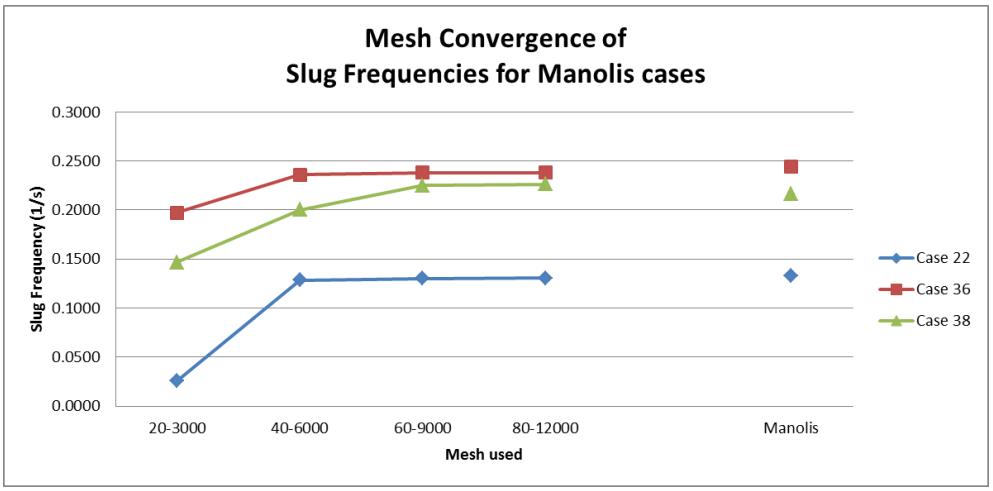


Figure 7.3: Mesh convergence for slug frequencies calculated for Manolis cases

So far two-dimensional CFD simulations have given good approximation to the predicted slug frequency, but the sample used for testing was very limited and in order to check whether this agreement will continue we need to extend our testing. Therefore it was decided to carry out simulations (Kalogerakos et al., 2011) on a further set of experimental cases, which were chosen for their difference in velocities and because the predicted slug frequencies are in the extreme end, i.e. either very high or very low. In this manner it should be possible to see whether a 2D simulation will still hold or break down. Mesh used was 40-6000, as it was shown earlier to be a satisfactory choice for two of the three cases, and for case 38 it may not be the optimal choice but it is a necessary compromise between accuracy and speed. The experimental cases were once more taken from Manolis (Manolis, 1995) and they are shown in Table 7.3 together with the simulation results. In the graph shown in Fig. 7.4 the 3D plot highlights the various degrees of agreement of Fluent 2D simulations with the experimental results by highlighting the discrepancy between simulation and experimental slug frequency, in a grid of liquid velocities vs. gas velocities, calculated using Matlab

Gas Velocity m/s	Liquid Velocity m/s	Experimental Slug Frequency	2D FLUENT Slug Frequency
6.109	1.220	0.683	0.728
6.937	1.282	0.817	0.824
6.051	0.509	0.074	0.120
1.548	0.519	0.244	0.230
2.058	0.498	0.217	0.200
8.572	1.081	0.461	0.333
8.344	0.490	0.194	0.101
10.436	0.528	0.156	0.083
10.474	0.761	0.174	0.108
7.907	0.745	0.278	0.185
6.837	0.772	0.287	0.221
6.532	0.532	0.333	0.125
6.541	1.065	0.200	0.452
1.945	0.751	0.506	0.470
2.285	1.001	0.444	0.442
3.830	1.241	0.415	0.566
2.846	1.026	0.541	0.477
3.649	1.0410	0.3459	0.389
5.024	1.300	0.339	0.638
5.108	0.999	0.346	0.490
1.796	0.751	0.794	0.296
4.054	1.025	0.486	0.426
7.037	0.817	0.287	0.215
6.981	0.534	0.083	0.087

Table 7.3: Cases used for further validation of hydrodynamic slug flow simulation, together with experimental and 2D FLUENT slug frequency

(MATLAB R2010a, 2010). In Fig. 7.5 the discrepancies between experiments and simulations can be visually appreciated: the horizontal axis is the gas velocity and this gives another view of the discrepancies related to the inlet gas velocity. The average discrepancy is 22.7% which may appear rather large but it is better than most comparisons found in recent literature. The worst disagreements appear to occur in correspondence of cases having the combination of high gas velocities and

low liquid velocities. As it can be seen in Fig. 7.6, slug frequency discrepancies of less than 20% account for 59% of all cases, therefore it indicates overall an acceptable agreement, considering the limitations of using a 2D channel instead of a full 3D pipe.

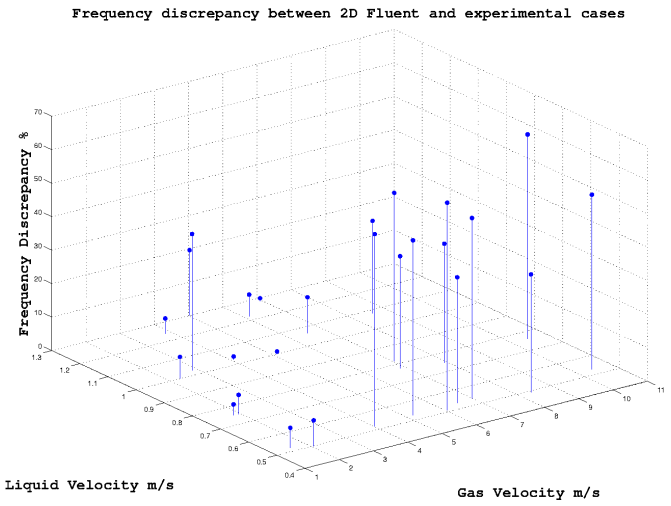


Figure 7.4: Frequency discrepancies in percentage between experimental and 2D CFD for small set of slugs.

Figures from the new cases are shown in Fig. 7.7. What can be appreciated in these graphs is the fact that slugs do occur, with varying frequency depending on the inlet conditions for velocities. It can also be seen that frequently the slugs reach the top of the pipe, thereby filling it up completely with liquid at that point. Although it is difficult to determine whether slugs are periodic by looking at the graphs, an average slug frequency is a useful indicator of how close the 2D CFD results are to the experimental results.

In order to carry out an extensive testing of the modelling capability of 2D CFD for two-phase channel flow, it was decided to carry out 2D CFD simulations on the largest possible sample of the recorded experimental cases (Manolis, 1995)

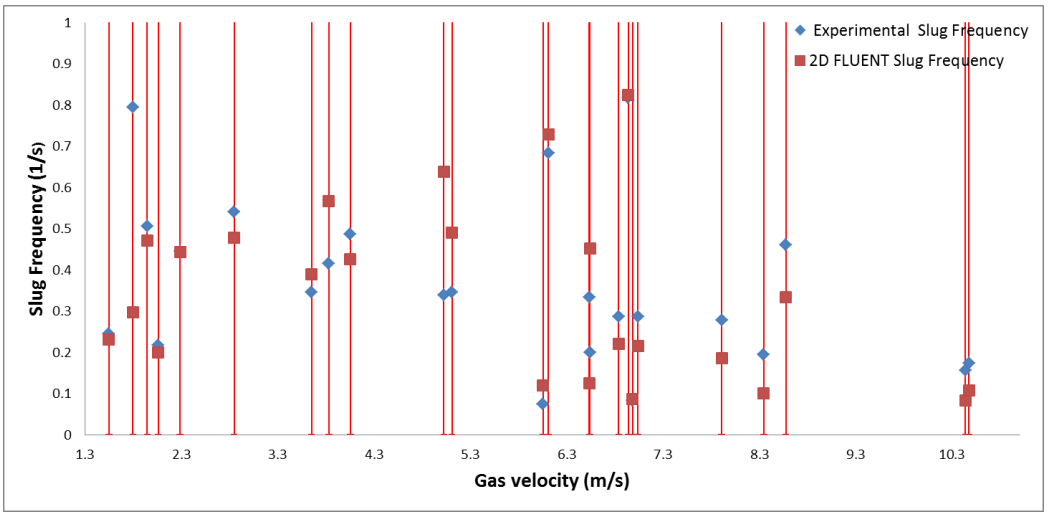


Figure 7.5: Experimental and 2D CFD calculated slug frequencies vs. inlet gas velocities. Each pair of experimental and 2D simulation results lie on the same vertical line.

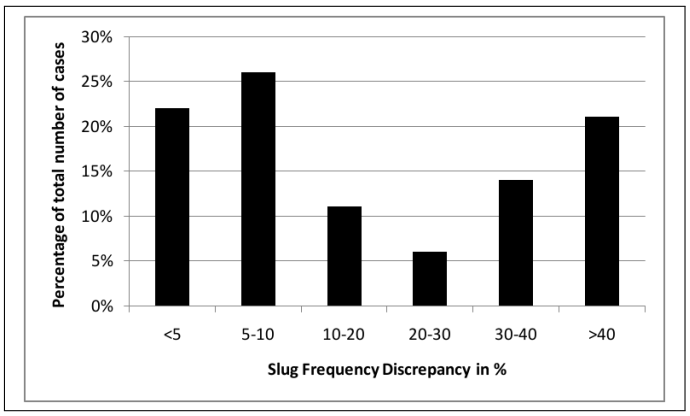


Figure 7.6: Range of slug frequency discrepancies between 2D Fluent and experimental data for small set of slugs.

which contain information also regarding slug frequencies. A full list of the cases is included in Appendix C. Moreover a small sets of simulations was carried on cases with initial conditions in the stratified regime, to check whether the simulations correctly predict the expected flow regime.

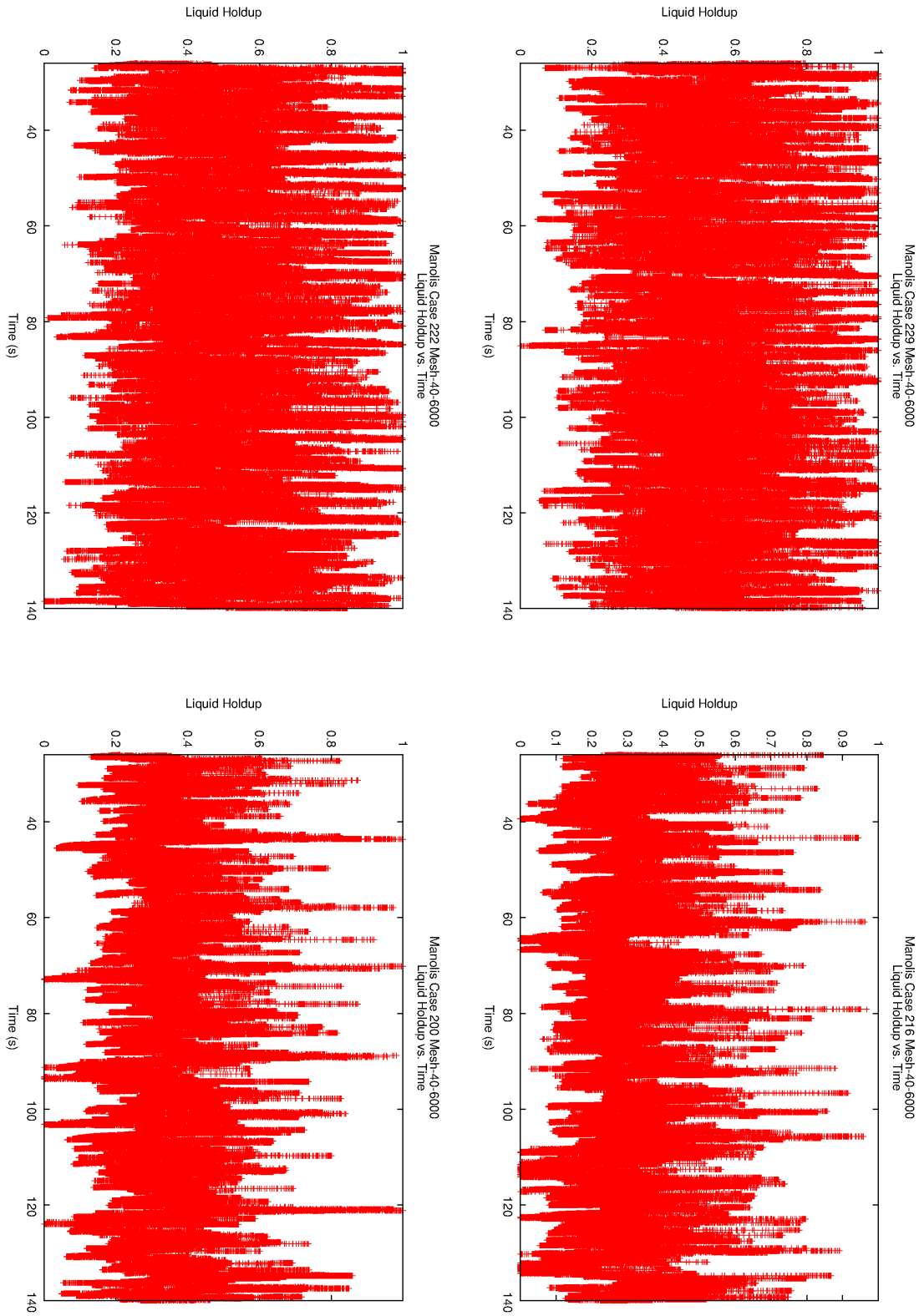


Figure 7.7: Liquid Height vs. Time for Manolis cases 222, 229, 200 and 216

7.4 Simulation of stratified flow

For stratified flow cases, initial conditions were taken from the flow regime map, and when running transient flow, the simulations reached very quickly a state of quasi-equilibrium. The graph in Fig. 7.8 shows the liquid volume fraction as it evolves during the first 90 seconds. To be noted that the simulation was carried out in transient mode, and not steady state, and so there was no convergence but still a state of quasi-equilibrium was reached. Therefore 2D CFD can correctly predict stratified flow as shown in the flow regime map.

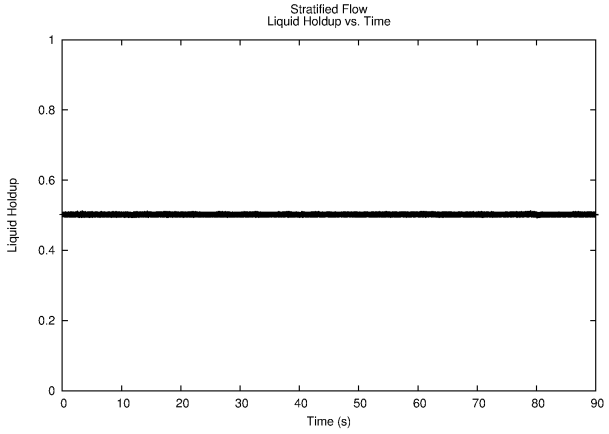


Figure 7.8: Liquid holdup vs. time for stratified flow.

7.5 Simulation of full set of slug experiments

In continuation from the results achieved using the steady state solution as an initial point for the transient simulations, it was decided to carry out simulations of all experimental cases with measured slug frequencies (Appendix C). The total number of cases investigated is 92. A simulation set of this kind of scale should give a better indication of the overall accuracy in the prediction of slug frequencies

using 2D CFD, and hopefully will allow also to identify any shortcomings in order to be able to make an informed choice when deciding on the methodology used.

7.5.1 Identification of individual slugs

Identifying slugs and hence calculating slug frequency is a process that normally is completed manually for each simulation. Since the current set of slug cases is a total of 92, it was necessary to devise an automated way of identifying slugs. An example of a slug with relatively high frequency can be seen in Fig. 7.9 and it is evident that an automated process is desirable as slug counting has to be carried out for over 92 cases. A more detailed view on a smaller time scale of the same experimental case is shown in Fig. 7.10, where again new details emerge that were not immediately visible from the previous graph. A close-up view of a single slug (for experimental case 22) can be seen in Fig. 7.11. The thought process of

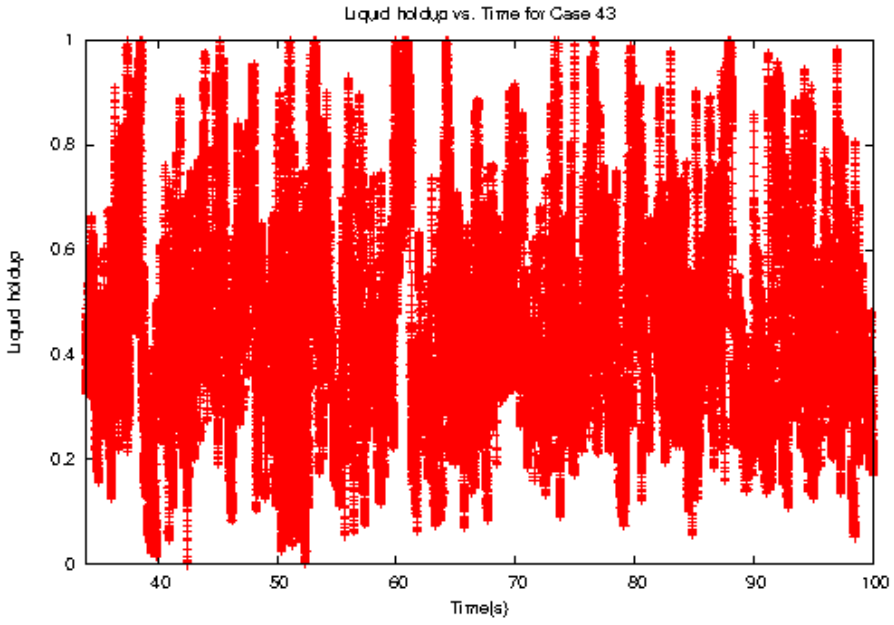


Figure 7.9: Liquid holdup vs. time for experimental case 43, overall view.

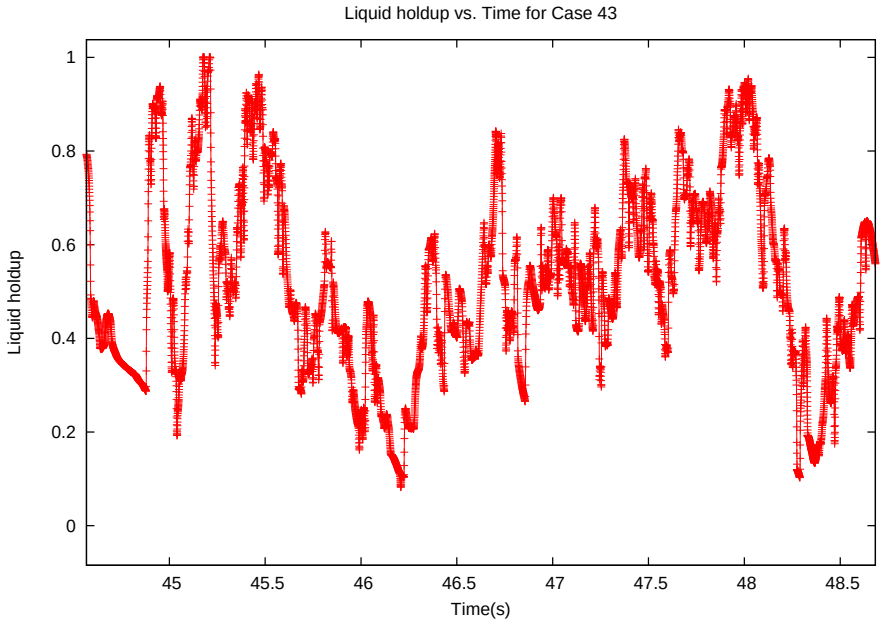


Figure 7.10: A detailed view of the graph of liquid holdup vs. time for experimental case 43.

identifying slugs had to be clearly set out, not only to decrease post-processing time but also to ensure repeatability and independence of results. In order for an algorithm to be written, the following constants need to be defined (here called threshold constants):

- What is the minimum value (slug_start) of liquid holdup above which a slug has formed?
- What is the maximum value (slug_end) of liquid holdup below which the previous slug is deemed to have finished?

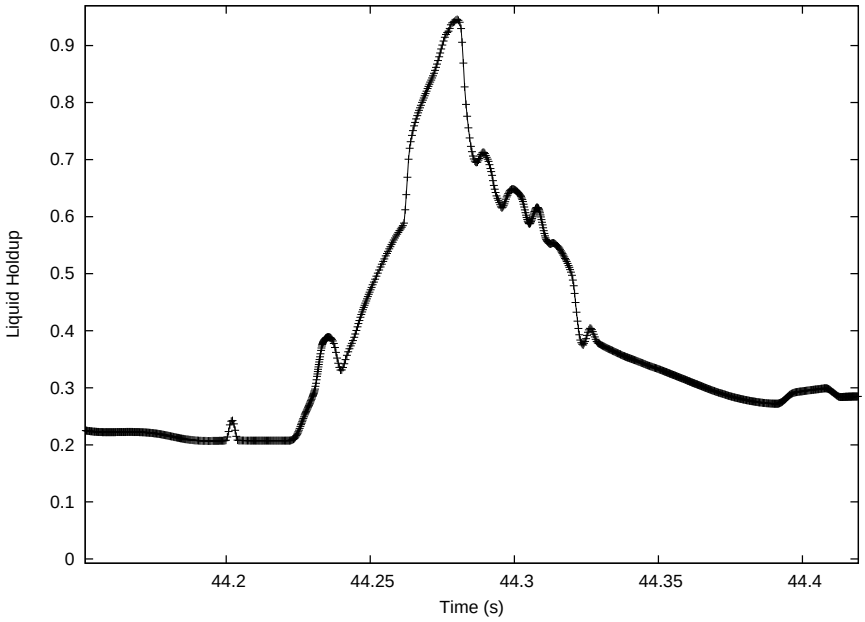


Figure 7.11: Close-up view of a single slug in experimental case 22.

The process itself consists of the following steps:

- Start looking for slug_start.
- Once slug_start has been reached, then increase count of slugs by 1, and start checking for slug_end.
- Once slug_end has been reached, then look again for slug_start and repeat process.

This is the process followed also when counting the slugs manually, and therefore assuming that the threshold constants are the same, then the results obtained with the algorithm should be the same as the ones obtained manually. A check was carried out on a set of 20 cases and the results were in agreement. All simulations were carried out for a total of 100s, and the first 20s were neglected

as a certain initial time was found to be necessary for the flow to be established. This was a decision based on empirical observations over many cases.

7.5.2 Results

Discrepancies between simulation and experimental slug frequencies, with different threshold constants chosen are shown in Table 7.4. A 3D view of the frequency

		Slug_start		
		0.80	0.85	0.90
Slug_end	0.20	22.3%	22.3%	20.5%
	0.30	18.8%	19.1%	21.4%
	0.40	15.4%	18.1%	19.4%

Table 7.4: Effect of use of different threshold constants on discrepancies of slug frequencies

discrepancies between experiments and 2D CFD simulations can be seen in Fig. 7.12, where a higher column indicates a worse result. A projection of that graph onto the plane of liquid velocity vs. gas velocity with colours indicating the level of discrepancy is shown in Fig. 7.13, and there the Kelvin-Helmholtz inviscid limit is also shown. This line indicates the narrow range of applicability of the two-fluid 1D model, and the importance of being able to use a well-posed 2D model to describe slugs outside that range.

As can be seen in Fig. 7.14, slug frequency discrepancies of less than 20% account for 57% of all cases, therefore it indicates overall a good agreement. It was also observed that at times large differences in discrepancies occurred between

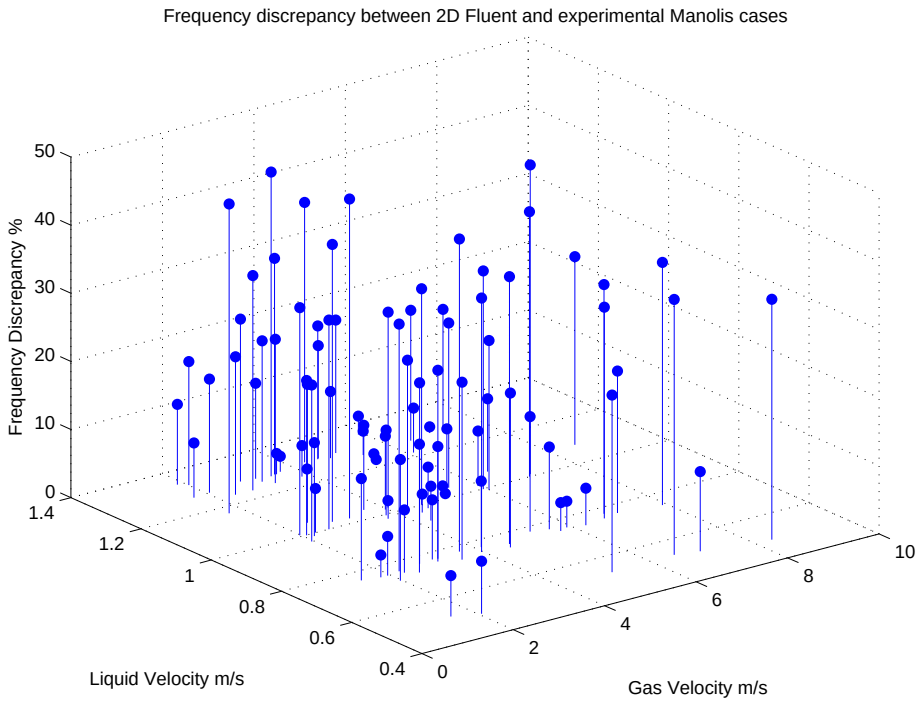


Figure 7.12: Frequency discrepancies in percentage between experiments and 2D CFD simulations, for full set of slug cases, viewed in 3D.

cases with very small difference in initial conditions. An example is shown in Table 7.5. This indicates that at least the CFD simulations are consistent, and that possibly more variables should have been included in order to have a more accurate description of the problem, or perhaps some experiments themselves could benefit from a reassessment.

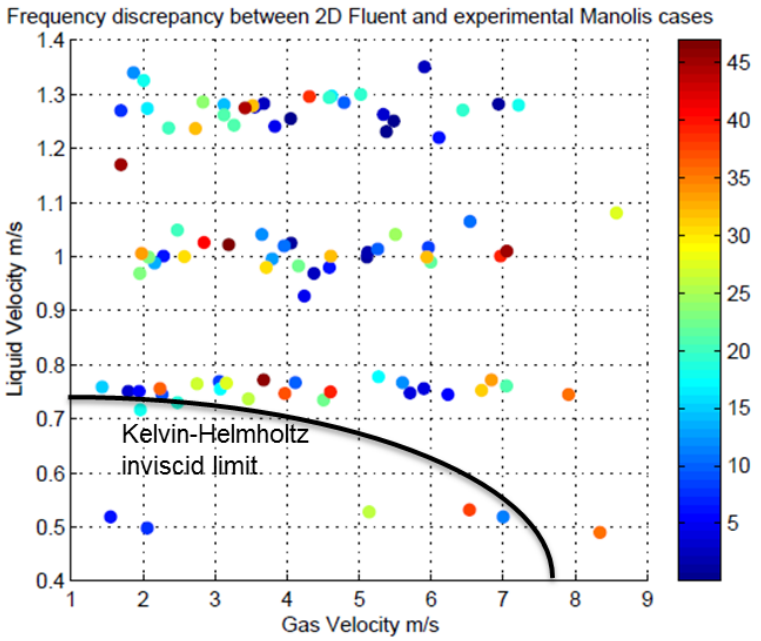


Figure 7.13: Frequency discrepancies in percentage between experimental and 2D CFD, for full set of slug cases. The Kelvin-Helmholtz inviscid limit is also shown - above this line the two-fluid model used in 1D simulations is not well-posed, while 2D CFD is.

