Technische Universiteit
Eindhoven
University of Technology

Department of Applied Physics

Machine Learning for Closure Models in Multiphase-Flow Applications

Master Thesis

J.F.H. Buist

Supervisors
ir. Y. van Halder
dr.ir. B. Sanderse
prof.dr.ir. B. Koren
prof.dr.ir. G.J.F. van Heijst

Report Number
R-1959-A

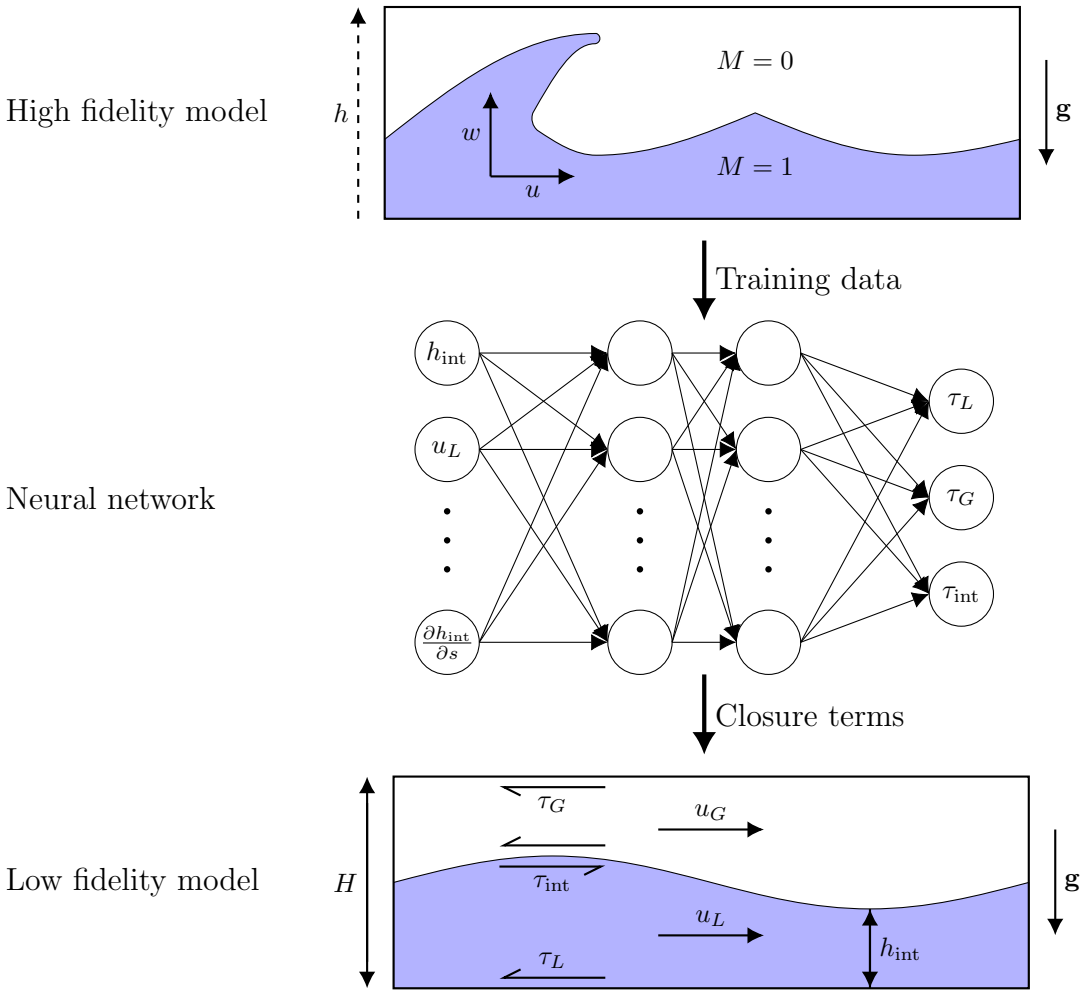
February 2019

CWI
Centrum Wiskunde & Informatica

Abstract

Multiphase flows are described by the multiphase Navier-Stokes equations. Numerically solving these equations is computationally expensive, and performing many simulations for the purpose of design, optimization and uncertainty quantification is often prohibitively expensive. A cheaper, simplified model, the so-called two-fluid model, can be derived from a spatial averaging process. The averaging process introduces a closure problem, which is represented by unknown friction terms in the two-fluid model. Correctly modeling these friction terms is a long-standing problem in two-fluid model development.

In this work we take a new approach, and learn the closure terms in the two-fluid model from a set of unsteady high-fidelity simulations conducted with the open source code Gerris. These form the training data for a neural network (NN). The NN provides a functional relation between the two-fluid model's resolved quantities and the closure terms, which are added as source terms to the two-fluid model. With the addition of the locally defined interfacial slope as an input to the closure terms, the trained two-fluid model reproduces the dynamic behavior of high fidelity simulations better than the two-fluid model using a conventional set of closure terms.



Preface

This thesis was submitted in partial fulfillment of the requirements for the degree of Master of Science in Applied Physics from the Eindhoven University of Technology. The work presented in this thesis was carried out in the Scientific Computing research group of the Centrum Wiskunde & Informatica (CWI) in Amsterdam. Here, I encountered an environment which challenged me to push my boundaries, and dive into new things at every corner of my broad research topic.

I am indebted to my daily supervisors at CWI, Yous van Halder and Benjamin Sanderse. I want to thank Yous for the introduction to multiphase flow CFD, and the engaging discussions on neural networks and new things to try with them. Benjamin I thank for his careful look at my work, his sharp and detailed feedback, and good suggestions for further steps in the project.

Thank you to my supervisor Barry Koren, from the Scientific Computing research group of the department of Applied Mathematics at the Eindhoven University of Technology, for introducing me to CWI. Thank you also for the talks on the way back from Cascade, and your engagement with my project and my future.

My supervisor GertJan van Heijst, from the Turbulence and Vortex Dynamics research group of the department of Applied Physics at the Eindhoven University of Technology, I thank for the critical questions on the physical aspects of my work. Thank you for taking on this new and different topic, and keeping it in line with the requirements for a degree in physics.

Jurriaan Buist
Eindhoven, February 2019

Contents

Chapter 1

Introduction

1.1 Multiphase flow

Gas-liquid multiphase flow is a problem of interest in the oil and gas industry. For example, oil and gas are often transported together in long pipelines from remote fields to an offshore platform or an onshore plant [1]. Ever more remote and deep underwater fields are being tapped as more easily accessible fields are depleted. Transport of liquefied natural gas (LNG) is another hot topic in the oil and gas industry; it is currently greatly on the rise [2], [3]. During the loading and unloading of a carrier, the carrier will be only partially filled with liquid[1] so that external waves may induce sloshing inside the carrier [2]. This in turn may induce harmful rolling motion of the ship [4] or damage the thermal insulation [5].

For these multiphase flows, accurate numerical models have been proposed [6], [7]. However, a trade-off will always have to be made between model accuracy and computational expense. This difficult trade-off has persisted to today.

Some applications require the numerical model to be solved repeatedly. An example is *Uncertainty Quantification* (UQ), which has become an active area of research in recent years. This field (see e.g. [8]) is concerned with determining the effect of model error or input parameter variation on the outcome of the simulation. Different algorithms exist with varying requirements on the number of model evaluations.

Another example where repeated model evaluations are required are optimization problems. Often, in order to find the optimal values of a set of performance indicators, it is desired to calculate these for different values of a potentially large set of tunable parameters. Only with a large number of model evaluations can one map all the local and global minima or maxima in the performance reliably.

For these problems where many potentially expensive model evaluations are required, it is necessary to resort to low fidelity models. These make a trade-off between accuracy and computational efficiency that leans more towards computational efficiency. In this thesis we will study low-fidelity models for the multiphase applications mentioned above.

1.2 Low fidelity model for two-phase pipe flow

One way to create a cheaper model is to reduce the dimensionality of the model. In the case of pipe flow, we are mainly interested in variation of the averaged flow properties along the pipe's axial direction. We therefore reduce a full three-dimensional (3D) model to a one-dimensional (1D) model. However, the effect of the flow structure in the cross-sectional plane - which the 1D model does not solve for - on the 1D flow variables, must be modeled in some way. We add so-called *closure terms* to the 1D model which approximate this effect. A different well-known example of closure terms is the concept of a turbulence closure model, which models the effect of unresolved turbulence on the averaged flow.

The focus in this project is the 1D stratified two-fluid model, a simplified model for stratified liquid-gas flow in a pipe or square cross-section duct geometry (Figure 1.1). In the 1D two-fluid model the velocity fields are not resolved along the direction normal to the duct wall; they only vary along the direction of flow through the duct (s in the figure). The wall and interface stresses, which are important terms in

[1] In order to liquefy the natural gas, so that it takes up less volume, it is cooled to low temperatures. Due to imperfect thermal insulation some of the LNG will evaporate and remain in the tank as a gas phase. Loading and unloading of a carrier may take up to 24 hours, at an offshore platform or at an onshore facility.

the governing equations, depend on the unresolved wall-normal velocity profile slope. Therefore the 1D two-fluid model requires closure terms for the wall and interface stresses.

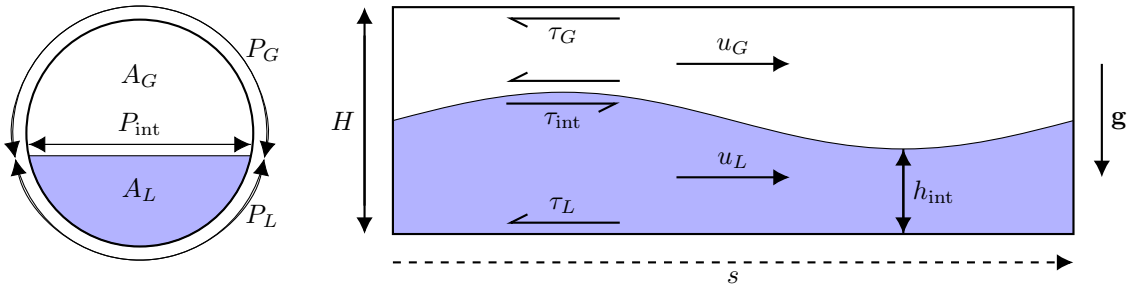


Figure 1.1: A schematic of the 1D two-fluid model for a circular pipe. Left: a cross-section, representing a single cell for which the areas A_L, A_G and the velocities u_L, u_G are defined. Right: a string of these cells together forming a length of the pipe. The total friction terms are calculated as products of the mean stresses τ_L, τ_G, τ_{int} and the perimeters over which they act.

For laminar single-phase flow, analytical closure relations are easily found (based on fully developed steady state flow). For turbulent single-phase flow these cannot be found and conventionally closure relations based on empirical data are used, of which a multitude exists (see section 3.4). They are mostly determined for a circular pipe geometry, but this equivalence is not exact and the error is large for the laminar case. The friction factors are usually determined for steady, fully developed flow. They all have a specific range of Reynolds numbers for which they are said to hold. The functional forms of common friction factors are not very insightful; they are simply fit to match data.

For two-phase flow empirical relations are harder to construct because the pressure drop in fully developed flow is now determined by the sum of three different stresses, instead of a single stress in the case of single-phase flow. Additionally, the interfacial stress is complicated to model; depending on the relative fluid viscosities the interface may be approximated as a free surface for the liquid and as a no-slip wall for the gas. The effect of interfacial waves on the interfacial stress is hard to quantify. Furthermore, the possibility of backflow in inclined pipes complicates the situation. The same average velocity may result from a very different flow profile with very different velocity gradients (even reversed), so that expressing the friction closure terms as a function merely of the average velocity becomes dubious, and a correction is hard to find empirically. Additionally, the vast majority of the literature concerns circular pipe flow and it can be hard to find good closure terms for a different geometry.

For two-phase laminar flow, analytical closure relations may be found [9]. In a 2D channel geometry elegant exact expressions are the result, but for pipe flow the expressions are not practical; they require numerical integration of complex integrals. It is great that these analytical solutions exist, but they hold only for steady state, fully developed, smoothly stratified (not wavy) flow.

In conclusion, modeling wall and interface stresses in terms of resolved quantities (averaged velocity fields), either by fitting experimental data or physical arguments, is a difficult task. Our proposed alternative is to the extract friction factors to close the 1D two-fluid model from high-fidelity simulation data. If we have highly resolved 2D or 3D simulation data, the wall and interface stresses can be calculated exactly. From this one can determine a closure relation relating the wall and interface stresses to the averaged flow variables which *are* resolved in the two-fluid model. The flow need not be steady state and fully developed; this operation is possible in any kind of flow conditions. With a simulation code the geometry, fluid properties, and initial conditions can be adjusted relatively easily. This would have the advantage of enabling friction factor calculation more specific to the geometry and flow parameters of interest to the user of the two-fluid code. More generally this approach can be useful for the closure of other low fidelity flow models; it was pioneered by Ma et al. [10], [11] for bubbly vertical channel flow.

Different strategies can be conceived for the determination of the closure relations from high fidelity simulation data. A straightforward approach is to use linear regression. Using this approach we would have to assume a functional relation for the closure term, of which the unknown coefficients could be fitted to the data. Having to choose a functional relation could be an advantage, since in defining it we can use our knowledge of the physics of the problem. However, it could also be a disadvantage, since we are in fact already determining the form of the solution, while the data might suggest a different functional form.

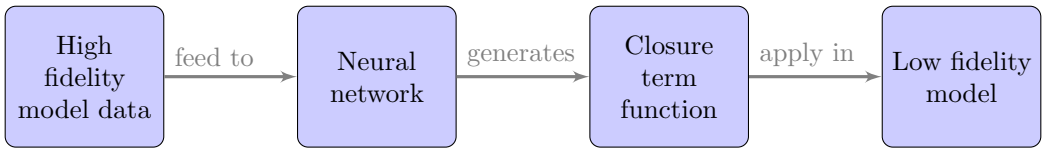


Figure 1.2: The general idea, applicable not only to the low fidelity 1D two-fluid model but to any low fidelity flow model requiring closure.

Therefore we propose to use instead artificial neural networks (ANN) to perform the fitting. The general approach is summarized in Figure 1.2. Using a neural network, we need only to specify a network architecture. The network training algorithm automatically finds a (possibly complex nonlinear) function which fits the data well.

1.3 Neural networks

Artificial neural networks have made a name for themselves in recent years through remarkable results in applications such as image classification [12], natural language processing [13], and learning to beat the masters at the complex Chinese chess-like game of 'Go' [14]. This diversity of applications illustrates their remarkable ability to represent complex relations between arbitrary inputs and outputs. Training a neural network is a form of function fitting, but with a complex nonlinear structure which allows the neural network to approximate a wide range of functions[2].

An artificial neural network is a computational graph of connected nodes, which individually perform straightforward operations. They generally take a weighted sum of the values of the incoming connections and apply some non-linear activation function to this sum. The original incarnation of these nodes was called the 'perceptron', by researchers who were trying to model the human brain [16]. The power of neural networks lies in the concept of connecting these nodes in large and diverse networks, in ways which model the network of neurons in the brain. A schematic of a simple (fully connected, feed-forward, non-convolutional) network structure is shown in Figure 1.3. The schematic shows one possible structure; the number of hidden layers and number of nodes in each layer can be chosen freely (besides more imaginative alterations).

Essential to their utility is their ability to be trained efficiently, via the backpropagation algorithm [17]. A loss function defined at the output layer measures the current predictive performance of the network for a given set of inputs, by comparing the difference between output training data and the model output. The backpropagation algorithm efficiently calculates the gradient of this loss function to the weights of the node connections, which are the free parameters of the network which determine the degree to which

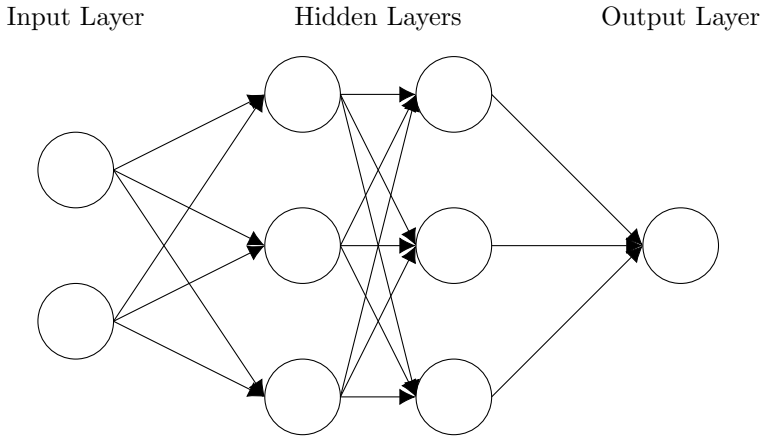


Figure 1.3: A schematic of a neural network.

[2]A neural network with a single hidden layer can approximate almost any function with an arbitrary degree of accuracy, depending on the number of nodes. Neural networks are a class of *universal approximators* [15].

a signal is passed on between two consecutive nodes. Using this gradient an optimization algorithm can tune the weights; this is called the 'training' or 'learning' of the network, in analogy to the brain.

Some recent introductions to neural networks are given by [18], [19]. An older book (1996 first edition) by the authors of the MATLAB shallow neural network implementation is [20].

1.4 Neural networks in fluid dynamics

As noted above, the need for closure terms for the shear stresses in 1D flow is analogous to the need for turbulence closure terms in insufficiently resolved turbulent flow. Neural networks have already been applied successfully in this area. Sargini et al. [21] used a neural network to create a subgrid scale (SGS) model for a Large Eddy Simulation (LES), which reproduces the dynamics of LES using an expensive SGS model (Bardina's scale similar (BFR) SGS model), at a lower computational cost. Their neural network output a turbulent viscosity coefficient (a.k.a. eddy viscosity) as a function of the gradients of the spatially averaged velocities, and products of the velocity fluctuations. Their learned closure term produced good results, for Reynolds numbers within and close to the range of Reynolds numbers used in the training data. Moreau et al. [22] used neural networks fed by pseudospectral DNS data of turbulent flow to model the subgrid variance in the concentration of an advected species for an LES, but did not test simulations with the learned closure term. An eddy viscosity coefficient for atmospheric flow over an urban boundary layer was obtained using a neural network by Esau [23].

Yarlanki et al. [24] tried an unusual inverse approach. The parameters of the $k-\epsilon$ turbulence model are to be determined and form the inputs to an ANN, and the differences between CFD results and experimental results are the outputs of the neural network. The neural network learns the error between simulation and experiment as a function of the turbulence model parameters, and thus the parameter set which yields the smallest error can be found indirectly. The found $k-\epsilon$ model parameters reduced the discrepancy between simulation and experiment significantly compared to standard parameters, for their specific test case. Tracey et al. [25] reproduced the Spalart-Allmaras turbulence closure model (without a specified functional form) from the output of simulations done with this closure model. They report very promising results, but stress the importance of choosing appropriate ANN inputs and cost function. Gamahara and Hattori [26] recently used DNS to obtain a functional relation for the Reynolds stress tensor directly, which shows performance close to that of a Smagorinsky SGS model.

In multiphase flow applications, the use of neural networks to identify closure terms is still in its infancy. One existing example is Lu et al. [27], [28], who trained a neural network with data from micro-scale DNS simulations of a gas-solid mixture under influence of a shock, to provide closure relations for the particle-particle and gas-particle interactions for use in coarse macro-scale simulations. The main inspiration for the current project is taken from Ma et al. [10], [11]. They consider dispersed liquid-gas flow, for which they take a reduced order model, averaged along the streamwise direction and one spanwise direction, and bounded by periodic boundaries in their 3D DNS simulations. The closure relations required for their simplified model are for the wall-normal liquid flux, the average of the product of the streamwise and wall-normal velocities, and the average surface tension. They obtain these using neural networks fed by their 3D DNS data, with good results. They also tried linear regression with a predetermined functional relation, which produced similar results to the neural network, but is deemed by them to be a far less general approach. Gibou et al. [29] review numerical methods for simulating multiphase flow and machine learning applied to computational physics. Their review confirms that there is only limited existing work connecting neural networks and multiphase flow, and raises a number of questions to be tackled.

There has been some work connecting pipe flow stress closure terms (in the form of friction factors) and neural networks. Neural networks were trained to replicate different implicit empirical friction factor correlations by [30]–[38]. For the studies which generate an explicit form of the Colebrook-White friction factor equation (see section 3.4), the inputs are the Reynolds number and the relative roughness and the output is the friction factor. The training data consists of iterative solutions of the Colebrook-White equation for different Reynolds numbers and relative roughnesses. The usefulness of this work is limited though, since simpler explicit approximations of the Colebrook-White equation with a smaller error than the deviation of the Colebrook-White equation from the empirical data have existed for some time (e.g. Haaland (1983) [39]).

An important question when using neural networks in fluid dynamics is if we can mold them to be

more than just a fit of the data, i.e. if their output can incorporate known physical principles. This is a challenging and outstanding problem, but one good illustration of this principle was given by Ling et al. [40]. They remarked that any scalar flow variable, such as pressure or velocity magnitude, will be invariant to rotations, reflections, or translations of the frame of reference. Without special care a machine learning algorithm output will not adhere to this perfectly. Ling et al. describe two ways to enforce rotational invariance on trained neural network outputs in the context of turbulence modelling. One way is to simply feed the neural network training data to which a number of different rotations have been applied. But a far more efficient way is to use as inputs to the neural network only quantities which are themselves rotationally invariant. Doing this naturally ensures that the output is rotationally invariant. Another strategy they employ is making the inputs non-dimensional. This ensures that the output is non-dimensional as well, so that if an output variable is chosen that should indeed be non-dimensional, there will be no problems of dimension in the output functional relation.

Overall, good results are reported in the literature on fluid dynamics with neural networks. In the range of the training data nonlinear relations are reproduced accurately, and simulations using learned closure terms produce results close to the original data. Currently the advantage of neural networks lies in their application to specific cases, to which they may be applied relatively easily to learn correlations specific to a certain set of conditions. However, their extrapolating qualities are still limited, and improvements in choice of network inputs and structure are needed if neural networks to one day outperform e.g. conventional closure terms in general cases.

Particularly in turbulence closure models neural networks have been applied successfully. In multiphase flow the first steps have been made, but the approach is still new, especially for the two-fluid model that we consider in this work. When training neural networks to produce closure terms many difficult choices and trade-offs have to be made, and a lot of work is left to be done before the level of the highly problem-specifically optimized convolutional neural networks such as those used in image classification [12] is reached.

1.5 Project plan

It was discussed in section 1.2 that existing friction closure terms for 1D two-phase flow are lacking. Cases for which analytical solutions cannot be found are usually closed by experiment. But conclusive empirical relations are hard to find in some cases. Therefore in this project the neural network approach to finding closure terms will be applied to this problem. We will train a neural network on high-fidelity computational model data. With this method, the aim of this project is to combine the easy general applicability and accuracy of a high fidelity model with the low computational cost of a simplified model. This would alleviate the existing necessity for trade-offs between accuracy and computational efficiency in the simulation of multiphase flow.

Figure 1.4 shows a schematic of the project plan. We restrict ourselves to (periodic boundary) laminar stratified channel flow, but the plan is applicable to circular pipe flow with some modifications. The figure shows two cases that are discussed in this work: smooth, fully developed, steady state flow and wavy, transient flow. High fidelity simulations can be conducted for both cases with the open-source code Gerris [6], which solves the full viscous incompressible Navier-Stokes equations. We start with smooth, fully developed, steady state channel flow, for which analytical solutions and closure relations are available for channel geometries (see section 3.4). These relations can be used to assess the accuracy of the high fidelity model simulations. Furthermore, we will train neural networks on steady state data and compare this to the analytical closure relations, so that the neural network can be tuned and the approach validated.