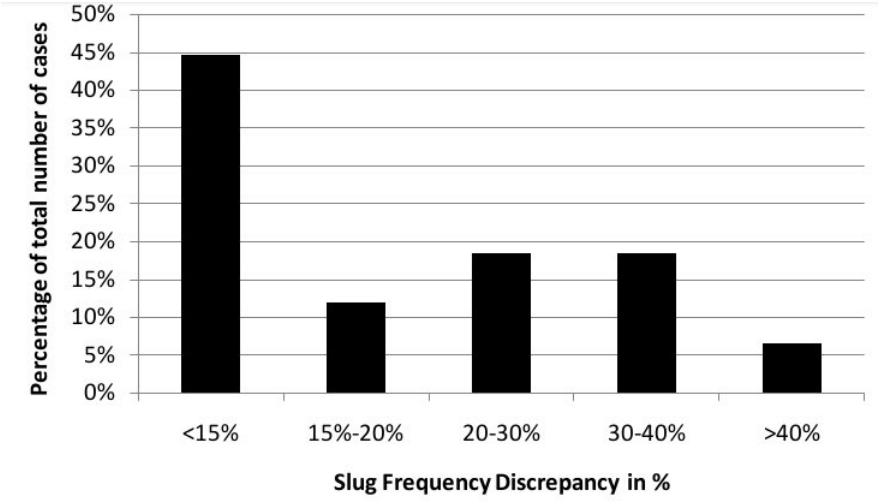


Figure 7.14: Range of slug frequency discrepancies between 2D Fluent and experimental data.

	Gas Velocity	Liquid Velocity	Experimental Frequency	2D CFD Frequency
Case 133	3.969	0.747	0.173	0.238
Case 194	4.116	0.767	0.289	0.261
% Diff.	3.7%	2.7%	67.1%	9.7%

Table 7.5: Two cases with similar initial conditions but large discrepancies

7.6 Conclusions

The use of 2D CFD to simulate two-phase flows in pipes (approximated by channels) gave satisfactory results for stratified flow and more importantly for transient flows, in particular slug flow. Results of mesh independence were used (Kalogerakos et al., 2010) and simulations results were compared with a regime flow map and with a set of experimental data Manolis (1995). Native 2D Fluent simulation with VOF model did not work due to the interfacial problems, in particular due to velocity differences between phases at the start. A methodology was presented here that allows this problem to be overcome, by carrying out an initial steady state simulation till convergence or quasi-equilibrium is reached, and subsequently the transient simulation is started.

The promising results from a limited sample (Kalogerakos et al., 2011) prompted for an extension to the largest sample of slugs available (from a unique source). Considering the size of the sample used, and also taking into account the limitations of the VOF model and the use of a channel instead of a 3D pipe, the simulation results show a good agreement with experiments (Kalogerakos et al., 2012c). Comparison with experimental cases showed that discrepancies between CFD values of frequencies and experimental measurements were worst for combination of high gas velocities and low liquid velocities. The use of 2D CFD has also been shown previously to give good results for wave-growth in two-phase flows in straight pipes (section 6, Kalogerakos et al. (2012a)). In cases where few slugs (<10) were observed over the duration of 100s, the choice of the threshold constants has a major effect on the resulting frequency.

Discrepancies between experimental and 2D CFD simulation frequencies range from an average of 14.9% to an average of 22.9%, depending on the threshold constants chosen.

Running times are approximately ten times quicker compared with 3D Fluent, and 2D Fluent can definitely be recommended both as an investigative tool but also a predictive tool. It can be used as a tool to predict flow in simple geometry cases, or perhaps as an initial tool to estimate flow properties in more complex cases. Moreover, use of 2D simulations offers also the possibility of measuring properties that then can be put back in 1D code in order to carry out calibrations, particularly for cases where there are discrepancies or outside the well-posedness region, in order to modify the source terms. Therefore 2D CFD is flexible and fast, provides a certain degree of accuracy, and can, under certain circumstances, be a valid substitute for 3D CFD.

As part of future work, the next step would be to carry out 2D CFD simulations on annular flow and dispersed bubble flow. A set of cases is ready for simulation, but it has been observed on some preliminary tests that the time-steps required are much smaller than the ones that have been encountered so far, and therefore more time will be needed to have a statistically significant sample.

References

Barnea, D. and Brauner, N. (1985), 'Holdup of the liquid slug in two phase intermittent flow', *International Journal of Multiphase Flow* **11**(1), 43–49. (cited at page xix, 179)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2010), Comparison between 2-D CFD and 1-D code for wave growth simulations, *in* 'International Conference on Multiphase Flow', University of Florida. (cited at page 178, 180, 195)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2011), Use of 2-D CFD for simulating two-phase flows in horizontal pipes, *in* 'International Association of Science and Technology for Development, Applied Simulation and Modelling'. (cited at page 182, 195)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012a), 'Y-dependent wave growth analysis for VOF model', *Multiphase Science and Technology* . In preparation. (cited at page 195)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012c), 'Comparison between 2D CFD and experiments for slug flow', *International Journal of Multiphase Flow* . In preparation. (cited at page 195)

REFERENCES

Manolis, I. G. (1995), High Pressure Gas-Liquid Slug Flow, PhD thesis, Department of Chemical Engineering and Chemical Technology, Imperial College of Science, Technology and Medicine, UK. (cited at page 178, 182, 184, 195)

MATLAB R2010a (2010), *MATLAB Documentation*, The MathWorks 2010. (cited at page 183)

Chapter 8

Industrial Analysis

8.1 Introduction

The aim of the current project is to develop methods to predict the initiation and development of slugs in oil pipelines. The framework within which the research is placed is that of the Transient Multiphase Flow Programme (TMF), which is sponsored by many companies from different countries, including oil companies. A list of the major sponsors of TMF is given below:

- ASCOMP
- GL Noble Denton
- BP Exploration
- CD-adapco
- Chevron
- ConocoPhillips
- ENI
- ExxonMobil
- FEESA
- IFP Energies nouvelles
- Institutt for Energiteknikk
- PDVSA (INTEVEP)

- Petrobras
- PETRONAS
- SPT Group
- Shell
- SINTEF
- Statoil
- TOTAL.

TMF is a framework consisting of consortia where pre-competitive research is managed and carried out. Pre-competitive research enables the collaboration between larger companies, smaller companies, and also government funded bodies and results in a productive engagement. The development of tools and methods is done in a cooperative manner and there is a management of knowledge creation and distribution between industry and academia.

BP is the industrial sponsor of the current project. In this research use was made of a direct communication channel with BP - data and reports were exchanged, and frequent visits with BP representatives took place. Communications also with other companies from the TMF consortium took place, albeit more rarely, and these also contributed to placing the work in context. A brief overview of BP will be given, together with the key financial aspects and its key areas of interest.

A basic view on oil economics will be given, including demand and supply of oil, and the trend in oil prices will also be analysed, in order to be correlated with the incentive for oil companies to invest in research and development. The latest events when there was a a large fluctuation in the oil price will also be mentioned.

Known effects of slugs in the oil industry will be shown and their cost implications will be given in detail. Advice on approaches to be taken to tackle the slugging phenomenon will also be given, starting from the results of the thesis. Moreover a process design discussion will be applied to pipe design, accompanied by some general considerations.

A general assessment of the possible impact of the use by BP of the outcome of the project will also be given, and a brief description of the areas where research and development has been focused is also provided.

The outcome of the current project has been a combination of new product development and a methodological innovation. The new product development is the new computing framework of the upgraded 1D multiphase code EMAPS, specifically:

- Optimised and upgraded version of 1D code EMAPS.
- Numerical enhancements with velocity profile coefficients.
- Validation with wave growth problem.
- Parallelisation of all models and sources in EMAPS.
- Testing suite for all sequential and parallel cases.
- Versioning control (SVN) and automatic testing upon code submission.

The methodological innovation consists of the use of 2D CFD VOF for channel simulation with:

- Special initialisation techniques to allow transient simulations.
- Validation with wave growth problem.
- Mathematical perturbation analysis.
- Simulations of 92 experimental slug flow cases.

Therefore the work presented in this thesis has direct relevance to oil industry, as well as to industry associated with pipeline design, installation and operation.

8.2 BP p.l.c.

BP is the industrial sponsor of the research carried out in the current Engineering Doctorate. There have been regular communications and exchanges of information with representatives from BP. The main contact was a senior flow assurance engineer, who was also responsible in the decision-making process regarding multiphase flow projects design and assignment. He was in communication with various departments in BP, including dealing with field data, experiments and simulations. The deliverables agreed for the project were:

- validating EMAPS through simulations of known problems and experimental and field data concerning slug flow
- introducing numerical enhancements to EMAPS
- decreasing computation times in EMAPS
- using multi-dimensional methods to investigate slug flow.

These deliverables would constitute a series of technology innovations for BP, even though they have the capabilities already in place and/or the availability of external consultancies in order to implement the solutions proposed, as these are software based and their uptake could be carried out in a small time-frame.

Apart from the financial contribution, BP has also provided us with a large data set from Prudhoe Bay (Alaska) oil field. A summary of BP's operations in Alaska are given below in order to provide the context of their work in terms of their global operations.

8.2.1 Overview

BP is a global oil and gas company with its main head-quarters in London. As shown in the summarised list below, it operates oil and gas fields in many countries and continents:

- United Kingdom
- Norway
- Trinidad and Tobago
- Gulf of Mexico
- Alaska
- Azerbaijan
- Egypt
- North America
- Vietnam
- Angola
- Colombia
- Indonesia
- Australia
- Russia

Its global presence is in more than 80 countries, if both operation and marketing processes are taken into consideration. In 2010 BP spent $780 million on research and development (R&D), compared with $587 million in 2009 and $595 million in 2008. Moreover BP has long-term research programs with universities and research institutions around the world, ranging from energy bioscience and conversion technology, to carbon mitigation and nanotechnology in solar power. BP is also at the forefront of the development and application of innovative exploration technologies, including seismic acquisition techniques, fibre optic pipeline monitoring technologies and enhanced oil recovery (EOR) technologies, eg. gas injection which reduces the viscosity of crude oil. In particular EOR technologies are believed to increase the overall recovery factor from oil fields by 1%.

	$ million				
	2010	2009	2008	2007	2006
Sales and other operating revenues	297,107	239,272	361,143	284,365	265,906
Profit (loss) for the year	(3,324)	16,759	21,666	21,169	22,601

Table 8.1: Brief summary of BP's financial information over the last 5 years (BP, 2010)

Due to the immense costs associated with the oil spill in the Gulf of Mexico, financial values for year 2010 show a loss (Table 8.1), but otherwise sales have increased compared to the average of previous years.

8.2.2 Alaska: Prudhoe Bay

Prudhoe Bay is the largest oil field in North America, measuring about 15 miles by 40 miles (Fig. 8.2). It is also the 18th largest field ever discovered worldwide, originally containing 25 billion barrels of oil. It is located in northern Alaska on the coast of the Arctic Ocean. The field was discovered in 1968 and production began in 1977 when the Alaska Pipeline was completed (BP, 2006). The maximum rate of production was reached in 1979 at 1.5 million barrels per day, which was maintained until 1989, after which a decline by 10% per year has been observed. BP operates the field, but a total of nine companies have also an interest in the field leases, including ConocoPhillips Alaska Inc. (36%) and ExxonMobil (36%); BP itself has 26% ownership. The major owners have invested more than $25 billion to developed the Prudhoe Bay field and the transport system necessary to move Prudhoe Bay crude oil to market.

Oil is currently being transported through the Trans-Alaska pipeline from Prudhoe

Bay to Valdez, Alaska 800 miles to the South (Fig. 8.1). The pipeline is four feet in diameter, and built about 5 feet above ground (Fig. 8.3). It is also designed to permit 5 feet of vertical movement and up to 20 feet laterally, in order to allow non-destructive movement during seismic events. Every 75 miles there are pump stations that help the oil move along the pipe. The oil takes approximately 5 days to traverse the pipeline.

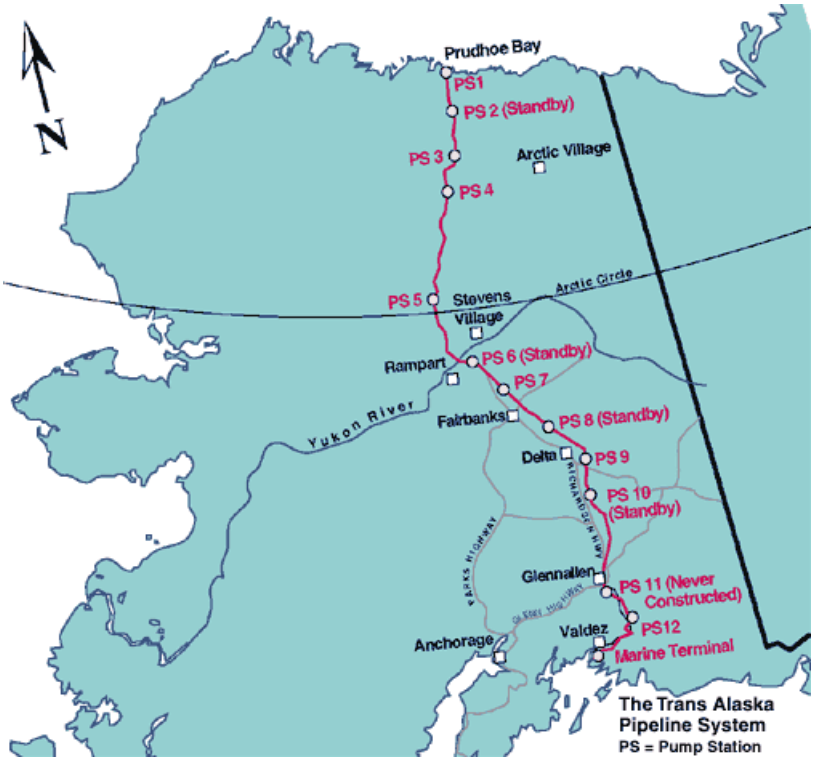


Figure 8.1: Map showing Prudhoe Bay and the pipe connecting it to Valdez in the south.

Figure 8.2: Aerial view of Prudhoe Bay (BP, 2006).

Figure 8.3: Snapshot of the Trans-Alaska pipeline (Smith, 1996)

8.3 Oil

The detailed effects of oil prices on the economy and on society are very complex and far-reaching, and they are outside the scope of this project. What is of more direct relevance is the effect of the oil price change on the incentive for investment for oil companies.

8.3.1 Oil Economics

As shown in Fig. 8.4 the price of oil has proved to be very volatile, reaching a maximum of $147 a barrel in July 2008, before dropping at $32 at the end of the same year, and then increasing again. Not only companies but even whole countries (especially the major oil exporters) base their budget forecast on income generated from oil export, therefore such price volatility can cause serious issues.

As can be seen in Fig. 8.5, production and supply of oil has not increased significantly since 2005, therefore, at least in the short term, it is unlikely that there will be a sudden increase of supply of oil, partly because it is considered that all "easy" oil has already been found (Wheatcroft, 2010). Life cycle patterns for oil reservoirs and fields have been well established (Sorrell et al., 2010), but their effect on future oil supply is still being debated. Normally, in a given field, a profile of initial increasing production flow rates is followed by eventual decline, and to keep total production increasing, it would be necessary to discover new fields continuously. High oil prices will motivate oil explorations normally riddled with high costs either due to difficult accessibility or due to different oil typology (eg. extremely heavy crude oil). Such investments may lead also to discovery of new technological developments which will allow extraction at a

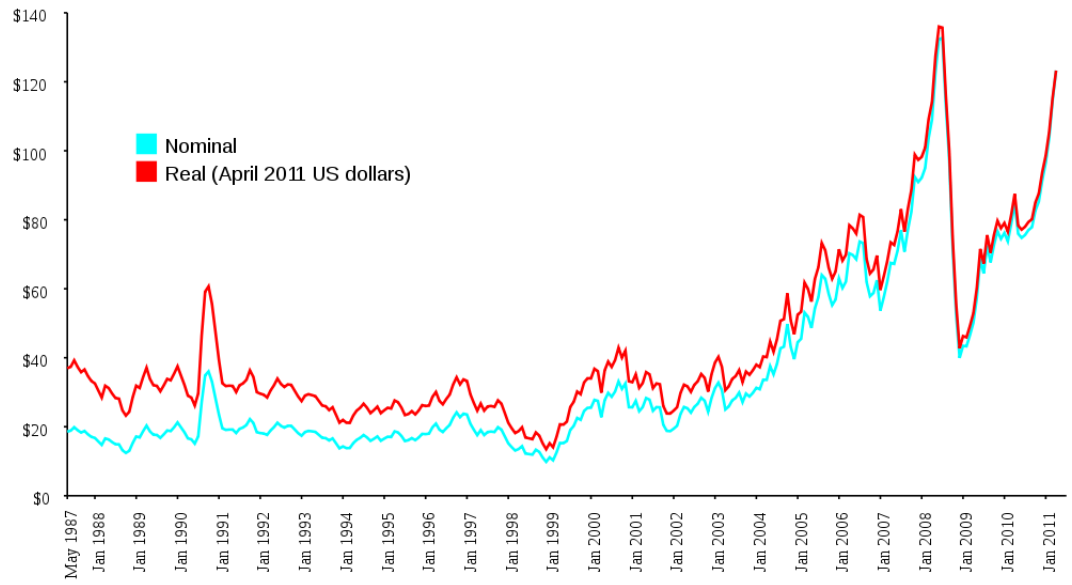


Figure 8.4: Monthly average Brent prices for the period May 1987 - April 2011 (EIA, 2011)

lower cost than initially forecast. Moreover developments of new technologies provide organisations with the opportunity for growing their businesses as new and improved technologies are usually recognised to be the drivers of competitive advantage (Rickard, 2006). But it is also important to remember that there are very long lead times between the initial discovery of a new oil reservoir and the actual time when the new oil will be delivered.

Although the oil price may be volatile, what should be kept in mind is that the cradle-to-grave timescale for the innovations of the current project is relatively short compared with changes to oil demand and therefore effects due to changes occurring in the project timescale may be rather limited. The estimates of the timescale are about 5 years for the innovations as they are at the moment and around 10 years allowing for small additions/upgrades.

Hereafter a brief explanation of the demand curve and price elasticity will be

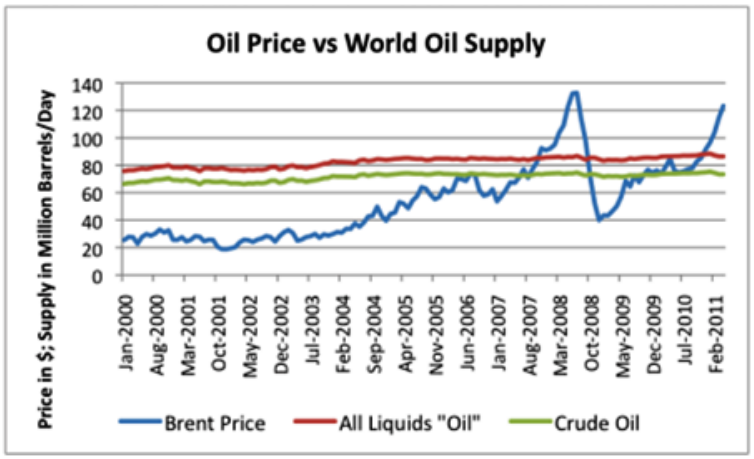


Figure 8.5: World oil production, compared to Brent price (EIA, 2011)

given, as its understanding is helpful in order to have an understanding of the relation between oil price and income.

The importance of the demand curve is that for any business, knowledge of its level is critical to the generation of revenue which is given by the product of the quantity sold (Q) and the price of the product (P). Hence an increase in the quantity sold (ΔQ) will increase total revenue. An increase in the price (ΔP) of the product is likely to reduce sales and the issue for the business is how responsive sales are to a change in price (Rickard, 2006). This responsiveness is measured by the price elasticity of demand (ϵ_P) as shown below (Rickard, 2006):

$$\epsilon_P = \left| \frac{\%\text{ change in } Q}{\%\text{ change in } P} \right| = \left| \frac{\Delta Q}{\Delta P} \times \frac{P}{Q} \right| \tag{8.1}$$

Thus for example if ϵ_P=2, and whatever the percentage reduction (or increase) in the price, the percentage change in the quantity demanded will increase (or decrease) by twice the percentage. In other words, a 10% reduction in price will lead to a 20% increase in the amount sold. According to the value of ϵ_P encountered, this relationship can be categorised as follows (Rickard, 2006):

- $1 > \epsilon_P > 0$: demand is price inelastic: an increase (or decrease) in price

will increase (or reduce) revenue.

- $\epsilon_P = 1$: demand is unit elastic: a change in price will lead to an identical percentage change in the quantity demanded therefore revenue will remain constant.

- $\epsilon_P > 1$ demand is price elastic: an increase (or reduction) in price will reduce (or increase) revenue.

The oil demand has been modelled (IMF, 2011) as a third-order polynomial:

$$o_{it} = \alpha_i + \lambda_t + \rho o_{it-1} + \beta \log(p_{it}) + \gamma \Delta \log(y_{it}) + P(y_{it}) + u_{it} \tag{8.2}$$

where o is oil per capita, y is real per capita GDP at purchasing power parity, $P()$ is a third-order polynomial, p is the real price of oil in local currency, fixed effects are captures by α_i and λ_t represents time dummies. More detailed explanation of the variables can be found in IMF (2011). This model was derived from a large set of data using a particular type of low-frequency filtering, called asymmetric filtering (Christiano and Fitzgerald, 2003).

The elasticity calculated by IMF (2011) disagrees with the value obtained by Hamilton (2008), whose estimate is 0.2 to 0.3, and who also calculates estimates for the US in the range of 0.4 to 0.5. But there is agreement on the presence of three key features:

- low price elasticity of demand

- strong growth in demand from China, the Middle East and other newly industrialised economies

- failure of global production to increase.

	Short-Term Elasticity	Long-Term Elasticity
	Price	Price
Combined OECD,		
Non-OECD,	-0.017	-0.067
and major oil-exporting economies[1]		

Table 8.2: Oil demand price elasticities, including oil-exporting economies (IMF, 2011)

Thus it is unlikely that, in the long run, there will be a decrease in pressure on oil prices. The continuous increase in oil consumption by developing countries can be seen in Fig. 8.6.

A list of oil demand price elasticities for a large sample (1990-2009) is shown in Table 8.2. Here a 10% increase in oil prices would lead to a reduction in oil demand of 0.2%.

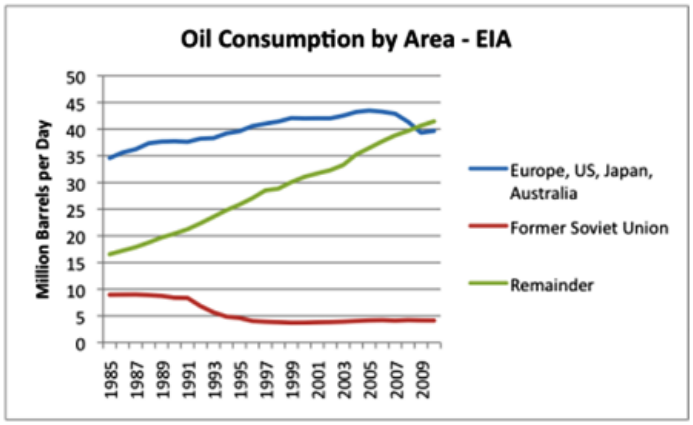


Figure 8.6: Oil Consumption per area (EIA, 2011)

[1]OECD stands for Organization for Economic Cooperation and Development, and comprises Australia, Austria, Belgium, Canada, Denmark, Finland, France, Germany, Greece, Ireland, Italy, Japan, Korea, Luxembourg, Netherlands, New Zealand, Portugal, Singapore, Spain,

The cost of production (including cost of extraction) cannot be averaged easily since the marginal cost of an oil barrel varies among producers and among different oil fields belonging to the same producer, due the different characteristics of the fields and also of the oil extracted itself. What can be said, however, is that over time the production of oil for a specific field rises up to a peak of production, before declining (Horsnell et al., 2008): as shown in Fig. 8.7 this follows a bell-shaped curve.

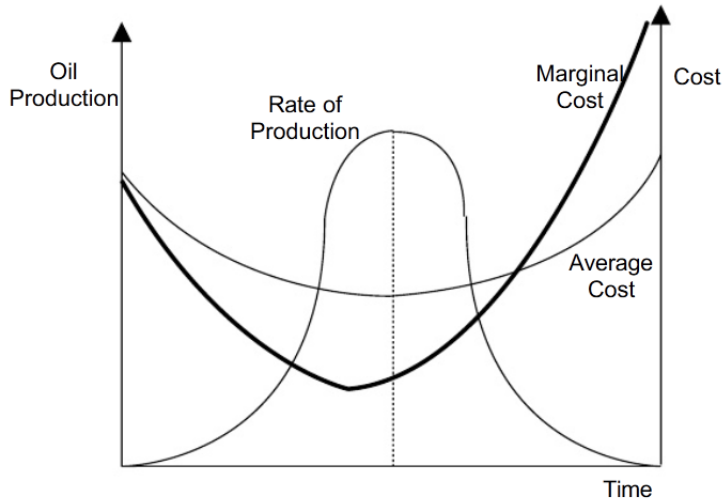


Figure 8.7: Oil production for a field, combined with the marginal cost and average cost (Horsnell et al., 2008)

After the peak of production has been reached, it will be necessary to use more energy in order to carry on extracting oil. Assuming that the marginal cost of oil for a specific well will increase in time (N. Hanley, 2007), this at some point

Sweden, Switzerland, United Kingdom, and United States. Non-OECD countries are Argentina, Bangladesh, Brazil, Bulgaria, Chile, China, Colombia, Egypt, Hong Kong SAR, Hungary, India, Indonesia, Malaysia, Mexico, Pakistan, Peru, Philippines, Poland, Romania, South Africa, former Soviet Union, Taiwan Province of China, Thailand, and Turkey. The oil-exporting countries comprise Algeria, Islamic Republic of Iran, Kuwait, Norway, Qatar, Saudi Arabia, United Arab Emirates, and Venezuela.

will reach market price, and at that point (or shortly before) it will be necessary to close the well. When the price of oil will be high enough to make extraction profitable, then the well the be reopened (Adelman, 1990).

8.3.2 Research and Development by Oil Companies

In Fig. 8.8 the possibilities for expanding production capacity for heavy oil and other forms of unconventional oil are shown. Thus there are still many untapped resources, and the question will be the amount of R&D (research and development) required in order to make extraction cost-effective. Of course a high oil price on the market would be a strong incentive for such explorations.

Graphs of R&D spending by natural resources companies are shown below and they include:

- Classification of Fortune Global 100 companies (list of companies ordered by highest revenue) by industry type: Natural resources companies, including oil companies, comprise 24% of all 100 companies (Fig 8.9).
- R&D expenditure by segment: Expenditure on R&D by natural resources companies appears to be only 4% (Fig. 8.10).
- R&D intensity: The ratio of R&D expenditure and revenue is 0.37%, which is below industry average (Fig. 8.11).
- R&D expenditure history for oil companies: There is a general increase in investment on R&D for oil companies. High oil prices appear to correspond to larger investment, as expected (Fig. 8.12).

In general it appears that oil companies invest in R&D less than industry average, and predictably tend to increase their investment when oil prices are higher.

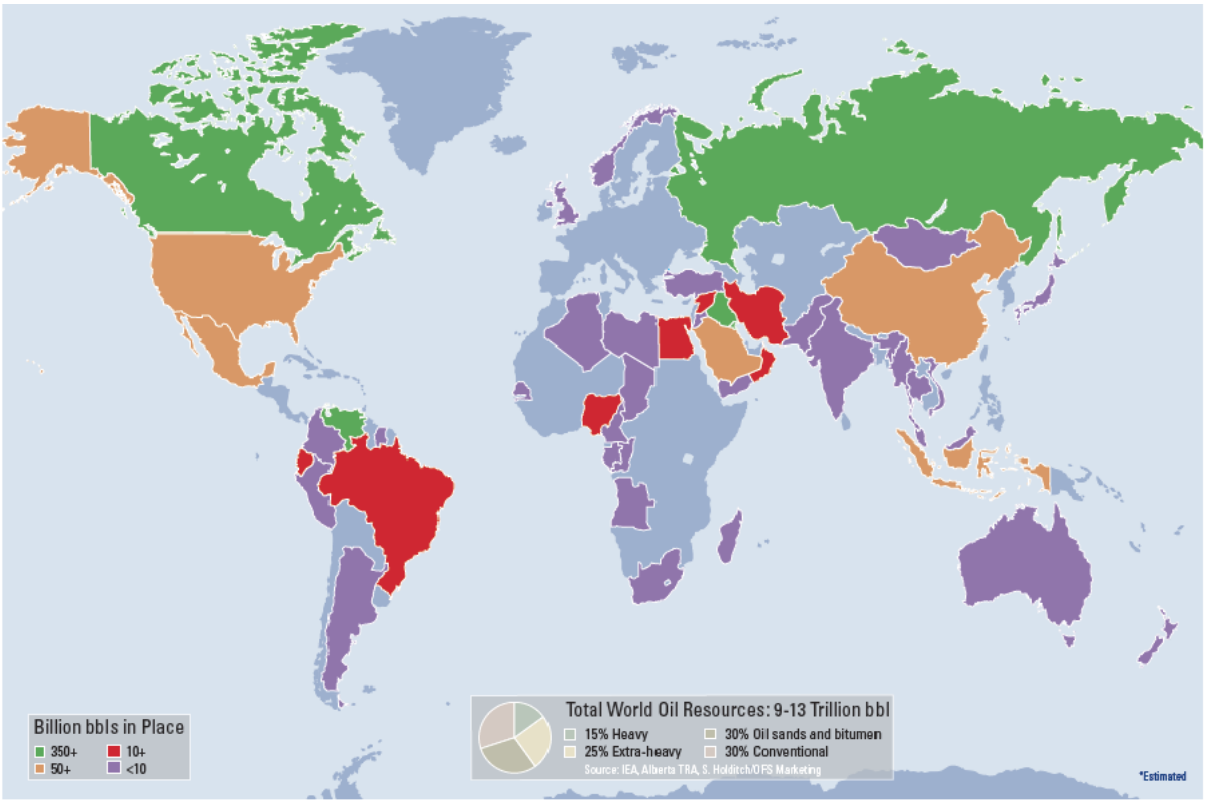


Figure 8.8: Estimated world oil resources (Shafei, 2011)

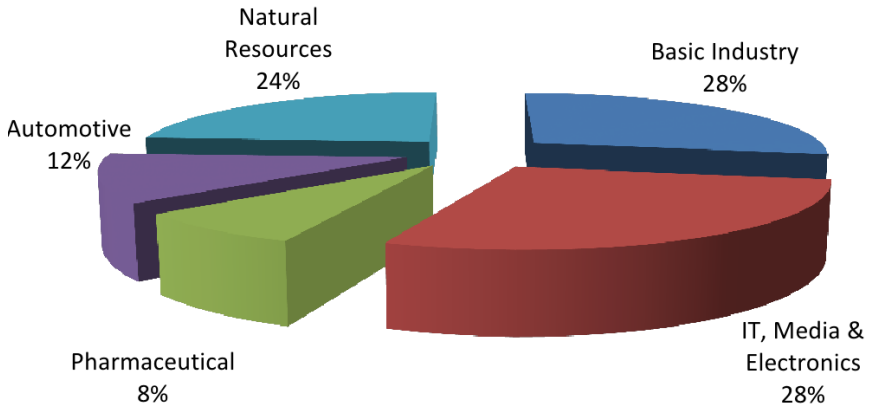


Figure 8.9: Classification of Fortune Global 100 companies by industry type (Shafei, 2011)

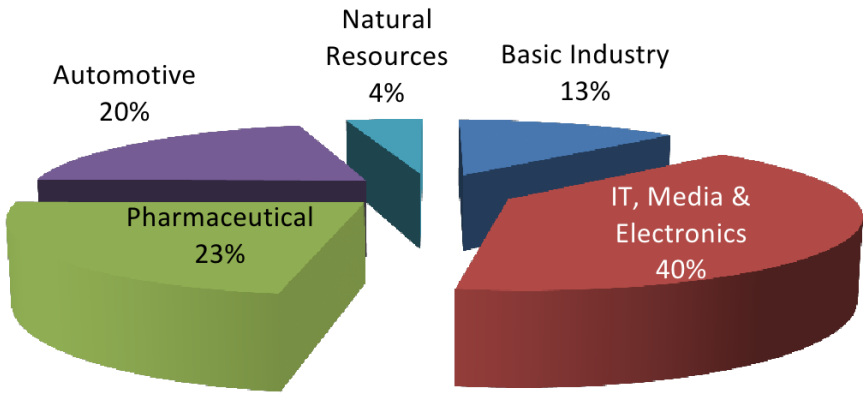


Figure 8.10: Research and development by industry segment (Shafei, 2011)

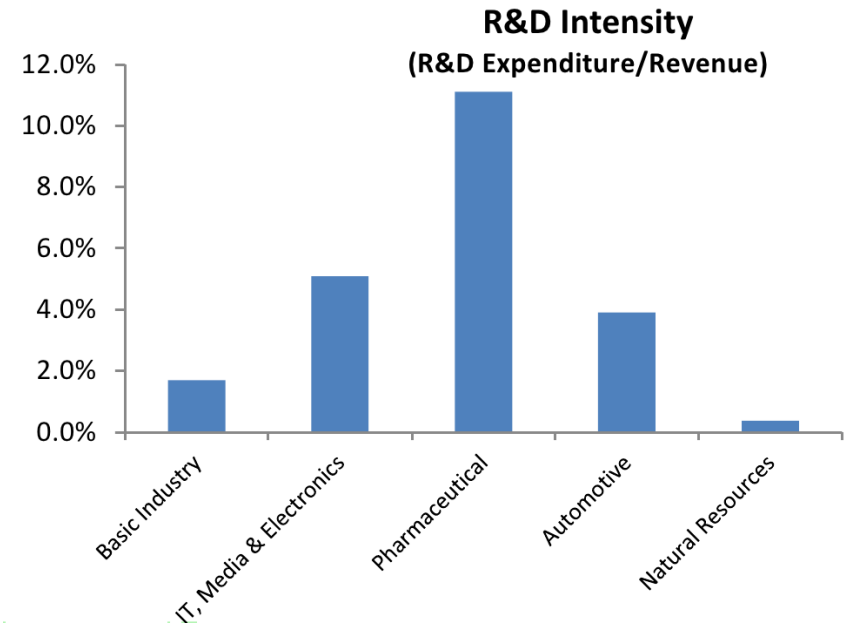


Figure 8.11: Research and development intensity (Shafei, 2011)

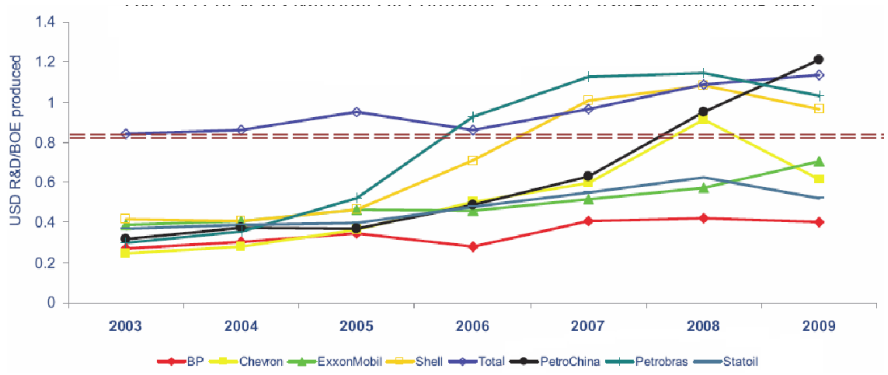


Figure 8.12: Research and development expenditure history for oil companies (Shafei, 2011)

The current innovations support decisions to increase investment in R&D. It may be observed that such investments may have been only a small part of the total expenditure for oil companies, but as recent events have shown, the consequences of just a few emergencies can lead to massive payouts and liability costs, plus maintenance costs may be much higher than anticipated, therefore it is imperative that no effort is spared in the initial design and investigation process which is part of research and development. In the long run this shift will lead to large cost savings.

8.4 Slugs

8.4.1 Slug types

In the current project, attention has been focussed on hydrodynamic slugs, as these are notoriously difficult to predict. There are four types of slugs encountered in pipelines, these are summarised in Table 8.3, along with a comment on associated modelling considerations. Hydrodynamic slugs are notoriously difficult to predict, and when they arise they have significant consequences on pipeline operation. The problems associated with hydrodynamic slugging can be very costly. Research into hydrodynamic slugging could offer a significant contribution to oil industry.

Slug Type	Description	Considerations
Terrain slugging	Caused by changes in height of the pipeline	Easily predicted
Pigging slugging	Created artificially to push liquid out (pigging)	Easily predicted
Severe slugging	Occurs in risers due to liquid accumulated at the bottom of the riser	Difficult prediction - high costs
Hydrodynamic slugging	Caused by Kelvin-Helmholtz instabilities (velocity shears)	Difficult prediction - very high costs and can happen in a variety of situations

Table 8.3: Types of slugs, descriptions and considerations

8.4.2 Economic considerations

It is estimated that each year losses of up to $6 billion are incurred by the oil industry due to slugging problems in pipelines (Sathananthan, 2007), therefore it is of crucial importance to be able to predict slug formation onset and evolution.

There are already some methods used to mitigate the effects of slugging in pipelines. One such method tackles slugging affecting the riser pipe leading to the topsides equipment, which can lead to a violent event with significant damage. By placing an 'n' shaped section just prior to the riser, the momentum of the rapidly moving liquid can be reduced in a controlled way, thus preventing slugging damage (BP patent WO2007/034142). But an improved understanding

of the slugging phenomenon would allow for a more general approach, with the possibility of more generalised applications.

There have been two outputs from the research:

- New product development: one-dimensional code EMAPS for multiphase flow calculations.
- Methodological innovation: combination of use of 1D code, 2D computational fluid dynamics (CFD) commercial code and 3D CFD commercial code.

Using the tools provided will allow:

- A better understanding of the provisioning required in order to counteract slugging
- Making an informed decision of the choices available (slug catcher, control of flow rates and so on)
- Designing of the slug catcher
- Designing of pipe (see next section).

8.5 Pipe design

The slug frequency calculations could be used to redesign the mass flow rate settings, slug catchers and/or pipe sections. Small changes in flow rates can have large effects on the production process. Design of the process can also constrain designs. It is important to balance capacity of pipe and demand, taking into account constraints set by the slugs.

Applying the principle that process design should be done for variety and volume, this is an example where design can be thought of as a way to prevent slugs of various slug lengths and frequencies to reach the outlet of the pipe.

Unlike a factory, in a pipe it can be assumed that the *throughput rate* is the same as the *throughput time*, as in general there should be no loss of mass. Moreover there is no significant transformation of material. However we could use the idea of work-in process but as unit of slugs that will have to be controlled and/or reduced, and in this case utilisation would be the proportion of time that is used to carry out slug catching/reducing. Moreover environmental issues may also affect the design processes and may have legal implications.

As discussed earlier, the volume-variety effect can have an important contribution on the process design, but in general no process design is suitable for all types of operations, therefore a compromise will have to be made, also keeping mind the total costs involved. It is difficult to give exact figures or even estimates due to the limitation of available data.

Specifically the tools derived from this project can be used in a variety of ways, including but not limited to:

- Carrying out a parametric study using EMAPS on the effect of different diameters
- Carrying out a parametric study using EMAPS on the effect of materials with different pipe wall roughness
- Carrying out investigations using EMAPS on the effect of a change of diameters during slug flow
- Designing a complex pipeline system, use EMAPS for long section and couple with 2D CFD for bends and/or connections

- Extending the complex study for varying flow rates and different fluid viscosities.

8.6 Market analysis

Due to the size of the losses currently incurred by oil companies with the slug problem, different approaches have been tried and also new software tools are being developed. It is very likely that at BP a variety of software is being trialled and tested in order to compare the results: that will include commercial CFD products and also one-dimensional codes.

Regarding one-dimensional codes, there are no figures publicly available, and due to lack of data estimates at this level would be misleading.

With the increased availability and decreasing price of parallel computing power, and with ever increasing costs of physical experiments, it can be safely presumed that for more future projects a simulation of the relevant problem will be considered at least an important part of the solution. Therefore the market of commercial codes regarding fluid flow will definitely increase in the near future, probably even faster than the average figure of 13% per year which has been found for the past few years.

There are no exact figures for the total market share for commercial CFD products, but it has been estimated to be around $700M (Hanna, 2011), keeping in mind that in 2009 it was known to be around $650M and assuming a compound average growth rate of 13% per year. Commercial CFD code FLUENT (part of ANSYS since 2006, and combined with CFX, acquired by ANSYS in 2003) is believed to be worth around 45% of the total market. There are other commercial CFD codes available, including Star-CCM+ (part of CD-adapco) and in the future the

combination of proliferation of more commercial codes and also some open-source software may change the balance.

Specifically regarding parallel computing, it is now possible to have easy access to large parallel clusters and pay rates depending on use - these infrastructures are already well-established (eg. Amazon EC2) and this reduces the need for large initial investments specifically aimed at parallel computing. Therefore the cost of uptake is again minimal, even if no pre-existing in-house infrastructure is present. Regarding the positioning of the current innovations in the context

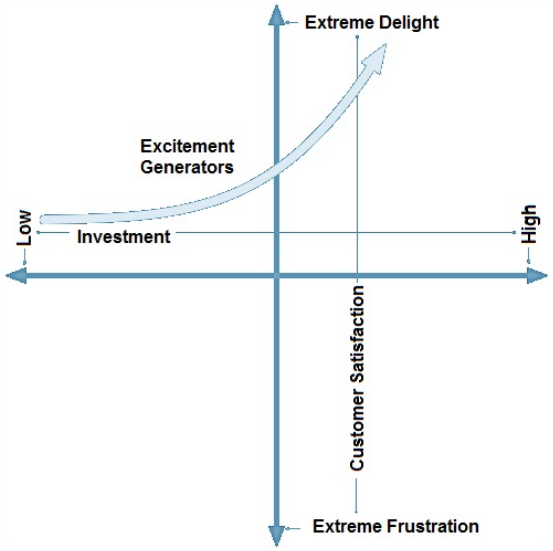


Figure 8.13: Position of the innovative tools in the Kano Model diagram.

of a Kano model (Ullah and Tamaki, 2011), it is expected (or hoped!) that the products are excitement innovators (Fig. 8.13), since it is well-known that hydrodynamic slug flow is notoriously difficult to simulate, and therefore a new framework composed of a parallelised 1D code with the addition of numerical enhancement and the new methodology of coupling with 2D code, combined with

low investment and uptake costs, should be a very welcome set of innovations.

8.7 Value chain

The range of activities starting from conception of an innovative tool up to its delivery is described by a value chain (Campbell, 2008). A value chain has both structural and dynamic components, and here a brief analysis will be given relevant to the current project. Starting from the structure, it is possible to identify the following:

- *End markets*: Here they are represented by BP but it could be applied also to other oil companies facing the same issues (slug flow) and even companies that are working hand in hand with oil companies to tackle slug flow.

- *Business enabling environment*: As part of the environment at the beginning of the project it was important to agree on the terms (including legal issues and disclosure agreements) that would then be mutually binding for the entire project duration. Sometimes there is the risk that such an environment may stifle innovation, but here there was reciprocal communication and few limitations otherwise.

- *Vertical linkages*: There is a direct vertical linkage with BP since they are the final recipient of the innovative tools. The relationship could be described as mutually beneficial, and also with knowledge transfer having taken place. There were however other forms of vertical linkage, including linkage with commercial software, hardware and service (parallel cluster) providers.

- *Horizontal linkages*: There were horizontal linkages with other universities including Imperial College. But also the relationship with BP itself could be described as having a property akin to a horizontal linkage because of the exchange of information and the environment that facilitated collaboration.

- *Supporting markets*: The TMF consortium and collaboration with other members could be considered as an example of supporting market that leads more value and facilitates the whole process.

Moving on to the analysis of the dynamic factors, the following elements can be identified:

- *Value chain governance*: It is important to have power and ability to exert control along the chain, as relationships may change with time. In this case the flexibility of the innovation tools makes them easily adaptable to new requirements should they arise, or new clients.

- *Inter-firm relationship*: The relationships developed with Imperial college and other members of the TMF consortium have been recurrent and mutually beneficial (in this case without financial exchanges).

- *Upgrading*: The opportunities for firm-level upgrading are multiple, and in this case by creating a flexible parallel computing framework with versioning control for the 1D code and also the combination of 2D CFD and 1D coupling, the tools are in place to facilitate upgrading.

The main elements of the value chain for the product have been explained above, and some of the complex underlying relationships and processes are outlined, this helps to put the process of research and development into an industrial context.

8.8 Performance characteristics

The 1D code EMAPS does not aim to directly compete with other major codes, instead it concentrates on the niche of slug flow prediction, even though it is capable of simulating other flows as well. It is a niche market because it is a particular segment of the (multiphase flow) market, and it is focussed on obtaining accurate and quick solutions to slug flow problems. Also there are not many other codes available that can provide similar capability. The characteristics that will make it attractive for BP and other oil companies facing the slugging problem are as follows:

- Proven track record of successfully simulating slugs in pipes: slug frequency predicted with an average discrepancy of 7%
- Tested on experimental data, including data provided by BP
- Adaptivity implemented: only when necessary, precision in calculations will be increased
- Highly parallelised, can be used on any shared memory architecture
- Able to simulate very long pipes - tested up to 100 KM
- Flexible: new physical/mathematical models can be added to the existing code

Use of CFD commercial software FLUENT is already widespread, and in this project a new methodology was proposed, which allows the use of two-dimensional (2D) CFD instead of the more standard three-dimensional simulations. Particular techniques have to be followed in order for 2D CFD in order to obtain slugs in a channel simulation, and these are explained in detail. The characteristics of using 2D CFD combined with the new techniques are as follows:

- On a sample of 92 experimental cases, 57% of slug frequencies showed discrepancies of less than 20%
- Up to ten times faster than 3D CFD
- Can be used as an investigative tool and also as a predictive tool
- Can be used to predict flow in simple geometry cases
- Can be used as an initial tool to estimate flow properties in more complex cases
- Can measure properties that then can be put back in 1D code in order to carry out calibrations
- Flexible and fast
- Valid substitute of 3D CFD in certain cases

8.9 Cost implications

The research project cost BP £8,500 per year for the duration of four years. To put that in perspective, BP spent a total of $780 million in 2010 in research and development. The implementation costs for BP will consist of man-hours required to learn to use to software involved, estimated at not more than one to two months, and the use of parallel infrastructures. On the other hand, it is not possible to make an accurate assessment of the likely financial return for BP, as it will depend on their speed in implementing solutions based on the predictions using EMAPS and/or 2D CFD, and too many unknown factors are involved. For the same reason, an accurate sensitivity analysis is not possible either at this stage. But it is clear that solving even a small percentage of the slugging problem

will bring major benefits to BP regarding oil flow, and this will bring a competitive advantage compared to other companies that will not have implemented those changes yet. It could also lead BP to create further patents based on results from EMAPS and then sell them on to other companies. Therefore the uptake of the innovations from this project should lead directly to decreased production costs due to better knowledge of the provision required to deal with hydrodynamic slugs, and it should encourage BP to invest in research that tackles engineering issues at the source and enables cost reduction and consequently increase in profit.

8.10 Conclusions

In this project methods based both on one-dimensional modelling and also on two-dimensional modelling are presented, together with comparison with full fledged three-dimensional simulations. Validation for all models are carried out and the associated limitations are also noted. Users are given different choices for different scenarios, depending on the approach required.

Oil companies and/or companies involved in pipe design can decide to undertake case and feasibility studies by choosing one of the various methods discussed in the current thesis. These methods range from fast and flexible approaches to more precise but also more time consuming analyses. The 1D code EMAPS will benefit BP directly by allowing long pipes to be simulated in parallel environments, in the confident knowledge that slug flow will be correctly predicted within the well-posedness region of the two-fluid model. A 2D CFD commercial code can also be used, combined with particular techniques, as an investigative and predictive tool for simple geometry cases. The benefits of using these two approaches are explained in the context of a market analysis. Uptake costs for BP are very low, as the solutions proposed are software based, use can be made of short-term

consultancies in order to help achieve a quick implementation, and hardware requirements for implementation can be fulfilled by taking advantage of the clusters available for hire. Moreover the total project cost is very small compared both with the losses incurred due to the slugging phenomenon and also compared with the total expenditure in R&D by BP.

References

Adelman, M. A. (1990), 'Mineral Depletion, with Special Reference to Petroleum.', *The Review of Economics and Statistics* **72**(1), 1–10. (cited at page 213)

BP (2006), Prudhoe Bay Fact Sheet, Technical report, BP. (cited at page xx, 204, 206)

BP (2010), BP Annual Report and Form 20-F 2010, Technical report, BP. (cited at page xxiv, 204)

Campbell, R. (2008), The Value Chain Framework Briefing Paper, Technical report, United States Agency for International Development. (cited at page 223)

Christiano, L. J. and Fitzgerald, T. J. (2003), 'The band pass filter', *International Economic Review* **44**(2), 435–465. (cited at page 210)

EIA (2011), Petroleum reports, Technical report, U.S. Energy Information Administration. (cited at page xx, 208, 209, 211)

Hamilton, J. D. (2008), Understanding crude oil prices, Technical report, Department of Economics, University of California, San Diego. (cited at page 210)

Hanna, K. (2011), Back to the Future - Trends in Commercial CFD, *in* 'NAFEMS World Congress', Boston, USA. (cited at page 221)

Horsnell, P., Jacazio, C., Norrish, K., Sen, A. and N. Snowdon, Y. Y. (2008), Barclays Capital Commodities Research, Technical report, Barclays Capital. (cited at page xx, 212)

IMF (2011), World Economic Outlook April 2011: Tensions from the Two-Speed Recovery Unemployment, Commodities and Capital Flows, Technical report, International Monetary Fund. (cited at page xxiv, 210, 211)

N. Hanley, J. F. Shogren, B. W. (2007), *Environmental Economics in Theory and Practice*, 2nd edn, Hampshire, Palgrave Macmillan. (cited at page 212)

Rickard, S. (2006), *The economics of organizations and strategy*, McGraw-Hill Companies, London. (cited at page 208, 209)

Sathananthan, R. (2007), Ensuring effective long distance subsea tie-backs in deep & ultra deepwater environments, The Prince Hotel, Kuala Lumpur, Malaysia. (cited at page 218)

Shafei, K. (2011), The role of R&D in the oil and gas industry, Technical report, KAPSARC. (cited at page xx, 214, 215, 216)

Smith, M. Q. (1996), Where the Pipeline Meets the Permafrost, Technical report, Technology Today, Southwest Research Institute. (cited at page xx, 206)

Sorrell, S., Miller, R., Bentley, R. and Speirs, J. (2010), 'Oil futures: A comparison of global supply forecasts', *Energy Policy* **38**(9), 49905003. (cited at page 207)

Ullah, A. M. M. S. and Tamaki, J. (2011), 'Analysis of Kano-model-based customer needs for product development', *Systems Engineering* **14**(2), 154–172. (cited at page 222)

Wheatcroft, P. (2010), 'The next crisis: Prepare for peak oil', *The Wall Street Journal* . (cited at page 207)

Chapter 9

Research Conclusions

This chapter summarises the overall conclusions from this research. These are presented following the aim and objectives as outlined in chapter 1. Practical recommendations are also given.

9.1 Conclusions of 1D code EMAPS

A new version of EMAPS was successfully created by porting versions from divergent branches into a single one. All models, together with geometry capability (inclined/horizontal/vertical pipes) and adaptive mesh refinement were migrated to Sub-version (SVN), a newer software versioning and revision control system, combined with a continuous integration environment (cruise control). After users submit their changes (integration) back to the server, an automated build of the software and a complete run of a newly automated test suite is completed.

All present models in EMAPS have been parallelised, and instructions on how to parallelise any future models are also provided, with minimal knowledge of parallelisation required for any future user. A full test suite has been written, both

for the sequential and the parallel code. Tests are automatically run whenever a code change is submitted. The efficiency of the parallel version may however vary from one model to another because of the details of the model implementation. Some parts of the parallel part of the code are highly dependent upon the grid size and other parameters determining the time the simulation runs, and it is possible that at a certain stage the efficiency of a parallel run will have reached its maximum with the chosen number of cores.

The speed-ups have been best for single pressure model, and even more so for long pipes and/or long simulation times. Further speed-ups were observed when adaptive mesh refinement was used. The parallelisation of EMAPS has enabled it to be used on long pipes (tested up to 100KM), which normally would have required an inordinate amount of time to run.

The journal publication of the above procedures (Kalogerakos et al., 2012d) can also be useful to anybody who is working on a code using iterative schemes and grid that wishes to parallelise it. The main issues faced and the reasons for the various approaches are given and it is quite likely that another project will face similar issues when parallelising.

Simulations with good agreements were carried out using the single pressure model on data from Imperial College (Manolis, 1995) and field data (X and R pad) from BP (Hill and Turner, 1988). It has to be kept in mind that EMAPS may not work outside the region of well-posedness of the two-fluid model, as the equations will become ill-posed. This is an important limitation which affects any one-dimensional code based on the two-fluid model.

9.2 Conclusions of use of velocity profile coefficients

The original implementation in EMAPS assumed a velocity profile coefficient equal to 1. This overlooks the ratio that arises due the averaging over the cross-sectional area. In the model development work a number of different velocity profile coefficients from the literature were tested. Then using 3D CFD simulations, a detailed analysis of correlations between velocity profile coefficients and flow variables was carried out, including pressure, phase height, pressure gradient and combinations of the previous variables. The best correlation was found with pressure, and this was implemented in EMAPS.

Comparisons were carried out with experimental results from WASP facility in Imperial College, London (Ujang et al., 2006). Using the fitted C_V, EMAPS simulations were repeated on the same case studies, and on pipe sections from BP's Prudhoe Bay. Fittings of C_V against pressure give better results for slug frequency and slug lengths (Kalogerakos et al., 2012b). Although it is not straightforward to generalise these correlations to other ranges of flow parameters, nevertheless the use of a modified velocity profile coefficient appears to be a necessary step when using 1D multiphase codes.

9.3 Conclusions of the wave growth problem

The wave growth problem analysis, a problem with known analytical solutions, consisting of flow determined by the input of an initial sine-wave, has been carried out with EMAPS and the wave growth rate obtained compared successfully with results from 1D code TRIOMPH, Imperial College. The use of 2D FLUENT to simulate two-phase flow in a channel was initially validated by repeating the wave growth problem but this time using the volume of fluid (VOF) model, both

by using incompressible and compressible gas flow (Kalogerakos et al., 2010). A full mathematical perturbation analysis on the VOF model was also carried out, in order to validate the simulation results of 2D FLUENT (Kalogerakos et al., 2012a). The wave growth rate calculated numerically as a first approximation was close to the one measured from the FLUENT simulation. Thus the theoretical validation gave a good preliminary agreement. A more detailed analysis is now being carried out where a y-dependency of the perturbation is assumed.