For wavy, transient flow, existing closure relations are lacking, and we will proceed to directly train a neural network based on the high-fidelity simulation results, with the same architecture that will have already been validated for the case mentioned above. The new closure terms can be validated by plugging them into the 1D two-fluid code (our low fidelity model) and evaluating if with these closure terms its behavior is similar to the 2D 'truth' given by the high fidelity model.

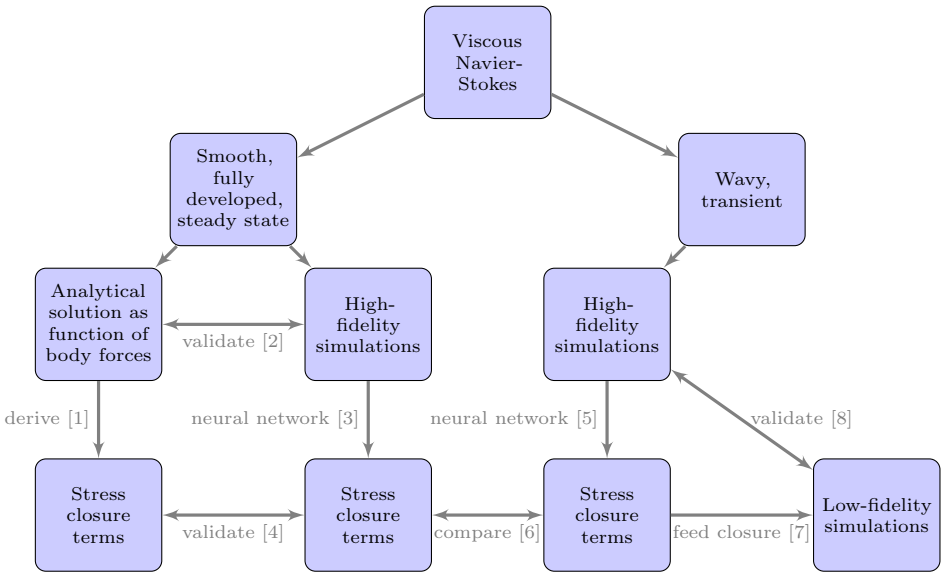


Figure 1.4: A flow chart of the project structure. We analyze two different sets of solutions of the Navier-Stokes equations, one of which has analytical solutions. These are used to validate the high-fidelity (Gerris) simulations and the extraction of closure terms using a neural network. Afterwards the same architecture is applied to construct closure terms for wavy, transient flow, which are then tested in a low-fidelity model.

1.6 Structure of the report

In chapter 2 we outline the physics of stratified multiphase flow, and our high fidelity computational model which we use to model these physics. Chapter 3 presents the 1D stratified two-fluid model and casts it into a form specialized to 2D channel flow. This chapter includes a detailed discussion of the closure terms for the wall and interface stresses and existing empirical relations for them.

In chapter 4, we analyze the stability and well-posedness of our flow problem. The employed two-fluid model turns out to only be well-posed under certain conditions. A linear stability analysis of the two-fluid model and of a 2D model for the same flow problem yields predictions for the propagation and amplification of small wave-like disturbances. These predictions are used to validate our computational models for the inviscid case, and to examine fundamental restrictions in getting the 1D model to emulate the dynamics of the 2D model. The inviscid dispersion relations may be used for qualitative explanations of phenomena occurring in viscous simulations in later chapters.

In chapter 5 we validate our high and low fidelity models for smoothly stratified (non-wavy) viscous flow. We discuss the difficulties in extracting the required training data for our neural network.

Chapter 6 provides a brief introduction to neural networks, before applying them to the case of fully developed, steady, laminar 2D channel flow. We discuss the choices made in defining our neural network and validate them using the analytical closure terms. Finally, we apply our neural network to the case of wavy, unsteady flow in chapter 7. We test our learned closure terms in the 1D two-fluid model.

Chapter 2

2D and 3D Multiphase Flow Models

2.1 Introduction

In this chapter we will discuss the physics and solution scheme of our high-fidelity multiphase flow model, which resolves the flow in two or three spatial dimensions. We will first go over the physical laws of fluid dynamics, paying special attention to their application to multiphase flow. These laws form the basis of our computational model. Afterwards we will see a method for solving the equations corresponding to these physical laws on a computational grid. We use the open source code Gerris [6], [41] as our high-fidelity model.

2.2 The Navier-Stokes equations for fluid flows

In the following, we employ the continuum hypothesis. This means averaging out individual interacting molecules into a continuous fluid with continuously defined properties such as ρ (density) and $\mathbf{u}$ (velocity). This allows us to convert the principles of mass and momentum conservation into a set of continuous equations; the Navier-Stokes equations. See also textbooks such as [42], [43].

2.2.1 Mass conservation

In any classical[1] physical system, mass must be conserved. This means that the change in time of the mass inside a control volume V, illustrated in Figure 2.1, is equal to the mass flowing in and out of the control volume at its boundaries. For the flow of a continuous fluid this can be written as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho \,\mathrm{d}V = -\oint_S \rho\mathbf{u}\cdot\mathbf{n}\,\mathrm{d}S, \tag{2.1}$$

where $\mathrm{d}/\mathrm{d}t$ is a derivative with respect to time, $\int_V \mathrm{d}V$ is an integral over the control volume V, $\rho = \rho(\mathbf{x}, t)$ is the density ($\mathbf{x}$ representing coordinates in three-dimensional space and t representing time), $\oint_S \mathrm{d}S$ is an integral over the closed surface S bounding V, $\mathbf{u} = \mathbf{u}(\mathbf{x}, t)$ is the velocity vector and $\cdot\,\mathbf{n}$ is the inner product with the vector normal to the surface S, pointing outward.

Since the control volume V is fixed in space, the time derivative on the left hand side can be brought inside the integral using Leibniz' rule. For the right hand side we can apply the divergence theorem, or Gauss' theorem, to obtain

$$\int_V \frac{\mathrm{d}\rho}{\mathrm{d}t}\,\mathrm{d}V = -\int_V \boldsymbol{\nabla}\cdot(\rho\mathbf{u})\,\mathrm{d}V. \tag{2.2}$$

In the limit $V \to 0$ the above implies

$$\frac{\partial\rho}{\partial t} + \boldsymbol{\nabla}\cdot(\rho\mathbf{u}) = 0. \tag{2.3}$$

This is a specific form of the generic conservation law

$$\frac{\partial q}{\partial t} + \boldsymbol{\nabla}\cdot\mathbf{f}(q) = c(q), \tag{2.4}$$

[1] In relativity theory, mass and energy are equivalent, so that mass is conserved *together with* the energy of the system.

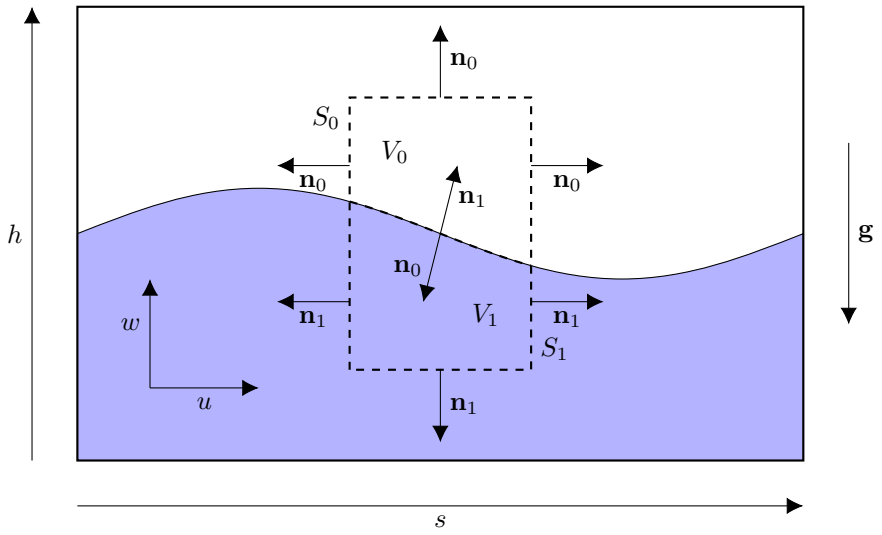


Figure 2.1: A schematic of two control volumes V_0, V_1 in two-phase flow. The control volumes are bounded by S_0 and S_1 respectively, with normal vectors pointing outward. Here we place the boundaries of the control volumes on the interface so that they only ever contain one fluid. The spatial coordinate vector $\mathbf{x}$ has components s, y (not pictured), and h. The velocities in these directions are the components u, v, and w of $\mathbf{u}$, respectively.

where $q = q(\mathbf{x}, t)$ is a conserved quantity, $\mathbf{f}$ is the flux of q and c is a source of q (which can also be a function of q).

In (2.3) the flux is only the convective flux of ρ

$$\mathbf{f}_C = \rho\mathbf{u}. \tag{2.5}$$

If there are multiple species, there may also be a diffusive flux, given by Fick's law, with a diffusion coefficient D:

$$\mathbf{f}_i = \mathbf{f}_{C,i} + \mathbf{f}_{D,i}, \quad \text{with} \quad \mathbf{f}_{C,i} = \rho_i\mathbf{u}, \quad \mathbf{f}_{D,i} = D\boldsymbol{\nabla}\rho_i. \tag{2.6}$$

In our model the flux is only the convective flux, since we make the *assumption of sharp interfaces* [44, p. 22]. The fluids are assumed immiscible, so they do not diffuse into each other. Then a change of ρ in the control volume can only result from a net flow of mass, i.e. (2.5).

The material derivative describes the change of a property of a fluid parcel as we follow it during its flow. It can be defined

$$\frac{\mathrm{D}}{\mathrm{D}t} = \frac{\partial}{\partial t} + \mathbf{u} \cdot \nabla(). \tag{2.7}$$

Using this and an expansion of the divergence operator:

$$\boldsymbol{\nabla} \cdot (\rho\mathbf{u}) = \mathbf{u} \cdot \boldsymbol{\nabla}\rho + \rho\boldsymbol{\nabla} \cdot \mathbf{u}, \tag{2.8}$$

(2.3) can be rewritten as

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = -\rho\boldsymbol{\nabla} \cdot \mathbf{u}. \tag{2.9}$$

This shows that the density of a fluid parcel can only change if the flow is divergent or convergent, in which case the fluid parcel expands or compresses. The flow can be said to be incompressible if this does not happen:

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0, \tag{2.10}$$

in which case the density of a fluid parcel traveling with the flow does not change. Though in the incompressible case a fluid parcel cannot be compressed or expanded, the density might still vary between different fluid parcels (e.g. between two different fluids in a multiphase flow simulation).

2.2.2 Momentum conservation

Momentum conservation for fluids is an application of Newton's second law. The change in time of the momentum contained in a control volume is dictated by the magnitude and direction of the forces acting upon it. Momentum can also enter or leave the control volume with the flow, just as mass can. These ideas can be expressed mathematically as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_V \rho\mathbf{u}\,\mathrm{d}V = -\oint_S \rho\mathbf{u}\,(\mathbf{u}\cdot\mathbf{n})\,\mathrm{d}S - \oint_S p\mathbf{n}\,\mathrm{d}S + \oint_S \boldsymbol{\tau}\cdot\mathbf{n}\,\mathrm{d}S + \int_V \rho\mathbf{g}\,\mathrm{d}V, \tag{2.11}$$

where $p = p(\mathbf{x}, t)$ is the pressure, $\boldsymbol{\tau} = \boldsymbol{\tau}(\mathbf{x}, t)$ is the stress tensor, and $\mathbf{g}$ is the gravity force vector. For any fixed control volume V this equation will hold so that via Gauss' theorem and Leibniz's rule the differential form

$$\frac{\partial\rho\mathbf{u}}{\partial t} = -\boldsymbol{\nabla}\cdot(\rho\mathbf{u}\mathbf{u}) - \boldsymbol{\nabla}p + \boldsymbol{\nabla}\cdot\boldsymbol{\tau} + \rho\mathbf{g} \tag{2.12}$$

can be found, where

$$\boldsymbol{\nabla}\cdot(\rho\mathbf{u}\mathbf{u}) = \rho\mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u} + \mathbf{u}\boldsymbol{\nabla}\cdot(\rho\mathbf{u})\,, \tag{2.13}$$

so that using (2.3) the balance is reduced to

$$\rho\frac{\partial\mathbf{u}}{\partial t} + \rho\mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u} = -\boldsymbol{\nabla}p + \boldsymbol{\nabla}\cdot\boldsymbol{\tau} + \rho\mathbf{g}. \tag{2.14}$$

On the left hand side we can now recognize the material derivative of $\mathbf{u}$ (multiplied by ρ).

The intermediate form (2.12) is an equation in conservative form (2.4), with $\mathbf{q} = \rho\mathbf{u}$,

$$\mathbf{f} = \mathbf{f}_C + \mathbf{f}_D, \quad \text{with} \quad \mathbf{f}_C = \rho\mathbf{u}\mathbf{u}, \quad \mathbf{f}_D = -\boldsymbol{\tau}, \tag{2.15}$$

and

$$\mathbf{c} = \rho\mathbf{g} - \boldsymbol{\nabla}p. \tag{2.16}$$

In this interpretation, $\rho\mathbf{u}\mathbf{u}$ is the convective flux of momentum, the stress term represents diffusion of momentum, and the pressure gradient and gravitational field are sources of momentum.

The interpretation of $-\boldsymbol{\tau}$ as a diffusive flux of momentum is justified by its constitutive law for Newtonian fluids

$$\boldsymbol{\tau} = 2\mu\left[\mathbf{D} - \frac{1}{3}(\boldsymbol{\nabla}\cdot\mathbf{u})\mathbf{I}\right] + \zeta(\boldsymbol{\nabla}\cdot\mathbf{u})\mathbf{I}, \tag{2.17}$$

with $\mathbf{I}$ the identity matrix and $\mathbf{D}$ the rate of strain deformation tensor. According to Stokes' hypothesis the bulk viscosity ζ is zero. If furthermore incompressibility is assumed, the stress tensor reduces to

$$\boldsymbol{\tau} = 2\mu\mathbf{D}, \tag{2.18}$$

where $\mathbf{D} = \frac{1}{2}\left(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^{\mathrm{T}}\right)$ and in Cartesian coordinates, with $\mathbf{x} = (s, y, h)$ and $\mathbf{u} = (u, v, w)$ (see Figure 2.1), and $\mathbf{D}$ is given by

$$\mathbf{D} = \begin{bmatrix} \frac{\partial u}{\partial s} & \frac{1}{2}(\frac{\partial v}{\partial s} + \frac{\partial u}{\partial y}) & \frac{1}{2}(\frac{\partial w}{\partial s} + \frac{\partial u}{\partial h}) \\ \frac{1}{2}(\frac{\partial v}{\partial s} + \frac{\partial u}{\partial y}) & \frac{\partial v}{\partial y} & \frac{1}{2}(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial h}) \\ \frac{1}{2}(\frac{\partial w}{\partial s} + \frac{\partial u}{\partial h}) & \frac{1}{2}(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial h}) & \frac{\partial w}{\partial h} \end{bmatrix}. \tag{2.19}$$

For incompressible flow with constant density and viscosity, by substitution of $\boldsymbol{\nabla}\cdot\mathbf{u} = 0$ and $\mu = \rho\nu$, it can be shown that

$$\boldsymbol{\nabla}\cdot\boldsymbol{\tau} = \boldsymbol{\nabla}\cdot\rho\nu\left[\boldsymbol{\nabla}\mathbf{u} + (\boldsymbol{\nabla}\mathbf{u})^{\mathrm{T}}\right] = \nu\nabla^2\rho\mathbf{u} = \boldsymbol{\nabla}\cdot\nu\boldsymbol{\nabla}\rho\mathbf{u}. \tag{2.20}$$

Thus, returning to (2.4) and (2.15), for all intents and purposes we could write $\mathbf{f}_D = -\boldsymbol{\tau} = -\nu\boldsymbol{\nabla}\rho\mathbf{u}$. This is exactly Fick's form of the diffusive flux (given in (2.6) for mass), but with the kinematic viscosity ν playing the role of diffusion coefficient.

2.2.3 Energy conservation

A similar expression as for mass and momentum conservation can be formulated for energy conservation. Alternatively, the flow can be assumed isothermal, and the system can be closed with an equation of state

$$\rho = f(p). \tag{2.21}$$

This eliminates the need for an energy equation.

For incompressible flows neither an energy equation nor an equation of state is needed. The density is determined by

$$\frac{\mathrm{D}\rho}{\mathrm{D}t} = 0, \tag{2.22}$$

as discussed in subsection 2.2.1. If a fluid starts out with uniform density, then in an incompressible flow the fluid will retain that density everywhere and (2.22) need not be solved.

The pressure must be such that it forces, via the momentum balance, the velocity to satisfy (2.10). The pressure is the only unknown besides the velocity in the momentum balance (2.14). With these multiple unknowns there are multiple combinations of $\mathbf{u}$ and p which satisfy the momentum balance. But the continuity equation (2.10) narrows the choice down to one possible combination. If we choose an appropriate pressure, then from the momentum balance we will obtain a velocity field that is incompressible. This can be viewed as the pressure projecting the velocity field into the space of functions that satisfy (2.10) [44].

2.3 Multiphase flow and the interface

The main difficulty in simulating multiphase flow lies in handling the interface. In this project, we deal with two fluids which, as stated before, are immiscible and have a sharp interface. These two fluids may be for example a liquid and a gas. The equations of motion as derived above apply to both fluids separately. They can in principle be solved for one of the fluids at a time, with the influence of the other fluid entering via the boundary conditions at the interface. In practice this can be complicated since the interface is part of the solution and can assume complex forms.

2.3.1 Kinematic boundary conditions

We consider a thin control volume centered on the interface between two fluids, illustrated in Figure 2.2. The thickness of the control volume tends to zero, so that no mass can accumulate inside of it. The control volume travels with the interface at velocity $\mathbf{u}_{\mathrm{int}}$. The integral formulation for mass conservation (2.1) then leads to the Rankine-Hugoniot condition

$$\rho_1 \left(\mathbf{u}_1 - \mathbf{u}_{\mathrm{int}}\right) \cdot \mathbf{n} = \rho_0 \left(\mathbf{u}_0 - \mathbf{u}_{\mathrm{int}}\right) \cdot \mathbf{n} = \dot{m}, \tag{2.23}$$

in which $\mathbf{u}_{\mathrm{int}}$ is the interface velocity, $\mathbf{u}_1$ is the velocity of fluid 1, $\mathbf{u}_0$ is the velocity of fluid 0 and $\dot{m}$ is the mass flow across the interface. A mass flow across the interface means that one phase gains mass at the expense of the other, and thus implies phase change. If there is no phase change, $\dot{m} = 0$.

If there is no phase change, (2.23) leads to the boundary condition

$$\mathbf{u}_{\mathrm{int}} \cdot \mathbf{n} = \mathbf{u}_1 \cdot \mathbf{n} = \mathbf{u}_0 \cdot \mathbf{n}, \tag{2.24}$$

which can also be applied at a solid boundary.

With the continuum hypothesis, there can be no slip between gas and liquid at the interface since a discontinuous velocity profile would result in infinite stress. The no-slip boundary condition can be combined with (2.24) to yield

$$\mathbf{u}_1 = \mathbf{u}_0. \tag{2.25}$$

We can obtain another boundary condition by applying the integral momentum balance (2.11) to the control volume in Figure 2.2. Again, no momentum can accumulate in the control volume due to its vanishing thickness. The gravitational term in (2.11) must be converted to a surface integral:

$$\int_V \rho \mathbf{g} \, \mathrm{d}V = -\int_V \boldsymbol{\nabla} U \, \mathrm{d}V = -\oint_S U \mathbf{n} \, \mathrm{d}S, \tag{2.26}$$

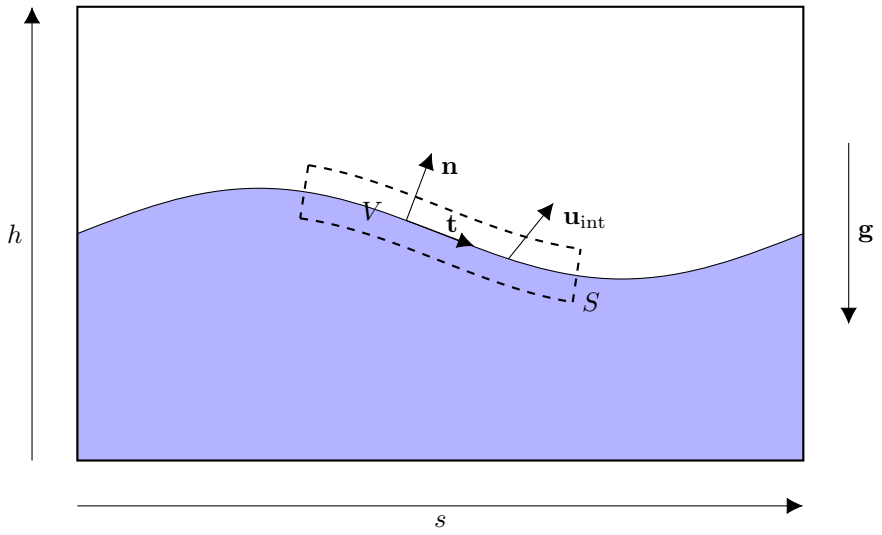


Figure 2.2: A schematic of thin control volume V with boundary S, centered around the interface. The normal to the interface is $\mathbf{n}$ and the tangent is $\mathbf{t}$.

with $U(\mathbf{x})$ the gravitational potential energy

$$U = \int_0^h \rho g \, \mathrm{d}h', \tag{2.27}$$

in which $\rho = \rho(\mathbf{x})$.

The inflow and outflow of momentum then balance in the following way:

$$\rho_1 \mathbf{u}_1 \left(\mathbf{u}_1 - \mathbf{u}_{\text{int}}\right) \cdot \mathbf{n} + p_1 \mathbf{n} - \boldsymbol{\tau}_1 \cdot \mathbf{n} + U_1 = \rho_0 \mathbf{u}_0 \left(\mathbf{u}_0 - \mathbf{u}_{\text{int}}\right) \cdot \mathbf{n} + p_0 \mathbf{n} - \boldsymbol{\tau}_0 \cdot \mathbf{n} + U_0 = \dot{M}, \tag{2.28}$$

where $\dot{M}$ is the momentum transfer across the interface. Since the control volume is vanishingly thin, the difference between U_1 at the boundary on one side of the interface and U_0 at the boundary on the other side of the interface is negligible (calculated via (2.27)), and so these terms cancel out. In the absence of phase change the advection terms are zero by virtue of (2.24). We are left with

$$(p_1 - p_0)\mathbf{n} - (\boldsymbol{\tau}_1 - \boldsymbol{\tau}_0) \cdot \mathbf{n} = 0, \tag{2.29}$$

By taking inner products with $\mathbf{n}$ and $\mathbf{t}$, this is split into two boundary conditions[2] :

$$p_1 - p_0 - \mathbf{n} \cdot (\boldsymbol{\tau}_1 - \boldsymbol{\tau}_0) \cdot \mathbf{n} = 0, \tag{2.30}$$

$$\mathbf{t} \cdot (\boldsymbol{\tau}_1 - \boldsymbol{\tau}_0) \cdot \mathbf{n} = 0. \tag{2.31}$$

The expression $\boldsymbol{\tau} \cdot \mathbf{n}$ signifies the stress acting on the interface, so that $\mathbf{n} \cdot \boldsymbol{\tau} \cdot \mathbf{n}$ is the stress acting on the interface in the direction of $\mathbf{n}$, and $\mathbf{t} \cdot \boldsymbol{\tau} \cdot \mathbf{n}$ is the stress acting on the interface in the direction parallel to the interface: the shear stress. These conditions express that the force that the first fluid exerts upon the second should be opposite but equal in magnitude to the force that the second exerts upon the first, i.e. Newton's third law.

With the identity (2.18), (2.30) and (2.31) can be written as

$$p_1 - \mathbf{n} \cdot \mu_1 (\boldsymbol{\nabla} \mathbf{u}_1 + \boldsymbol{\nabla} \mathbf{u}_1^{\mathrm{T}}) \cdot \mathbf{n} = p_0 - \mathbf{n} \cdot \mu_0 (\boldsymbol{\nabla} \mathbf{u}_0 + \boldsymbol{\nabla} \mathbf{u}_0^{\mathrm{T}}) \cdot \mathbf{n}, \tag{2.32}$$

$$\mathbf{t} \cdot \mu_1 (\boldsymbol{\nabla} \mathbf{u}_1 + \boldsymbol{\nabla} \mathbf{u}_1^{\mathrm{T}}) \cdot \mathbf{n} = \mathbf{t} \cdot \mu_0 (\boldsymbol{\nabla} \mathbf{u}_0 + \boldsymbol{\nabla} \mathbf{u}_0^{\mathrm{T}}) \cdot \mathbf{n}. \tag{2.33}$$

In 2D, for a flat interface at constant h, these reduce to

$$p_1 + 2\mu_1 \frac{\partial w_1}{\partial h} = p_0 + 2\mu_0 \frac{\partial w_0}{\partial h} \tag{2.34}$$

$$\mu_1 \left(\frac{\partial u_1}{\partial h} + \frac{\partial w_1}{\partial s} \right) = \mu_0 \left(\frac{\partial u_0}{\partial h} + \frac{\partial w_0}{\partial s} \right), \tag{2.35}$$

[2] In 3D, the second boundary condition holds for two different tangent vectors.

of which

$$\frac{\partial w_1}{\partial s} = \frac{\partial w_0}{\partial s} \tag{2.36}$$

due to the no-slip condition (2.25); more generally $\mathbf{t} \cdot \boldsymbol{\nabla} \mathbf{u}^{\mathrm{T}} \cdot \mathbf{n}$ is continuous across the interface. Similarly, $\mathbf{t} \cdot \boldsymbol{\nabla} \mathbf{u} \cdot \mathbf{t} = \mathbf{t} \cdot \boldsymbol{\nabla} \mathbf{u}^{\mathrm{T}} \cdot \mathbf{t}$, which reduces to $\partial u / \partial s$ for this geometry, is continuous due to the no-slip condition. As a result, incompressibility ($\boldsymbol{\nabla} \cdot \mathbf{u} = 0$) implies the continuity of $\mathbf{n} \cdot \boldsymbol{\nabla} \mathbf{u} \cdot \mathbf{n} = \mathbf{n} \boldsymbol{\nabla} \mathbf{u}^{\mathrm{T}} \cdot \mathbf{n}$, $\partial w / \partial h$ for this geometry. Summarized, we have continuity between the two fluids of the following velocity gradient components:

$$\mathbf{t} \cdot \boldsymbol{\nabla} \mathbf{u}_1^{\mathrm{T}} \cdot \mathbf{n} = \mathbf{t} \cdot \boldsymbol{\nabla} \mathbf{u}_0^{\mathrm{T}} \cdot \mathbf{n} \qquad \rightarrow \qquad \frac{\partial w_1}{\partial s} = \frac{\partial w_0}{\partial s}, \tag{2.37}$$

$$\mathbf{t} \cdot \boldsymbol{\nabla} \mathbf{u}_1 \cdot \mathbf{t} = \mathbf{t} \cdot \boldsymbol{\nabla} \mathbf{u}_1^{\mathrm{T}} \cdot \mathbf{t} = \mathbf{t} \cdot \boldsymbol{\nabla} \mathbf{u}_0 \cdot \mathbf{t} = \mathbf{t} \cdot \boldsymbol{\nabla} \mathbf{u}_0^{\mathrm{T}} \cdot \mathbf{t} \qquad \rightarrow \qquad \frac{\partial u_1}{\partial s} = \frac{\partial u_0}{\partial s}, \tag{2.38}$$

$$\mathbf{n} \cdot \boldsymbol{\nabla} \mathbf{u}_1 \cdot \mathbf{n} = \mathbf{n} \boldsymbol{\nabla} \mathbf{u}_1^{\mathrm{T}} \cdot \mathbf{n} = \mathbf{n} \cdot \boldsymbol{\nabla} \mathbf{u}_0 \cdot \mathbf{n} = \mathbf{n} \boldsymbol{\nabla} \mathbf{u}_0^{\mathrm{T}} \cdot \mathbf{n} \qquad \rightarrow \qquad \frac{\partial w_1}{\partial h} = \frac{\partial w_0}{\partial h}. \tag{2.39}$$

If the terms (2.39) are small[3], $\mathbf{n} \cdot \boldsymbol{\tau} \cdot \mathbf{n}$ will be small for both fluids and (2.32) determines that the pressure will be continuous over the interface; $p_1 = p_0$[4]. In this case the boundary conditions (2.32) and (2.33) express the existence of a single interfacial stress

$$\boldsymbol{\tau}_{\mathrm{int}} = \boldsymbol{\tau}_1 \cdot \mathbf{n} = \boldsymbol{\tau}_0 \cdot \mathbf{n}, \tag{2.40}$$

with only a tangential component:

$$\boldsymbol{\tau}_{\mathrm{int}} = \mathbf{t} \cdot \boldsymbol{\tau}_{\mathrm{int}} = \mathbf{t} \cdot \boldsymbol{\tau}_1 \cdot \mathbf{n} = \mathbf{t} \cdot \boldsymbol{\tau}_0 \cdot \mathbf{n}, \tag{2.41}$$

where $\mathbf{n}$ is the interface normal seen from the liquid like $\mathbf{n}_1$ in Figure 2.1, so that this stress enters the momentum balance (2.11) for control volume 1 in Figure 2.1 directly, but requires an added minus sign to be applied to control volume 0. Thus, the forces on the control volumes in Figure 2.1 are opposite but equal in magnitude, as required by Newton's third law, and consist now solely of the shear stress.

If the characteristic horizontal length scale is much larger than the vertical length scale: $L \gg H$[5], a further simplification of the interfacial stress can be achieved. In this case the terms (2.37) will be small compared to $\mathbf{t} \cdot \boldsymbol{\nabla} \mathbf{u} \cdot \mathbf{n} \rightarrow \partial u / \partial h$. The boundary condition (2.33) $\rightarrow$ (2.35) then simplifies to

$$\mathbf{t} \cdot \mu_1 \boldsymbol{\nabla} \mathbf{u}_1 \cdot \mathbf{n} = \mathbf{t} \cdot \mu_0 \boldsymbol{\nabla} \mathbf{u}_0 \cdot \mathbf{n} \qquad \rightarrow \qquad \mu_1 \frac{\partial u_1}{\partial h} = \mu_0 \frac{\partial u_0}{\partial h}. \tag{2.42}$$

This relation will lead to a sharp gradient in velocity for the gas and a relatively low velocity gradient for the liquid since the liquid viscosity is generally much higher than the gas viscosity. The large gradient in the gas velocity may make it appear that there is a jump between the liquid and gas velocity when a discretization is performed with a limited grid resolution, however this is not allowed by (2.25).

At solid boundaries the terms (2.37) and (2.39) will be firmly zero so that like explained above the stress will consist solely of a shear stress of the form (2.42). However the stress in a solid cannot be modeled in this way, but we do not need to explicitly model it if we simply assume the solid to be stationary and non-changing.

2.3.2 One-fluid formulation

For incompressible flow we now have a complete set of equations and boundary conditions. The mass and momentum balances are solved separately for both fluids. In a 2D or 3D numerical solver this means that the grid must be adapted each time step so that the boundaries of the grid cells line the interface. Considering that the interface geometry may be very complex, this is not always practical. It can be beneficial to write the equations in such a way that the same equations apply to the entire domain, and not just in the area where one of the fluids presides. This is the so-called one-fluid formulation.

[3] For example if the interface is approximately parallel to some impenetrable solid boundaries (which is likely if we consider a shear flow with a long wavelength perturbation), so that we have approximate hydrostatic balance.

[4] Alternatively, for irrotational flow this condition will hold.

[5] For example if we have a shear flow with a long wavelength perturbation applied to it. This, along with pure parallel shear flow, is the case we shall consider.

In the one-fluid formulation, the mass and momentum balances that have been derived will hold for a control volume containing two different fluids. What changes is that the material properties such as viscosity and density will jump abruptly at the interface. All forces in the momentum balance apply to an arbitrary control volume taken anywhere in the computational domain. We neglect surface tension here because capillary waves are not of interest in our simulations.

The one-fluid version of the Navier-Stokes equations for multiphase incompressible flow becomes

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \tag{2.43}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u} = \frac{1}{\rho} \left(-\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \left[\mu \boldsymbol{\nabla} \mathbf{u} + \mu (\boldsymbol{\nabla} \mathbf{u})^{\mathrm{T}} \right] \right) + \mathbf{g} \tag{2.44}$$

with at solid walls the boundary conditions

$$\mathbf{u} \cdot \mathbf{n} - \mathbf{u}_{\mathrm{wall}} \cdot \mathbf{n} = 0, \tag{2.45}$$

and

$$\mathbf{u} \cdot \mathbf{t} - \mathbf{u}_{\mathrm{wall}} \cdot \mathbf{t} = 0. \tag{2.46}$$

In these equations, ρ and μ are functions of $\mathbf{x}$ and for their determination it is still necessary to know the shape of the interface.

A graphical representation of this model applied to 2D channel flow is given in Figure 2.3. In the figure, M is a marker function which marks which fluid is located at a particular point. Where the marker function is 1, the viscosity and density will have the values corresponding to fluid 1, and where it is zero the viscosity and density will be those of fluid 0.

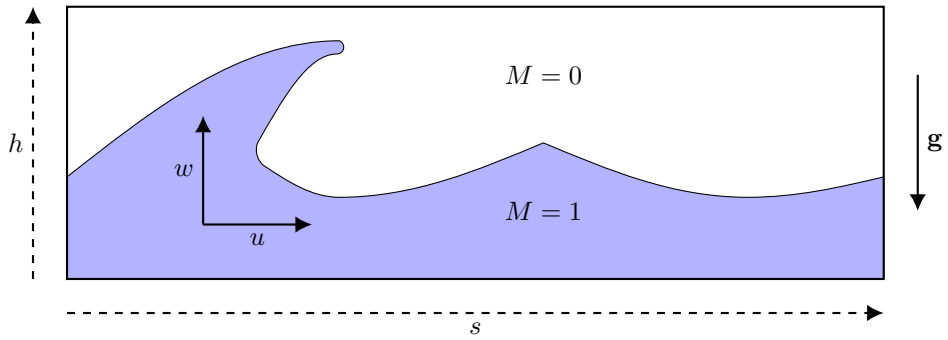


Figure 2.3: A schematic of the one-fluid model for 2D channel flow.

The marker function is advected with the fluid via an equation of the form of (2.4):

$$\frac{\partial M}{\partial t} + \boldsymbol{\nabla} \cdot (M \mathbf{u}) = 0, \tag{2.47}$$

which corresponds with the advection of mass via (2.3). Like the mass it represents, the marker function does not diffuse and has no source term. For the marker function in incompressible flow it therefore also holds that

$$\frac{\mathrm{D}M}{\mathrm{D}t} = 0. \tag{2.48}$$

In single phase flow, the second term in (2.3), corresponding to the second term in (2.47), is zero and the equation is trivial. But in the one-fluid formulation for multiphase flow either (2.47) or (2.48) must be solved explicitly to determine the evolution of the location of the interface.

2.4 Direct numerical simulation of multiphase flow

Using the one-fluid formulation of subsection 2.3.2, it is possible to solve the flow equations with methods developed for single phase flow. The important difference is that we must allow for variable material properties. The determination of the material properties requires solving for the location of the interface. The evolution of the location of the interface is usually determined by advecting a marker function for it. Effectively, we have an extra coupled equation (2.47) to solve, along with the mass and momentum equations.

2.4.1 Spatial discretization

The system (2.43) and (2.44) needs to be discretized in order to be solved numerically. Derivatives with respect to time and space are not mixed in these equations and thus they can be considered separately. A method of lines approach is adopted here, in which we first discretize the spatial derivatives and only then consider the problem of how to integrate the equations in time.

Spatial discretization of the Navier-Stokes equations is often done using finite volume methods. These have the advantage that they are conservative by design. This is because they are formulated using the integral form of the momentum balance, e.g. (2.11), applied to the individual grid cells. The expression used for the inflow of mass or momentum at a cell boundary is identical to the expression used for the outflow of the neighboring cell.

When applying a finite volume method to incompressible flow, it is natural to use a staggered grid, illustrated for 1D in Figure 2.4. This means that grid points where the velocities are defined are shifted relative to the points where pressure and material properties are stored. For the solution of the equation for mass conservation, a control volume is used which is centered around the point where the pressure is defined, with the velocities defined at the centers of the edges of the control volume. If there is a net inflow, the pressure of the control volume must increase, and if there is a net outflow the pressure must decrease. For incompressible flow this must happen instantaneously so that there is never a net mass flux into a control volume.

For the solution of the equation for momentum conservation, a control volume is used which is centered around the velocity for which, after being multiplied by the density, we are enforcing the conservation law. That is, the momentum fluxes through the control volume boundaries must balance (or, for a non-steady problem, alter the momentum of the control volume).

Finite volume methods can also be used with colocated grids, in which velocities and the pressure are defined at the same grid points. The main comparative advantage of a staggered grid is that the coupling between the velocity and the pressure at different grid points is increased. The momentum balance (2.44) links the velocity and the pressure gradient. On a 1D staggered grid this means concretely that $u_{i+1/2}$ is related to p_i and p_{i+1}. Therefore p_i and p_{i+1} both depend on $u_{i+1/2}$; they are coupled. Also, $u_{i+3/2}$ is related to p_{i+1} and p_{i+2}, so that p_{i+1} and p_{i+2} are coupled and by extension p_{i+2} and p_i are coupled, and all pressure points can be shown to be coupled in this way. On a 1D colocated grid u_i is related to p_{i-1} and p_{i+1}, and u_{i+1} is related to p_i and p_{i+2}. Therefore p_{i-1} and p_{i+1} are coupled, and p_i and p_{i+2} are coupled, but the pairs are not coupled to one another! The grid can be shown to consist of two separate sets of coupled points. This lack of coupling can lead to wild oscillations in the pressure field, even while the continuity equation is satisfied [45], [46].

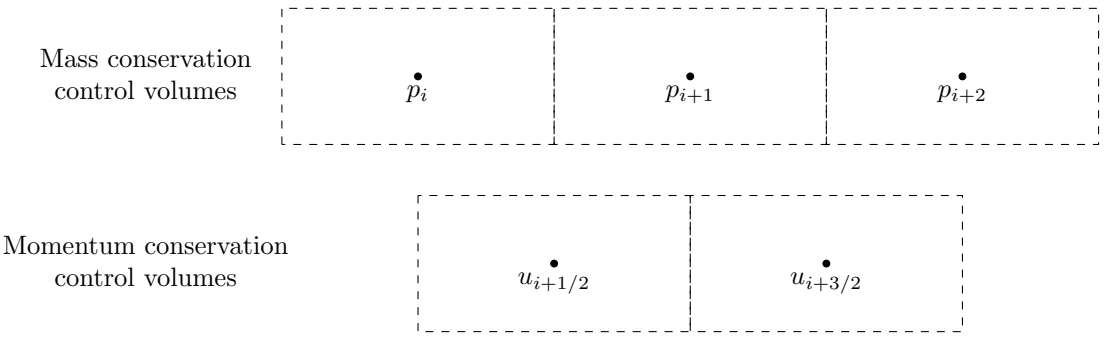


Figure 2.4: A schematic of a 1D staggered grid. The momentum conservation control volumes are shifted by half a grid cell with respect to the mass conservation control volumes. In a mass conservation control volume, the evolution of the pressure is determined by the difference between the flow at the boundaries, which for incompressible flow should sum to zero so that the pressure remains constant. In a momentum conservation control volume, the evolution of the flow is determined in part by the difference in pressure between the cell boundaries (which comes down to the pressure gradient which appears in (2.44)).

However, colocated grids are simpler to use when solving equations in complex geometries [45]. Therefore methods were developed to get around the coupling problem and according to Ranade (2002) [45] most commercial CFD codes now use colocated grids.

2.4.2 Time integration

With the spatial discretization completed one can write the most basic time integration scheme as

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} = -\mathbf{A}_h^n + \frac{1}{\rho^n}\left(-\nabla_h p^{n+1} + \mathbf{D}_h^n + \mathbf{f}^n\right), \tag{2.49}$$

in which n denotes the time step and Δt is the length of the time step. $\mathbf{A}_h^n$ is the discrete form of the advection term at time step n, $\nabla_h p^{n+1}$ is the discrete form of the pressure gradient at time step $n+1$, $\mathbf{D}_h^n$ is the same for the diffusion, and $\mathbf{f}^n$ is the same for the body forces (e.g. gravity, surface tension) at time n.

As explained in subsection 2.2.3, for incompressible flow it is necessary to find the pressure, which projects the velocity such that the continuity equation (2.43) is satisfied. This can be done using a so-called projection method, introduced by Chorin (1968) [47]. A basic example of a projection method which does what was explained in words in subsection 2.2.3 is given by [44].

It begins by splitting (2.49) in two parts. The first part is the predictor step, where an intermediate $\mathbf{u}^*$ is found by solution of (2.49), but leaving out the pressure:

$$\frac{\mathbf{u}^{*}-\mathbf{u}^n}{\Delta t} = -\mathbf{A}_h^n + \frac{1}{\rho^n}\left(\mathbf{D}_h^n + \mathbf{f}^n\right), \tag{2.50}$$

In the second step, the projection step, the pressure adjusts ('projects') the velocity $\mathbf{u}^*$ to the new velocity $\mathbf{u}^{n+1}$ via the formula

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{*}}{\Delta t} = -\frac{1}{\rho^n}\nabla_h p^{n+1}. \tag{2.51}$$

Solving this equation for $\mathbf{u}^*$ and substituting into (2.50) yields exactly the original equation (2.49), so $\mathbf{u}^{n+1}$ satisfies the original equation.

In order to make $\mathbf{u}^{n+1}$ also satisfy the continuity equation

$$\nabla_h \cdot \mathbf{u}^{n+1} = 0, \tag{2.52}$$

the divergence of (2.51) is taken and the continuity equation for $\mathbf{u}^{n+1}$ is substituted to yield

$$\nabla_h \cdot \left(\frac{1}{\rho^n}\nabla_h p^{n+1}\right) = \frac{1}{\Delta t}\nabla_h \cdot \mathbf{u}^{*}. \tag{2.53}$$

In the above equation, $\mathbf{u}^*$ is already known from the predictor step so that $\nabla_h p^{n+1}$ can be found directly and substituted in (2.51) to enable calculation of $\mathbf{u}^{n+1}$.

In single phase flow, ρ^n can usually be taken out of the divergence in (2.53), yielding a Poisson equation for the pressure, for which many solution methods exist. The variable density in the case of multiphase flow leads to some added difficulty. The solution of the pressure equation is often the most time-consuming part of a simulation, since it generally needs to be solved iteratively.

Note that in this method, we are only solving the momentum balance explicitly; the mass balance only enters the solution as a constraint which determines the pressure. This pressure then modifies the velocity field to make it divergence-free. With the staggered grid described in subsection 2.4.1 the pressure at the center of mass conservation cells is directly coupled, without need for interpolation, to the velocities which are defined at the mass conservation cell boundaries (see Figure 2.4), via (2.51)[6]. With (2.51), a divergence-free velocity field is directly computed at the mass conservation cell boundaries, so that mass is conserved.

The given method is only first-order in time. Kim and Moin (1985) [48] describe a projection method with second-order accuracy in time. Other methods of integrating the incompressible Navier-Stokes equations in time exist, a notable one being the PISO method [49].

2.4.3 Advecting a fluid interface

The most apparent extra requirement of a multiphase flow simulation code is the need to keep track of the location of the interface between the two fluids. In the one-fluid approach, this is done by advecting a marker function representing one or the other fluid, as discussed in subsection 2.3.2. It is not practical to solve the transport equation (2.47) directly. This is because any finite difference scheme would numerically diffuse the discontinuity in the marker function at the interface [44].

[6]The mass conservation cell boundary centers are the momentum conservation cell centers. In the finite volume formulation the pressure gradient is equivalent to the difference in pressure between the cell boundaries; see (2.11).

The volume-of-fluid method

In the volume-of-fluid (VOF) method [50], the marker function is averaged over the grid cells to define the color function

$$C = \frac{1}{V}\int_V M \,\mathrm{d}V. \tag{2.54}$$

The color function is a function which gives the volume fraction of the reference fluid in a grid cell. The material properties in grid cells i can then be expressed as functions of this color function. For example

$$\rho_i = C_i\rho_1 + (1 - C_i)\rho_0, \tag{2.55}$$
$$\mu_i = C_i\mu_1 + (1 - C_i)\mu_0, \tag{2.56}$$

with ρ_1 and μ_1 the density and viscosity of the fluid indicated by $M = 1$ and ρ_0 and μ_0 the fluid indicated by $M = 0$. For the density, this formulation is necessary in order for mass to be conserved[7]. For the viscosity, different averaging and interpolation methods may be used: see section 5.4.

The color function advection is performed in two steps, which are shown in Figure 2.5. If we know the location of the interface and the velocities, we can calculate the amount of fluid 1 that is advected to the next grid cell. Then we know the color function C of that grid cell. After the interface advection step, we need to reconstruct the interface using C, for use in the next advection step.

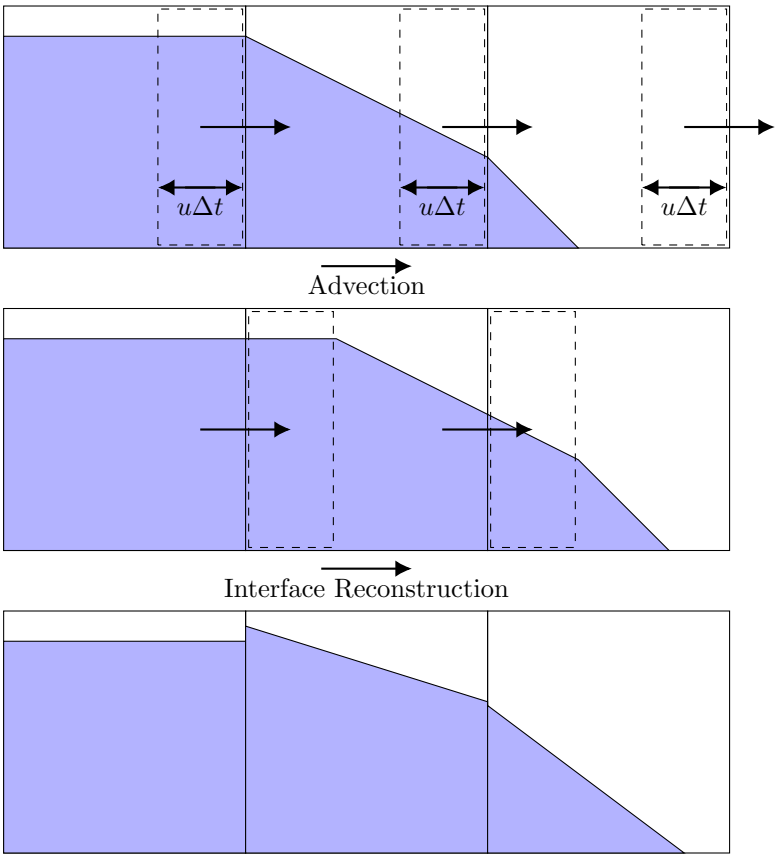


Figure 2.5: An illustration of 2D VOF color function advection, for the simple case $w = 0$. The values of the color function in each grid cell are used to construct an interface, via PLIC. This allows geometric advection of the color function, which is conservative. The new values of the color function in each grid cell are used to construct a new interface, for the next time step.