9.4 Conclusions of 2D CFD

The use of 2D CFD to simulate two-phase flows in pipes (approximated by channels) gave satisfactory results for stratified flow and more importantly for transient flows, in particular slug flow. Results of mesh independence were used (Kalogerakos et al., 2010) and simulations results were compared with a regime flow map and with a set of experimental data Manolis (1995). Native 2D Fluent simulation with VOF model did not work due to the interfacial problems, in particular due to velocity differences between phases at the start. A methodology was presented here that allows this problem to be overcome, by carrying out an initial steady-state simulation till convergence or quasi-equilibrium is reached, and subsequently the transient simulation is started.

The promising results from a limited sample (Kalogerakos et al., 2011) prompted for an extension to the largest sample of slugs available (from a unique source). It was necessary to write a code that would calculate slug frequencies in an automated way, depending on the choice of threshold constants related to slug formation. Considering the size of the sample used, and also taking into account the limitations of the VOF model and the use of a channel instead of a 3D pipe, the simulation results show a good agreement with experiments (Kalogerakos

et al., 2012c). Comparison with experimental cases showed that discrepancies between CFD values of frequencies and experimental measurements were worst for combination of high gas velocities and low liquid velocities. The use of 2D CFD has also been shown previously to give good results for wave-growth in two-phase flows in straight pipes (section 6, Kalogerakos et al. (2012a)). In cases where few slugs (<10) were observed over the duration of 100s, the choice of the threshold constants has a major effect on the resulting frequency.

Discrepancies between experimental and 2D CFD simulation frequencies range from an average of 14.9% to an average of 22.9%, depending on the threshold constants chosen.

Running times for 2D simulations are approximately ten times quicker compared with 3D simulations. 2D CFD can definitely be recommended both as an investigative tool but also a predictive tool. Moreover, use of 2D simulations offers also the possibility of measuring properties that then can be put back in 1D code in order to carry out calibrations, particularly for cases where there are discrepancies. Therefore 2D CFD is flexible and fast, and can, under certain circumstances, be a valid substitute for 3D CFD. It also has a much wider applicability range compared ti 1D as it does not have the limitation of the well-posedness region.

2D CFD is an alternative to 3D CFD that should be kept under consideration, either as a tool to predict flow in simple geometry cases, or perhaps as an initial tool to estimate flow properties in more complex cases.

9.5 Practical recommendations

When trying to decide on the most appropriate method of simulating slugs, it is useful to keep in mind the following:

- For pipes longer than 100m and in the ranges allowed by the two-fluid model as shown in the flow regime map, it is advisable to use EMAPS directly. It will correctly predict the onset of slug flow and slug frequency, and it will be the quickest method.

- For pipes of length between 30m and 100m, a combination of EMAPS (depending of flow properties) and 2D CFD can be used. A decision will have to be made on whether it is important to have any detail of the actual slugs.

- For pipes smaller than 30m, the choice can be between EMAPS, 2D CFD or 3D CFD. At this stage a full 3D simulation can be carried out, although a good reason will have to be to put forward in order to prefer that over the faster 1D or 2D simulations (for example some irregularities in the input flow, the presence of a particular geometry, or well-posedness region issues).

For the above recommendation, the availability of a 16 core architecture for the CFD and a 4 core for EMAPS was assumed. Combinations of different simulation methods are also possible, as parts with "difficult" geometry can be simulated with 2D or 3D separately, and coupled with 1D for the rest of the pipe. The flexibility that is available is very important in order to obtain the most robust results in the quickest time possible, especially when time is of essence in decision making.

9.6 Future work

Future work arising from this research will include:

- Completing the y-dependency wave growth analysis.
- Extending the dataset for the velocity profile coefficient analysis.
- Carrying out an MPI parallelisation of EMAPS.
- Deriving interfacial stresses from 2D calculations and using them in EMAPS to possibly extend the well-posedness region.
- Investigating the validity of 2D CFD for annular and dispersed bubbly flow.

References

Hill, T. and Turner, P. (1988), Prudhoe Bay Western operating area multiphase flow data August - September 1987, Technical report, BP International, Research Centre. (cited at page 234)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2010), Comparison between 2-D CFD and 1-D code for wave growth simulations, *in* 'International Conference on Multiphase Flow', University of Florida. (cited at page 236)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2011), Use of 2-D CFD for simulating two-phase flows in horizontal pipes, *in* 'International Association of Science and Technology for Development, Applied Simulation and Modelling'. (cited at page 236)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012a), 'Y-dependent wave growth analysis for VOF model', *Multiphase Science and Technology* . In preparation. (cited at page 236, 237)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012b), 'Adjustment of velocity profile coefficients for one-dimensional multiphase flow', *Journal of Computational Physics* . In preparation. (cited at page 235)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012c), 'Comparison between

2D CFD and experiments for slug flow', *International Journal of Multiphase Flow* . In preparation. (cited at page 236)

Kalogerakos, S., Gourma, M. and Thompson, C. P. (2012d), 'Use of OpenMP to parallelise a one-dimensional multiphase code', *Computer Physics Communication* . In preparation. (cited at page 234)

Manolis, I. G. (1995), High Pressure Gas-Liquid Slug Flow, PhD thesis, Department of Chemical Engineering and Chemical Technology, Imperial College of Science, Technology and Medicine, UK. (cited at page 234, 236)

Ujang, P. M., Lawrence, C. J., Hale, C. P. and Hewitt, G. F. (2006), 'Slug initiation and evolution in two-phase horizontal flow', *International Journal of Multiphase Flow* **32**, 527–552. (cited at page 235)

Appendices

Appendix A

Graphs of EMAPS simulations

A.1 Graphs for EMAPS simulations of Manolis cases using C_V=1

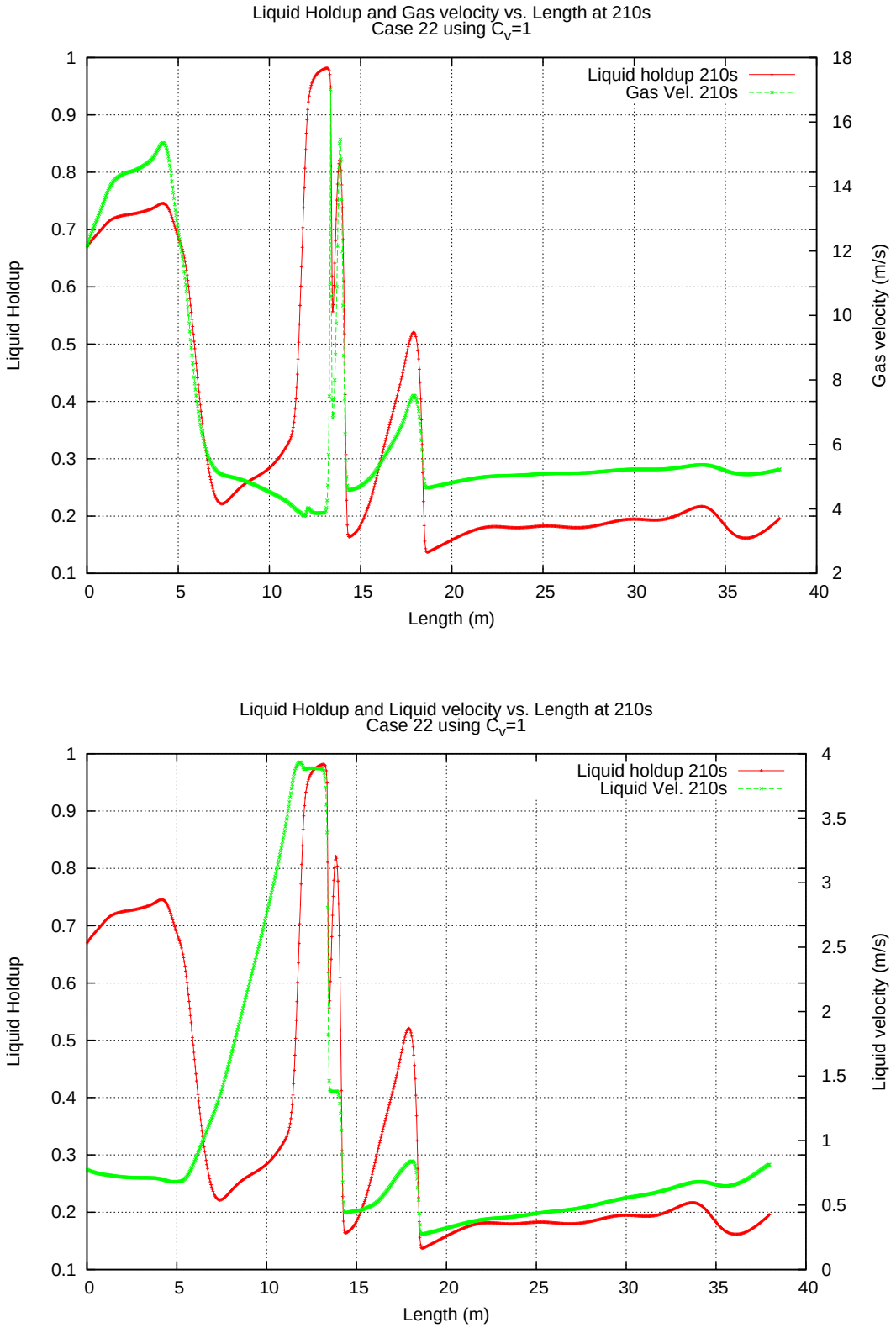


Figure A.1: Case 22 with C_V=1. Top: liquid holdup and gas velocity vs. pipe length at 210s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 210s.

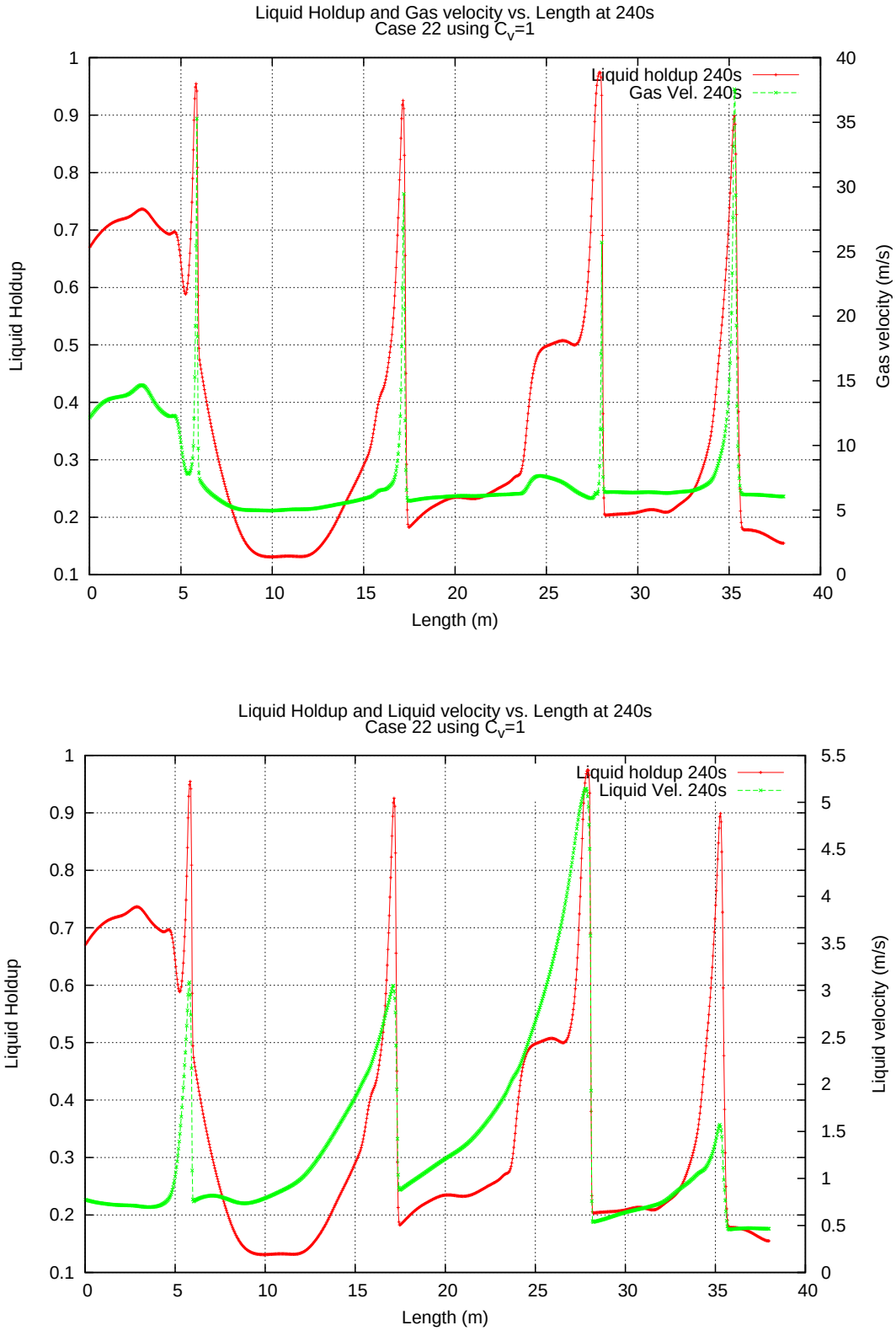


Figure A.2: Case 22 with C_V=1. Top: liquid holdup and gas velocity vs. pipe length at 240s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 240s.

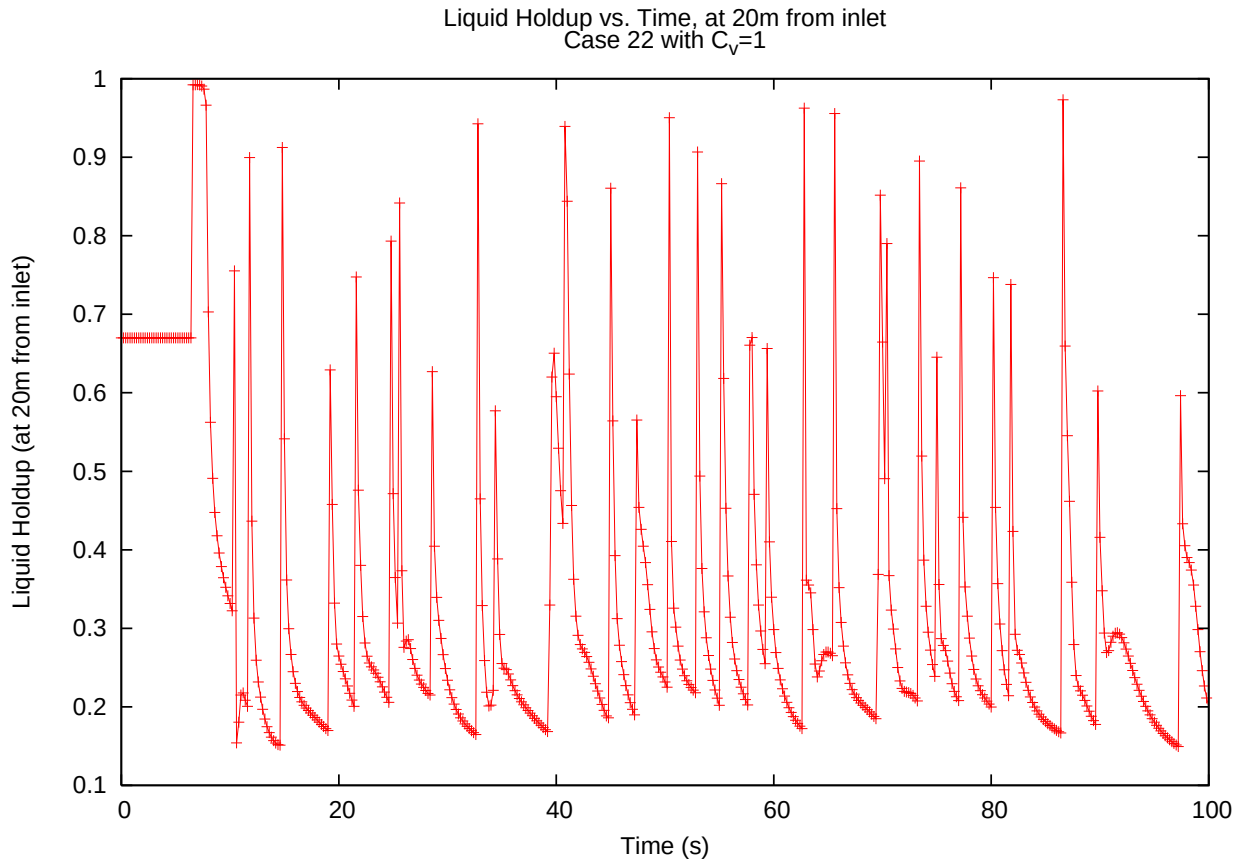


Figure A.3: Case 22 with C_V=1. Liquid holdup vs. time, measured at 20m from inlet.

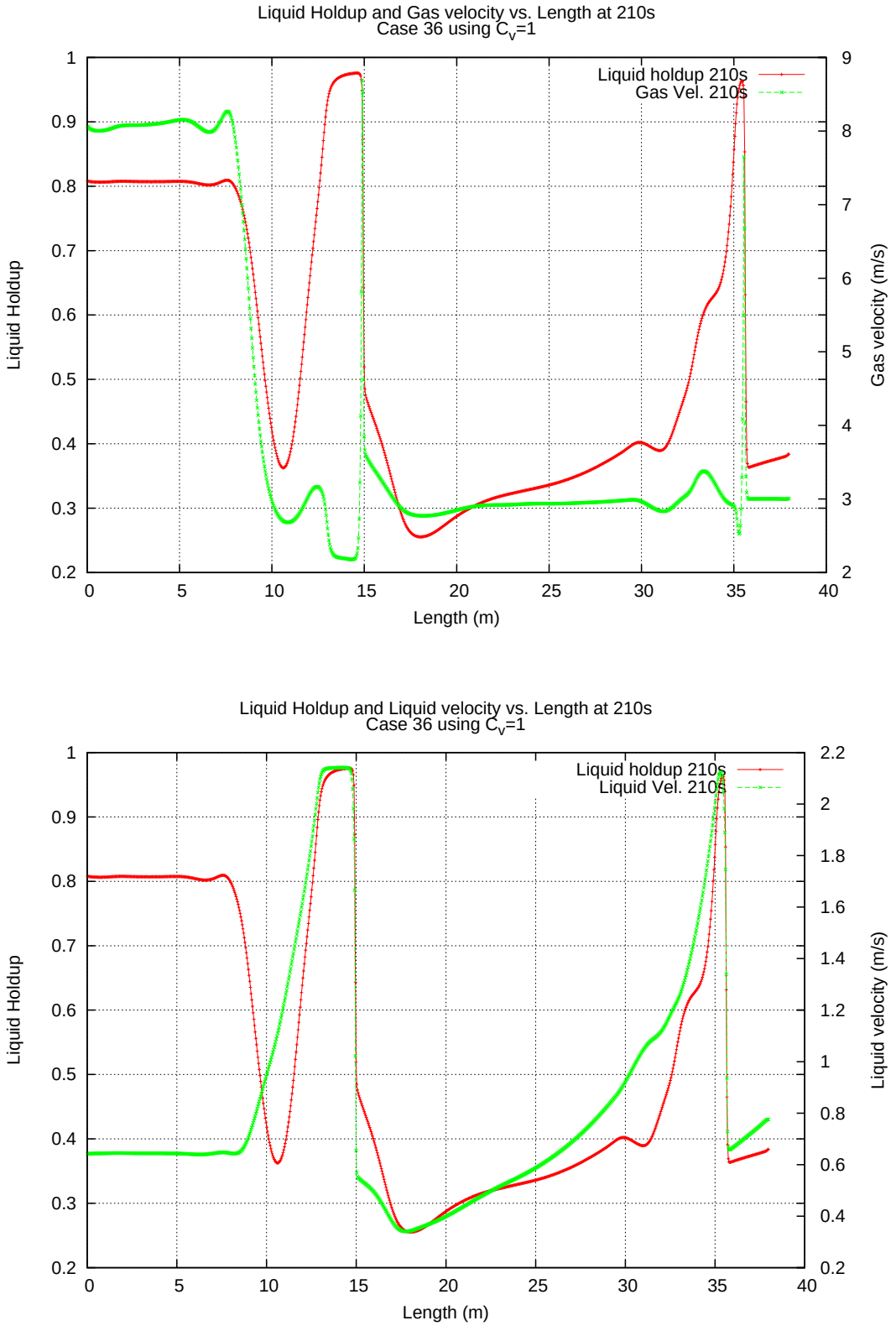


Figure A.4: Case 36 with C_V=1. Top: liquid holdup and gas velocity vs. pipe length at 210s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 210s.

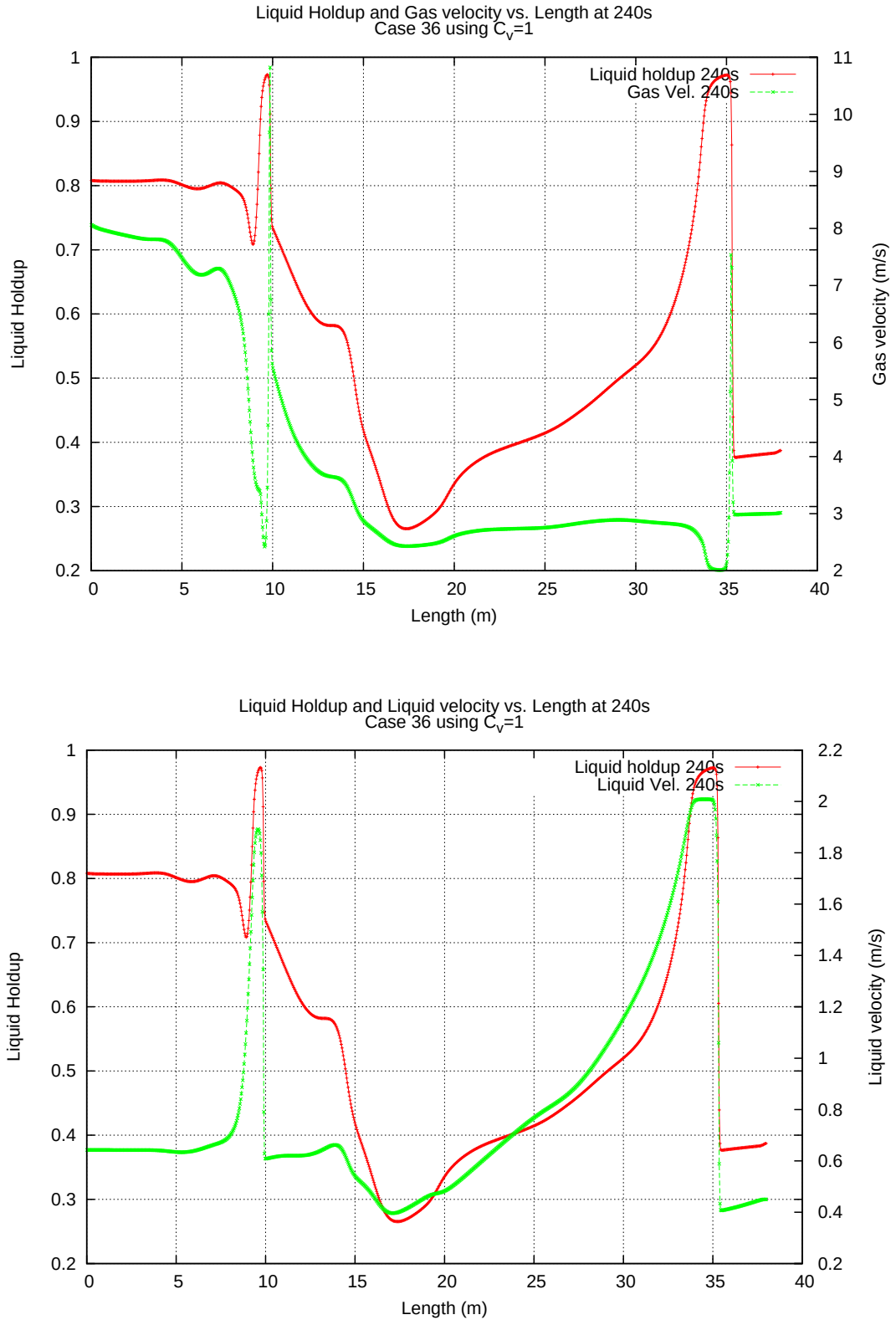


Figure A.5: Case 36 with C_V=1. Top: liquid holdup and gas velocity vs. pipe length at 240s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 240s.

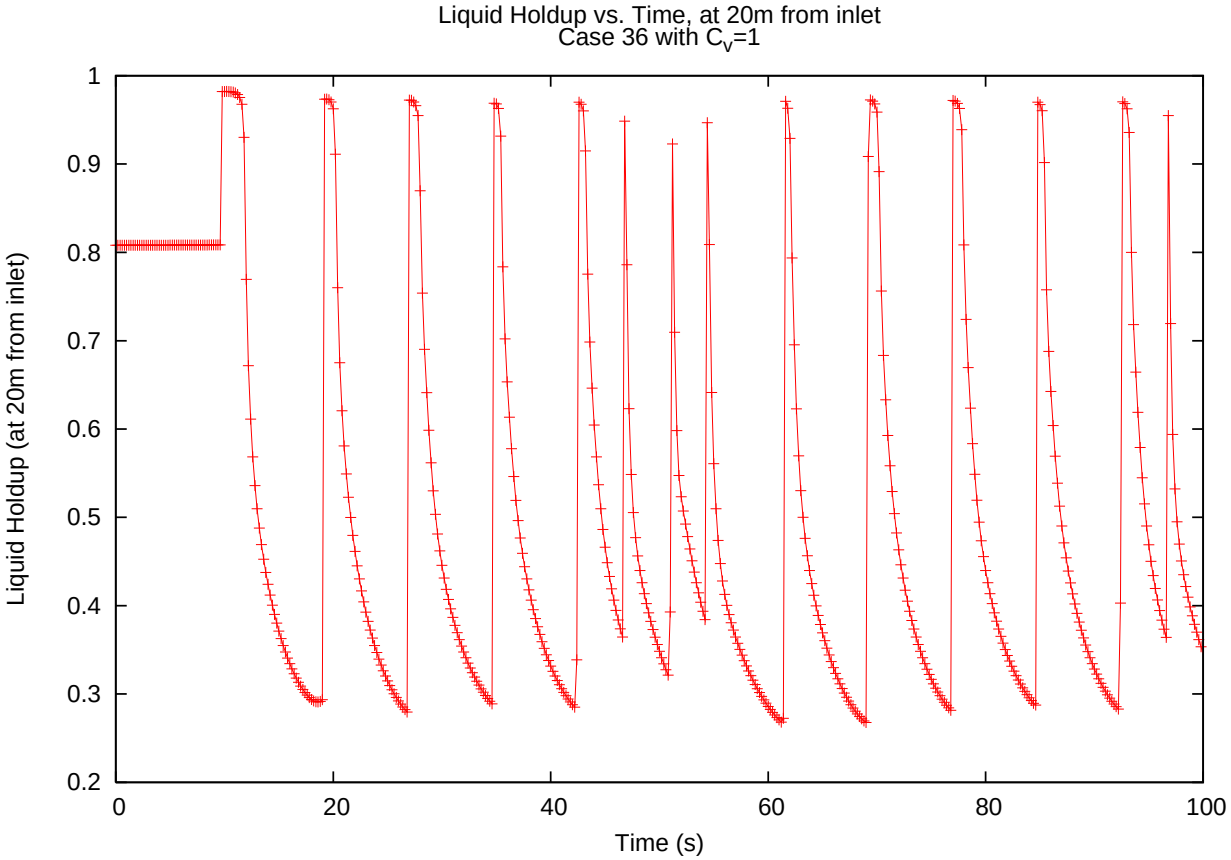


Figure A.6: Case 36 with C_V=1. Liquid holdup vs. time, measured at 20m from inlet.