In 1D the interface reconstruction step is not necessary since the color function already determines the interface uniquely. In 2D the relation between color function and interface geometry is not unique.

[7] The mass present in a grid cell is the unweighted average of the product of the density and the marker function over the grid cell, multiplied by the size of the grid cell. This formulation for ρ_i yields the mass in the grid cell divided by the grid cell size.

We have to make a choice in some way; a commonly used method is piecewise linear interface calculation (PLIC) [51]. In PLIC the interface is represented as a single diagonal line drawn through the grid cell. This is an approximation which leads to sharp corners (seen in the last part of Figure 2.5) which can break away as 'flotsam' or 'jetsam' [52].

To determine this line first the normal of the line must be computed. This is done using the value of the color function in the grid cell and in the eight neighboring cells. After the normal is determined, the offset (distance from the cell boundaries) of the interface is determined by demanding that the area of the resulting polygon matches the color function of the cell.

Once the interface is known, two points which describe this line uniquely (e.g. the intercepts of the line with the cell boundaries) can be advected exactly using the local velocity, which should be known from the solution of the momentum balance. This advection method can be regarded to be based on (2.48). In 2D it is done in two steps; one for each velocity component. These steps can be described as linear mappings of the original area occupied by the reference fluid to the new time step. The appropriate combination of an explicit mapping (using the velocity at the previous time step) and an implicit mapping (using the velocity at the new time step) yields a conservative scheme (without diffusion). The result of the linear mappings is the value of the color function in each grid cell. A detailed discussion is given by Rider and Kothe [53].

Front tracking

The first development of front tracking methods for viscous multiphase flow was done by [54]. In front tracking methods, marker *points* are advected instead of a marker function. This can be done simply and exactly, for example with the formula

$$\mathbf{x}_f^{n+1} = \mathbf{x}_f^n + \mathbf{u}_f^n \Delta t, \tag{2.57}$$

in which $\mathbf{x}_f^n$ is the location of front point f at time step n, and $\mathbf{u}_f^n$ its velocity. The interface need not be reconstructed; the advected points directly define the interface. The points live separately from the grid points, and carry the information of their exact location. In 2D the front can be structured, which means the the front points are ordered and know which front point comes before and which comes after on the interface. In 3D unstructured fronts are used, in which the points carry no information on their connections but instead triangular surface elements store the indexes of the front points which are their vertices.

Information (such as the local velocity) is passed from the grid points to the front points by interpolation, in which close by grid points are weighted most. Precisely the other way around the grid points are assigned values for the gradient of a marker function. The marker function, just as before, denotes the presence of one or the other fluid and at the interface its gradient is a constant, pointed normal to the interface. When each grid point knows the local gradient of the marker function, the marker function can be reconstructed at each grid point starting from a grid point where the marker function is known. This marker function then determines the local values of the material properties.

Level-set methods

A third important class of interface advection methods is the level-set method, introduced by Osher et al. [55]–[57]. In this method a marker function is used which is a smooth function of the distance to the interface. It is zero at the interface, negative on one side and positive on the other. No interface reconstruction is needed in this method. The marker function is smooth and thus can be advected by standard numerical methods applied to (2.47).

The full advection equation can be simplified by substitution of $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$ and the observation that with the current definition for the marker function the interface normal (see Figure 2.2) is given by

$$\mathbf{n} = -\frac{\boldsymbol{\nabla} M}{|\boldsymbol{\nabla} M|}. \tag{2.58}$$

With these substitutions, (2.47) simplifies to

$$\frac{\partial M}{\partial t} - (\mathbf{u} \cdot \mathbf{n})|\boldsymbol{\nabla} M| = 0. \tag{2.59}$$

The density and viscosity are defined as smooth functions (with continuous first derivatives) of the advected marker function. The velocities of the next time step can be calculated using the new $\rho(\mathbf{x})$ and $\mu(\mathbf{x})$ via (2.43) and (2.44). The new velocity field can then be used to advance (2.59) in time. In practice the time stepping does not have to be purely explicit though.

2.4.4 Comparison

All these three methods for interface advection operate within the one-fluid framework discussed in subsection 2.3.2. But still the approaches differ significantly.

Changes in the front topology must be handled explicitly in front tracking methods. Since the front points (or the elements in the case of unstructured grids) carry information to which front points they are connected, it must be explicitly determined if these connections should be altered at the new time step. This is the main downside of this method relative to VOF, where the color function at the new time step is the result and the interface is calculated from this naturally.

The front tracking method is also not naturally mass conservative, whereas the VOF method is. Level-set methods have the same problem of being non-conservative. Adjustments can be made to improve the mass conservation property, but these complicate the method, while the main advantage of level-set methods are their comparative simplicity in simply requiring another PDE (2.59) to be solved [44].

The main advantage of VOF methods is their natural conservative quality, while their main downside is that the interface reconstruction can be quite laborious, and introduces 'jetsam' and 'flotsam'. The calculation of the surface tension involves similar processes and is also time-consuming. In front-tracking methods surface tension can be calculated on the front and this is more natural. However, we will not consider surface tension in this project.

Other classes of simulation methods exist. For example Smoothed Particle Hydrodynamics (SPH). The main difference with the methods described above is that it is not grid-based, but particle-based. Good reviews are given by [58] and [59].

This method has an efficient, highly parallel GPU implementation in the form of DualSPHysics [7], which, like Gerris, is open source. A further advantage of the particle-based method is that discontinuities, large perturbations and complex geometries are dealt with naturally. However the treatment of boundary conditions is quite complex (and different to the way our low-fidelity model treats them) and the calculation of stresses is not straightforward and susceptible to fluctuations in the particle density. Furthermore the method is weakly compressible and not incompressible like our low-fidelity model (discussed in chapter 3).

Lattice Boltzmann methods are also widely used in multiphase flow. Their main advantages are their potential to model complex physics at the meso-scale, their suitability to complex geometries and again their capacity for mass parallelization. An open source implementation is Palabos [60], [61]. We do not quite need the qualities listed above. We do not need to deal with probability distributions for particles; a macro-scale continuous Newtonian fluid description is preferable since it is more in line with our low-fidelity 1D two-fluid model.

By choosing the DNS approach, we keep the physics simple and general, and similar to those of our low-fidelity 1D two-fluid model. Using VOF interface advection, mass conservation is ensured. The essential step of calculating the stresses is straightforward; we just need to calculate the velocity gradients at the boundaries numerically using velocities defined at the centers of fixed grid cells (further discussion can be found in section 3.4 and section 5.4). DNS was also used as the high-fidelity model for the generation of data for the learning of closure terms in multiphase flow by Ma et al. [10], [11]. The VOF method is implemented in commonly used CFD codes such as Fluent [62] and OpenFoam [63].

2.5 Gerris

The Gerris flow solver [6], [41], is an open source solver which works according to the principles explained in section 2.4. In this project we use the 2D implementation of Gerris. The finite volume approach to spatial discretization is taken. A colocated grid is used, since the grid can be complicated in Gerris. Gerris allows local quadtree grid refinement in two dimensions, which means that some of the root cells can be split in four, and the resulting cells can be split in four, etc. The grid is structured (all grid cells are square) and there are limits to the jump in refinement between grid cells. Gerris has the ability to refine the grid adaptively according to some preset condition (e.g. refine to a set level all grid cells where the absolute vorticity exceeds some set value). In our simulations, we will use a uniform refinement.

It is often necessary to convert the cell-centered values of $\mathbf{u}$, ρ, and μ into face-centered values. This can be done using central interpolation, i.e. a face-centered value is obtained by averaging the cell-centered values of the two neighboring cells. The fluid-dependent parameters ρ, and μ at the cell faces are based on the color function C interpolated in this manner. The gradients at cell boundaries are calculated as

$$\frac{\partial p_{i+1/2}}{\partial s} = \frac{p_{i+1} - p_i}{\Delta s}, \tag{2.60}$$

when the two grid cells are of the same level. Gerris also offers limiters to calculate gradients at cell faces, for a good balance between stability and convergence. We use the Van Leer generalized minmod limiter with $\theta = 2$ [64].

For temporal discretization Gerris uses a second order projection method [65], in which a multilevel Gauss-Seidel iterative method is used to solve the pressure Poisson equation (e.g. (2.53)). The velocity advection term is discretized according to the second order unsplit upwind scheme of [66], and for the diffusion term a Crank-Nicholson discretization is employed.

Using a colocated grid complicates the projection scheme compared to the principles described in subsection 2.4.1 and subsection 2.4.2. In order to couple the pressures and velocities for this colocated grid an approximate projection method is used for the cell-centered velocities [67]. The face-centered velocities are projected to be exactly (discretely) divergence-free.

For the interface advection a Volume of Fluid approach, as described above, is taken. The VOF advection makes use of the exactly divergence-free face-centered velocities. Gerris takes the one-fluid model of multiphase flow and solves the dimensional equation

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \mathbf{u} = \frac{1}{\rho} \left[-\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot (\mu(\boldsymbol{\nabla} \mathbf{u} + \boldsymbol{\nabla} \mathbf{u}^T)) \right] + \texttt{Source}(\mathbf{u}), \tag{2.61}$$

with as the source term normally $\mathbf{g}$.

A dimensionless form of the equations can be derived by substituting for the dimensional variables their dimensionless equivalents multiplied by some characteristic value. Then the simulations can be run in dimensionless form by giving for ρ the density divided by a reference density (possibly ρ_L), for μ similarly a Reynolds number depending on the local viscosity, and for $\mathbf{g}$ a term $-(1/\mathrm{Fr})\mathbf{z}$. This can be done, but with large density and viscosity differences the difference in dimensionless numbers between the two fluids remains large, and the choice of characteristic values is difficult, certainly when considering different initializations and boundary conditions. The non-dimensionalization is further complicated if it is to match with the non-dimensionalization of the low fidelity model discussed in chapter 3. This is then also connected to the non-dimensionalization of closure term inputs and outputs, to be learned by the neural network. To keep the framework general, consistent, and simple, we do not non-dimensionalize the simulations.

We use all standard Gerris settings, except that we lower the tolerance of the projection and approximate projection from $1 \cdot 10^{-3}$ to $1 \cdot 10^{-6}$. The grid spacing $\Delta s = \Delta h$ is a user input, and the time step is set so that the maximum value of

$$\texttt{CFL} = \frac{|\mathbf{u}| \Delta t}{\Delta s} \tag{2.62}$$

anywhere in the simulation is 0.8. However, there is an additional constraint that in mixed VOF cells the maximum value should be 0.5. We do not filter the color function (i.e. averaging over multiple cells), to keep the interface relatively sharp.

Gerris has been validated against viscous linear instability theory by Fuster et al. [68] and Bagué et al. [69]. It has been experimentally validated for a number of cases, including sloshing in a rectangular tank [70] and liquid jet atomization [71]. Wroniszewski et al. [72] compare Gerris favorably to a few different multiphase DNS codes for the case of runup of a coastal wave.

2.6 Conclusion

The Navier-Stokes equations and their application to multiphase flow were laid out. We consider incompressible flow without surface tension. We have derived the boundary conditions at solid boundaries and at the interface between two fluids. We can in principle constantly adjust the boundaries of the domains in which the two fluids reside and require mass and momentum conservation for these two

domains separately, with the effect of the other fluid entering through the boundary conditions. But it is more practical to consider the one-fluid formulation, in which the equations are solved for a single, unified domain, with the material properties being functions of the spatial coordinates, according to which fluid is currently present at those coordinates.

We have discussed different methods for the numerical solution of the equations for multiphase flow. The domain is split into finite volume cells with exact mass and momentum conservation properties. In the time integration mass conservation for the next time step is enforced via a fractional step projection method, in which a first guess for the velocity is computed from the momentum balance, and subsequently adjusted to be incompressible by an appropriate pressure field. This pressure field is calculated via an implicit Poisson-like equation, which is more complicated than the single-phase equation due to the inclusion of the density in the argument of the divergence operator. The calculation of this pressure field is the most time-consuming step in the calculation.

The VOF advection method is defined by the geometric advection of a color function, which defines the fraction of a grid cell filled by a reference fluid. In VOF mass conservation during the advection of the interface comes naturally.

Gerris is an open source code which uses the one-fluid formulation on a colocated grid with the VOF method for interface advection. The use of a colocated grid complicates the projection method but allows for complex geometries and locally refined grids. Gerris may be less efficient in terms of computational cost than codes which make use of SPH or LBM methods, but it contains the physics we need to model our pipe flow and which correspond to those of our low-fidelity 1D two-fluid model, discussed in chapter 3.

Gerris may be viewed in some sense as a 2D extension of our 1D two-fluid model. This is what we want for our high fidelity model. For with this 2D extension, we can calculate stresses explicitly using the 2D resolved velocity field. These explicitly calculated stresses can be linked to averaged velocities and flow parameters, to form closure terms for our 1D low fidelity model.

Chapter 3

1D Two-Fluid Model

3.1 Introduction

In this chapter, we discuss our low fidelity model. This is our 1D model, which can be evaluated at low computational cost, at the expense of requiring closure and neglecting some of the physics. In this chapter the assumptions made to arrive at this 1D model will be laid out and it will be shown where the need for closure terms arises. We will discuss the meaning of the closure terms, and conventional versions and their limitations. Afterwards we will discuss the connection between the model equations for a 3D geometry and a 2D geometry, the latter of which we focus on in this project.

The two-fluid model that will be described here is a quasi-1D model. It was introduced by Wallis [73], and worked on further by e.g. Ishii [74]. It is used by many companies as a basis for the simulation of multiphase pipe flow [75]. An insightful derivation of the two-fluid model is offered by [76], which is repeated below. For a more extensive discussion of the two-fluid model see e.g. [77].

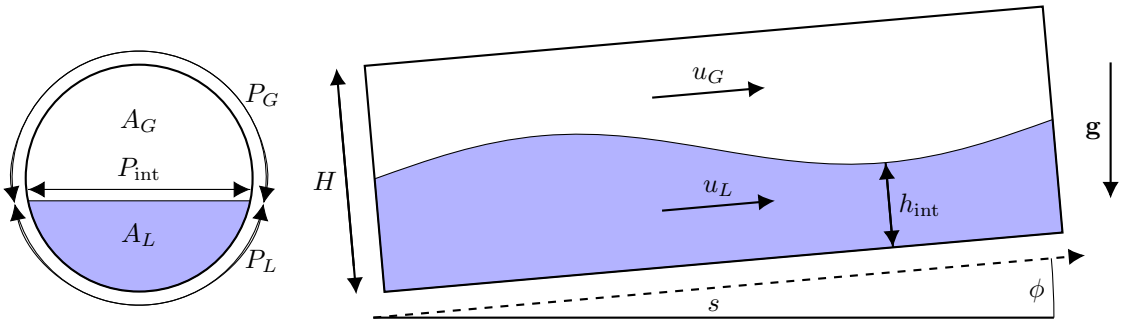


Figure 3.1: A schematic of the geometry of the 1D two-fluid model for pipe flow.

Figure 3.1 shows the geometry of the problem of interest. We have a long pipe with horizontal coordinate s, that may be tilted at an angle ϕ. As a function of s we may define the cross-sectional areas A_L and A_G of two different fluids, with the subscripts L and G suggesting 'liquid' and 'gas'. The liquid and the gas interact with each other along the interfacial perimeter P_{int} and with the wall along the wall perimeters P_L and P_G respectively.

3.2 Single-phase flow

We start the derivation considering only single-phase flow. The starting points for the derivation of Sanderse [76] are the integral mass (2.1) and momentum (2.11) balances. The balances are applied to a three-dimensional pipe segment of arbitrary geometry. Fluid can only flow in or out of the pipe segment at the streamwise boundaries. The other boundaries are impenetrable and will in principle be no-slip boundaries.

We define

$$\{f\}(t) = \frac{1}{V}\int_V f(\mathbf{x},t)\,\mathrm{d}V, \qquad V = \int_V \mathrm{d}V, \tag{3.1}$$

or, in words, the volume average of a scalar quantity f in a volume V is equal to the inverse of the volume multiplied by the integral of the quantity over the volume V. The original quantity is a function of time and position, but the volume average, for a given volume, is only a function of time. The area average (often a cross-sectional average) is defined similarly:

$$\langle f \rangle (s,t) = \frac{1}{A} \int_A f(\mathbf{x},t)\,\mathrm{d}A, \qquad A = \int_A \mathrm{d}A. \tag{3.2}$$

The cross-sectional average is still a continuous function of the streamwise coordinate s. Using these definitions the integral balances can be rewritten in terms of the volume and area averages of the various unknowns (ρ, $\mathbf{u}$).

For the mass balance of a small volume $V \approx A(s)\delta s$, pictured in Figure 3.2, we obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\{\rho\}(t)A_1\delta s\right) + \langle \rho u \rangle (s+\delta s,t)A_2 - \langle \rho u \rangle (s,t)A_1 = 0, \tag{3.3}$$

where u is the streamwise velocity, arising from the $\mathbf{u}\cdot\mathbf{n}$ term in the integral mass balance, and $A_1 = A(s)$ and $A_2 = A(s+\delta s)$. This equation is divided by δs and the limit $\delta s \to 0$ is taken. In this limit the volume average becomes an area average which is a function of s. The approximation

$$\langle fg \rangle \approx \langle f \rangle \langle g \rangle \tag{3.4}$$

is used to simplify the expression. The equation for mass conservation then becomes

$$\frac{\partial}{\partial t}\left(\rho A\right) + \frac{\partial}{\partial s}\left(\rho u A\right) = 0, \tag{3.5}$$

in which all variables now represent cross-sectional averages ($u = \langle u \rangle$) and are functions of s and t only.

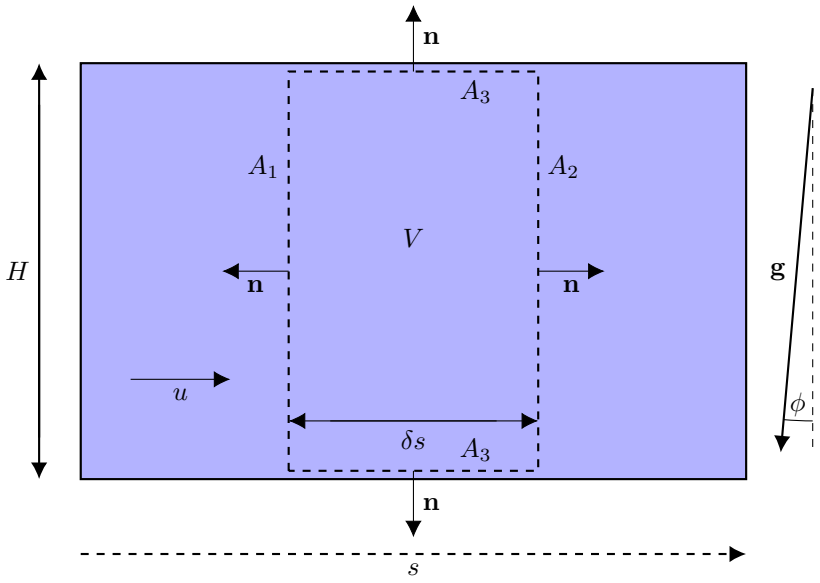


Figure 3.2: A small ($\delta s \ll H$) control volume for single-phase pipe flow. At the top and bottom the control volume is bounded by impenetrable no-slip boundaries. The reference frame is rotated compared to Figure 3.1, so that the gravitational vector is viewed as standing at an angle ϕ relative to the domain.

The derivation of the momentum balance from (2.11) is exactly analogous. The only notable difference is that the no-slip walls enter the equation. They enter through the stresses that are nonzero there, and the wall pressure that acts upon the control volume and may have an influence on the streamwise momentum balance if the normal $\mathbf{n}$ of the wall has a streamwise component. The intermediate equation for a small control volume with $V \approx A(s)\delta s$ is [76]

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\left(\{\rho u\}V\right) + \langle \rho u^2 \rangle_2 A_2 - \langle \rho u^2 \rangle_1 A_1 = &-\left(\langle p \rangle_2 A_2 - \langle p \rangle_1 A_1\right) - \langle p\mathbf{e}_s \cdot \mathbf{n} \rangle_3 A_3 \\ &- g\sin(\phi)\{\rho\}V + \langle \tau_{ss} \rangle_2 A_2 - \langle \tau_{ss} \rangle_1 A_1 + \langle \tau_{\mathrm{tot}} \rangle_3 A_3,\end{aligned} \tag{3.6}$$

in which A_1 is the inflow boundary at s, A_2 is the outflow boundary at $s + \delta s$, and A_3 is the no-slip boundary. The angle ϕ is the inclination of the pipe segment in the streamwise direction. The term τ_{tot} is the total streamwise stress acting at the pipe wall

$$\tau_{\text{tot}} = (\boldsymbol{\tau} \cdot \mathbf{n}) \cdot \mathbf{e}_s, \tag{3.7}$$

with $\mathbf{e}_s$ the unit vector in the streamwise direction.

The average streamwise stress at the no-slip boundary is $\langle \tau_{\text{tot}} \rangle_3$ and it can be written as

$$\langle \tau_{\text{tot}} \rangle_3 A_3 = \int_s \bar{\tau}_{\text{tot}} P \, \mathrm{d}s \approx \bar{\tau}_{\text{tot}} P \delta s, \tag{3.8}$$

with $\bar{\tau}_{\text{tot}}$ the average stress along the perimeter (a function of s) and P the perimeter length. The viscous normal stress at inflow and outflow $\langle \tau_{ss} \rangle_1 A_1$ and $\langle \tau_{ss} \rangle_2 A_2$ are neglected (see section 3.4).

Like for the mass balance, (3.6) reduces to

$$\frac{\partial}{\partial t}(\rho u A) + \frac{\partial}{\partial s}\left(\rho u^2 A\right) = -\frac{\partial p}{\partial s} A - \rho g A \sin\phi + \tau_{\text{tot}} P, \tag{3.9}$$

in which again all variables are area averages, except for the stress which is a perimeter average. Note that the pressure term differs from the convection term in the sense that it is not in conservative form. This is because of the wall pressure term at A_3 in (3.6), which is nonzero for the case of pipes with changing cross-sectional area.