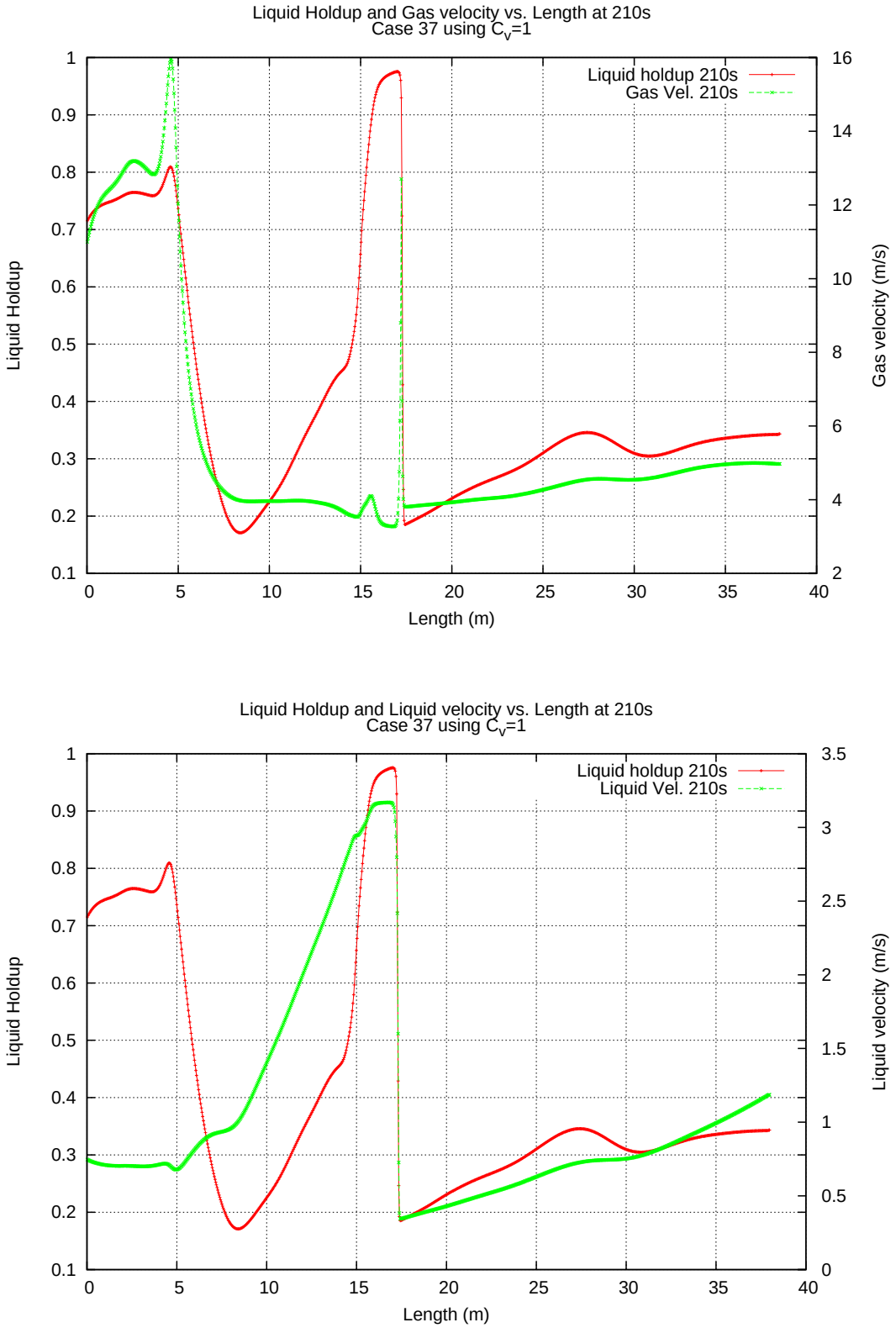


Figure A.7: Case 37 with C_V=1. Top: liquid holdup and gas velocity vs. pipe length at 210s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 210s.

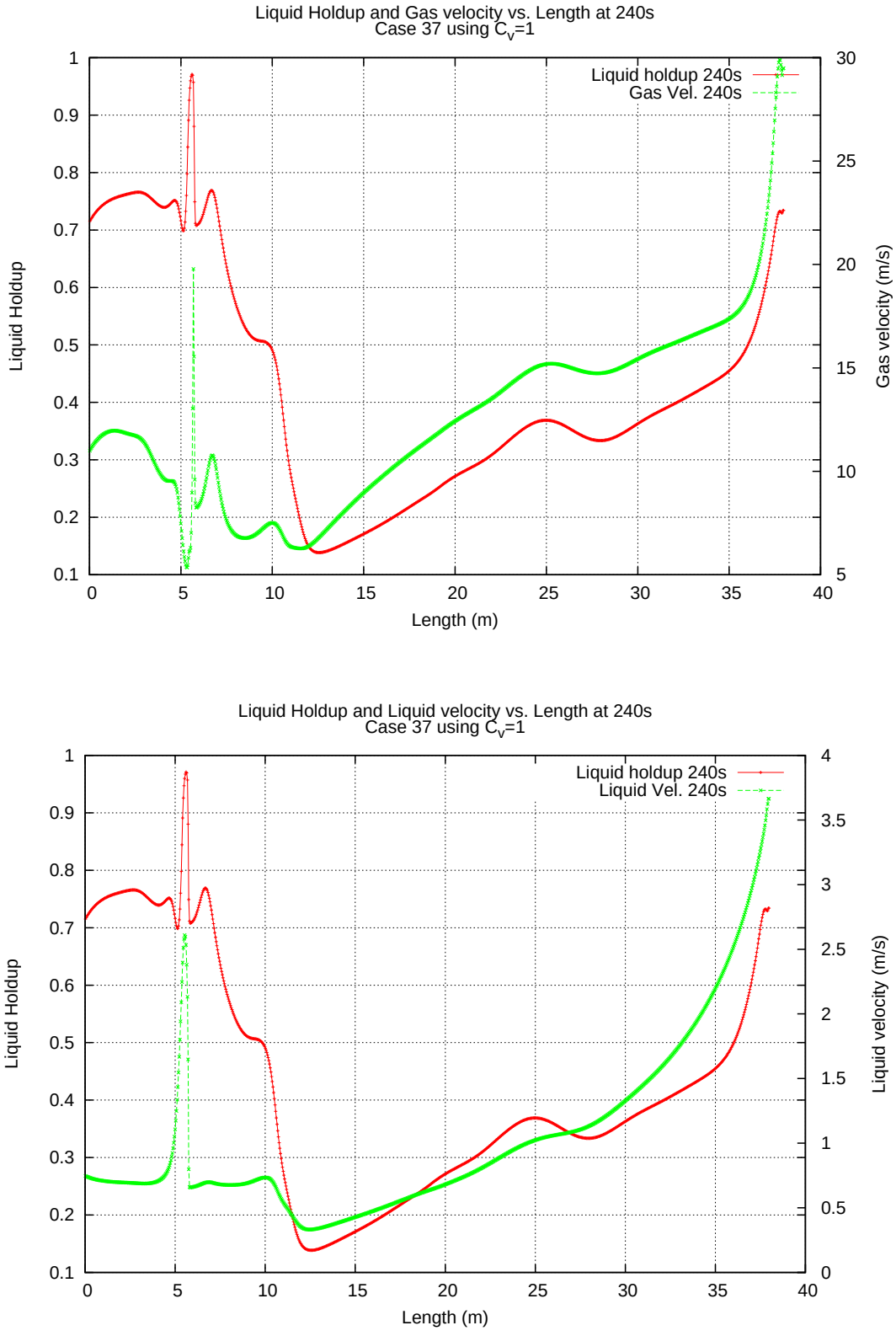


Figure A.8: Case 37 with C_V=1. Top: liquid holdup and gas velocity vs. pipe length at 240s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 240s.

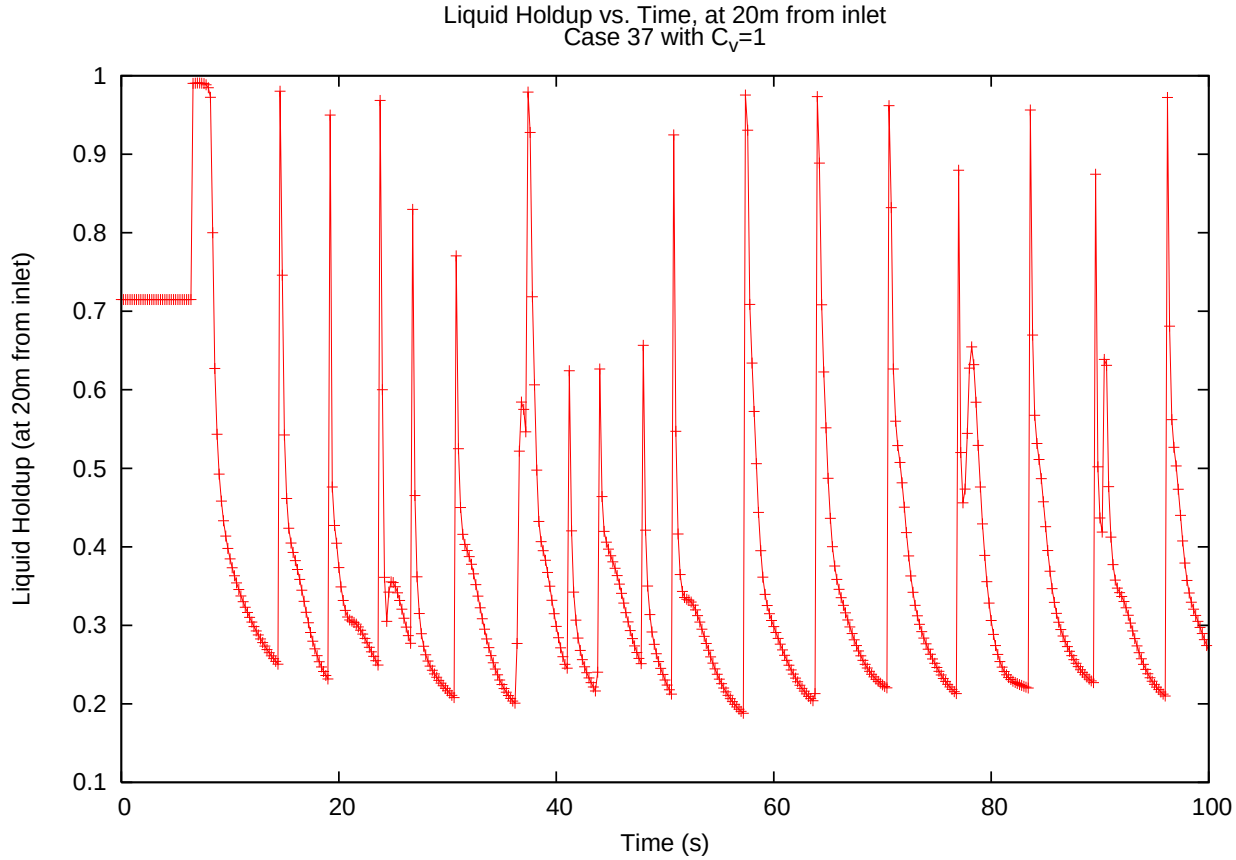


Figure A.9: Case 37 with C_V=1. Liquid holdup vs. time, measured at 20m from inlet.

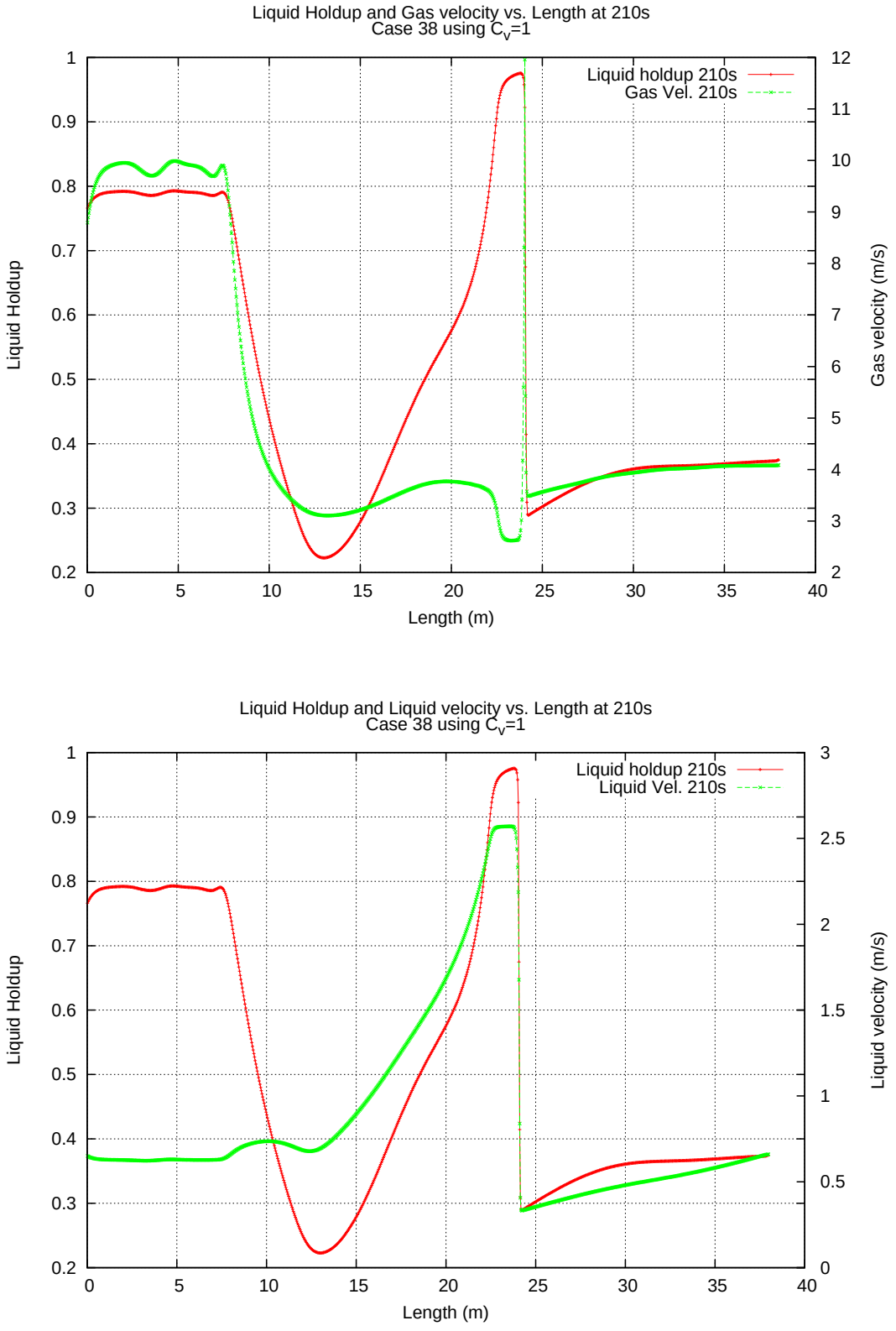


Figure A.10: Case 38 with C_V=1. Top: liquid holdup and gas velocity vs. pipe length at 210s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 210s.

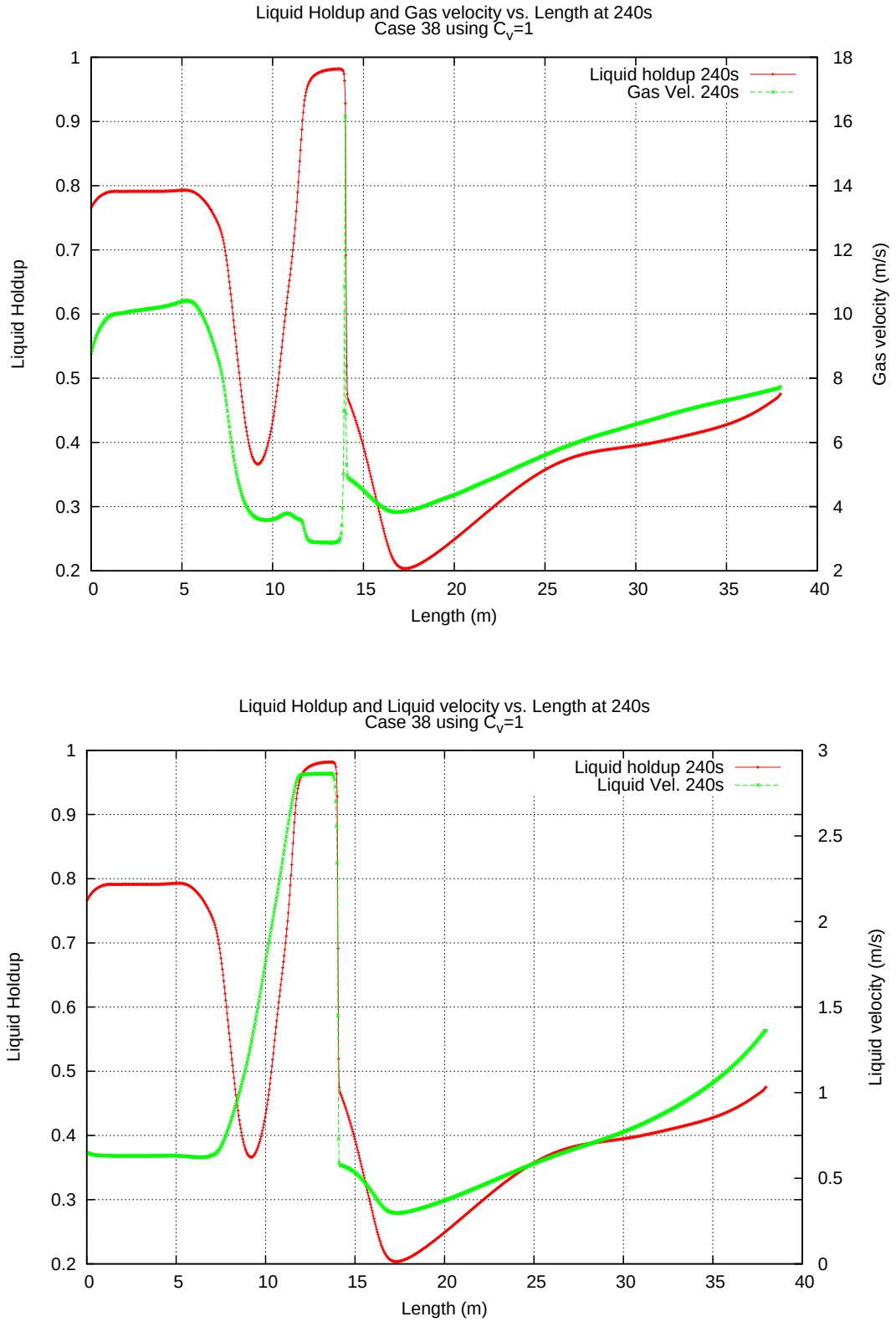


Figure A.11: Case 38 with C_V=1. Top: liquid holdup and gas velocity vs. pipe length at 240s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 240s.

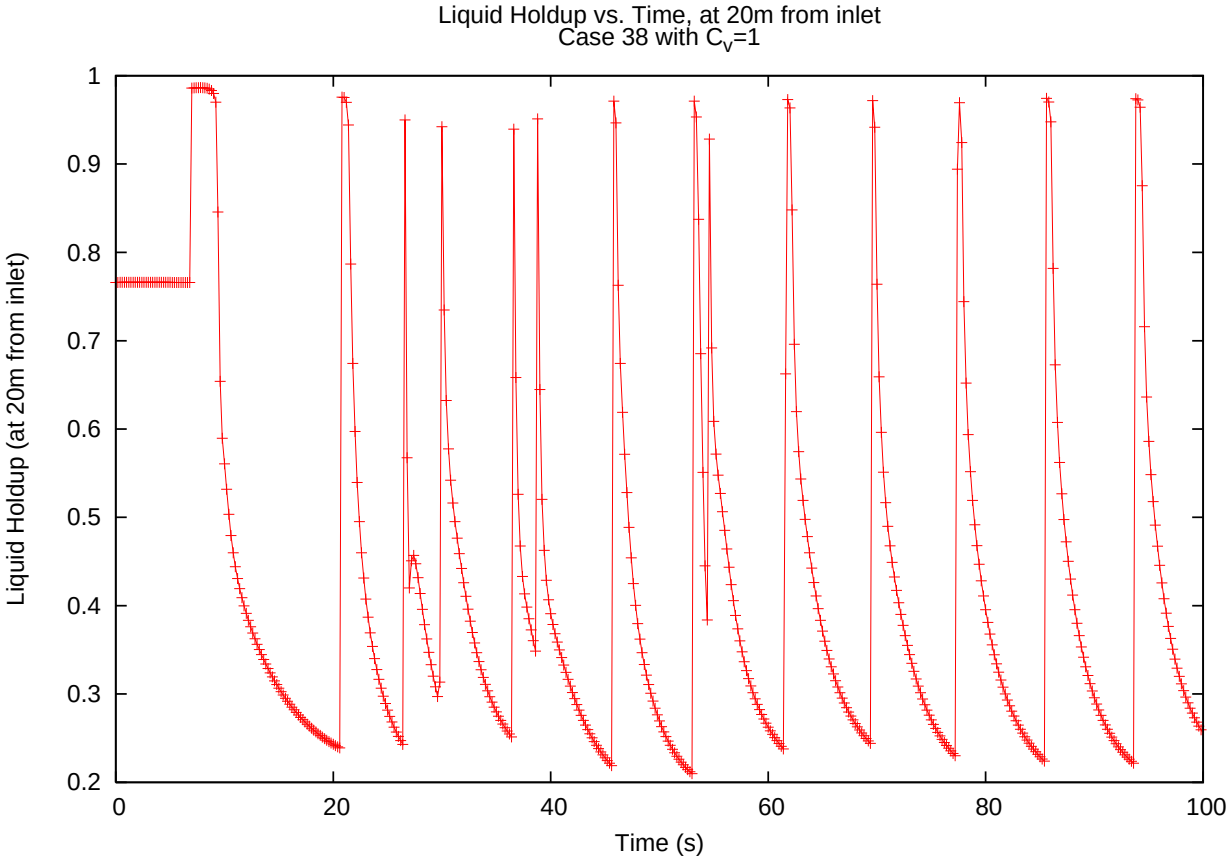


Figure A.12: Case 38 with C_V=1. Liquid holdup vs. time, measured at 20m from inlet.

A.2 Graphs for EMAPS simulations of Manolis cases using Reynolds based C_V

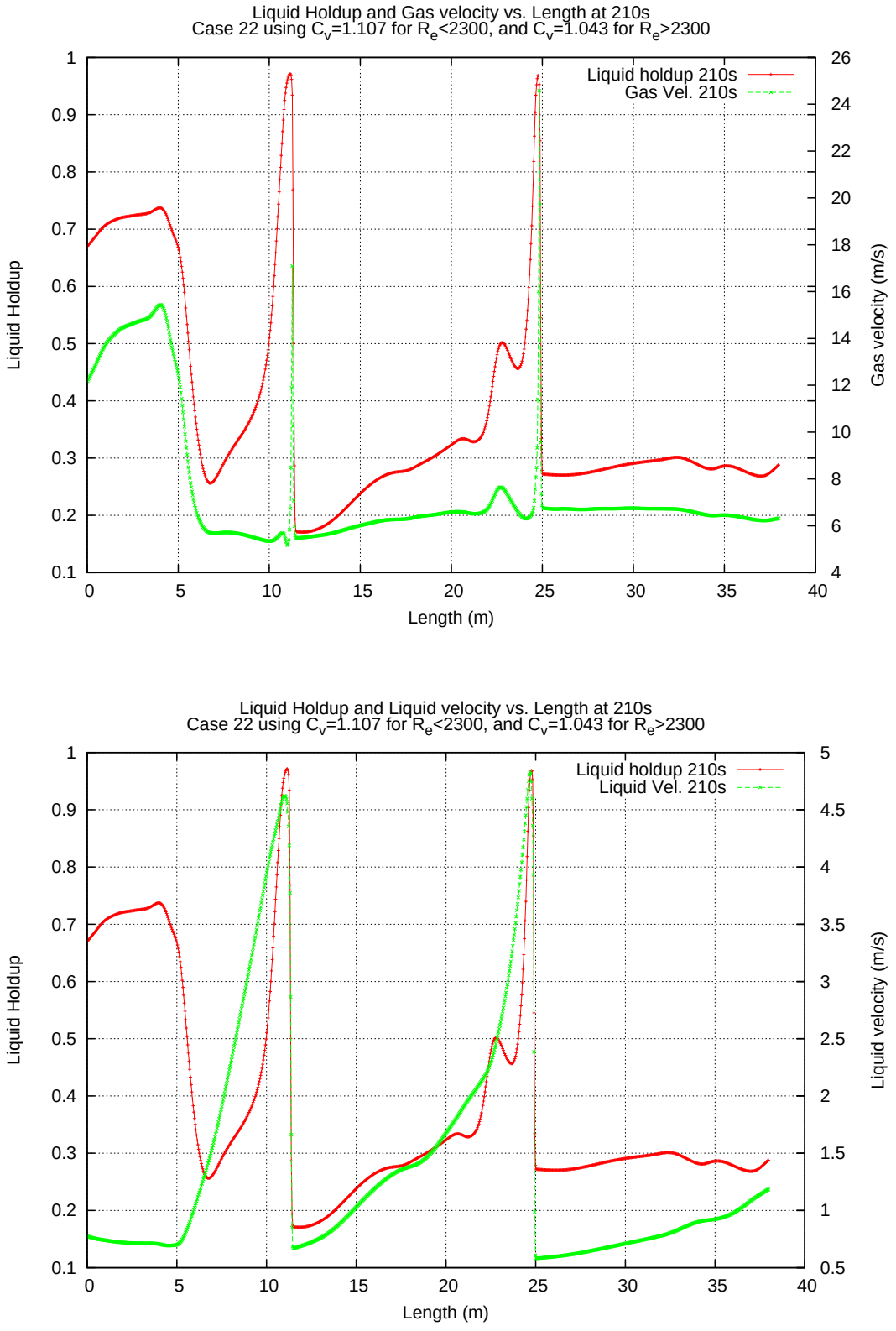


Figure A.13: Case 22 with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300. Top: liquid holdup and gas velocity vs. pipe length at 210s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 210s.

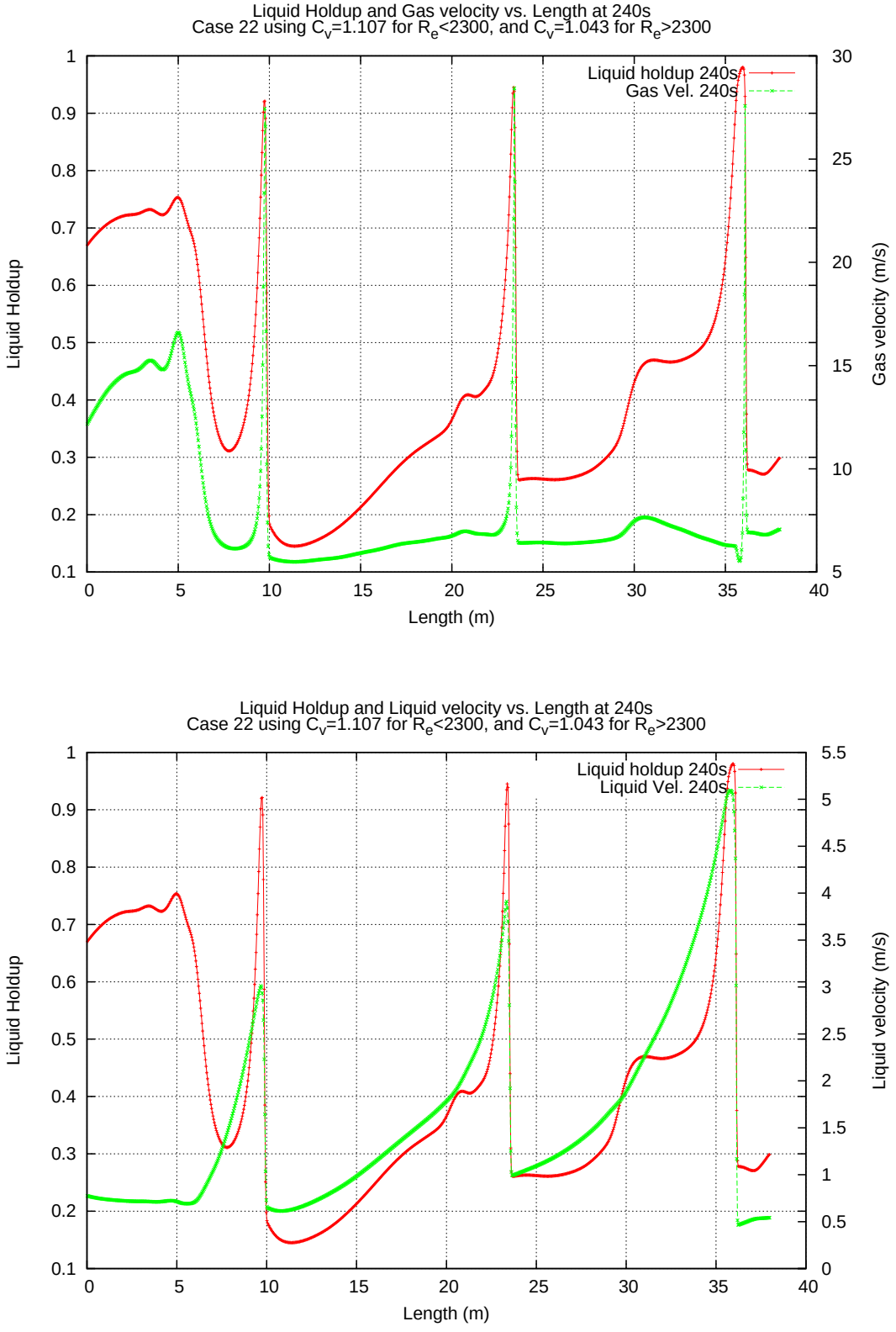


Figure A.14: Case 22 with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300. Top: liquid holdup and gas velocity vs. pipe length at 240s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 240s.

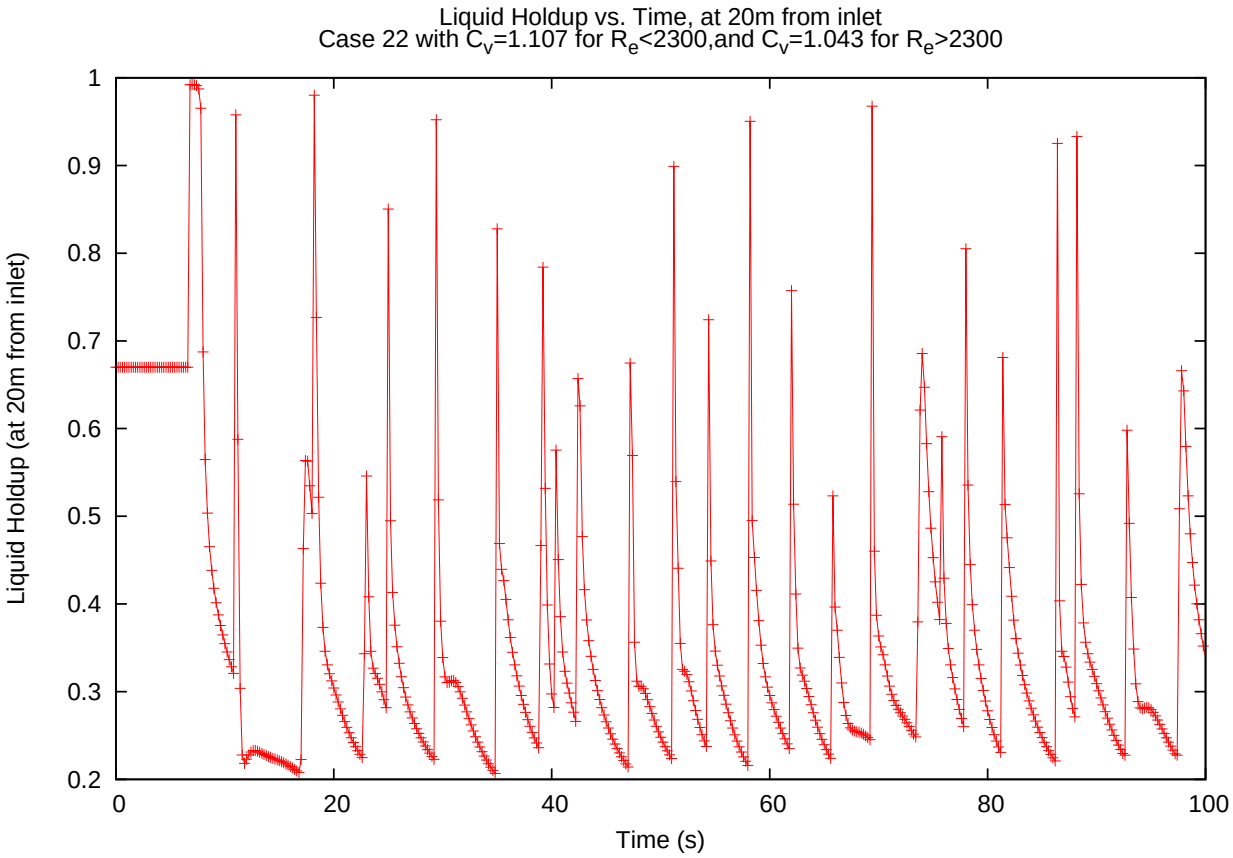


Figure A.15: Case 22 with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300. Liquid holdup vs. time, at 20m from the inlet.

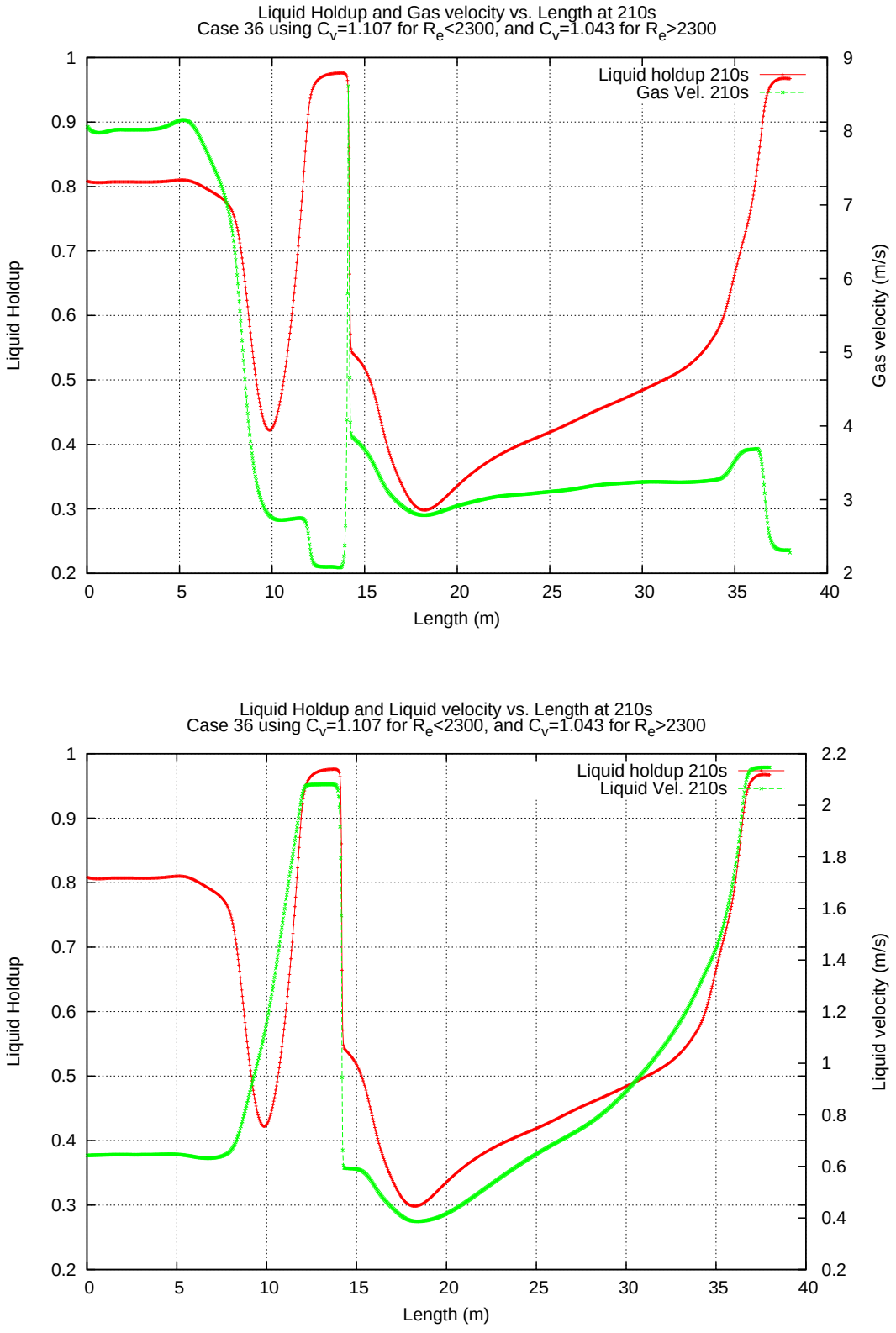


Figure A.16: Case 36 with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300. Top: liquid holdup and gas velocity vs. pipe length at 210s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 210s.

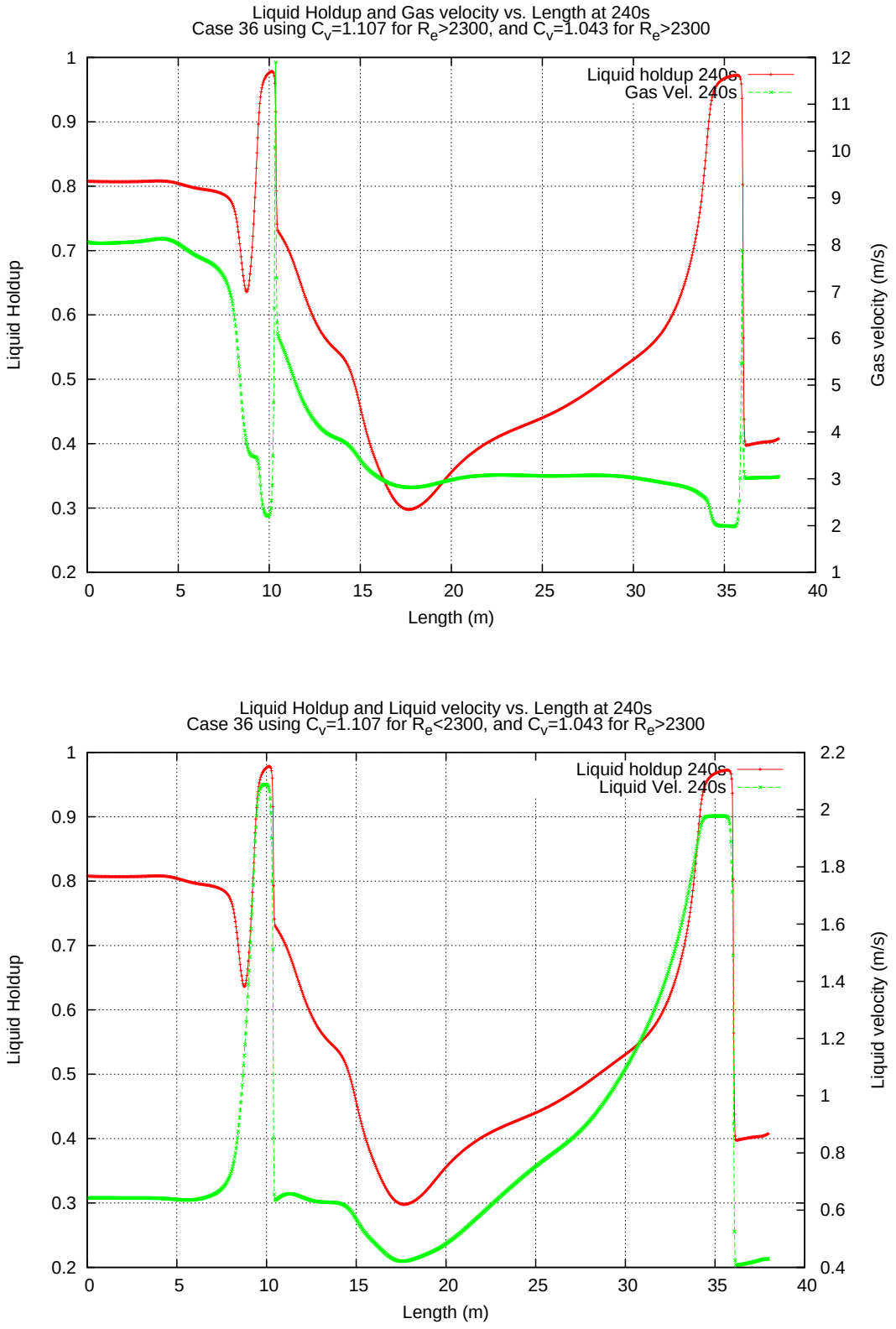


Figure A.17: Case 36 with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300. Top: liquid holdup and gas velocity vs. pipe length at 240s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 240s.

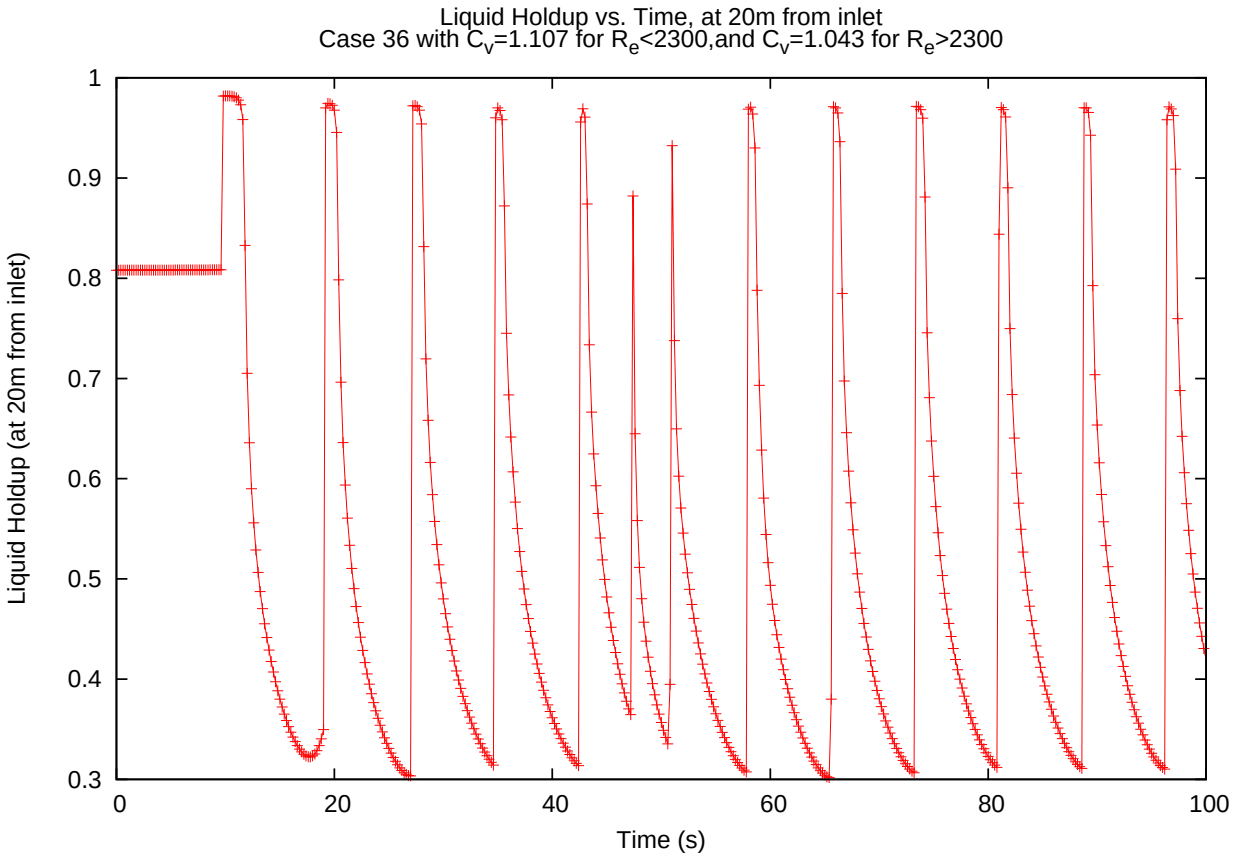


Figure A.18: Case 36 with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300. Liquid holdup vs. time, at 20m from the inlet.

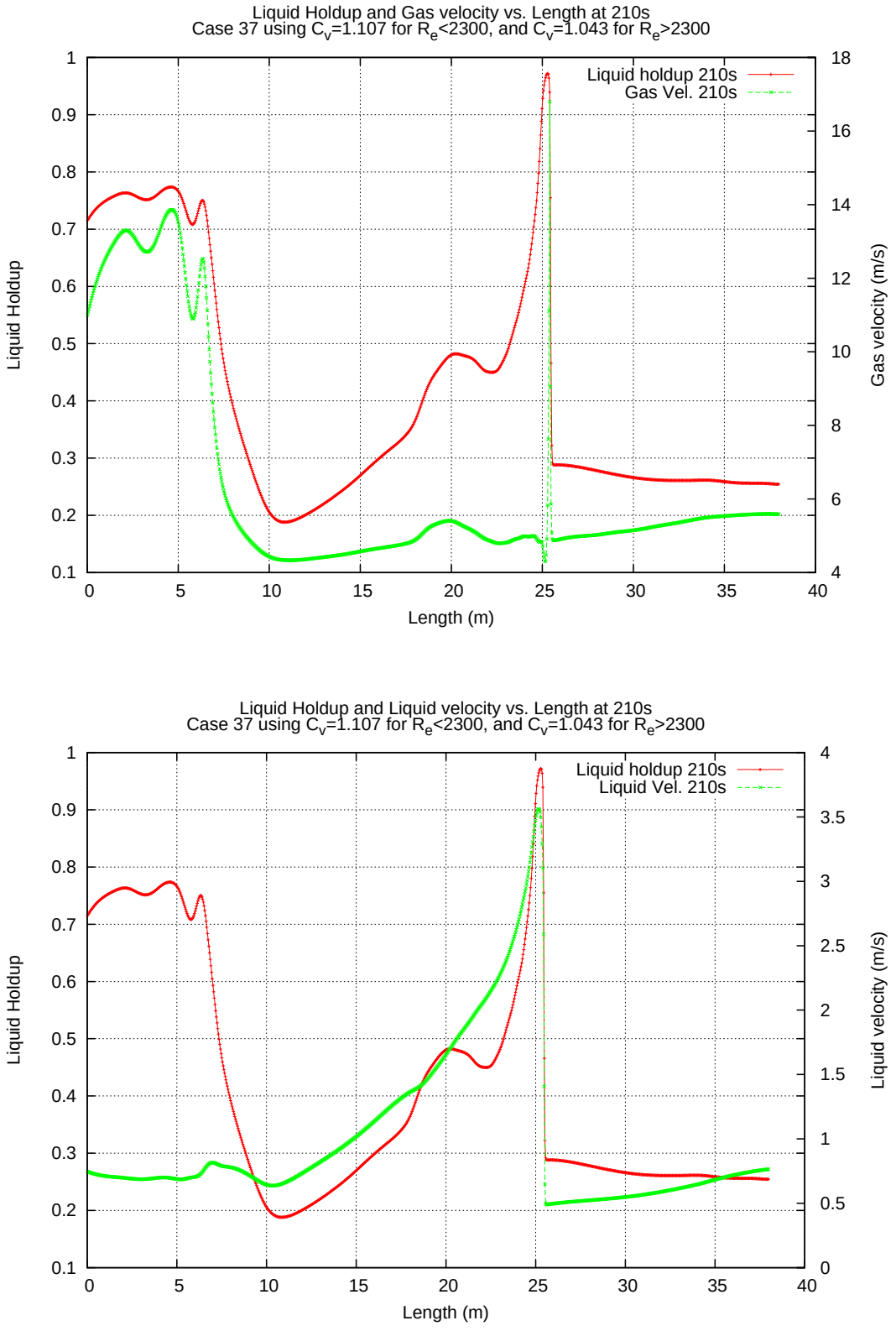


Figure A.19: Case 37 with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300. Top: liquid holdup and gas velocity vs. pipe length at 210s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 210s.

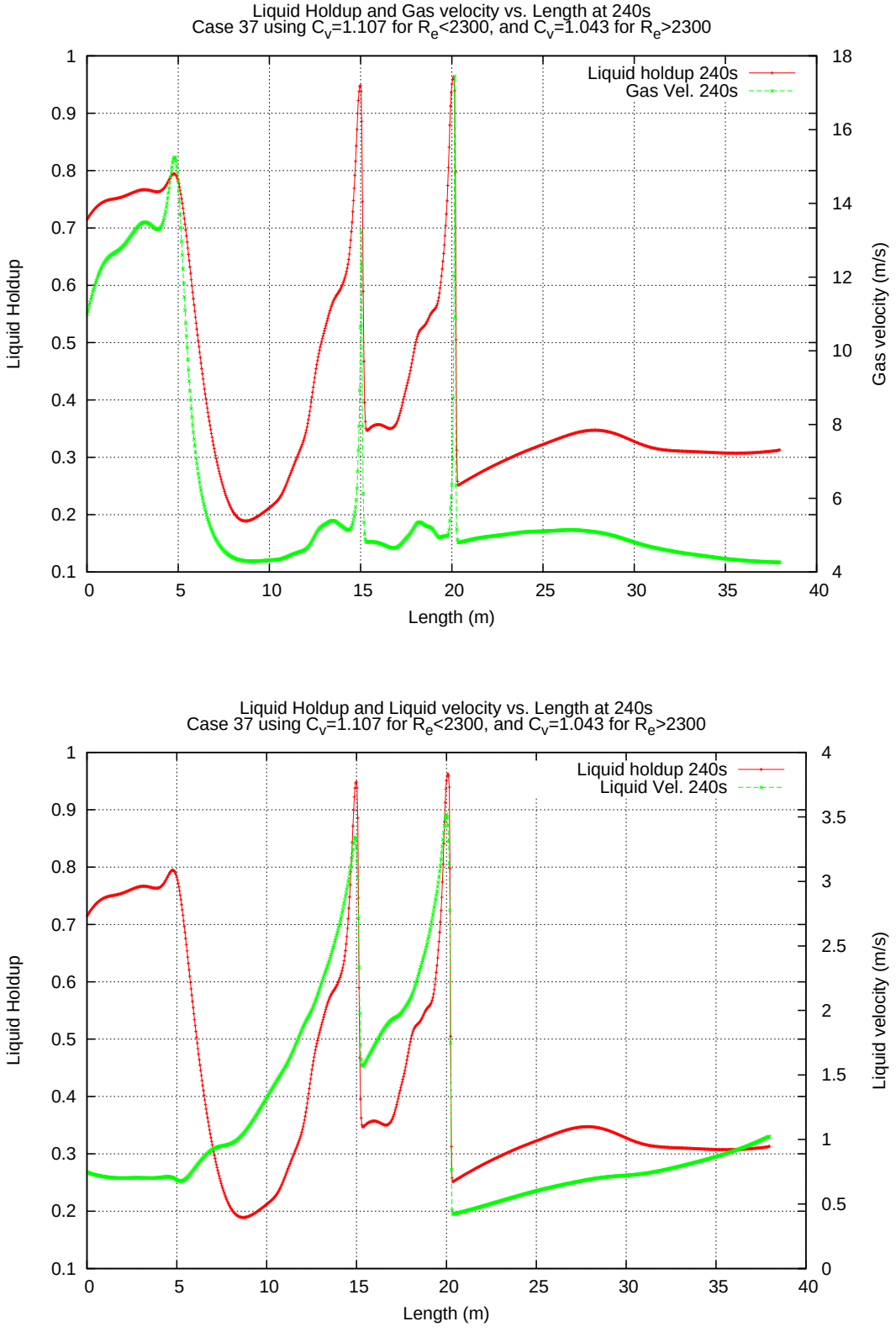


Figure A.20: Case 37 with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300. Top: liquid holdup and gas velocity vs. pipe length at 240s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 240s.

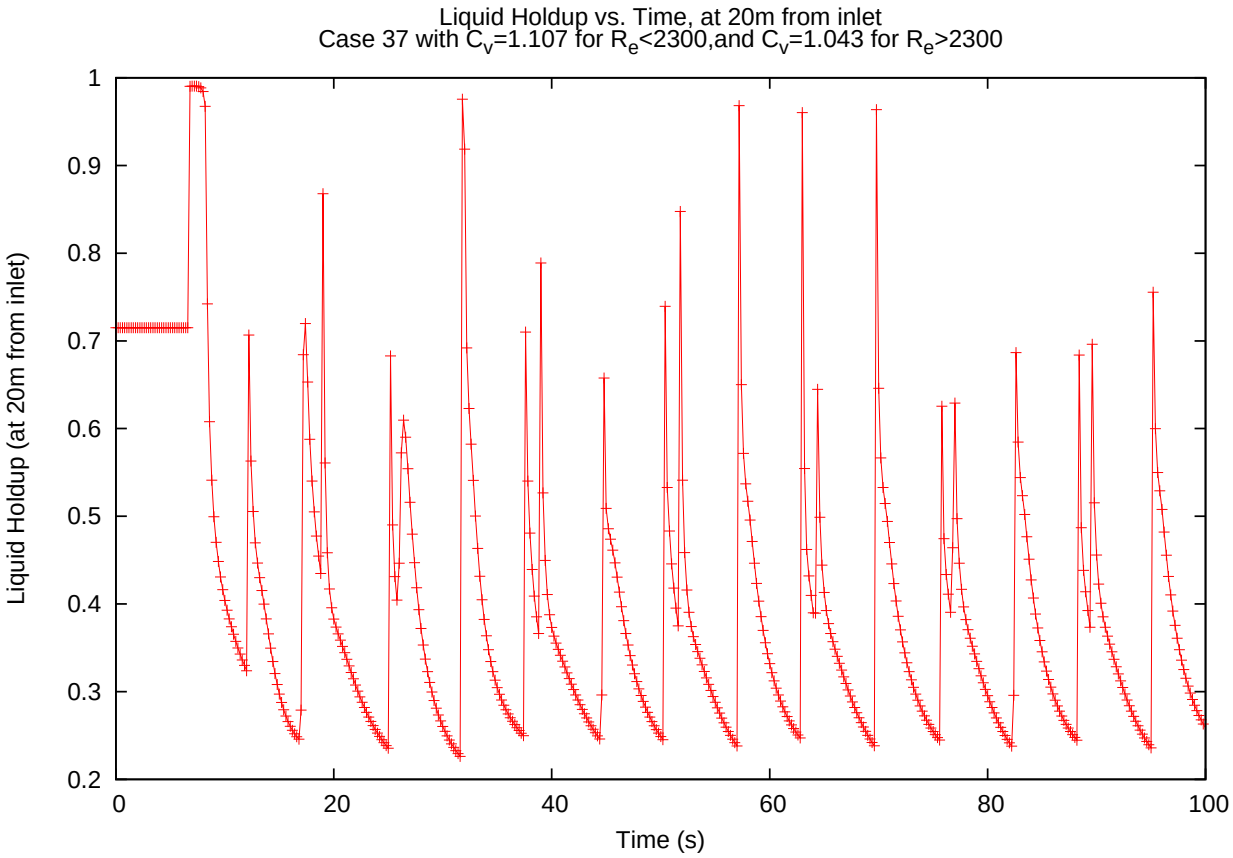


Figure A.21: Case 37 with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300. Liquid holdup vs. time, at 20m from the inlet.