3.3 Stratified two-phase flow

In this thesis we will consider stratified multiphase flows, to which the equation describing mass conservation (3.5) can be applied to each phase individually. The control volume considered before can be split into two at the interface between the two fluids, as illustrated in Figure 3.3. Since no mass can traverse the interface (see subsection 2.3.1), for the purpose of mass conservation this boundary behaves exactly like the pipe wall, with boundary condition (2.24). The mass balance thus applies to both fluids separately, with the understanding that the cross-sectional area in the equation represents the cross-sectional area of the concerned fluid. The density and velocity also differ per fluid. We note the two mass balances for the different fluids:

$$\frac{\partial}{\partial t}(\rho_L A_L) + \frac{\partial}{\partial s}(\rho_L u_L A_L) = 0, \tag{3.10}$$

$$\frac{\partial}{\partial t}(\rho_G A_G) + \frac{\partial}{\partial s}(\rho_G u_G A_G) = 0, \tag{3.11}$$

where we choose to refer to one fluid with the subscript L (liquid) and the other with subscript G (gas).

For the momentum balance (3.9) the extension is slightly less straightforward. Momentum also cannot traverse the interface, since momentum is carried by a mass flow. Pressure and friction forces do act upon the split up control volumes at this interface, according to boundary conditions (2.30) and (2.31). The A_3 in the intermediate equation (3.6) now includes the interface, and excludes the pipe wall in contact with the other fluid.

As in the single phase case, the wall pressure term combines with the inflow and outflow pressures into a $(\partial p/\partial s)A$ term. However, this pressure is now the average pressure in the fluid under consideration. The wall friction term in (3.6) is split into contributions from the pipe wall and from the interface. The result for the two fluids can be written

$$\frac{\partial}{\partial t}(\rho_L u_L A_L) + \frac{\partial}{\partial s}\left(\rho_L u_L^2 A_L\right) = -\frac{\partial p_L}{\partial s} A_L + \tau_L P_L - \tau_{\text{int}} P_{\text{int}} - \rho_L A_L g \sin\phi, \tag{3.12}$$

$$\frac{\partial}{\partial t}(\rho_G u_G A_G) + \frac{\partial}{\partial s}\left(\rho_G u_G^2 A_G\right) = -\frac{\partial p_G}{\partial s} A_G + \tau_G P_G + \tau_{\text{int}} P_{\text{int}} - \rho_G A_G g \sin\phi, \tag{3.13}$$

in which again all variables are area averages (e.g. $(\partial p_L/\partial s)\, A_L \to \langle (\partial p_L/\partial s)\, A_L \rangle = \langle \partial p_L/\partial s \rangle A_L = \int_{A_L} (\partial p_L/\partial s)\, \mathrm{d}A$). At the interface between the two fluids we may expect to see the effects of surface tension. However for large Weber numbers (which arise in large scale flows) and low surface curvature, its effects may be neglected.

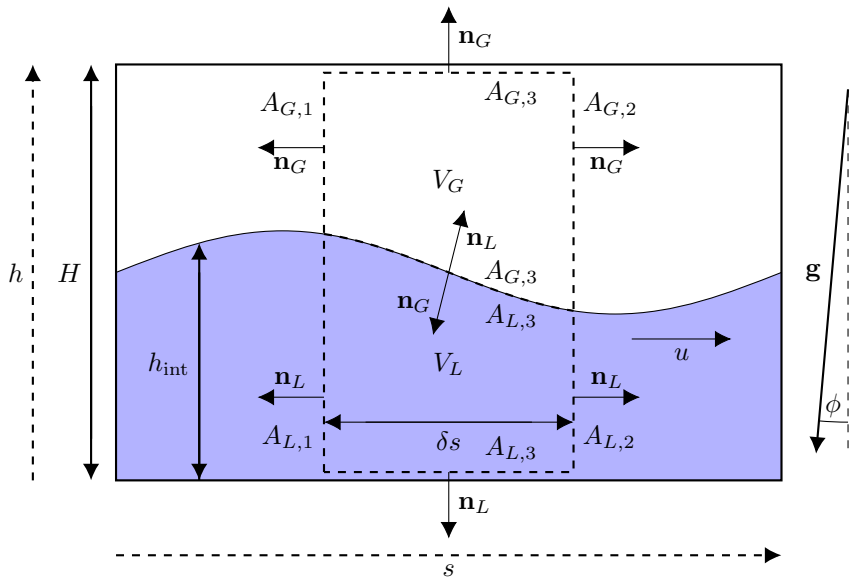


Figure 3.3: Two small ($\delta s \ll H$) control volumes for two-phase pipe flow. At the top and bottom the control volume is bounded by impenetrable no-slip boundaries. At the interface the two control volumes meet. Mass and momentum are conserved for the two control volumes separately.

We can reduce the number of unknowns in (3.12) and (3.13) using the vertical momentum balance. We assume a hydrostatic balance, i.e.

$$\frac{\partial p}{\partial h} = -\rho g \cos(\phi), \tag{3.14}$$

with h the vertical coordinate. This is possible if the vertical velocity is negligible. Then the pressure of the two fluids as a function of height (these are not averaged cross-sectionally like the p_L and p_G in (3.12) and (3.13)) can be defined relative to the interface pressure p_{int} at height h_{int}:

$$p_L(h) = p_{\text{int}} - \rho_L g \cos(\phi)\,(h - h_{\text{int}})\,, \tag{3.15}$$

$$p_G(h) = p_{\text{int}} - \rho_G g \cos(\phi)\,(h - h_{\text{int}})\,. \tag{3.16}$$

Normally the liquid will be below the gas so that the pressure in the liquid will be higher.

Now we want to have (3.12) and (3.13) be functions of only the interface pressure and not the average pressure in both fluids. To achieve this the integral $\langle \partial p_L/\partial s \rangle A_L$ is split into a term $(\partial p_{\text{int}}/\partial s)\, A_L$ and a remainder, using (3.15) (and the same for the gas). The result is

$$\frac{\partial}{\partial t}\left(\rho_L u_L A_L\right) + \frac{\partial}{\partial s}\left(\rho_L u_L^2 A_L\right) = -\frac{\partial p_{\text{int}}}{\partial s} A_L + LG_L + \tau_L P_L - \tau_{\text{int}} P_{\text{int}} - \rho_L A_L g \sin\phi, \tag{3.17}$$

$$\frac{\partial}{\partial t}\left(\rho_G u_G A_G\right) + \frac{\partial}{\partial s}\left(\rho_G u_G^2 A_G\right) = -\frac{\partial p_{\text{int}}}{\partial s} A_G + LG_G + \tau_G P_G + \tau_{\text{int}} P_{\text{int}} - \rho_G A_G g \sin\phi, \tag{3.18}$$

in which the two remainder terms LG_L and LG_G have appeared. These are the so-called level gradient terms, and they represent the effect of streamwise pressure variation due to deformation of the interface. They are given by

$$LG_L = \frac{\partial}{\partial s}\int_{A_L} \rho_L g \cos(\phi)(h - h_{\text{int}})\,\mathrm{d}A, \tag{3.19}$$

$$LG_G = \frac{\partial}{\partial s}\int_{A_G} \rho_G g \cos(\phi)(h - h_{\text{int}})\,\mathrm{d}A, \tag{3.20}$$

and in a rectangular duct geometry with width W, height H, and area $A = WH$ (see Figure 3.4), they evaluate to

$$LG_L = -\frac{\partial}{\partial s}\left[\frac{1}{2}\rho_L g \cos\phi\, \frac{A}{H} h_{\text{int}}^2\right] = -\frac{\partial}{\partial s}\left[\frac{1}{2}\rho_L g \cos\phi\, \frac{H}{A} A_L^2\right], \tag{3.21}$$

$$LG_G = \frac{\partial}{\partial s}\left[\frac{1}{2}\rho_G g \cos\phi\, \frac{A}{H}\left(H - h_{\text{int}}\right)^2\right] = \frac{\partial}{\partial s}\left[\frac{1}{2} g \cos\phi \frac{H}{A} \rho_G A_G^2\right]. \tag{3.22}$$

For incompressible flow in a cylindrical geometry the level gradients are often expressed as [75]

$$LG_L = -\rho_L g A_L \frac{\partial h_{\text{int}}}{\partial s} \tag{3.23}$$

$$LG_G = -\rho_G g A_G \frac{\partial h_{\text{int}}}{\partial s}. \tag{3.24}$$

The above expressions need to be closed by a relation $h_{\text{int}} = h_{\text{int}}(A_L)$. If we fill in $h_{\text{int}} = (H/A)A_L = (H/A)(A - A_G)$ for rectangular duct flow, we recover (3.21) and (3.22). For the cylindrical geometry the connection $h_{\text{int}} = h_{\text{int}}(A_L)$ can be made via the so-called wetted angle according to Biberg's relation [78], but we will consider the rectangular duct geometry from now on for simplicity.

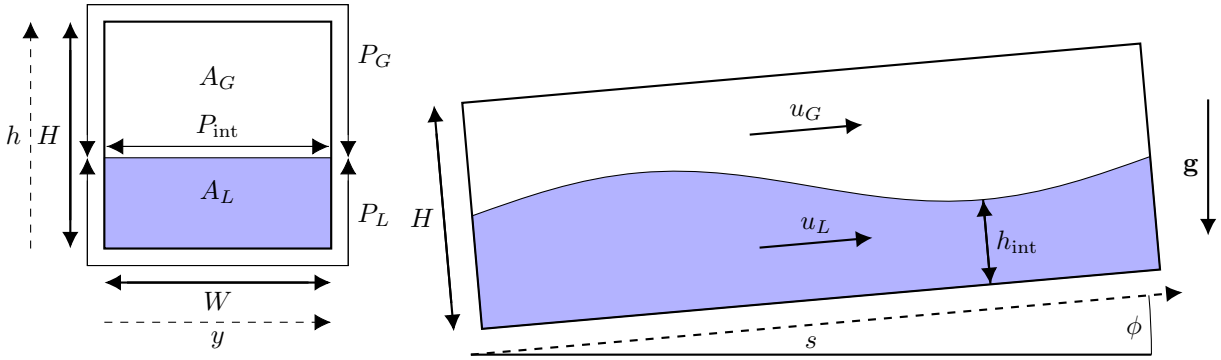


Figure 3.4: A schematic of the geometry of the 1D two-fluid model for rectangular duct flow, with total cross-sectional area $A = WH$.

3.4 Stress terms

3.4.1 Introduction

In this section we first expand our understanding of the stress terms appearing in the momentum balances (3.17) and (3.18). We focus on the rectangular duct geometry of Figure 3.4 for simplicity. Afterwards we discuss conventional relations for these stresses, called closure terms. We discuss differences between the closure terms for single-phase and two-phase flow, and for various duct geometries and flow conditions. Finally we examine an analytical solution for laminar, fully developed, steady state two-phase flow and the closure terms which can be derived from it.

3.4.2 Interpretation of the stress terms

The total stress at the no-slip boundaries of the two-phase control volumes are $\langle \tau_{\text{tot}} \rangle_{L,3} A_{L,3}$ for the liquid and $\langle \tau_{\text{tot}} \rangle_{G,3} A_{G,3}$ for the gas and they can be written as

$$\langle \tau_{\text{tot}} \rangle_{L,3} A_{L,3} = \int_s \tau_L P_L \, \mathrm{d}s + \int_s -\tau_{\text{int}} P_{\text{int}} \, \mathrm{d}s \approx \tau_L P_L \delta s - \tau_{\text{int}} P_{\text{int}} \delta s, \tag{3.25}$$

$$\langle \tau_{\text{tot}} \rangle_{G,3} A_{G,3} = \int_s \tau_G P_G \, \mathrm{d}s + \int_s \tau_{\text{int}} P_{\text{int}} \, \mathrm{d}s \approx \tau_G P_G \delta s + \tau_{\text{int}} P_{\text{int}} \delta s \tag{3.26}$$

(compare to (3.8)). The stresses acting at the solid boundaries are given by

$$\tau_L = \frac{1}{P_L} \int_{P_L} \tau_{\text{tot}} \, \mathrm{d}P \tag{3.27}$$

$$\tau_G = \frac{1}{P_G} \int_{P_G} \tau_{\text{tot}} \, \mathrm{d}P, \tag{3.28}$$

if the duct cross-section A is constant, and the interfacial stresses are given by

$$\tau_{L,\text{int}} = \frac{1}{P_\text{int}} \int_{P_\text{int}} \frac{\tau_\text{tot}}{|\mathbf{e}_h \cdot \mathbf{n}_L|} \, \mathrm{d}P, \tag{3.29}$$

$$\tau_{G,\text{int}} = \frac{1}{P_\text{int}} \int_{P_\text{int}} \frac{\tau_\text{tot}}{|\mathbf{e}_h \cdot \mathbf{n}_G|} \, \mathrm{d}P, \tag{3.30}$$

with τ_tot calculated from the respective control volumes. The cross-sectionally averaged stresses which appear in the momentum balances (3.17) and (3.18) are

$$\langle \tau_\text{tot} \rangle_{L,3} = \tau_L P_L + \tau_{L,\text{int}} P_\text{int} \tag{3.31}$$

$$\langle \tau_\text{tot} \rangle_{G,3} = \tau_G P_G + \tau_{G,\text{int}} P_\text{int}. \tag{3.32}$$

All these expressions depend on the local (unaveraged) streamwise stress τ_tot

$$\tau_\text{tot} = (\boldsymbol{\tau} \cdot \mathbf{n}) \cdot \mathbf{e}_s. \tag{3.33}$$

In the above expression, $\boldsymbol{\tau} \cdot \mathbf{n}$ is the stress acting at the boundary A_3 with normal $\mathbf{n}$, and the inner product with $\mathbf{e}_s$ is taken because we need its influence on the streamwise momentum balance. The other momentum balances are not considered. Regardless, the vertical momentum balance is trivial with $w = 0$, since we assume hydrostatic balance.

The form of the stress tensor for incompressible flow is (2.18), with the deformation tensor (2.19). Using these definitions one can see that in a Cartesian coordinate system, with the x coordinate replaced by the streamwise coordinate s and the z coordinate replaced by the vertical coordinate h

$$\tau_\text{tot} = \tau_{ss}(\mathbf{e}_s \cdot \mathbf{n}) + \tau_{ys}(\mathbf{e}_y \cdot \mathbf{n}) + \tau_{hs}(\mathbf{e}_h \cdot \mathbf{n}). \tag{3.34}$$

The expression for the streamwise stress acting at A_3 is thus determined by the (local) orientation $\mathbf{n}$ of A_3.

For a rectangular duct geometry one of the three stresses will act at each bounding plane[1]:

$$\tau_{ss} = 2\mu \frac{\partial u}{\partial s}, \quad \text{with } \mathbf{e}_s \cdot \mathbf{n} = 1 \text{ at the positive } (h, y) \text{ plane}, \tag{3.35a}$$

$$\tau_{ys} = \mu \left(\frac{\partial v}{\partial s} + \frac{\partial u}{\partial y} \right), \quad \text{with } \mathbf{e}_y \cdot \mathbf{n} = 1 \text{ at the positive } (s, h) \text{ plane}, \tag{3.35b}$$

$$\tau_{hs} = \mu \left(\frac{\partial w}{\partial s} + \frac{\partial u}{\partial h} \right), \quad \text{with } \mathbf{e}_h \cdot \mathbf{n} = 1 \text{ at the positive } (s, y) \text{ plane}, \tag{3.35c}$$

and at the opposing bounding planes the inner products are negative. Generally, the no-slip walls need not align with any bounding plane, but for a rectangular duct geometry they will align with the (s, h) and (s, y) planes, with τ_{ys} and τ_{hs} the corresponding shear stresses.

At the inflow and outflow of a control volume (in either a cylindrical or rectangular duct geometry) we have only $\tau_\text{tot} = \mp\tau_{ss}$, which is a normal stress (not a shear stress). As stated in section 3.2 this stress is neglected. If it were not neglected, it would gives rise to a term

$$\frac{\partial \tau_{ss} A}{\partial s} = \frac{\partial \tau_{ss}}{\partial s} A + \tau_{ss} \frac{\partial A}{\partial s} = 2\mu \frac{\partial^2 u}{\partial s^2} A + 2\mu \frac{\partial u}{\partial s} \frac{\partial A}{\partial s} \tag{3.36}$$

in the final equation for single phase flow (3.9), and likewise for (3.17) and (3.18). This term must thus compete with the pressure gradient and the shear stresses. In the single-phase cross-sectionally averaged model this term must be identically zero to satisfy the assumption of incompressibility, which with $v = w = 0$ reduces to $\partial u / \partial s = 0$. In two-phase flow $\partial u / \partial s$ may be nonzero but will be small if the characteristic horizontal length scale is much larger than the characteristic vertical length scale, i.e. $L \gg H$.

As derived in subsection 2.3.1, at the solid walls the stress is purely in the tangential direction and of the form

$$\tau = \mathbf{t} \cdot \mu \boldsymbol{\nabla} \mathbf{u} \cdot \mathbf{n}. \tag{3.37}$$

[1]Here the notation '(x_1, x_2) plane' means the plane at which x_1 and x_2 are variable and x_3 is fixed. The normal of the plane thus lies along x_3.

At the (s,h) and (s,y) planes this reduces to

$$\tau_{\text{tot}} = \mu\frac{\partial u}{\partial y} \quad \text{at the positive } (s,h) \text{ plane}, \quad \tau_{\text{tot}} = -\mu\frac{\partial u}{\partial y} \quad \text{at the negative } (s,h) \text{ plane} \tag{3.38}$$

$$\tau_{\text{tot}} = \mu\frac{\partial u}{\partial h} \quad \text{at the positive } (s,y) \text{ plane}, \quad \tau_{\text{tot}} = -\mu\frac{\partial u}{\partial h} \quad \text{at the negative } (s,y) \text{ plane} \tag{3.39}$$

i.e. τ_{tot} represents the wall friction due to the streamwise velocity and its gradient normal to the no-slip walls. The found τ_{tot} can be filled into (3.27) and (3.28) and the integrals can be calculated.

In two-phase flow the interface behaves like a no-slip wall for the fluids considered separately, except for the fact that this wall can move. The general form of the streamwise stress is given by (3.34), (3.35). Since in the two-fluid model h_{int} is only a function of s (not of y), the interface normal vector cannot have a component in the $\mathbf{e}_y$ direction so that $(\mathbf{e}_y \cdot \mathbf{n}) = 0$ and only the first and third terms in (3.34) remain. We have already said to neglect τ_{ss} and thus only $\tau_{hs}(\mathbf{e}_h \cdot \mathbf{n})$, given by (3.35c) remains. With the assumption $w = 0$ (due to hydrostatic balance) or $L \gg H$ (see subsection 2.3.1) this reduces to

$$\tau_{\text{tot}} = \mu\frac{\partial u}{\partial h}(\mathbf{e}_h \cdot \mathbf{n}). \tag{3.40}$$

In subsection 2.3.1 we had already found that under the condition of hydrostatic balance (2.41) holds, with $\mathbf{n}_L$ as in Figure 3.3 and $\mathbf{t}$ as in Figure 2.2:

$$\boldsymbol{\tau}_{\text{int}} = \mathbf{t} \cdot \boldsymbol{\tau}_{\text{int}} = \mathbf{t} \cdot \boldsymbol{\tau}_L \cdot \mathbf{n}_L = \mathbf{t} \cdot \boldsymbol{\tau}_G \cdot \mathbf{n}_L, \tag{3.41}$$

i.e. the interfacial stress acts tangential to the interface and the stress acting on the gas and the stress acting on the liquid are opposite but equal in magnitude. Thus, filling in (3.40) in (3.29) and (3.30), the factor $(\mathbf{e}_h \cdot \mathbf{n})$ cancels against $|(\mathbf{e}_h \cdot \mathbf{n})|$ except for a minus sign for the gas, and we find

$$\tau_{G,\text{int}} = -\tau_{L,\text{int}} = \frac{1}{P_{\text{int}}}\int_{P_{\text{int}}} -\mu_G\frac{\partial u_G}{\partial h}\,\mathrm{d}P = \frac{1}{P_{\text{int}}}\int_{P_{\text{int}}} -\mu_L\frac{\partial u_L}{\partial h}\,\mathrm{d}P. \tag{3.42}$$

To agree with convention, we define

$$\tau_{\text{int}} = \tau_{G,\text{int}} \tag{3.43}$$

so that we can rewrite (3.31) and (3.32) as

$$\langle\tau_{\text{tot}}\rangle_{L,3} = \tau_L P_L - \tau_{\text{int}} P_{\text{int}}, \tag{3.44}$$

$$\langle\tau_{\text{tot}}\rangle_{G,3} = \tau_G P_G + \tau_{\text{int}} P_{\text{int}}, \tag{3.45}$$

which is what has been used in writing the intermediate versions (3.12) and (3.13) of the momentum balances.

Even in the greatly simplified expressions (3.38), (3.39), and (3.40), we have a problem. They require knowledge of the dependence of u on h and y. But in the two-fluid model the momentum balances (3.17), (3.18) are derived for control volumes containing the whole domain from the interface to the pipe wall; only in the s direction the limit to $\delta s = 0$ is taken. The equations form a model for the average liquid and gas velocities $\langle u_L\rangle$ and $\langle u_G\rangle$; which are functions only of s and t. Therefore we cannot compute the stresses directly. We need to make approximations for them in terms of known variables, also called 'closure terms', much like the closure terms needed to close the Reynolds-averaged Navier-Stokes equations.

Note that the stresses (3.38), (3.39), and (3.40) may be functions of h and y in a 3D simulation. However, in the 1D two-fluid model, we only require the perimeter averaged stresses τ_L, τ_G, and τ_{int}.

3.4.3 Single-phase closure terms

Definitions

Friction closure terms are conventionally based on the fully developed steady state force balance

$$P\tau = AB, \tag{3.46}$$

with B the body force

$$B = \frac{\partial p}{\partial s} + \rho g \sin(\phi). \tag{3.47}$$

The force balance can equivalently be written as

$$P\tau = A\frac{\partial p_{\text{mod}}}{\partial s}, \tag{3.48}$$

with p_{mod} the modified pressure including the gravitational potential:

$$p_{\text{mod}} = p + \rho g \sin(\phi) s. \tag{3.49}$$

The force balance yields an expression for the stress at steady state as a function of the body force. But it gives no information on the averaged fluid velocity. We therefore need to conduct experiments, or analyze the 2D or 3D equations, to obtain expressions relating the body force and the fluid velocity, and by extension expressions relating the stress and the averaged fluid velocity.

A relation of the form $\tau(B)$ will not generalize to unsteady flow, since B will not change throughout the evolution of a flow to steady state. A stress that does *not* increase with increasing velocities also means that an unsteady simulation starting from a zero velocity field will never reach a steady state; the velocities can keep growing without a balancing stress.

A relation of the form $\tau(u)$ however, has potential for generalizing to unsteady flow, if the form of the velocity profile is similar between the unsteady flow under consideration and the fully developed steady state solution. This is because the stresses are directly determined by the slope of the velocity profile at the boundaries, via expressions like (3.38), (3.39), and (3.40). The averaged velocity u will correlate with the slopes of the velocity profile, though we cannot hope for a unique relation, because different velocity profiles are possible which yield the same averaged velocity.

The general (conventional) form for the wall friction in the case of positive velocity is

$$\tau = -\frac{1}{2} f \rho u^2, \tag{3.50}$$

with u the average velocity. This form can be applied to different flows as long as we adjust the Fanning friction factor to the flow at hand. In this light the Fanning friction factor of a steady fully developed flow through a conduit is *defined* [79]

$$f = -\frac{\tau}{\frac{1}{2}\rho u^2} = -\frac{1}{4} D_{\text{h}} \frac{\partial p_{\text{mod}}}{\partial s} \frac{1}{\frac{1}{2}\rho u^2}. \tag{3.51}$$

Note that for $u > 0$, $\partial p_{\text{mod}}/\partial s < 0$. The hydraulic diameter D_{h} is defined as

$$D_{\text{h}} = \frac{4A}{P}. \tag{3.52}$$

Sometimes the Darcy (a.k.a. Darcy-Weisbach, or Moody) friction factor is used, which is just four times the Fanning friction factor. The Darcy friction factor ($4f$) appears in the Darcy-Weisbach equation for head loss $\Delta h = \Delta p_{\text{mod}}/\rho g$ which can be obtained from (3.51) by substituting $\partial p_{\text{mod}}/\partial s \to \Delta p_{\text{mod}}/\Delta s$:

$$\Delta h = 4f \frac{\Delta s}{D_{\text{h}}} \frac{u^2}{2g}. \tag{3.53}$$

The friction factor can be derived analytically when assuming incompressible, fully developed, stationary, laminar flow in a pipe geometry (Poiseuille flow). Then we can use the 3D steady state balance to find a direct relation between the pressure gradient and the averaged velocity, and then via (3.51) an equation for the friction factor. In this case the Fanning friction factor f can be shown analytically to be [80, p. 181]

$$f = \frac{16}{\text{Re}}, \tag{3.54}$$

(i.e. $C = 16$ and $n = 1$, as noted above) with the Reynolds number defined

$$\text{Re} = \frac{\rho u D_{\text{h}}}{\mu}. \tag{3.55}$$

For a circular pipe the hydraulic diameter given by (3.52) is just the pipe diameter, and for a rectangular channel with width W it is given by $D_{\text{h}} = \frac{2HW}{H+W}$. In the limiting case $W \gg H$ we get

$D_\mathrm{h} = 2H$. For laminar rectangular cross-section pipe flow an analytical friction factor as a function of the aspect ratio W/H has been derived by Cornish (1928) [81]. For a square cross-section ($W = H$) the friction factor is

$$f = \frac{\sqrt{\pi}}{2} \cdot \frac{16}{\mathrm{Re}}, \tag{3.56}$$

and for a pipe with an infinite width to depth ratio ($W/H \to \infty$) the friction factor becomes

$$f = \frac{3}{2} \cdot \frac{16}{\mathrm{Re}}. \tag{3.57}$$

These relations have been experimentally validated by [82], with some discrepancy. There are two differences in these equations with respect to (3.54); they contain a constant pre-factor close to 1 and the Reynolds number is based on a different hydraulic diameter (via (3.52) and (3.55)).

Pipe flow empirical friction factors

For turbulent flow the friction factor can not be derived analytically but must be determined experimentally using (3.51). Already in 1913, Blasius [83] determined f experimentally for turbulent pipe flow ($\mathrm{Re} > 3000$) to be

$$f = \frac{0.0791}{\mathrm{Re}^{1/4}}. \tag{3.58}$$

This relation is still considered to be a very accurate approximation to the experimental data for $\mathrm{Re} < 10^5$[84]. The Blasius equation holds for a smooth pipe. For laminar flow the pipe roughness has no influence on the friction factor, but for turbulent flow it does [85]. A widely used equation which takes into account the roughness of the pipe was introduced by Colebrook and White [86]:

$$\frac{1}{\sqrt{f}} = -1.737 \ln\left(\frac{\epsilon}{3.707 D_\mathrm{h}} + \frac{1.254}{\mathrm{Re}\sqrt{f}}\right), \tag{3.59}$$

in which ϵ represents the effective surface roughness height. The Colebrook-White equation is not practical to use directly since it is an implicit equation, but good explicit approximations exist (see e.g. [39]).

In the limit of fully developed turbulent flow, with $\mathrm{Re} \to \infty$, the second term in the logarithm in (3.59) becomes negligible and the equation reduces to the correlation found by Nikuradse [87]:

$$\frac{1}{\sqrt{f}} = 1.737 \ln\left(3.707 D_\mathrm{h}/\epsilon\right). \tag{3.60}$$

In the smooth pipe limit, $\epsilon/D_\mathrm{h} \to 0$ and the first term in the logarithm in (3.59) becomes negligible:

$$\frac{1}{\sqrt{f}} = 1.737 \ln\left(0.7962\,\mathrm{Re}\sqrt{f}\right). \tag{3.61}$$

This is the friction factor correlation found by Prandtl and Von Kármán [88]. It is discussed in [89].

In [90] (after the work done in [91], [92]) Churchill combined experimental correlations for the friction factor in different flow regimes to obtain an equation for the friction factor which is valid in all flow regimes. He combined (3.54), (3.60) and another correlation by Colebrook [86] for smooth pipes, though he notes that in place of this last correlation (3.61) may also be used. He furthermore added an empirical relation for the friction factor in the transitional regime based on data by Wilson and Azad [93]. The combined result is

$$f = 2\left[\left(\frac{8}{\mathrm{Re}}\right)^{12} + \frac{1}{(A+B)^{3/2}}\right]^{1/12}, \tag{3.62}$$

with

$$A = \left[2.457 \ln \frac{1}{\left(\frac{7}{\mathrm{Re}}\right)^{0.9} + \frac{0.27\epsilon}{D_\mathrm{h}}}\right]^{16} \quad \text{and} \quad B = \left(\frac{37,530}{\mathrm{Re}}\right)^{16}, \tag{3.63}$$

which can readily be seen to reduce to (3.54) for $\mathrm{Re} \to 0$.

Channel and rectangular duct flow friction factors

The Churchill correlation does not hold for channel flow in the laminar regime; for this case the friction factor is missing the factor (3/2) from (3.57). However, in early literature it is postulated that in the turbulent regime the friction factors for pipe flow and rectangular cross-section duct flow follow approximately the same relation, given that the hydraulic diameter given by (3.52) is used [82], [94]–[96]. In this context the hydraulic diameter is known as the *equivalent* diameter. Vennard (1940) [95] explains that for turbulent flow in a rough pipe the equivalence holds because for this type of flow the resistance is mainly determined by the extent of the wetted perimeter, of which the effect is incorporated in the friction factor relation through the hydraulic diameter. The resistance is mainly determined by the extent of the wetted perimeter since the friction phenomena are confined to a thin boundary layer at the pipe surface. For laminar flow the equivalence does not hold since in this case friction phenomena result from the fluid viscosity and are distributed throughout the whole pipe cross-section.

In later work though, for rectangular duct flow with high aspect ratios W/H the friction factor is observed to tend upwards of the circular pipe flow correlations at constant Reynolds number, with the conventional hydraulic diameter given by (3.52) [97]. Jones (1976) [97], following a similar idea by Rehme (1973) [98], introduced a modified Reynolds number for use in the friction factor relations as

$$\mathrm{Re}^* = \frac{\rho u D_\mathrm{L}}{\mu}, \tag{3.64}$$

with a *laminar-equivalent* diameter

$$D_\mathrm{lam} = \phi^* D_\mathrm{h}, \tag{3.65}$$

with ϕ^* a geometry factor which is a function of W/H, defined such that for laminar flow all geometries satisfy the friction factor relation

$$f = \frac{16}{\mathrm{Re}^*}. \tag{3.66}$$

By comparison with (3.54), (3.56), and (3.57), we conclude that for circular pipe flow $\phi^* = 1$, for square cross sections $\phi^* = 2/\sqrt{\pi}$ and for infinite width channel flow $\phi^* = 2/3$. These geometry factors are defined for arbitrary rectangular duct flow aspect ratios using the general laminar analytical friction factor solution of Cornish (1928) [81].

Substituting his modified Reynolds number in the smooth pipe form of the Colebrook-White equation (3.61), Jones finds better agreement with a range of experimental studies of turbulent flow in rectangular ducts. Mishima et al. [99] also found experimental agreement to Jones' model. We may expect that in any case for smooth pipes the Churchill correlation (3.62) with Jones' modified Reynolds number will produce sufficiently accurate friction factors, since in the laminar limit $\mathrm{Re}^* \to 0$ the correlation is exact and in the turbulent regime it is based on an equation similar to the Colebrook-White equation. In the turbulent fully rough regime the friction factor is independent of Reynolds number and of the duct geometry, except for the direct dependence on wetted perimeter through the hydraulic diameter [100].

Sadotomi et al. [101] employ a similar approach to Jones but takes the geometry factor out of the Reynolds number (and thus out of the power). In this approach their geometry factor for turbulent flow is not equal to that of laminar flow (as in Jones' method) but is an implicit function of it. He and Gotts (2004) [102] provide a literature review of different friction factor adaption methods and introduce a new method in which the laminar flow solution is not needed. Duan et al. [100] regard fully developed turbulent flow in non-circular geometries and propose a definition of the Reynolds number with the length scale being the square root of the duct cross-sectional area.

3.4.4 Two-phase closure terms

Conventional closure terms

For multiphase flow the situation is far more complicated. Instead of one steady state force balance, we now have two separate force balances, coupled by the interfacial stress:

$$P_L \tau_L - P_\mathrm{int} \tau_\mathrm{int} = A_L B_L, \tag{3.67}$$

$$P_G \tau_G + P_\mathrm{int} \tau_\mathrm{int} = A_G B_G, \tag{3.68}$$

with the body forces

$$B_L = \frac{\partial p}{\partial s} + \rho_L g \sin(\phi), \tag{3.69}$$

$$B_G = \frac{\partial p}{\partial s} + \rho_G g \sin(\phi). \tag{3.70}$$

This means that we do not have a fully determined system, so that the three stresses (as functions of the fluid velocities) cannot be determined directly from experimental data[2]. Now, if we experimentally or analytically determine expressions for the body forces as functions of the (now two) averaged velocities, these do not directly yield expressions for the stresses as functions of the velocities. In order to determine the stresses, assumptions must be made about the form of the stresses, or a relation between the stresses must be assumed. The forms of the stresses are therefore based on the single-phase forms, and the interfacial friction factor is often based on the gas friction factor.

The conventional closure terms for the perimeter averaged stresses in 1D stratified gas-liquid flow are given by Taitel and Dukler (1976) [103]. A slight modification to make them directional yields (compare to (3.50))

$$\tau_L = -\frac{1}{2} f_L \rho_L u_L |u_L|, \quad \tau_G = -\frac{1}{2} f_G \rho_G u_G |u_G|, \quad \tau_{\text{int}} = -\frac{1}{2} f_{\text{int}} \rho_G \left(u_G - u_L\right) |u_G - u_L|. \tag{3.71}$$

The Taitel and Dukler Fanning friction factors are given by

$$f_L = \frac{C}{\text{Re}_L^n}, \quad f_G = \frac{C}{\text{Re}_G^n}, \tag{3.72}$$

with Reynolds numbers

$$\text{Re}_L = \frac{\rho_L |u_L| D_L}{\mu_L}, \quad \text{Re}_G = \frac{\rho_G |u_G| D_G}{\mu_G}, \tag{3.73}$$

hydraulic diameters

$$D_L = \frac{4A_L}{P_L}, \quad D_G = \frac{4A_G}{P_G + P_{LG}}, \tag{3.74}$$

and coefficients

$$C = 0.046 \text{ for turbulent flow}, \quad \text{or} \quad C = 16 \text{ for laminar flow}, \tag{3.75}$$

and

$$n = 0.2 \text{ for turbulent flow}, \quad \text{or} \quad n = 1 \text{ for laminar flow}. \tag{3.76}$$

The coefficients C and n for turbulent flow given here come from the McAdams correlation, found in e.g. [104] (noted here to hold for smooth pipe flow with Reynolds numbers between 5000 and 200,000). Choosing the hydraulic diameters in this way means that the gas-liquid interface is taken to act as a (nearly) free surface for the liquid and as a no-slip surface for the gas [105]. In a 2D channel flow the hydraulic diameters evaluate to

$$D_L = 4h_{\text{int}}, \quad D_G = 2(H - h_{\text{int}}). \tag{3.77}$$

It was established by Gazley (1949) [106] that for smooth, stratified, cocurrent flow, the interfacial friction factor is equal to the gas friction factor. Therefore Taitel and Dukler originally took the interfacial friction factor to simply be equal to the gas friction factor:

$$f_{\text{int}} = f_G. \tag{3.78}$$

If the interface is wavy the interfacial friction is increased relative to an identical case but with a smooth interface [107]. Johnston (1984) [108] even observes that the interfacial stresses can act opposite to the predicted direction ($-\text{sgn}\left(u_G - u_L\right)$), when the interface height moves towards the top of the pipe and the gas velocity increases. For small and regular amplitude wavy flow, [109] tested the use of a friction factor of $f_i = 0.014$, based on the data of [110]. Cheremisinoff and Davis (1979) [109] find that another relation is required for large-amplitude roll waves, which is linearly dependent on the liquid

[2]The variability of P_L, P_G, P_{int}, A_L, A_G, and the possibility of different flow regimes including wavy flow, annular flow and dispersed flow further complicate experiments.

Reynolds number. Andritsos and Hanratty (1987) [107] find results that counter this, but suggest that the the ratio f_i/f_g increases with the ratio of the wave amplitude to the wavelength and with the gas velocity. They conclude that $f_i = f_g$ if the velocity difference is lower than the critical gas velocity for Kelvin-Helmholtz instability, and that a relation for f_i can be derived to be f_g plus a term increasing with the ratio of the liquid holdup to the pipe diameter and with the ratio of the gas velocity to the critical gas velocity. Based on these considerations [111] propose an interfacial friction factor of

$$f_{\text{int}} = \max\left(f_G, 0.014\right). \tag{3.79}$$

Together the presented closures form a relatively simple set used by e.g. [112], [113].

Other empirical closure terms

The set of relations and coefficients given above is but one of many possible choices. Agrawal et al. [105] present a few different models and propose their own, in which the liquid wall friction is based on the average velocity that a pipe filled entirely with the liquid would have if the average velocity in the part actually occupied by the liquid is u_L. For turbulent flow the friction factor is then calculated using the Blasius correlation (3.58). Their interfacial friction factor carries coefficients $C = 1.3$, $n = 0.57$ and uses the gas velocity in the definition of the Reynolds number and in the shear stress expression (3.71).

Kowalski [114] performs experiments for liquid-wall, gas-wall, and interfacial friction in smooth and wavy stratified pipe flow and compares the results to a number of different models. Agreement with Agrawal et al. [105] or Taitel and Dukler [103] is only obtained for the gas-wall friction; for the other two stresses Kowalski proposes new relations. Brauner and Maron (1993) [115] propose an expression for the interfacial stress (like (3.71)) which is directly dependent on the local streamwise interfacial height gradient. An important more recent contribution to the collection of two-phase pipe friction closure models, based on theoretical considerations, was made by Biberg [116].

Using most closure models one must switch between closure terms for laminar flow and turbulent flow in some way when using them in a computer code. A simple way which avoids a sudden jump in the friction factor is to just choose which one of the two is higher for the current Reynolds number [117]. A more intricate method, employed by [113], is to use Churchill's correlation (3.62) instead of (3.72). The approach is to calculate Churchill's friction factor for the liquid and gas separately, using (3.73) and (3.74). The interface friction factor can then still be defined as (3.79). In using the Churchill correlation one is in fact using the Colebrook-White equation (3.59) on which it is partly based.

Reviews of interfacial friction factors are given by [117]–[119] and [117] also provides a review of gas and liquid wall friction factors.

Analytical closure terms

For the special case of laminar two-phase pipe flow analytical solutions exist, which can be used to derive friction factor expressions. An early analysis is given by [120]. A more recent analysis considering two-phase horizontal pipe flow with interfaces curved in the spanwise direction is given by [121]. Biberg and Halvorsen [122] considered flat interfaces in inclined pipe flow and calculated the mean interfacial and wall stresses as a function of the (constant) body forces in an insightful manner. They did this for a circular pipe geometry, and also for the considerably simpler case of a channel geometry. For the latter case, as a function of the vertical coordinate z defined relative to the position of the interface ($z = h - h_{\text{int}}$), the velocity profiles are

$$u_L(z) = -\frac{B_L h_{\text{int}}^2}{2\mu_L}\left[1 - \left(\frac{z}{h_{\text{int}}}\right)^2\right] - \frac{\tau_{\text{int}} h_{\text{int}}}{\mu_L}\left[1 + \frac{z}{h_{\text{int}}}\right], \tag{3.80}$$

$$u_G(z) = -\frac{B_G (H - h_{\text{int}})^2}{2\mu_G}\left[1 - \left(\frac{z}{H - h_{\text{int}}}\right)^2\right] + \frac{\tau_{\text{int}} (H - h_{\text{int}})}{\mu_G}\left[1 - \frac{z}{(H - h_{\text{int}})}\right], \tag{3.81}$$

with stresses

$$\tau_{\text{int}} = -\frac{\mu_G B_L h_{\text{int}}^2 - \mu_L B_G (H - h_{\text{int}})^2}{2\left[\mu_G h_{\text{int}} + \mu_L (H - h_{\text{int}})\right]}, \tag{3.82}$$

$$\tau_L = B_L h_{\text{int}} + \tau_{\text{int}}, \quad \tau_G = B_G h_{\text{int}} - \tau_{\text{int}}, \tag{3.83}$$

and body forces given by (3.69) and (3.70).

Ullmann et al. [9], [123] note that for closure of the two-fluid model, expressions for the mean stresses as functions of the averaged fluid velocities and holdup are needed, as has been discussed above. For the circular pipe geometry case, obtaining these requires numerical integration of complex expressions. However, for channel geometries, with the velocity profiles (3.80) and (3.81), exact expressions can be obtained. These are given by Ullmann et al. [9] as

$$\tau_L = -\frac{1}{2} f_L \rho_L u_L |u_L| \cdot F_L^*, \quad \tau_G = -\frac{1}{2} f_G \rho_G u_G |u_G| \cdot F_G^*, \quad \tau_{\text{int}} = -\frac{1}{2} f_{\text{G}} \rho_G \left(u_G - u_L\right) |u_G| F_{\text{int},G}^*. \tag{3.84}$$

Note the (small) difference between (3.84) and (3.71) in the definition of the interfacial stress, where the term $|u_G - u_L|$ is replaced by $|u_G|$. The friction factors are given by

$$f_L = \frac{3}{2} \cdot \frac{16}{\text{Re}_L}, \quad f_G = \frac{3}{2} \cdot \frac{16}{\text{Re}_G}, \tag{3.85}$$

with Reynolds numbers

$$\text{Re}_L = \frac{\rho_L |u_L| D_L}{\mu_L}, \quad \text{Re}_G = \frac{\rho_G |u_G| D_G}{\mu_G}, \tag{3.86}$$

hydraulic diameters

$$D_L = 2h_{\text{int}}, \quad D_G = 2(H - h_{\text{int}}). \tag{3.87}$$

These are evidently just the analytical expression found for single-phase flow in a channel geometry, regarding the gas-liquid interface as a wetted perimeter in the calculation of the hydraulic diameter. However (3.84) also includes the two-phase correction factors for the wall friction

$$F_L^* = \frac{1 + \frac{1}{2}\frac{u_G}{u_L}\left[\frac{\mu_L}{\mu_G}\frac{u_L}{u_G}\frac{H-h_{\text{int}}}{h_{\text{int}}} - 1\right]}{1 + \frac{\mu_L}{\mu_G}\frac{H-h_{\text{int}}}{h_{\text{int}}}}, \quad F_G^* = \frac{1 + \frac{1}{2}\frac{u_L}{u_G}\left[\frac{\mu_G}{\mu_L}\frac{u_G}{u_L}\frac{h_{\text{int}}}{H-h_{\text{int}}} - 1\right]}{1 + \frac{\mu_G}{\mu_L}\frac{h_{\text{int}}}{H-h_{\text{int}}}}, \tag{3.88}$$

and for the interfacial friction

$$F_{\text{int},G}^* = \frac{1}{1 + \frac{\mu_G}{\mu_L}\frac{h_{\text{int}}}{H-h_{\text{int}}}}. \tag{3.89}$$

The interfacial friction can equally be written as

$$\tau_{\text{int}} = -\frac{1}{2} f_{\text{L}} \rho_L \left(u_G - u_L\right) |u_L| F_{\text{int},L}, \tag{3.90}$$

with

$$F_{\text{int},L}^* = \frac{1}{1 + \frac{\mu_L}{\mu_G}\frac{H-h_{\text{int}}}{h_{\text{int}}}}. \tag{3.91}$$

Ullmann et al. remark that in the limit of $u_G \gg u_L$ and $\frac{\mu_L}{\mu_G}\frac{H-h_{\text{int}}}{h_{\text{int}}} \gg 1$, $F_G^* \to 1$ and $F_L^* \to 1/2$, thus reducing the effect of the correction factors simply to a doubling of the liquid hydraulic diameter (to $4h_{\text{int}}$). This is insightful, since the same liquid hydraulic diameter results if it is defined as in (3.74), i.e. if the gas-liquid interface is regarded as a free surface for the liquid. The correction factors also enable reversal of the shear stresses (the correction factors can be negative) in the case of back-flow or counter-current flow.

Ullmann et al. propose closure terms for circular pipe flow, which are not exact but which are based on the principles of the exact closure terms found for channel flow. They compare the predictions of a 1D two-fluid model incorporating these new closure terms to the exact 3D analytical solution and obtain good results. Ullmann and Brauner generalize these closure relations to turbulent flow in [124]. It is important to note that these closure terms are derived for the case of smooth, flat interfaces. Yu and Sparrow [125] conducted experiments that showed that laminar flow theory is still valid in the presence of very small interfacial waves, but in the presence of large amplitude waves the closure relations will probably break down, in view of the results given by the literature discussed above (e.g. [107]).

3.4.5 Conclusion

The stresses τ_L, τ_G, and τ_{int} appearing in the 1D two-fluid model are perimeter-averaged stresses, which represent the effect of gradients in the streamwise velocity normal to pipe walls and the interface. These gradients are not resolved in the 1D two-fluid model, and therefore closure terms are needed from external sources.

The friction closure of flow through a conduit is a difficult problem, for which a great deal of literature exists. Closure relations aim to relate the averaged flow variables (such as averaged velocity) and the friction force, via the velocity profile. Considering only steady state flow, a direct relation between the averaged velocity and the velocity profile can be established, but when including unsteady flow the relation becomes non-unique.

The single-phase closure relations depend on the pipe geometry, the level of turbulence and the wall roughness. For single-phase flow through a circular pipe the choice is easy enough, but for more complex geometries there is no definite answer on how to model the friction factor.

For two-phase flow the problem is even more difficult. The steady state force balances, of which we now have two, do not directly determine expressions for the stresses even if expressions for the two body forces as functions of holdup and the two velocities are established. Particularly the modeling of the interfacial stress in two-fluid models is a controversial and difficult problem [115]. It is usually based on the gas wall friction factor, but this approach loses accuracy for increasingly wavy flow. The liquid and especially the gas friction factor modelling are often based upon correlations for single-phase flow (e.g. the Blasius correlation (3.58)), although more complicated relations also exist.

Fortunately, for fully developed, smooth, flat interface, laminar flow in a channel geometry, analytical closure terms can be formulated [9]. The same is possible for a pipe flow geometry but expensive to calculate integrals remain in the expressions [9]. In this work, we will turn our attention to laminar flow in a channel geometry and use the exact Ullmann closure terms [9] as a reference.