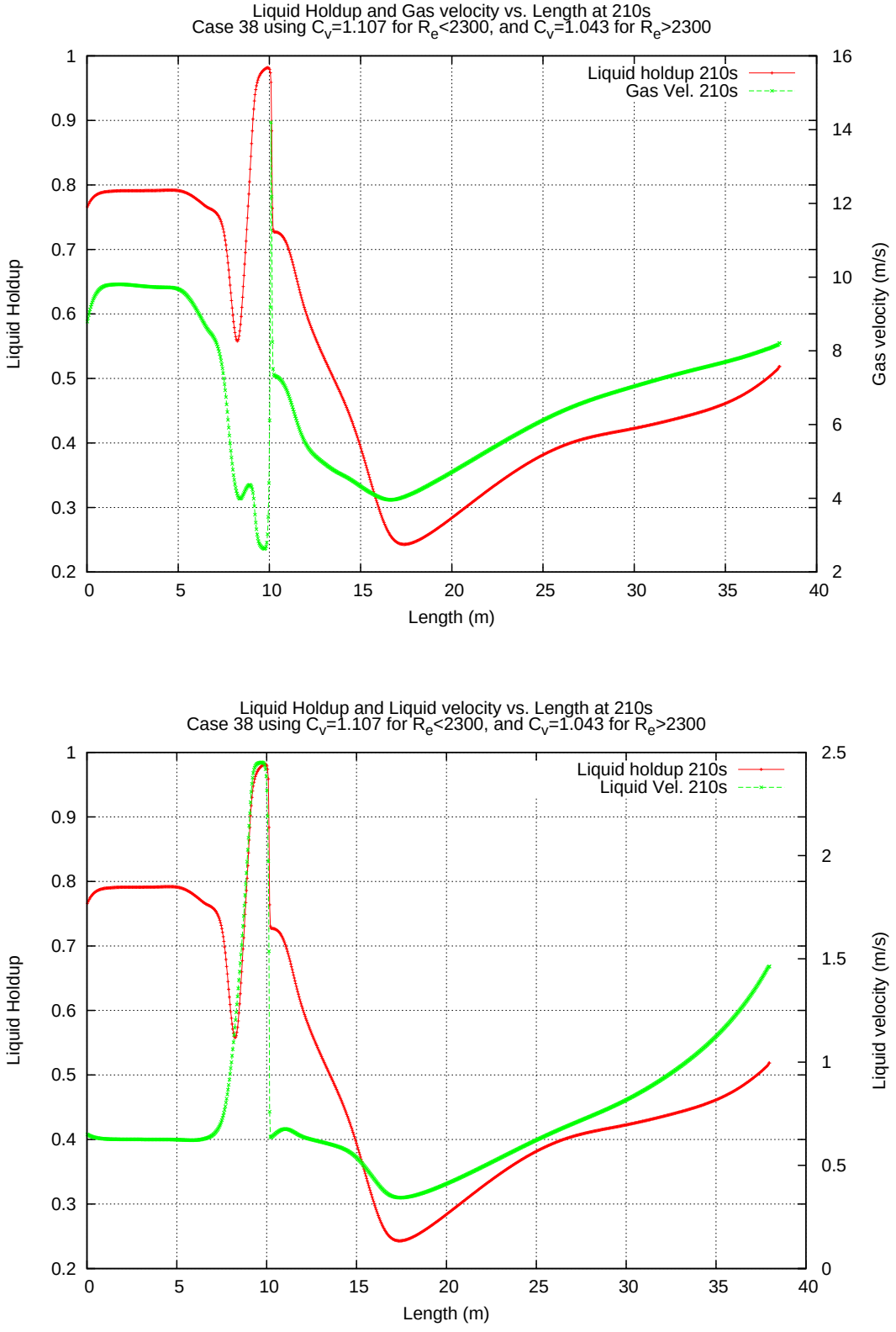


Figure A.22: Case 38 with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300. Top: liquid holdup and gas velocity vs. pipe length at 210s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 210s.

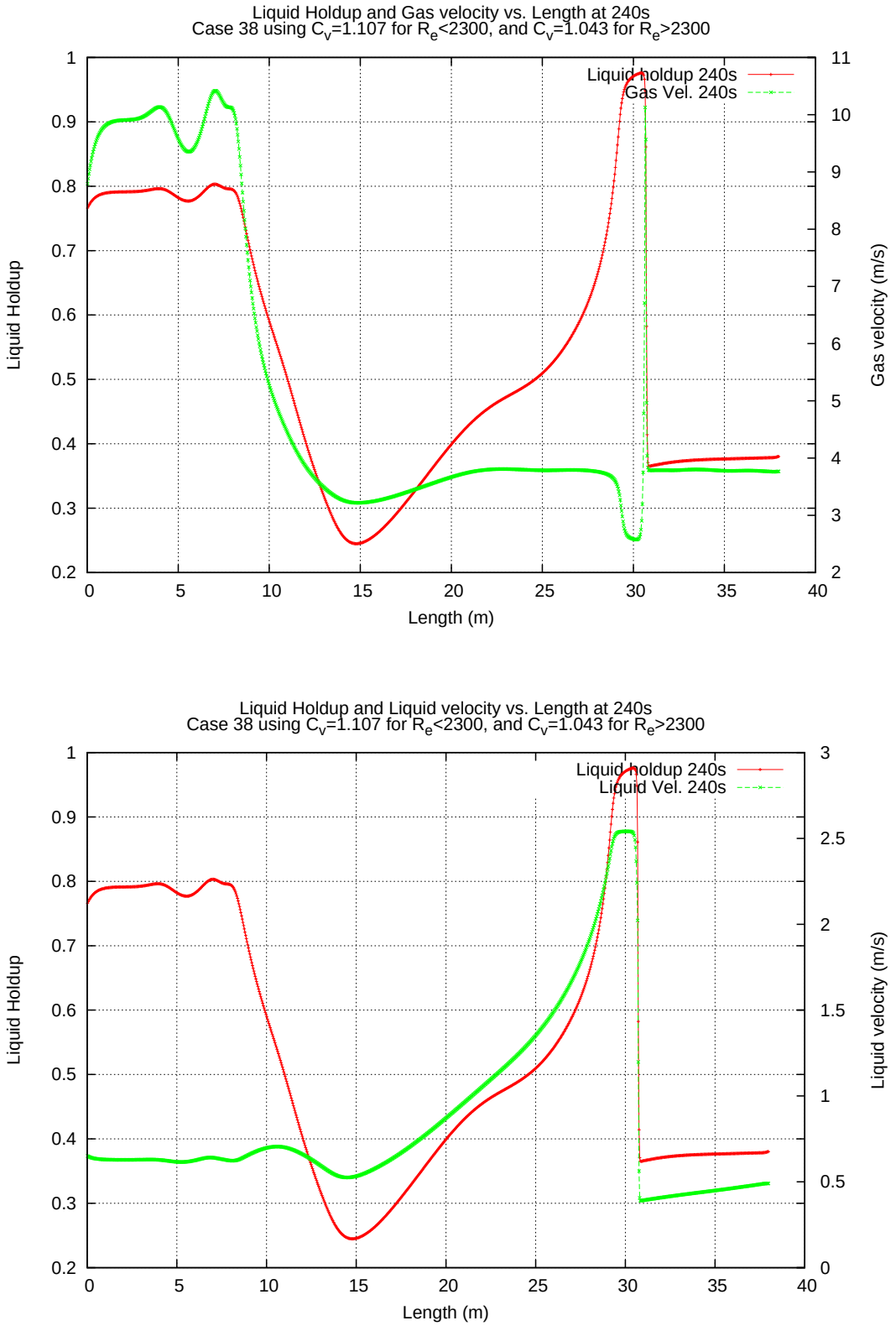


Figure A.23: Case 38 with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300. Top: liquid holdup and gas velocity vs. pipe length at 240s. Bottom: Liquid holdup and liquid velocity vs. pipe length at 240s.

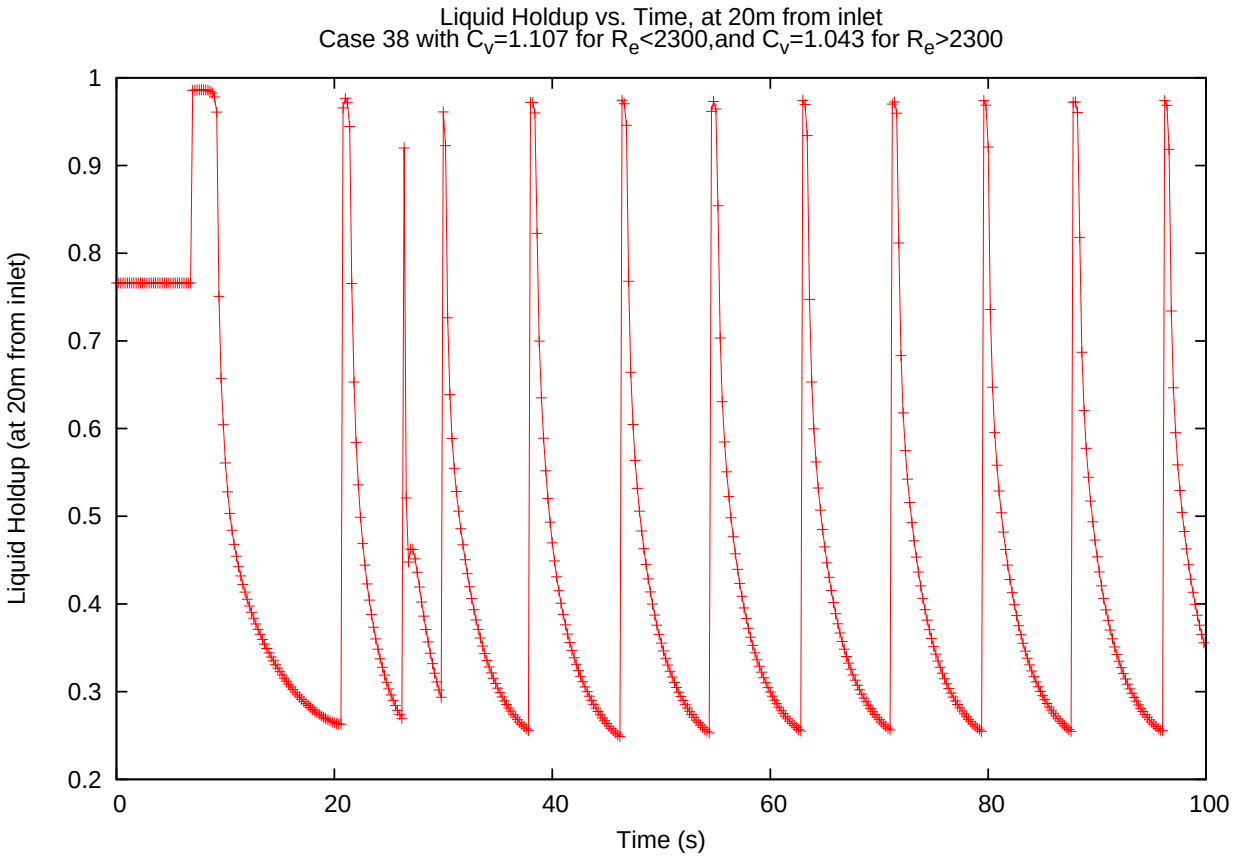


Figure A.24: Case 38 with C_V=1.107 for R_e<2300, and C_V=1.043 for R_e>2300. Liquid holdup vs. time, at 20m from the inlet.

Appendix B

User Defined Functions and Instructions for Fluent

B.1 UDF in FLUENT for sine-wave at t=0

```
#include "udf.h"

#define amp     0.05      /* Disturbance Amplitude */
#define diam    0.078       /* Pipe Diameter */
#define alpha1 0.758       /* Equilibrium Volume Fraction of heavy fluid */
#define alpha2 0.242       /* Volume Fraction of light fluid */

#define xcen 19.0 /*3.15*/
#define xini 15.0 /*2.48*/
#define xfin 23.0  /*3.82*/

#define PI 3.14159

DEFINE_ON_DEMAND(cvof_vel)

{

Domain *d_mix, *d_phase1,  *d_phase2;             /* declare domain pointer since
                    it is not passed as an argument to the DEFINE macro  */

Domain *domain[3];
```

B. USER DEFINED FUNCTIONS AND INSTRUCTIONS FOR FLUENT

```
d_mix = Get_Domain(1);
d_phase1 = Get_Domain(2);
d_phase2 = Get_Domain(3);

domain[0]=d_mix;
domain[1]=d_phase1;
domain[2]=d_phase2;

 real xc[ND_ND];
 real y;
 Thread **pt;
 Thread *thread;
 int i;
 cell_t cell;

{
 mp_thread_loop_c(thread, d_mix, pt)


  begin_c_loop_int (cell, thread)
  {
     C_CENTROID(xc,cell,thread);

  if (xc[1] < (alpha1) * diam)

        /* set volume fraction to 1 for centroid*/
        {C_VOF(cell,pt[0]) =1.;
        C_VOF(cell,pt[1])  = 0.;
        C_U(cell,pt[0])=0.4;


        }
  else
        /* otherwise initialize to zero */
        {C_VOF(cell,pt[0]) = 0.;
        C_VOF(cell,pt[1])  = 1.;
          C_U(cell,pt[1])=2.0;

        }


if ((xc[0] > xini) && (xc[0] < xfin))

{
y = alpha1 *(1.0  +   amp * sin(2./(xfin-xini) * PI * (xc[0]-xini)));


if (xc[1] < (y)*diam)

        /*set volume fraction to 1 for centroid */
        {C_VOF(cell,pt[0]) = 1.;
        C_VOF(cell,pt[1])  = 0.;
        C_U(cell,pt[0])=0.4;

        }
else
```

```
        /* otherwise initialize to zero */
        {C_VOF(cell,pt[0]) = 0.;
        C_VOF(cell,pt[1])  = 1.;
          C_U(cell,pt[1])=2.;

        }
}

 }

  end_c_loop_int(cell, thread)

}


}
```

B.2 Setting compressible air flow in FLUENT 2D

Define air as phase-1, compressible (doing the inverse usually creates problems).

Define Boundary Conditions
Loinlet Mass Flow Inlet
mixture set: Turbulence Specification method Intensity and
     Hydraulic Diameter 0.075
Thermal 300 K
  phase-1 set: Mass Flux 0
phase-2 set: Mass flux = physical velocity*density
UpInlet   Mass Flow Inlet as before: Diameter 0.003
   Outlet Pressure outlet, mixture intensity and hydraulic diameter 0.078
phase-2 back flow volume fraction 0

B. USER DEFINED FUNCTIONS AND INSTRUCTIONS FOR FLUENT

Appendix C

Details of slug cases

C.1 List of all experimental slug cases used for 2D CFD simulations

Experimental cases reported by Manolis (1995) have been used extensively in setting up both 1D simulations in EMAPS and 2D CFD simulations. A full list of the cases used is reported below, together with the initial gas and liquid superficial velocities, as well as the discrepancy (in percentage) between the experimental frequency and the 2D CFD frequency. Refer to Chapter *2D CFD simulation of slugs* for more details.

C. DETAILS OF SLUG CASES

Table C.1: List of all experimental slug cases used for 2D CFD simulations

Case number	Gas Superficial Velocity	Liquid Superficial Velocity	Discrepancy %
23	5.141	0.528	25.99
25	3.463	0.737	26.04
26	5.707	0.748	4.17
30	5.505	1.041	24.17
39	1.432	0.759	14.95
41	2.750	0.765	26.56
42	2.476	1.049	20.31
53	1.962	0.717	17.79
54	2.478	0.730	18.83
59	2.077	0.999	22.72
64	4.373	0.969	2.70
68	3.073	0.755	16.88
73	2.161	0.988	13.71
86	3.553	1.277	3.06
87	3.123	1.262	20.70
88	4.310	1.296	38.13
89	3.263	1.243	21.10
94	2.354	1.238	20.24
97	3.674	0.772	45.90
99	2.727	1.237	31.54
101	6.700	0.753	30.99
102	5.963	1.017	8.08
103	5.378	1.231	0.01
108	5.338	1.263	3.52

C.1 List of all experimental slug cases used for 2D CFD simulations

Continued

Case number	Gas Superficial Velocity	Liquid Superficial Velocity	Discrepancy %
109	5.259	1.014	9.70
110	5.602	0.767	12.09
113	3.416	1.275	44.65
120	3.522	1.279	31.70
131	4.049	1.255	0.41
132	3.959	1.020	10.78
133	3.969	0.747	37.28
138	3.124	1.281	14.02
139	2.573	1.000	30.70
149	3.678	1.283	2.30
150	3.711	0.980	30.32
155	4.160	0.983	22.42
161	4.618	1.297	16.55
162	4.588	0.980	6.04
163	4.602	0.750	39.23
165	3.793	0.996	12.50
166	2.010	1.326	18.13
171	1.867	1.340	11.79
176	1.981	1.006	33.42
180	1.693	1.170	45.40
181	1.956	0.969	22.96
188	3.159	0.766	27.68
189	3.190	1.022	46.88
190	2.832	1.286	23.84
196	4.584	1.294	19.58
197	4.792	1.285	9.80
198	5.121	1.008	1.71
202	5.908	1.351	2.69
203	4.606	1.001	31.69
208	7.214	1.280	17.72
209	6.964	1.001	38.86
210	7.047	0.761	20.83
211	6.999	0.519	11.68

Continued

Case number	Gas Superficial Velocity	Liquid Superficial Velocity	Discrepancy %
217	6.439	1.271	19.14
218	5.480	1.251	0.21
219	5.994	0.990	21.93
220	5.900	0.756	3.84
221	6.233	0.745	5.43
223	7.052	1.010	45.39
230	5.940	0.999	32.03
250	3.059	0.769	8.76
254	2.060	1.274	16.67
256	2.264	0.745	9.22
258	2.235	0.756	36.31
229	6.109	1.220	6.54
222	6.937	1.282	0.89
36	1.548	0.519	6.00
38	2.058	0.498	7.71
31	8.572	1.081	27.61
6	8.344	0.490	35.30
14	7.907	0.745	35.63
15	6.837	0.772	33.65
24	6.532	0.532	37.50
29	6.541	1.065	10.68
40	1.945	0.751	5.76
74	2.285	1.001	6.45
90	3.830	1.241	4.54
93	2.846	1.026	40.73
96	3.649	1.041	12.43
142	5.024	1.300	19.54
143	5.108	0.999	0.80
177	1.796	0.751	3.29
195	4.054	1.025	1.03
27	4.506	0.735	22.67
81	4.240	0.927	5.05
194	4.116	0.767	9.70
199	5.268	0.778	16.87
180	1.693	1.270	8.04

References

Manolis, I. G. (1995), High Pressure Gas-Liquid Slug Flow, PhD thesis, Department of Chemical Engineering and Chemical Technology, Imperial College of Science, Technology and Medicine, UK. (cited at page 277)

Appendix D

Scripts

D.1 Various scripts written for file processing

```
# script written by S Kalogerakos
# removes specified characters from files
#!/bin/bash

P=*.prof

for file in $P
do
 sed -n '/y/,/z/p' $file > col1_.txt              #output lines inclusive between y and z
 sed -n '/absolute/,/)/p' $file > col2_.txt          #output lines inclusive between absolute and )
 sed '/^(/d;/^)/d' col1_.txt > col1.txt          #remove lines containing ( and )
 sed '/^(/d;/^)/d' col2_.txt > col2.txt
 file_new=`echo "$file" | sed 's/prof/xy/'`;
 awk '{getline col1<"'col1.txt'"; print col1+0.039, $1} ' col2.txt > $file_new
 rm col1_.txt col1.txt col2_.txt col2.txt
done
```

```
# SCRIPT written by S Kalogerakos

convert animated.gif %08d.jpg
#need to have imagemagick installed, search on yast.
#This will convert animated.gif to many jpegs

ffmpeg -r 15 -i %08d.jpg -y -an animated.avi
```

```
#convert the created jpg files into avi, at a speed of 15fps.
#Need to have ffmpeg installed

_____________________________________

# SCRIPT written by S Kalogerakos

mplayer -vo jpeg animated.gif
#need to have mplayer installed, search on google mplayer opensuse.
#This will convert animated.gif to many jpegs


ffmpeg -r 15 -i %08d.jpg -y -an animated.avi
#convert the created jpg files into avi, at a speed of 15fps

_____________________________________

#/bin/ksh

# Script by S. Kalogerakos February 2009
# Written in order to analyse FLUENT output files
# Particles and radiation simulation


rm -rf intensity.txt
(ls output-*.txt | while read filename ;
do
awk
'FNR==1
{timestep=1.0E-5;particle_id=$3;previous_time=$11; intensity12_previous=$12}
FNR>1 {duration11=$11-previous_time; nsteps=duration11/timestep; previous_time=$11;
intensity12+=nsteps*(intensity12_previous+$12)*0.5; intensity_previous=$12}
END { print particle_id, intensity12}' $filename >> intensity.txt; done )


#(FNR>1) avoid first row header
# BEGIN - do things before processing file
# END - do things after having finished processing file
# getline var < file     get whole row from file into variable var
# split (var,a) split string var into array a
#NR is number of row

_____________________________________

#/bin/ksh

# Script by S. Kalogerakos February 2009
# Written in order to get post-processing properties for radiation in particle simulation


rm -rf intensity.txt
(ls output-*.txt | while read filename ;
do awk 'FNR==1{particle_id=$3;previous_time=$11; intensity12_previous=$12}
FNR>1 {duration11=$11-previous_time; previous_time=$11;
intensity12+=duration11*intensity12_previous; intensity_previous=$12}
```

```
END { print particle_id, intensity12}' $filename >> intensity.txt; done )

#alternative
awk 'BEGIN{print"Particle_ID","Intensity"}
{getline intensity<"'intensity_"$filename"'";
split(var5,a);
print col1,col5-a[1],a[1],
((col5-a[1])^2)*a[3],col6-a[2],a[2],
(col6-a[2])^2*a[4]}' $filename> fluctuations_$filename; done)


#(FNR>1) avoid first row header
# BEGIN - do things before processing file
# END - do things after having finished processing file
# getline var < file     get whole row from file into variable var
# split (var,a) split string var into array a
#NR is number of row
```

```
#/bin/ksh

# Script by S. Kalogerakos March 2009
# Written in order to get particle and radiation properties from FLUENT

rm -rf intensity.txt
(ls output-*.txt | while read filename ;
do awk
'FNR==1{previous_time=$11}
FNR>1{particle_id=$3; duration11=$11-previous_time; previous_time=$11;
intensity12+=duration11*$12}
END { print particle_id, intensity12}' $filename >> intensity.txt; done )

#alternative
awk 'BEGIN{print"Particle_ID","Intensity"} {getline intensity<"'intensity_"$filename"'";
split(var5,a);print col1,col5-a[1],a[1],((col5-a[1])^2)*a[3],col6-a[2],a[2],
(col6-a[2])^2*a[4]}' $filename> fluctuations_$filename; done)



#(FNR>1) avoid first row header
# BEGIN - do things before processing file
# END - do things after having finished processing file
# getline var < file     get whole row from file into variable var
# split (var,a) split string var into array a
#NR is number of row
```

```
#script written by S Kalogerakos
# Sums columns and extracts their average

#/bin/ksh
(ls  soln.* | while read filename ; do awk '{sum5+=$5} {sum6+=$6} {sum7+=$7}
END { print sum5/NR}' $filename ; done > $filename.averages)
```

```
# SCRIPT written by S Kalogerakos

#Find maximum and minimum values present in column 1

awk 'NR == 1 {m=$1 ; p=$1}
$1 >= m {m = $1}
$1 <= p {p = $1}
END { print "Max = " m, "    Min = " p }' file.txt
```

```
# SCRIPT written by S Kalogerakos

# sorting according to numerical value, 2 for second column

sort -nk 2 filename
```

```
#Script written by S Kalogerakos
# Order files by columns
cd temp
# Changing Directory: Sorting & Removing duplicate Rows
Q=output*
#pattern it is searching for

for file in $Q

do
rm -rf temp_output;
sort -ruk 11n $file > temp_output;                    # Compare the 11th Column " time"
mv temp_output "$file"

done
```

```
#SCRIPT written by S Kalogerakos
# Read variables from multiple files, file1.txt and file2.txt
awk '{getline var1<"file1.txt"; print $1, var1}' file2.txt
```

```
#SCRIPT written by S Kalogerakos
# remove duplicate lines comparing to second file
awk 'BEGIN{
while ((getline < "file1") > 0)
    list50[$1] = 1}
!list50[$1] {print}' file2 > file2new
```

```
# SCRIPT written by S Kalogerakos
# Remove duplicate lines
awk '{
if ($0 in stored_lines)
   x=1
else
   print
   stored_lines[$0]=1
}' $1 > $2
```

```
#              SCRIPT DEVELOPED BY STAMATIS KALOGERAKOS
#                   Cranfield University AMAC Group
#                        DATE  : 01 July 2009
# File and string manipulation
#!/bin/bash

P=pressure-*.xy

mkdir -p backup
cp -f $P backup/

for file in $P
do
sed -e '1,4d' -e '/)/d' $file > temp_$file      # Removing first 4 lines and Brackets
done

for file in temp_$P
do
 file_new=`echo "$file" | sed 's/temp_//'`;             # Moving Back Temporary File To Original File
 mv "$file" "$file_new"
done
```

```
#              SCRIPT DEVELOPED BY STAMATIS KALOGERAKOS
#                   Cranfield University AMAC Group
#                        DATE  : 02 February 2009
# Resize PPM  files, which are image files output by FLUENT

P=*.ppm

for file in $P
do
file_new=`echo "$file" | sed 's/.ppm//'`
ppmtogif $file >temp1.gif
convert  -crop '680x35!+0+220'   temp1.gif temp2.gif
convert     -resize '680x35!' temp2.gif GIF/"$file_new".gif
rm -rf temp1.gif temp2.gif
done
```

```
#             SCRIPT DEVELOPED BY STAMATIS KALOGERAKOS
#                  Cranfield University AMAC Group
#                       DATE  : 02 February 2009
# Resize and convert TIF  files

P=*.tif

for file in $P
do
file_new=`echo "$file" | sed 's/.tif//'`
convert $file temp1.gif
convert  -crop '1360x70!+0+440'   temp1.gif temp2.gif
convert     -resize '1360x70!' temp2.gif GIF/"$file_new".gif
rm -rf temp1.gif temp2.gif
done
```

```
#             SCRIPT DEVELOPED BY STAMATIS KALOGERAKOS
#                  Cranfield University AMAC Group
#                       DATE  : 06 February 2009
# Program to scan IP address and select according to name
#!/bin/bash
i=1

while [ $i -lt 250 ]
do
nmap -sP 138.250."$i".* | grep -i imesh
echo "$i"
i=$[i+1]
done
```

```
#             SCRIPT DEVELOPED BY STAMATIS KALOGERAKOS
#                  Cranfield University AMAC Group
#                       DATE  : 02 February 2008
# Program to create animations from saved images of FLUENT

# TO EXECUTE THIS SCRIPT TYPE: ./SCRIPT  1  ,MAXNUMBER OF FILES,  NAMEMOVIE
#/bin/ksh
i=1

while [ "$i" -le "9" ] && [ "$i" -le "$2" ]
do

pnmquant 256 $3_000$i.ppm > $3_000$i_.ppm
ppmtogif  $3_000$i_.ppm >$3000$i.gif
rm $3_000$i_.ppm

let "i = $i + 1"
done


while [ "$i" -le "99" ] && [ "$i" -le "$2" ]
```

```
do

pnmquant 256 $3_00$i.ppm > $3_00$i_.ppm
ppmtogif  $3_00$i_.ppm>$300$i.gif
rm $3_00$i_.ppm

let "i = $i + 1"
done


#pause -1 "press return to continue"
```

```
#             SCRIPT DEVELOPED BY STAMATIS KALOGERAKOS
#                  Cranfield University AMAC Group
#                       DATE  : 17 May 2008
#
# show the name of the most recent case FLUENT file

ls -FAt *.cas.* | head -n 1
```

```
#             SCRIPT DEVELOPED BY STAMATIS KALOGERAKOS
#                  Cranfield University AMAC Group
#                       DATE  : 17 May 2009
#
#To execute this script, ./substitute.sh 22 36
#It will substitute all 22 with 36 in all .plt files

#!/bin/bash
OLD="$1"
NEW="$2"
DPATH="*.plt"
TFILE="out.tmp.$$"
for f in $DPATH
do
  if [ -f $f -a -r $f ]; then
   sed "s/$OLD/$NEW/g" "$f" > $TFILE && mv $TFILE "$f"
  else
   echo "Error: Cannot read $f"
  fi
done
rm $TFILE
```

```
#/bin/ksh

# Script by S. Kalogerakos February 2009
# Written in order to get fluctuating velocities and densities from soln.* files from EMAPS output
# Column $1 is time, $5 and $7 are gas velocities and densities, $6 and $8 are liquid velocities and densities
```

```
(ls soln.* | while read filename ;
do awk
'{FNR>1} {sum5+=$5; sum6+=$6; sum7+=$7; sum8+=$8; ++i}
END { print sum5/(i-1),sum6/(i-1),sum7/(i-1),sum8/(i-1)}'
$filename > averages_$filename;
awk 'BEGIN{print"Time","Gas Vel. Fluctuation",
" Gas Vel. Avg.","Gas Reynold Stress","Liq. Vel. Fluctuation",
" Liq. Vel. Avg.","Liq. Reynold Stress"}
NR<2 {next}
{ col1=$1;col5=$5;col6=$6;getline var5<"'averages_"$filename"'";
split(var5,a);print col1,col5-a[1],a[1],
((col5-a[1])^2)*a[3],col6-a[2],a[2],(col6-a[2])^2*a[4]}'
$filename> fluctuations_$filename; done)



#FNR>1 avoid first row header
# BEGIN - do things before processing file
# END - do things after having finished processing file
# getline var < file     get whole row from file into variable var
# split (var,a) split string var into array a



#NR is number of row
#
```

```
# SCRIPT written by S Kalogerakos

#!/bin/bash
# type ./text_replace.sh FILENAME
# this will expect the id number to be in the third column
# change the variables if necessary, i.e. e1 for first column.  If fourth column, then it needs to be
# while read e1 e2 e3 e4 junk
# do
# grep e4" $file > output-$file-$e4.txt

P=*.dpm.*                          #pattern it is searching for

for file in $P
do
tr -d '(\)' < $file > temp_$file
done


for file in temp_$P
do
 file_new=`echo "$file" | sed 's/temp_//'`;
 mv "$file" "$file_new"
done


for file in $P
```

```
do
#we store the original IFS
  O=$IFS; IFS=" ";
  cp "$file" temp2;
  while  read e1 e2 e3 junk
  do

     grep " $e3 " $file > temp/output-$e3.txt;     #create output file with id number
     rm -rf temp3;
     grep -v " $e3 " temp2 > temp3;
     mv temp3 temp2;
  done < temp2;
#revert back the IFS
IFS=$O;
rm temp2;
done
```

```
#              SCRIPT DEVELOPED BY STAMATIS KALOGERAKOS
#                   Cranfield University AMAC Group
#                        DATE  : 04 March 2009
#!/bin/bash
# type ./text_replace.sh FILENAME
# this will expect the id number to be in the third column
# change the variables if necessary, i.e. e1 for first column.
#If fourth column, then it needs to be
# while read e1 e2 e3 e4 junk
# do
# grep e4" $file > output-$file-$e4.txt

P=*.dpm.*                                              # Pattern it is searching for

for file in $P
do
tr -d '(\)' < $file > temp_$file                       # Removing Brackets
done


for file in temp_$P
do
 file_new=`echo "$file" | sed 's/temp_//'`;            # Moving Back Temporary File To Origial File
 mv "$file" "$file_new"
done


for file in $P
do
#we store the original IFS
  O=$IFS; IFS=" ";
  cp "$file" temp2;
  while  read e1 e2 e3 junk
  do

     grep " $e3 " temp2 >> temp/output-$e3.txt;        #create output file with id number
```

```
    rm -rf temp3;
    grep -v " $e3 " temp2 > temp3;                    #Create Temporary File after Removing Processed Line
    mv temp3 temp2;
    echo "Particle $e3 done";
  done < temp2;
#revert back the IFS
IFS=$O;
rm -rf temp2;
echo "----------------------File $file finished";
done
```