3.5 Equations for a 2D domain

The governing equations (3.10), (3.11), (3.17), (3.18) describe 1D flow in a 3D geometry, with walls on all sides. If we consider the rectangular duct geometry of Figure 3.4, with width W and height H, we can substitute in these equations the level gradients given by (3.21) and (3.22):

$$\frac{\partial}{\partial t}\left(\rho_L A_L\right) + \frac{\partial}{\partial s}\left(\rho_L u_L A_L\right) = 0, \tag{3.92a}$$

$$\frac{\partial}{\partial t}\left(\rho_G A_G\right) + \frac{\partial}{\partial s}\left(\rho_G u_G A_G\right) = 0, \tag{3.92b}$$

$$\frac{\partial}{\partial t}\left(\rho_L u_L A_L\right) + \frac{\partial}{\partial s}\left(\rho_L u_L^2 A_L\right) = -\frac{\partial p_{\text{int}}}{\partial s} A_L - \frac{\partial}{\partial s}\left[\frac{1}{2}\rho_L g \cos\phi \frac{H}{A} A_L^2\right] + \tau_L P_L - \tau_{\text{int}} P_{\text{int}} - \rho_L A_L g \sin\phi, \tag{3.92c}$$

$$\frac{\partial}{\partial t}\left(\rho_G u_G A_G\right) + \frac{\partial}{\partial s}\left(\rho_G u_G^2 A_G\right) = -\frac{\partial p_{\text{int}}}{\partial s} A_G + \frac{\partial}{\partial s}\left[\frac{1}{2} g \cos\phi \frac{H}{A} \rho_G A_G^2\right] + \tau_G P_G + \tau_{\text{int}} P_{\text{int}} - \rho_G A_G g \sin\phi, \tag{3.92d}$$

We then make the substitutions

$$\begin{aligned} A_L \to W h_{\text{int}}, \quad A_G \to W(H - h_{\text{int}}), \quad A \to WH \\ P_L \to W + 2h_{\text{int}}, \quad P_G \to W + 2(H - h_{\text{int}}), \quad P_{\text{int}} = W, \end{aligned}$$

in which we use the fact that h_{int} is only a function of s and not of the depth coordinate y. If the width of the channel W is assumed constant, it can be divided out of the equations and the system can be

written as

$$\begin{aligned}
\frac{\partial}{\partial t}\left(\rho_L h_{\text{int}}\right) + \frac{\partial}{\partial s}\left(\rho_L u_L h_{\text{int}}\right) &= 0, \\
\frac{\partial}{\partial t}\left(\rho_G (H - h_{\text{int}})\right) + \frac{\partial}{\partial s}\left(\rho_G u_G (H - h_{\text{int}})\right) &= 0, \\
\frac{\partial}{\partial t}\left(\rho_L u_L h_{\text{int}}\right) + \frac{\partial}{\partial s}\left(\rho_L u_L^2 h_{\text{int}}\right) &= -\frac{\partial p_{\text{int}}}{\partial s} h_{\text{int}} - \frac{\partial}{\partial s}\left[\frac{1}{2}\rho_L g \cos\phi\, h_{\text{int}}^2\right] \\
&\quad + \tau_L \frac{W + 2h_{\text{int}}}{W} - \tau_{\text{int}} - \rho_L h_{\text{int}} g \sin\phi, \\
\frac{\partial}{\partial t}\left(\rho_G u_G (H - h_{\text{int}})\right) + \frac{\partial}{\partial s}\left(\rho_G u_G^2 (H - h_{\text{int}})\right) &= -\frac{\partial p_{\text{int}}}{\partial s}(H - h_{\text{int}}) + \frac{\partial}{\partial s}\left[\frac{1}{2}\rho_G g \cos\phi\, (H - h_{\text{int}})^2\right] \\
&\quad + \tau_G \frac{W + 2(H - h_{\text{int}})}{W} + \tau_{\text{int}} - \rho_G (H - h_{\text{int}}) g \sin\phi.
\end{aligned}$$

Now, considering W to be the length scale of variation along the depth coordinate y, in the limit $W/H \to \infty$, h_{int} will hardly vary along y, and will *naturally* only be a function of s. In this limit the fractions that appear in the above system tend to 1:

$$\frac{W + 2h_{\text{int}}}{W} \to 1, \quad \frac{W + 2(H - h_{\text{int}})}{W} \to 1. \tag{3.93}$$

Taking this limit entails the approximation of the 3D domain (with walls on all sides) as a 2D domain. The 2D domain has no walls (or walls infinitely far away such that they have no influence) with normals pointing along the depth coordinate y, but only top and bottom walls and possibly left and right walls (see Figure 3.5). The system for 2D channel flow can thus be written as

$$\frac{\partial}{\partial t}\left(\rho_L h_{\text{int}}\right) + \frac{\partial}{\partial s}\left(\rho_L u_L h_{\text{int}}\right) = 0, \tag{3.94a}$$

$$\frac{\partial}{\partial t}\left(\rho_G (H - h_{\text{int}})\right) + \frac{\partial}{\partial s}\left(\rho_G u_G (H - h_{\text{int}})\right) = 0, \tag{3.94b}$$

$$\begin{aligned}
\frac{\partial}{\partial t}\left(\rho_L u_L h_{\text{int}}\right) + \frac{\partial}{\partial s}\left(\rho_L u_L^2 h_{\text{int}}\right) &= -\frac{\partial p_{\text{int}}}{\partial s} h_{\text{int}} - \frac{\partial}{\partial s}\left[\frac{1}{2}\rho_L g \cos\phi\, h_{\text{int}}^2\right] \\
&\quad + \tau_L - \tau_{\text{int}} - \rho_L h_{\text{int}} g \sin\phi,
\end{aligned} \tag{3.94c}$$

$$\begin{aligned}
\frac{\partial}{\partial t}\left(\rho_G u_G (H - h_{\text{int}})\right) + \frac{\partial}{\partial s}\left(\rho_G u_G^2 (H - h_{\text{int}})\right) &= -\frac{\partial p_{\text{int}}}{\partial s}(H - h_{\text{int}}) + \frac{\partial}{\partial s}\left[\frac{1}{2}\rho_G g \cos\phi\, (H - h_{\text{int}})^2\right] \\
&\quad + \tau_G + \tau_{\text{int}} - \rho_G (H - h_{\text{int}}) g \sin\phi.
\end{aligned} \tag{3.94d}$$

By comparison with the system (3.92) for 3D channel flow, an important observation can be made. The 2D system of equations is the same as the system of equations for 3D channel flow, but with the substitutions

$$\begin{aligned}
A_L \to h_{\text{int}}, \quad A_G \to H - h_{\text{int}}, \quad A \to H \\
P_L \to 1, \quad P_G \to 1, \quad P_{\text{int}} \to 1.
\end{aligned} \tag{3.95}$$

We can thus use the same solver for 2D channel flow as for 3D channel flow, if only we

- interpret the solution for A_L as h_{int}, A_G as $H - h_{\text{int}}$,
- interpret the input for A (the specification of the domain) as the input for H, and
- set $P_L, P_G, P_{\text{int}} = 1$ for the friction calculation.

Figure 3.5 shows the 2D geometry graphically.

For our 2D geometry we only have wall stresses at the (s, y) planes, given by (3.39). Since we know (or require) that the liquid lies below the gas, we can directly write

$$\tau_L = -\mu_L \frac{\partial u_L}{\partial h} \tag{3.96}$$

$$\tau_G = \mu_G \frac{\partial u_G}{\partial h}, \tag{3.97}$$

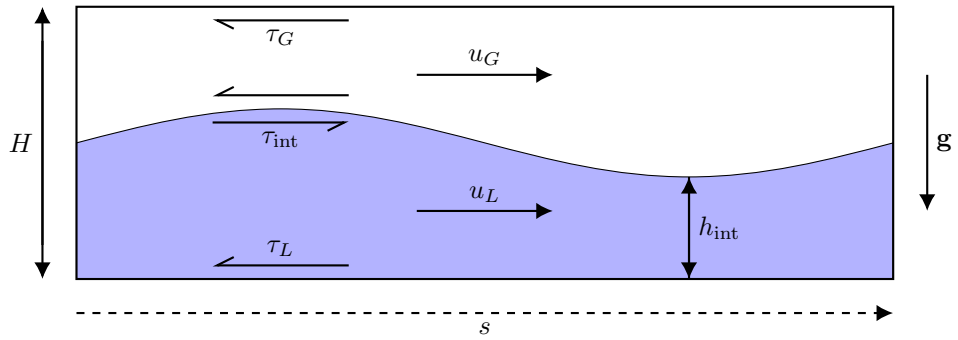


Figure 3.5: A schematic of the 1D two-fluid model for 2D channel flow.

where the perimeter averages (3.27) and (3.28) are trivial because the stress no longer varies along the perimeters. At the interface, like before in (3.42) but without a need for averaging, the stress is given by

$$\tau_{\text{int}} = \tau_{\text{int,G}} = -\tau_{\text{int,L}} = -\mu_G \frac{\partial u_G}{\partial h} = -\mu_L \frac{\partial u_L}{\partial h} \tag{3.98}$$

The two-fluid model system (3.94) may also be used to model free surface flow, in which the gas exerts no force on the liquid. For this case the equations are shown to reduce to the shallow water equations in Appendix A.

For convenience, the system (3.94) may be abbreviated as

$$\frac{\partial}{\partial t}\left(\rho_L h_{\text{int}}\right) + \frac{\partial}{\partial s}\left(\rho_L u_L h_{\text{int}}\right) = 0, \tag{3.99a}$$

$$\frac{\partial}{\partial t}\left(\rho_G (H - h_{\text{int}})\right) + \frac{\partial}{\partial s}\left(\rho_G u_G (H - h_{\text{int}})\right) = 0, \tag{3.99b}$$

$$\frac{\partial}{\partial t}\left(\rho_L u_L h_{\text{int}}\right) + \frac{\partial}{\partial s}\left(\rho_L u_L^2 h_{\text{int}}\right) = -\frac{\partial p_{\text{int}}}{\partial s} h_{\text{int}} + LG_L + F_L - \rho_L h_{\text{int}} g \sin\phi, \tag{3.99c}$$

$$\frac{\partial}{\partial t}\left(\rho_G u_G (H - h_{\text{int}})\right) + \frac{\partial}{\partial s}\left(\rho_G u_G^2 (H - h_{\text{int}})\right) = -\frac{\partial p_{\text{int}}}{\partial s}(H - h_{\text{int}}) + LG_G + F_G - \rho_G (H - h_{\text{int}}) g \sin\phi. \tag{3.99d}$$

Here the stresses are bundled into closure terms for this system

$$F_L = \tau_L - \tau_{\text{int}} \tag{3.100}$$

$$F_G = \tau_G + \tau_{\text{int}} \tag{3.101}$$

and the level gradient terms are not written in full:

$$LG_L = -\frac{\partial}{\partial s}\left[\frac{1}{2}\rho_L g \cos\phi\, h_{\text{int}}^2\right], \tag{3.102}$$

$$LG_G = \frac{\partial}{\partial s}\left[\frac{1}{2}\rho_G g \cos\phi\, \left(H - h_{\text{int}}\right)^2\right]. \tag{3.103}$$

3.6 Numerical two-fluid model

In this research the 'Rosa' code developed by Sanderse et al. [113], [126], [127] is employed for solving the incompressible form of (3.92), under the constraint $A_L + A_G = A$. With the substitutions (3.95) the code can be used for 2D channel flow.

The code discretizes the equations using a finite volume method on a staggered grid (recall Figure 2.4). This means that the mass conservation equations of (3.92) are integrated over a certain set of grid cells, and the momentum conservations are integrated over a set of grid cells which is displaced by half a grid cell relative to the previous set. This allows for a strong and straightforward coupling between the two. Still interpolation is needed for some variables: here we employ a central interpolation scheme, which ensures second order spatial accuracy.

After the system is discretized spatially, the time stepping is considered. We use the explicit three-stage, third order strong-stability preserving Runge-Kutta method referenced in [113], which follows Gottlieb et al. [128]. Explicit time integration methods are generally not feasible for the compressible two-fluid model due to the high velocities of the acoustic waves. For incompressible flow this is of no concern; the pressure field is calculated implicitly such that velocity field is divergence-free and the volume constraint $A_L + A_G = A$ is satisfied, following the same principles presented in subsection 2.4.2. Details on the constraint-consistent time integration scheme are available in [126].

3.7 Conclusion

The 1D two-fluid model is a model for stratified two-phase flow which can be derived by averaging the equations over the cross-section of an arbitrarily shaped duct. The same equations hold for 2D channel flow, with appropriate substitutions. In the derivation, we assume hydrostatic balance, so that the vertical velocity is negligible. The streamwise normal stress is neglected with the argument that we consider a horizontal length scale far larger than the vertical length scale.

The shear stresses in pipe flow depend on the streamwise velocity and its gradient normal to the pipe walls, and the interface. In the 1D two-fluid model the velocity field is not resolved; only the averaged streamwise velocities are known. Therefore closure terms, based on experiments or analysis, are needed to relate the averaged velocities (and material properties and pipe geometry) to the stresses. In general there is no one-to-one relation for this. However, narrowing the scope down to fully developed steady state flow there is, and this is what is used in conventional practice.

The closure terms found in literature for stratified two-phase flow have more problems. There is no consensus on closure terms for wide ranges of pipe geometries, interface waviness, and turbulence regimes. However, for the case of laminar, fully developed, flat interface steady state flow in a 2D channel the stresses can be calculated via an analytical algebraic expression. The derivation of these closure terms is referred to as step [1] in the flow chart Figure 1.4.

Chapter 4

Stability Analysis

4.1 Introduction

Our goal in this project is to introduce closure terms in the 1D two-fluid model which let its predictions match those of a 2D model. That is, we aim for step [8] in our project flow chart Figure 1.4 to show the best possible agreement between our high- and low-fidelity models, with the low-fidelity model still being more computationally efficient than our high-fidelity model. In this chapter, we analyze some fundamental differences between the two models, which define limits to our approach.

In this chapter, we perform linear stability analysis on both the 1D two-fluid model and on 2D potential flow. We analyze the inviscid dispersion relations, which predict how a small disturbance of the interface (and associated disturbances in the velocities) will evolve in time. Comparing these dispersion relations for the two models yields insight in the fundamental differences between them. These are differences that closure terms cannot overcome.

We find that the 1D and 2D models agree theoretically when the horizontal length scale (in the form of the wavelength of a perturbation) is much larger than the vertical length scale. Additionally, we show that 1D and 2D simulations agree with the theoretical dispersion relations for short periods of time.

In the first part of this chapter, we will furthermore discuss an important property of the 1D two-fluid model: its tendency to become ill-posed for certain initial conditions. This is another fundamental limitation of the two-fluid model, for when the model becomes ill-posed its predictions carry little meaning. Thus for certain regions of the parameter space the 1D two-fluid model cannot be used. We need to map this region so that it can be avoided in later chapters.

4.2 Well-posedness

The two-fluid model, comprised of the equations (3.94), is our main object of study. There are some mathematical properties we need to know about this model before we blindly discretize it and run a simulation. For one, we need to know if the model is well-posed in the sense of Hadamard. In this sense a problem can be said to be well-posed if

1. a solution exists,
2. the solution is unique,
3. the solution depends in a continuous manner on the initial and boundary conditions.

According to [129], these conditions imply

> 1. that one should not have too many (conflicting) boundary conditions, 2. that we should not have too few, and 3. that the effect of small perturbations is also small.

Thus if our boundary conditions are appropriate, only the third condition is of concern.

The third condition does *not* state that a small perturbation of the initial conditions cannot grow at all. It is sufficient if a small perturbation of the initial and boundary conditions does not grow in an unbounded manner. The vague phrase 'an unbounded manner' can be understood as 'with a growth rate tending to infinity'.

We may test the equations by substituting a wave-like trial solution and seeing if, and how much, it grows in time. A system of quasi-linear partial differential equations of which the homogeneous part propagates a wave-like solution without growth or damping can be said to be *hyperbolic* [130, p. 112]. Thus, for a time-dependent problem, hyperbolicity of the equations is a sufficient (but not necessary) condition for well-posedness, while an elliptic system is ill-posed [131]. Here, 'the homogeneous part' indicates the part of the equation containing derivatives of the unknown variables.

Hyperbolic equations require (for well-posedness) initial conditions for each unknown, and boundary conditions along each boundary, corresponding to the number of characteristics which propagate from that boundary. The solution may evolve in time without a final boundary. Therefore if we can prove our equations to be hyperbolic, we can set the appropriate number of boundary conditions, and give an initial solution and watch its evolution in time.

We consider the system of first order PDEs

$$\mathbf{A}\frac{\partial}{\partial t}\mathbf{u} + \mathbf{B}\frac{\partial}{\partial s}\mathbf{u} = 0. \tag{4.1}$$

We substitute the trial wave-like solution

$$\mathbf{u} = \hat{\mathbf{u}} \exp\left[i\left(ks - \omega t\right)\right], \tag{4.2}$$

with k the wavenumber and ω the angular frequency, which dictates the change of the solution with time (and is thus of interest in determining whether the system is hyperbolic). This yields

$$\left[-i\omega\mathbf{A} + ik\mathbf{B}\right]\hat{\mathbf{u}} = \mathbf{0}, \tag{4.3}$$

which has non-trivial solutions for $\hat{\mathbf{u}}$ (that is, solutions other than $\hat{\mathbf{u}} = \mathbf{0}$) if the determinant of $-\omega\mathbf{A} + k\mathbf{B}$ is zero. This condition can be written as the generalized eigenvalue problem

$$\det\left|\mathbf{B} - c\mathbf{A}\right| = 0, \tag{4.4}$$

with $c = \omega/k$. In Appendix B we demonstrate the solution of this problem with an example, and determine the characteristics of the system.

The problem of determining if the system is hyperbolic reduces to finding the eigenvalues which satisfy

$$\det\left|\mathbf{B} - c\mathbf{A}\right| = 0. \tag{4.5}$$

A necessary and sufficient general condition for hyperbolicity of the first order system (4.1) can be posed as: all eigenvalues c of the system (found via (4.5)) are real and there is a complete system of linearly independent eigenvectors [42]. Note that the source term contains no derivatives and has no influence on the classification. If none of the eigenvalues are real, the system is elliptic. If all of the eigenvalues are real but there is no complete system of linearly independent eigenvectors, the system is parabolic.

The classification is based solely on the homogeneous part of the system; the inhomogeneous part has no influence on the characteristic analysis [131]. Therefore the classification will also apply to systems of the form

$$\mathbf{A}\frac{\partial}{\partial t}\mathbf{u} + \mathbf{B}\frac{\partial}{\partial s}\mathbf{u} = \mathbf{c}(\mathbf{u}) \tag{4.6}$$

(in which $\mathbf{c}(\mathbf{u})$ has nothing to do with the eigenvalue c). Since the source term has no influence on the determination of the hyperbolicity of the system, it also has no influence on the well-posedness [131]. However, the source terms will influence the stability of the model.

4.3 Analysis of the two-fluid model

We recall the governing equations of the two-fluid model for 2D channel flow (3.99). Unfortunately, the system *cannot* be written in the conservative form

$$\frac{\partial \mathbf{u}}{\partial t} + \boldsymbol{\nabla}\cdot\mathbf{f}(\mathbf{u}) = \mathbf{c}(\mathbf{u}), \tag{4.7}$$

due in part to the non-conservative streamwise pressure gradient term. However, the system *can* be written in the quasi-linear form

$$\mathbf{A}(\mathbf{w})\frac{\partial}{\partial t}\mathbf{w} + \mathbf{B}(\mathbf{w})\frac{\partial}{\partial s}\mathbf{w} = \mathbf{c}(\mathbf{w}). \tag{4.8}$$

Assuming incompressible flow, the constituents of (4.8) are given by

$$\mathbf{w} = \begin{bmatrix} h_{\text{int}} \\ u_L \\ u_G \\ p \end{bmatrix}, \tag{4.9}$$

$$\mathbf{A} = \begin{bmatrix} \rho_L & 0 & 0 & 0 \\ -\rho_G & 0 & 0 & 0 \\ \rho_L u_L & \rho_L h_{\text{int}} & 0 & 0 \\ -\rho_G u_G & 0 & \rho_G(H - h_{\text{int}}) & 0 \end{bmatrix}, \tag{4.10}$$

$$\mathbf{B} = \begin{bmatrix} \rho_L u_L & \rho_L h_{\text{int}} & 0 & 0 \\ -\rho_G u_G & 0 & \rho_G(H - h_{\text{int}}) & 0 \\ \rho_L u_L^2 + \rho_L h_{\text{int}} g \cos\phi & 2\rho_L u_L h_{\text{int}} & 0 & h_{\text{int}} \\ -\rho_G u_G^2 + \rho_G(H - h_{\text{int}}) g \cos\phi & 0 & 2\rho_G u_G (H - h_{\text{int}}) & (H - h_{\text{int}}) \end{bmatrix}, \tag{4.11}$$

and

$$\mathbf{c} = \begin{bmatrix} 0 \\ 0 \\ -F_L - \rho_L h_{\text{int}} g \sin\phi \\ -F_G - \rho_G (H - h_{\text{int}}) g \sin\phi \end{bmatrix}. \tag{4.12}$$

4.3.1 Well-posedness of the two-fluid model

We need to determine if this system is well-posed in the sense of Hadamard. To this end we classify the system as discussed in section 4.2. Solving

$$\det |\mathbf{B} - c\mathbf{A}| = 0 \tag{4.13}$$

with $c = \omega/k$ yields four eigenvalues leading to four possible dispersion relations. Two of them are given by

$$\omega = k \frac{\rho_L u_L + \rho_G u_G \xi_1 \pm \sqrt{(\rho_L + \rho_G \xi_1)(\rho_L - \rho_G) h_{\text{int}} g \cos\phi - \rho_L \rho_G \xi_1 (u_G - u_L)^2}}{\rho_L + \rho_G \xi_1}, \tag{4.14}$$

with

$$\xi_1 = \frac{h_{\text{int}}}{H - h_{\text{int}}}.$$

The same derivation can be done for the 1D two-fluid model with a 3D (e.g. pipe or rectangular duct) geometry; the result is

$$\omega = k \frac{\rho_L u_L A_G A + \rho_G u_G A_L A \pm \sqrt{\left[(\rho_L A_G + \rho_G A_L)(\rho_L - \rho_G) H g \cos\phi - \rho_L \rho_G (u_G - u_L)^2 A\right] A_L A_G A}}{\rho_L A_G A + \rho_G A_L A}, \tag{4.15}$$

and making the same substitutions as in section 3.5 one retrieves (4.14). An alternative form in terms of the holdups $\alpha_L = h_{\text{int}}/A$, $\alpha_G = (H - h_{\text{int}})/A$ is given by [75] and [112]

$$\omega = k \frac{\frac{\rho_L u_L}{\alpha_L} + \frac{\rho_G u_G}{\alpha_G} \pm \sqrt{\left(\frac{\rho_L}{\alpha_L} + \frac{\rho_G}{\alpha_G}\right)(\rho_L - \rho_G) H g \cos\phi - \frac{\rho_L \rho_G}{\alpha_L \alpha_G} (u_G - u_L)^2}}{\frac{\rho_L}{\alpha_L} + \frac{\rho_G}{\alpha_G}}. \tag{4.16}$$

From (4.14) one can see that ω, and thus the eigenvalue, has an imaginary component if

$$\Delta U^2 = (u_G - u_L)^2 > \left(\frac{h_{\text{int}}}{\rho_L} + \frac{H - h_{\text{int}}}{\rho_G}\right)(\rho_L - \rho_G) g \cos\phi, \tag{4.17}$$

in which case one of the two possible wave-like solutions of the homogeneous system will grow exponentially in time and the system cannot be said to be hyperbolic. The system can also directly said to be ill-posed

in this case since the wavenumber k will tend to infinity for small wavelengths, leading to unbounded growth.