D.2 Scripts and Journals to be used with Ansys Fluent

```
# SCRIPT written by  S Kalogerakos
# Calculates maximum liquid holdup from FLUENT simulations

P=WAVGROWTH-*.cas.gz

rm -rf maxliqheights.txt flowtime.txt liqheight.txt

for file_cas in $P
do
mv $file_cas temp.cas.gz
file_dat=`echo "$file_cas" | sed 's/cas/dat/'`;
mv $file_dat temp.dat.gz

rm -rf liqheight.txt

/home/fn081840/fluent/fluent_2ddp_jou.sh liqheight.jou

#extract flow-time from created files



echo "$file_cas" | sed -e 's/WAVGROWTH-28M-II-SteadyState-//' -e 's/.cas.gz//' > flowtime.txt

# Find maximum values present in column 2 and store position
awk 'NR == 1 {m=$2 ; x=$1}      $2 >= m {m = $2 ; x = $1}
    END {getline flowtime<"flowtime.txt"; print flowtime, x, m  }' liqheight.txt >> maxliqheights.txt


mv temp.cas.gz $file_cas
mv temp.dat.gz $file_dat

done
```

```
# Script written by S Kalogerakos
# Output PPM images are automatically created for all existing cases, with matching names

P0=*-?.*.dat.gz
P1=*-??.*.dat.gz
initial_case=WAVGROWTH-28M-II-SteadyState-0.0000.cas.gz              #change this appropriately
mv $initial_case initial.cas.gz
j=1

for file in $P0
do
 file_old_dat=$file
 mv "$file" "current.dat.gz"
 fluent_2ddp_jou.sh contour.jou
 if [ "$j" -lt 10 ]; then
  mv "current.ppm" "WAVGROWTH-00$j.ppm"
 elif [ "$j" -lt 100 ]; then
  mv "current.ppm" "WAVGROWTH-0$j.ppm"
 elif [ "$j" -lt 1000 ]; then
  mv "current.ppm" "WAVGROWTH-$j.ppm"
 fi
mv current.dat.gz $file_old_dat
j=$(echo "$j+1"|bc)
done

for file in $P1
do
 file_old_dat=$file
 mv "$file" "current.dat.gz"
 fluent_2ddp_jou.sh contour.jou
 if [ "$j" -lt 10 ]; then
  mv "current.ppm" "WAVGROWTH-00$j.ppm"
 elif [ "$j" -lt 100 ]; then
  mv "current.ppm" "WAVGROWTH-0$j.ppm"
 elif [ "$j" -lt 1000 ]; then
  mv "current.ppm" "WAVGROWTH-$j.ppm"
 fi
mv current.dat.gz $file_old_dat
j=$(echo "$j+1"|bc)
done

mv initial.cas.gz  $initial_case
```

```
# Script written by S Kalogerakos
# Output TIF images are automatically created for all existing cases, with matching names

P0=*-?.*.dat.gz
P1=*-??.*.dat.gz
initial_case=WAVGROWTH-28M-II-SteadyState-0.0000.cas.gz              #change this appropriately
mv $initial_case initial.cas.gz
j=1
```

```
for file in $P0
do
 file_old_dat=$file
 mv "$file" "current.dat.gz"
 fluent_2ddp_jou.sh contour_tif.jou
 if [ "$j" -lt 10 ]; then
  mv "current.tif" "WAVGROWTH-00$j.tif"
 elif [ "$j" -lt 100 ]; then
  mv "current.tif" "WAVGROWTH-0$j.tif"
 elif [ "$j" -lt 1000 ]; then
  mv "current.tif" "WAVGROWTH-$j.tif"
 fi
mv current.dat.gz $file_old_dat
j=$(echo "$j+1"|bc)
done

for file in $P1
do
 file_old_dat=$file
 mv "$file" "current.dat.gz"
 fluent_2ddp_jou.sh contour_tif.jou
 if [ "$j" -lt 10 ]; then
  mv "current.tif" "WAVGROWTH-00$j.tif"
 elif [ "$j" -lt 100 ]; then
  mv "current.tif" "WAVGROWTH-0$j.tif"
 elif [ "$j" -lt 1000 ]; then
  mv "current.tif" "WAVGROWTH-$j.tif"
 fi
mv current.dat.gz $file_old_dat
j=$(echo "$j+1"|bc)
done

mv initial.cas.gz  $initial_case
```

```
# Script written by S Kalogerakos
# Allows seamless restarting of simulations
# Checks for existence of saved files
# Asks uses to set a max limit of iterations
#!/bin/bash
echo "ATTENTION!  Make sure it was Name (from Name-1000.cas.gz)"
echo "If it is first file, the two files need to be Name-0.cas.gz and Name-0.dat.gz"
echo "Otherwise exit with Ctrl+C"
read -p "If you are sure, then press enter to continue..."


          if [ -z "$1" ]; then
              echo "ERROR! usage: $0 Name (from Name-1000.cas.gz)"
              exit
          fi
N_Iterations=0

file=$1
initialname=$file
```

```
#show the name of the most recent case file
lastcasefile=`ls -FAt $file-*.cas.gz | head -n 1`;
lastdatafile=`ls -FAt $file-*.dat.gz | head -n 1`;

#care with the quotes in order for sed to use a variable $file
initialcounter=`echo "$lastcasefile" | sed -e 's/.cas.gz//' -e "s/$file-//"`;

echo "Last iteration is $initialcounter"
echo "If this is correct, then press enter to continue..."
read -p "Otherwise exit with Ctrl+C"

echo "How many total iterations do you want to carry out?"
echo "Files will be saved and re-partitioned every 50 iteration"
read N_Iterations

let "N_Iterations=$N_Iterations"                #force to become integer



if [ "$N_Iterations" -lt 50 ]; then   #change to 50
                echo "ERROR! Number of iterations must be larger than 50"
                exit
        fi
if [ -z "$N_Iterations" ]; then
                echo "ERROR! Number of iterations must be larger than 50"
                exit
        fi

let "TotalIterations=$N_Iterations+$initialcounter"
echo "Final Iteration will be $TotalIterations"



cp $lastcasefile backup_$lastcasefile
cp $lastdatafile backup_$lastdatafile


counter=$initialcounter


while [ "$counter" -lt "$TotalIterations" ]
do
        mv $lastcasefile temp1.cas.gz
        mv $lastdatafile temp1.dat.gz

        fluent 3ddp -i partition.jou

        #while [ ! -e "temp2.dat.gz" ]
        #  do
        #     sleep 1
        #  done

        mv temp1.cas.gz $lastcasefile
        mv temp1.dat.gz $lastdatafile
```

```
        echo "Partition successful at $counter"
        fluent 3ddp -t8 -i iteration.jou



        let "counter=$counter+50"             #change to +50



        lastcasefile=`echo "$initialname"-"$counter".cas.gz`
        lastdatafile=`echo "$initialname"-"$counter".dat.gz`
        mv temp2.cas.gz $lastcasefile
        mv temp2.dat.gz $lastdatafile
done
```

```
; Written by S Kalogerakos
; journal to plot liquid height

file read-case-data temp.cas.gz

plot plot yes liqheight.txt yes no no mixture y-coordinate yes 1 0 liq-vof ()

quit
quit
quit
quit
exit
```

```
; Written by S Kalogerakos
; journal to automatically partition case file

/define/user-defined/compiled-functions
compile
"libudf"
yes
"UDF_DOSE_UPDATE.c"

/define/user-defined/compiled-functions load libudf



/file/read-case-data temp1.cas.gz

(cx-gui-do cx-activate-item "MenuBar*ParallelMenu*Partition...")
(cx-gui-do cx-activate-item "Partition Grid*PanelButtons*PushButton1(Partition)")
(cx-gui-do cx-activate-item "Partition Grid*PanelButtons*PushButton1(Close)")



/file/write-case-data temp2.cas.gz yes
quit
```

```
quit
quit
quit
quit
exit yes
```

```
; Written by S Kalogerakos
; journal to write profiles of pressure during simulations
file

read-case-data
N_WASP_C36_water-15.194051.cas.gz

/file/write-profile pressure-01.0m.prof (line-1m) absolute-pressure () ()
/file/write-profile pressure-01.5m.prof (line-1.5m) absolute-pressure () ()
/file/write-profile pressure-02.0m.prof (line-2m) absolute-pressure () ()
/file/write-profile pressure-02.5m.prof  (line-2.5m) absolute-pressure () ()
/file/write-profile pressure-03.0m.prof  (line-3m) absolute-pressure () ()
/file/write-profile pressure-03.5m.prof  (line-3.5m) absolute-pressure () ()
/file/write-profile pressure-04.0m.prof  (line-4m) absolute-pressure () ()
/file/write-profile pressure-04.5m.prof  (line-4.5m) absolute-pressure () ()
/file/write-profile pressure-05.0m.prof  (line-5m) absolute-pressure () ()
/file/write-profile pressure-05.5m.prof  (line-5.5m) absolute-pressure () ()
/file/write-profile pressure-06.0m.prof  (line-6m) absolute-pressure () ()
/file/write-profile pressure-06.5m.prof  (line-6.5m) absolute-pressure () ()
/file/write-profile pressure-07.0m.prof  (line-7m) absolute-pressure () ()
/file/write-profile pressure-07.5m.prof  (line-7.5m) absolute-pressure () ()
/file/write-profile pressure-08.0m.prof  (line-8m) absolute-pressure () ()
/file/write-profile pressure-08.5m.prof  (line-8.5m) absolute-pressure () ()
/file/write-profile pressure-09.0m.prof  (line-9m) absolute-pressure () ()
/file/write-profile pressure-09.5m.prof  (line-9.5m) absolute-pressure () ()
/file/write-profile pressure-10.0m.prof  (line-10m) absolute-pressure () ()
/file/write-profile pressure-10.5m.prof  (line-10.5m) absolute-pressure () ()
/file/write-profile pressure-11.0m.prof  (line-11m) absolute-pressure () ()
/file/write-profile pressure-11.5m.prof  (line-10.5m) absolute-pressure () ()
/file/write-profile pressure-12.0m.prof  (line-12m) absolute-pressure () ()

quit
quit
quit
quit
exit
```

```
# SCRIPT written by S Kalogerakos
# To be used on a SGE environment
#$ -N Helix
# request a total of 4 processors for this job (2 nodes and 2 processors per node)
#$ -pe mpi 2
# combine PBS standard output and error files
#$ -j y
# specify your email address
#$ -m e
#
#$ -cwd
```

```
#$ -S /bin/sh
module load /usr/lib64/mpi/gcc/openmpi/lib64/openmpi

#echo "$TMPDIR/machines" > machines.txt
#fluent 3ddp -g -ssh -sge -t10 -sgepe fluent_pe 10 -i WASP.jou
#fluent 3ddp -g -ssh -sge -t10 -sgepe fluent_pe 10 -sgeq parallel.q -pethernet -mpi=net -i WASP.jou
fluent 3ddp -g -ssh -sge -t$NSLOTS -cnf=./machines -mpi=net -i WASP.jou
#fluent 3ddp -g -sge -t$NSLOTS -cnf=./machines -mpi=net -i WASP.jou
```

Appendix E

Sun Grid Engine configurations

E.1 Preliminary steps

Due to the availability of Desktop PCs with multi-core CPUs and also hyperthreading, and because of the need of using parallel processing in CFD simulations, it was decided to build a pool of computers, place them on a network and harness their processing power by installing a Sun Grid Engine configuration. The operating system used on all machines was Opensuse 11.2, but any Linux installation should be equivalent.

When running a simulation, the data files need to be read and written on a common shared drive, therefore it is necessary to mount a common drive. An easy way of accomplishing that is to install Samba. These are the steps:

- Create folder /mnt/W to link to /home/<username> on the node that will be chosen to save all data
- Use Yast to Set Samba Server
- Add users as follows:
  - useradd -c "<name surname>" "<username>" // if it does not exist already
  - smbpasswd -a <username>
- Add the following line to /etc/fstab

```
//<machine name>/users/<username> /mnt/W/<usrname>
        cifs rw,noperm,users,exec,dev,suid,credentials=/home/<username>/bin/.sambapasswd 0 0
```

where .sambapasswd is a file with the following content:

```
username=<username>
password=<password>
```

After the installation of Samba, it is necessary to set up SSH keys so that the nodes talk to each other in a secure manner but without having to manually input the password each time. The following steps should be followed:

- Set up OpenSSH and openssh server on all machines that need to work as nodes/hosts.
- Add user sgeadmin:users on all hosts/nodes
- Assuming machine1 and machine2 are the two nodes, then type in machine1 for the following users (root, sgeadmin, <user>):

  ```
  ssh -X machine2
  ```

  and connect with the required password. This will create a .ssh dir in your home directory with the proper permissions. After that exit back to machine1.
- On machine 1, type:

  ```
  ssh-keygen -t dsa
  ```

  This will prompt for a secret passphrase. Then two files called id_dsa and id_dsa.pub will be created in /home/<username>/.ssh dir. Note: it is possible to just press the enter key when prompted for a passphrase, which will make a key with no passphrase.
- In machine1 execute the following:

  ```
  scp ~/.ssh/id_dsa.pub machine2:~/.ssh/authorized_keys2
  ```

  Copy the id_dsa.pub file to the other host's .ssh dir with the name authorized_keys2. If authorized_keys2 exists already (check first), then add the contents of id_dsa.pub manually on a new line of the old authorized_keys2 file on machine2.
- Check by executing on machine1:

```
ssh -X machine2
```

It should be possible to access machine2 with no password request.

E.2 Sun Grid Engine installation

The user is advised to download the latest SGE version from their website

`http://www.sun.com/software/sge/get_it.jsp`.

It should be decided which machine will be master host and which will be nodes. The best solution is to use a normal machine as master node, and the (fast!) machines as execution nodes. Of course it is also possible to have the master node as execution node.

As root type:

```
mkdir -p /opt/sge
cd /opt/sge
```

Unzip the installation files.

```
gunzip Linux24_amd64/tar/sge-6_2-bin-linux24-ia64.tar.gz | tar xvpf -
          //Linux (Itanium platform) binaries for the 2.4 and 2.6 kernel
gunzip Linux24_amd64/tar/sge-6_2-bin-linux24-x64.tar.gz  | tar xvpf -
          //Linux binaries for the 2.4 and 2.6 kernel
```

Or the appropriate files to the architecture.
Repeat the following two steps for users: root, sgeadmin and any users that will be submitting
- - - - - - - - - - - - - - - - - -
1. Add the following line to ∼/.profile
SGE_ROOT=/opt/sge; export SGE_ROOT

2. Modify (or create) ∼/.bashrc
source ∼/.profile

- - - - - - - - - - - - - - - - - -

then run in terminal
source ∼/.profile

Follow the installation from the GUI as in
`http://wikis.sun.com/display/gridengine62u5/Custom+Installation`
Select Qmaster and Execution Host. Choose custom installation. Unselect Shadow host and Berkeley db host. Choose the following parameters:

```
admin user: sgeadmin
qmster host: machine name for master (let's say machine1)
grid engine root directory: /opt/sge
cell name: leave default
cluster name: can be left as it is
```

All the rest can be left as it is.
Unselect JMX.

In Spooling configuration, choose Classic Spooling method.

In Select Hosts, select for masternode also exec if you want to execute on that node. It should say reachable.

Click install and print the information page at the end. Close the GUI wizard.
Add the following lines to ∼/.profile
. /usr/share/modules/init/bash
. /opt/sge/default/common/settings.sh

Execute the following in the commandline
source ∼/.profile
qconf -ah machine2 machine3 #this is to add machine2 and machine3 as nodes

Check that the daemon is running. Type the following command:

ps -ef | grep sge

You should see output similar to the following example.

root 439 1 0 Jun 2 ? 3:37 /opt/sge/bin/sge_qmaster

If you don't, then start it manually:

```
$SGE_ROOT/$SGE_CELL/common/sgemaster  start
```

Now go to each node (eg machine2 and machine3 in our case) and carry out the following:
- - - - - - - - - - - - - - - - - -
su mkdir -p /opt scp -r machine1:/opt/sge /opt # this command should be carried out without asking for password prompt. If it asks for password, then you need to carry out the steps at the beginning of the tutorial regarding creating ssh keys - - - - - - - - - - - - - - - - - -

Add the following lines to ∼/.profile
SGE_ROOT=/opt/sge; export SGE_ROOT
. /usr/share/modules/init/bash
. /opt/sge/default/common/settings.sh

Modify (or create) ∼/.bashrc
source ∼/.profile SGE_ROOT=/opt/sge; export SGE_ROOT

Most of the installation of the execution host is described in

```
http://wikis.sun.com/display/gridengine62u5/How+to+Install+Execution+Hosts
```

from point 6 onwards, with the command
./inst_sge -x
Choose default cell name. Also check that execd will start automatically at start time. All other settings should be default.

Once installation is complete, check that sge_execd is running as a process. ps -aef — grep sge_execd If it doesn't, then restart it with

```
$SGE_ROOT/$SGE_CELL/common/sgeexecd  start
```

- -
Various details of the parallel queue setup:

```
Parallel Queue
Start up qmon
1. Parallel environment configuration
Add mpi
Slots <up to max processors>
Start Proc args  /opt/sge/mpi/startmpi.sh -catch_rsh $pe_hostfile
Stop Proc args /opt/sge/mpi/stopmpi.sh
Allocation rule   $round_robin (overall balancing)
          $fill_up (fill each node first)
  $pe_slots (use only one node)
Check Control slaves and Accounting summary

2. Queue control-> Cluster Queues -> Add
Name: parallel.q
Hostlist: @allhosts
Atributes for Host/Hostrgoup:
@/
@allhosts
<node1>
<node2>
<etc>
Click on @/
General configuration
Sequence nr. 0
Processors <total number of processors>
Shell /bin/bash
Shell Start Mode script_from_stdin
Initial state disabled
Slots           <up to max processors>
Batch and Interactive
Parallel environment
Add mpi to referenced PEs
Click on machine1
```

```
General Configuration
Sequence nr. 1
Processors <total number of processors on that machine>
Slots <total number of slots allowed on that machine>
Click on machine2
General Configuration
Sequence nr. 2
Processors <total number of processors on that machine>
Slots <total number of slots allowed on that machine>



Command-line commands:
Add queue:
qconf -Aq name.q

See all queues:
qconf -sql

Check hosts and load
qhost

Submit a job
qsub <scriptname>

Check jobs on hosts for user <username>
qconf -f | grep <username>
```

OPENMPI

Add the following line to ∼/.profile

```
module load /usr/lib64/mpi/gcc/openmpi/lib64/openmpi
                    #check paths on machine
```

E.3 Parallel Ansys Fluent

Make sure that machine1 can communicate with machine2 via rsh with no request password. This is the quickest way for cluster simulations. From machine1 try "rsh machine2 fluent -t0 -v". If it takes too long or times out, then you need to carry out the following steps:
1) Add file .rhosts to home directory in machine2, containing on each line the hostnames it needs to allow access from. In this case machine1.
2) Try to connect again. If it fails, then add (as root) the following lines to /etc/hosts.equiv on machine2:
machine1 +
+@machine1
3) Try again. If it fails, then you need to open the appropriate ports firewall, both outgoing from machine1 and incoming from machine2.

Example of a bash script

```
#$ -N Helix       #Name of case
# request a total of 16 processors for this job
#$ -pe mpi 16
# combine PBS standard output and error files
#$ -j y
# specify your email address
#$ -m e
#$ -M user@cranfield.ac.uk
#$ -cwd               #use current working directory
#$ -S /bin/sh
module load /usr/lib64/mpi/gcc/openmpi/lib64/openmpi
        #load mpi - check paths are correct

fluent 3ddp  -g  -sge -t$NSLOTS  -cnf=$TMPDIR/machines   -i journal.jou
```

You may also specify your own file *machines* which should have the following format:

```
machine1
machine1
machine2
```

This will use two CPU instances of machine1 and one CPU instance of machine2.

Nomenclature

Roman Symbols

ΔP	pressure drop	Pa
ΔP_k	pressure correction term for phase k	Pa
A	flow cross-section area	m^2
B_{fk}	body term due to gravity force for phase k	Pa/m
C_V	velocity profile coefficient	
C_{Vk}	velocity profile coefficient for phase k	
D	pipe diameter	m
D_k	hydraulic diameter of phase k	m
E	energy	$kg \cdot m^2/s^2$
F	force	$kg \cdot m/s^2$
f_k	friction factor for phase k	
Fr	mixture Froude number	
Fr_{cr}	critical Froude number	
g	gravitational acceleration	m/s^2
h	phase height	m
l	liquid length	m

Nomenclature

M_{ki}, τ_{ki}	interfacial stress term in momentum of phase k	Pa/m
M_{kw}, τ_{kw}	wall stress term in momentum of phase k	Pa/m
n	direction normal to face	
p	pressure	Pa
$Q(t)$	mass flow rate	kg/s
Re	Reynolds number	
S_W	wetted perimeter	m
u	flow velocity	m/s
u_d	drift velocity	m/s
U_M	mixture superficial velocity	m/s
u_t	translational velocity	m/s
u_k	velocity of k-phase	m/s
u_{SG}	superficial velocity of gas phase	m/s
u_{SL}	superficial velocity of liquid phase	m/s

Greek Symbols

α_k	volume fraction of phase k	
β	angle of inclination of the pipe	rad
ϵ_P	price elasticity of demand	
η	intensive quantity	
Γ_k	mass transfer term for phase k	$kg/(m^3 s)$
κ_i	interfacial curvature	
λ_L	slug length	m
μ	viscosity	$kg/(ms)$

ω	slug frequency	s^{-1}
Φ	dimensionless slug frequency	
ϕ	frequency	s^{-1}
ρ	density	kg/m^3

Subscripts

f	liquid film in slug unit
G	gas phase
GW	gas wall
i	interfacial
k	phase k, L for liquid, G for gas
ki	interfacial and phase k
kW	phase k and wall
L	liquid phase
LW	liquid wall
s	slug body in slug unit
u	slug unit