Two of the four eigenvalues are neglected in this analysis. They are related to the speed of sound of the flow, which is infinite in the incompressible limit, leading to elliptic behavior of the pressure. These eigenvalues do not need to be considered explicitly here, if incompressibility is treated as a constraint, instantaneously satisfied at each moment in time. Sanderse and Veldman [126] provide a thorough discussion.

4.3.2 Linear stability analysis

The linear stability analysis is similar to the classification of quasi-linear PDE's based on substitution of a wave-like solution (see section 4.2). The difference is that we now take into account the inhomogeneous part of the equations, and that we perform a linearization. This is as done by Liao et al. [112], Fullmer et al. [132], and Sanderse [75].

The linearization can be done by substituting

$$\mathbf{w} = \mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w}, \tag{4.18}$$

with $\boldsymbol{\Delta}\mathbf{w}$ a small disturbance ($\boldsymbol{\Delta}\mathbf{w} \ll \mathbf{w}^0$), into the system of governing equations (4.8) to yield

$$\mathbf{A}(\mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w})\frac{\partial}{\partial t}\left(\mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w}\right) + \mathbf{B}(\mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w})\frac{\partial}{\partial s}\left(\mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w}\right) = \mathbf{c}(\mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w}), \tag{4.19}$$

where $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{c}$ are functions of $\mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w}$. This means that they are just as they were defined earlier but with the four variables h_{int}, u_L, u_G, and p replaced by $h^0_{\text{int}} + \Delta h_{\text{int}}$, $u^0_L + \Delta u_L$, $u^0_G + \Delta u_G$, and $p^0 + \Delta p$ respectively.

We consider a reference state $\mathbf{w}^0$ that is in steady state, so that $\partial \mathbf{w}^0/\partial t = 0$. The reference state is also fully developed, so that the spatial derivatives of the liquid and gas holdups and of the velocities are zero (but not that of the pressure). These terms can thus be dropped from the equations. The remaining system for the reference state is

$$\mathbf{B}(\mathbf{w}^0)\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{\partial p^0}{\partial s} \end{bmatrix} = \mathbf{c}(\mathbf{w}^0). \tag{4.20}$$

The pressure gradient is constant and is necessary to drive the viscous fluids through the pipe/tank. We define the left hand side of the above equation as $\mathbf{d}(\mathbf{w}^0)$. It is given by

$$\mathbf{d}(\mathbf{w}^0) = \mathbf{B}(\mathbf{w}^0)\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{\partial p^0}{\partial s} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ h^0_{\text{int}}\frac{\partial p^0}{\partial s} \\ (H - h^0_{\text{int}})\frac{\partial p^0}{\partial s} \end{bmatrix}. \tag{4.21}$$

Similarly, we define

$$\mathbf{d}(\mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w}) = \mathbf{B}(\mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w})\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{\partial p^0}{\partial s} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ (h^0_{\text{int}} + \Delta h_{\text{int}})\frac{\partial p^0}{\partial s} \\ (H - h^0_{\text{int}} - \Delta h_{\text{int}})\frac{\partial p^0}{\partial s} \end{bmatrix}. \tag{4.22}$$

We apply the assumptions about the reference state in (4.19) ($\partial \mathbf{w}^0/\partial t = 0$, $\partial h_{\text{int}}/\partial s = 0$, $\partial u_L/\partial s = 0$, $\partial u_G/\partial s = 0$) and subtract the steady state balance (4.20). This can be written as

$$\mathbf{A}(\mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w})\frac{\partial}{\partial t}\boldsymbol{\Delta}\mathbf{w} + \mathbf{B}(\mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w})\frac{\partial}{\partial s}\boldsymbol{\Delta}\mathbf{w} + \mathbf{d}(\mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w}) - \mathbf{d}(\mathbf{w}^0) = \mathbf{c}(\mathbf{w}^0 + \boldsymbol{\Delta}\mathbf{w}) - \mathbf{c}(\mathbf{w}^0). \tag{4.23}$$

Then terms with products of disturbances are neglected (e.g. $\Delta u_L \frac{\partial \Delta h_{\text{int}}}{\partial s}$ or $g\Delta\alpha_L\Delta\rho_L \sin\phi$). The resulting system of linearized equations is

$$\mathbf{A}(\mathbf{w}^0)\frac{\partial}{\partial t}\boldsymbol{\Delta}\mathbf{w} + \mathbf{B}(\mathbf{w}^0)\frac{\partial}{\partial s}\boldsymbol{\Delta}\mathbf{w} = \tilde{\mathbf{c}}, \tag{4.24}$$

with

$$\mathbf{\Delta w} = \begin{bmatrix} \Delta h_{\text{int}} \\ \Delta u_L \\ \Delta u_G \\ \Delta p \end{bmatrix}, \tag{4.25}$$

and

$$\tilde{\mathbf{c}} = \mathbf{c}(\mathbf{w}^0 + \mathbf{\Delta w}) - \mathbf{c}(\mathbf{w}^0) - \mathbf{d}(\mathbf{w}^0 + \mathbf{\Delta w}) + \mathbf{d}(\mathbf{w}^0) = \begin{bmatrix} 0 \\ 0 \\ -\Delta h_{\text{int}} \frac{\partial p^0}{\partial s} - \Delta F_L - \rho_L \Delta h_{\text{int}} g \sin\phi \\ \Delta h_{\text{int}} \frac{\partial p^0}{\partial s} - \Delta F_G + \rho_G \Delta h_{\text{int}} g \sin\phi \end{bmatrix}. \tag{4.26}$$

The system (4.24) is not yet a linear system since $\tilde{\mathbf{c}}$ is still a function of $\mathbf{\Delta w}$.

We can make first order approximations for F_L and F_G in terms of the perturbations $\mathbf{\Delta w}$;

$$F_L \approx F_L^0 + \Delta h_{\text{int}} \frac{\partial F_L}{\partial h_{\text{int}}} + \Delta u_L \frac{\partial F_L}{\partial u_L} + \Delta u_G \frac{\partial F_L}{\partial u_G} + \Delta p \frac{\partial F_L}{\partial p} \tag{4.27}$$

$$F_G \approx F_G^0 + \Delta h_{\text{int}} \frac{\partial F_G}{\partial h_{\text{int}}} + \Delta u_L \frac{\partial F_G}{\partial u_L} + \Delta u_G \frac{\partial F_G}{\partial u_G} + \Delta p \frac{\partial F_G}{\partial p}, \tag{4.28}$$

with the derivatives taken at the reference state $\mathbf{w}^0$. Then $\Delta F_L = F_L - F_L^0$ and $\Delta F_G = F_G - F_G^0$, and we can rewrite $\tilde{\mathbf{c}}$:

$$\tilde{\mathbf{c}} = \mathbf{C}(\mathbf{w}^0)\mathbf{\Delta w}, \tag{4.29}$$

with $\mathbf{C}(\mathbf{w}^0)$ the sum of two Jacobians

$$\mathbf{C}(\mathbf{w}^0) = \frac{\partial \mathbf{c}(\mathbf{w}^0)}{\partial \mathbf{w}} - \frac{\partial \mathbf{d}(\mathbf{w}^0)}{\partial \mathbf{w}} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{\partial p^0}{\partial s} - \rho_L g \sin\phi - \frac{\partial F_L}{\partial h_{\text{int}}} & -\frac{\partial F_L}{\partial u_L} & -\frac{\partial F_L}{\partial u_G} & -\frac{\partial F_L}{\partial p} \\ \frac{\partial p^0}{\partial s} + \rho_G g \sin\phi - \frac{\partial F_G}{\partial h_{\text{int}}} & -\frac{\partial F_G}{\partial u_L} & -\frac{\partial F_G}{\partial u_G} & -\frac{\partial F_G}{\partial p} \end{bmatrix}, \tag{4.30}$$

where it is required that $\partial p^0/\partial s$ is constant. Now (4.24) can be approximated as

$$\mathbf{A}(\mathbf{w}^0)\frac{\partial}{\partial t}\mathbf{\Delta w} + \mathbf{B}(\mathbf{w}^0)\frac{\partial}{\partial s}\mathbf{\Delta w} = \mathbf{C}(\mathbf{w}^0)\mathbf{\Delta w}, \tag{4.31}$$

which is a linear system since now none of the coefficients are functions of $\mathbf{\Delta w}$ (they are only functions of the constant reference state).

With the linearized system (4.31) we can perform a stability analysis by substituting for the disturbance the wave form

$$\mathbf{\Delta w} = \mathbf{\Delta \hat{w}} \exp\left[i\left(ks - \omega t\right)\right], \tag{4.32}$$

with $\mathbf{\Delta \hat{w}}$ the amplitude. In fact, through a Fourier transformation, any disturbance can be written as a sum of functions in the above form (with different amplitudes, frequencies and wavenumbers). For each of the wave functions (4.31) will hold separately, since the equation is linear. Therefore an analysis based on the substitution of the given wave-like function can be used to make conclusions for an arbitrarily shaped continuous function.

Proceeding with the substitution, we get

$$\left[-i\omega\mathbf{A} + ik\mathbf{B} - \mathbf{C}\right]\mathbf{\Delta \hat{w}} = \mathbf{0}, \tag{4.33}$$

which has nontrivial solutions if

$$\det\left|k\mathbf{B} - \omega\mathbf{A} + i\mathbf{C}\right| = 0. \tag{4.34}$$

If both fluids are inviscid and gravity has no component in the streamwise direction ($\phi = 0$), then via (4.20) $\partial p/\partial s$ is also zero and $\mathbf{C} = \mathbf{0}$. In this special case the same results are found as in the analysis of the well-posedness of the system; the dispersion relation is (4.14) and the condition for instability is (4.17).

If $\mathbf{C}$ is nonzero, extra imaginary terms are added to the dispersion relation. Writing the dispersion relation out in full, these terms are seen to contain the factor k, and thus lead to infinite growth for

small wavelengths. This does not contradict the assertion that the inhomogeneous term has no effect on the well-posedness of the system, since here we are not considering the original system. We are instead considering the linearized system.

The allowed relative amplitudes $\boldsymbol{\Delta}\hat{\mathbf{w}}$ are the eigenvectors of $k\mathbf{B} - \omega\mathbf{A} + i\mathbf{C}$ corresponding to the two different dispersion relations found via (4.34). For the inviscid case ($\mathbf{C} = \mathbf{0}$) the expressions for the velocity perturbations as a function of the freely chosen $\Delta\hat{h}_{\text{int}}$ are relatively simple:

$$\begin{bmatrix} \Delta\hat{h}_{\text{int}} \\ \Delta\hat{u}_L \\ \Delta\hat{u}_G \\ \Delta\hat{p} \end{bmatrix} = \begin{bmatrix} \Delta\hat{h}_{\text{int}} \\ -\frac{ku_L-\omega}{kh_{\text{int}}}\Delta\hat{h}_{\text{int}} \\ \frac{ku_G-\omega}{k(H-h_{\text{int}})}\Delta\hat{h}_{\text{int}} \\ \cdots \end{bmatrix} \tag{4.35}$$

The pressure perturbation can be expressed using either the third or fourth row of (4.33):

$$\begin{aligned} \Delta\hat{p} &= \rho_L \frac{k^2u_L^2 - k^2h_{\text{int}}g\cos\phi - 2k\omega u_L + \omega^2}{k^2h_{\text{int}}}\Delta\hat{h}_{\text{int}}, \\ &= -\rho_G \frac{k^2u_G^2 + k^2(H-h_{\text{int}})g\cos\phi - 2k\omega u_G + \omega^2}{k^2(H-h_{\text{int}})}\Delta\hat{h}_{\text{int}}. \end{aligned} \tag{4.36}$$

The evolution in time of a disturbance with wavenumber k can then be written as

$$\boldsymbol{\Delta}\mathbf{w} = \text{Re}\left\{\boldsymbol{\Delta}\hat{\mathbf{w}}_1 \exp\left[i\left(ks - \omega_1 t\right)\right] + \boldsymbol{\Delta}\hat{\mathbf{w}}_2 \exp\left[i\left(ks - \omega_2 t\right)\right]\right\}, \tag{4.37}$$

in which $\boldsymbol{\Delta}\hat{\mathbf{w}}_1$ is the amplitude vector with relative amplitudes determined by the eigenvector corresponding to ω_1, which is the angular frequency determined by k and the first dispersion relation. Similarly, ω_2 is determined by k and the second dispersion relation and $\boldsymbol{\Delta}\hat{\mathbf{w}}_2$ is its eigenvector. This disturbance may alternatively be written as

$$\boldsymbol{\Delta}\mathbf{w} = \left|\boldsymbol{\Delta}\hat{\mathbf{w}}_1\right| \mathrm{e}^{\text{Im}\{\omega_1\}t} \cos\left(ks - \text{Re}\left\{\omega_1\right\}t + \boldsymbol{\theta}_1\right) + \left|\boldsymbol{\Delta}\hat{\mathbf{w}}_2\right| \mathrm{e}^{\text{Im}\{\omega_2\}t} \cos\left(ks - \text{Re}\left\{\omega_2\right\}t + \boldsymbol{\theta}_2\right). \tag{4.38}$$

with the angles

$$\boldsymbol{\theta}_1 = \begin{bmatrix} \arctan\left(\text{Im}\left\{\Delta\hat{h}_{\text{int},1}\right\}/\text{Re}\left\{\Delta\hat{h}_{\text{int},1}\right\}\right) \\ \arctan\left(\text{Im}\left\{\Delta\hat{u}_{L,1}\right\}/\text{Re}\left\{\Delta\hat{u}_{L,1}\right\}\right) \\ \arctan\left(\text{Im}\left\{\Delta\hat{u}_{G,1}\right\}/\text{Re}\left\{\Delta\hat{u}_{G,1}\right\}\right) \\ \arctan\left(\text{Im}\left\{\Delta\hat{p}_1\right\}/\text{Re}\left\{\Delta\hat{p}_1\right\}\right) \end{bmatrix}, \qquad \boldsymbol{\theta}_2 = \begin{bmatrix} \arctan\left(\text{Im}\left\{\Delta\hat{h}_{\text{int},2}\right\}/\text{Re}\left\{\Delta\hat{h}_{\text{int},2}\right\}\right) \\ \arctan\left(\text{Im}\left\{\Delta\hat{u}_{L,2}\right\}/\text{Re}\left\{\Delta\hat{u}_{L,2}\right\}\right) \\ \arctan\left(\text{Im}\left\{\Delta\hat{u}_{G,2}\right\}/\text{Re}\left\{\Delta\hat{u}_{G,2}\right\}\right) \\ \arctan\left(\text{Im}\left\{\Delta\hat{p}_2\right\}/\text{Re}\left\{\Delta\hat{p}_2\right\}\right) \end{bmatrix}. \tag{4.39}$$

The arctan function maps all angles to the range $[-\pi/2, \pi/2]$. If the real part of the complex amplitude is negative the actual angle should fall outside that range. In this case a constant π must be added to the result of the arctan to obtain the correct angle. The complex amplitude absolute magnitudes are given by

$$\left|\boldsymbol{\Delta}\hat{\mathbf{w}}_1\right| = \begin{bmatrix} \sqrt{\text{Re}\left\{\Delta\hat{h}_{\text{int},1}\right\}^2 + \text{Im}\left\{\Delta\hat{h}_{\text{int},1}\right\}^2} \\ \sqrt{\text{Re}\left\{\Delta\hat{u}_{L,1}\right\}^2 + \text{Im}\left\{\Delta\hat{u}_{L,1}\right\}^2} \\ \sqrt{\text{Re}\left\{\Delta\hat{u}_{G,1}\right\}^2 + \text{Im}\left\{\Delta\hat{u}_{G,1}\right\}^2} \\ \sqrt{\text{Re}\left\{\Delta\hat{p}_1\right\}^2 + \text{Im}\left\{\Delta\hat{p}_1\right\}^2} \end{bmatrix}, \quad \text{and} \quad \left|\boldsymbol{\Delta}\hat{\mathbf{w}}_2\right| = \begin{bmatrix} \sqrt{\text{Re}\left\{\Delta\hat{h}_{\text{int},2}\right\}^2 + \text{Im}\left\{\Delta\hat{h}_{\text{int},2}\right\}^2} \\ \sqrt{\text{Re}\left\{\Delta\hat{u}_{L,2}\right\}^2 + \text{Im}\left\{\Delta\hat{u}_{L,2}\right\}^2} \\ \sqrt{\text{Re}\left\{\Delta\hat{u}_{G,2}\right\}^2 + \text{Im}\left\{\Delta\hat{u}_{G,2}\right\}^2} \\ \sqrt{\text{Re}\left\{\Delta\hat{p}_2\right\}^2 + \text{Im}\left\{\Delta\hat{p}_2\right\}^2} \end{bmatrix}. \tag{4.40}$$

The initial disturbance at $t = 0$ is given by

$$\boldsymbol{\Delta}\mathbf{w} = \text{Re}\left\{\boldsymbol{\Delta}\hat{\mathbf{w}}_1 \exp\left[iks\right] + \boldsymbol{\Delta}\hat{\mathbf{w}}_2 \exp\left[iks\right]\right\}, \tag{4.41}$$

or equivalently

$$\boldsymbol{\Delta}\mathbf{w} = \left|\boldsymbol{\Delta}\hat{\mathbf{w}}_1\right| \cos\left(ks + \boldsymbol{\theta}_1\right) + \left|\boldsymbol{\Delta}\hat{\mathbf{w}}_2\right| \cos\left(ks + \boldsymbol{\theta}_2\right). \tag{4.42}$$

Any initial disturbance can now be written as a sum of initial disturbances of this form, with different wavenumbers, and as stated before, the analysis will still hold.

4.3.3 Discrete stability analysis

Fullmer et al. [132] show that, for a specific finite discretization, the *numerical* two-fluid model is unconditionally well-posed, in contrast to the analytical model (where 'well-posed' is taken strictly to mean the absence of unbounded growth). This is evidenced by the fact that there is a cutoff wavelength below which no growth takes place, whereas for the analytical model the growth rate tends to infinity for wavelengths tending to zero. The shortest waves are damped most strongly by numerical diffusion.

However, discretizations of the two-fluid model can still be numerically unstable, even if they are well-posed. In the ill-posed regime of the continuous two-fluid model, any consistent discretization will have large positive growth rates for waves slightly longer than a certain cut-off wavelength, determined by the grid resolution [132]. Even though simulations can be run in the ill-posed regime (since the numerical model is unconditionally well-posed due to numerical diffusion) numerical solutions of the two-fluid model in this regime are generally regarded to have no physical meaning [112], and will not converge upon grid refinement.

Instead, some authors interpret a migration of the numerical solution towards the ill-posed regime as triggering the transition to slug flow [111], [133], related to the fact that the inherent assumptions in the two-fluid model break down. Slug flow is characterized by the occurrence of pockets of liquid alternated by pockets of gas, such that the interface height is locally equal to the pipe diameter. In order to correctly predict the transition to slug flow (which is important for practical applications) it is desirable that the numerical two-fluid model becomes unstable close to the stability limit of the analytical two-fluid model. This requires a sufficiently fine grid and time step and appropriate discretization method.

4.4 2D Linear stability analysis

Our high-fidelity model Gerris solves the 2D incompressible Navier-Stokes equations (2.44) and (2.45). This can be seen as a 2D extension of the two-fluid model, not assuming hydrostatic balance or $L \gg H$. The Euler equations are the inviscid form of the Navier-Stokes equations, in which the viscous terms are neglected. We will show the stability analysis of the 2D equations, so that we can see some of what the 1D two-fluid equations are missing.

4.4.1 Unbounded domain

The classic Kelvin-Helmholtz instability for incompressible horizontal shear flow of two fluids is found by considering inviscid flow, which is described by the incompressible Euler equations. If the flow is initially irrotational, the flow will remain irrotational by virtue of Kelvin's circulation theorem. This means that the flow can be described as a potential flow with a potential ϕ, $\mathbf{u} = \boldsymbol{\nabla}\phi$, and $\nabla^2\phi = 0$. For irrotational flow one may derive the Bernoulli equation from the Euler equations.

The boundary conditions are

- At all times the flow infinitely far away from the interface remains in the initial state.
- Kinematic: the fluid parcels must move with the interface.
- Dynamic: the pressure is continuous over the interface.

Furthermore the resulting system of equations is linearized by assuming that the velocity perturbations on the initial flow remain small, along with the perturbation of the interface. The resulting system of equations (adapted from [134], [135]), which holds at $z = 0$, is

$$\frac{\partial \phi'_G}{\partial z} - \frac{\partial \eta}{\partial t} - u_G \frac{\partial \eta}{\partial s} = 0, \tag{4.43a}$$

$$\frac{\partial \phi'_L}{\partial z} - \frac{\partial \eta}{\partial t} - u_L \frac{\partial \eta}{\partial s} = 0, \tag{4.43b}$$

$$\rho_G \left(u_G \frac{\partial \phi'_G}{\partial s} + \frac{\partial \phi'_G}{\partial t} + g\eta \right) = \rho_L \left(u_L \frac{\partial \phi'_L}{\partial s} + \frac{\partial \phi'_L}{\partial t} + g\eta \right). \tag{4.43c}$$

In this system η is the perturbation of the interface height, z is the vertical coordinate defined relative to the interface, t is the time coordinate, and s is the horizontal coordinate which is parallel to the interface and the basic velocity u. ϕ'_G is the perturbation of the velocity potential in the upper part of the domain,

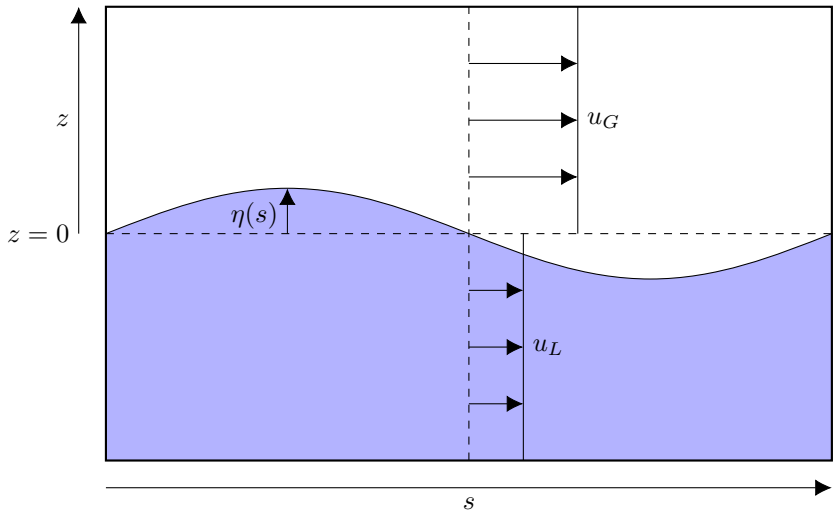


Figure 4.1: A schematic of a 2D flow with a perturbation applied to the interface.

u_G is the unperturbed velocity here, and ϕ'_L and u_L are the same for the lower fluid. Figure 4.1 shows the flow schematically.

The interface height η and the perturbations of the velocity potentials in the upper ϕ_G and lower ϕ_L fluids are then written as

$$\eta = \hat{\eta} \mathrm{e}^{i(ks-\omega t)}, \tag{4.44a}$$

$$\phi'_G = B_G \mathrm{e}^{-kz} \mathrm{e}^{i(ks-\omega t)}, \tag{4.44b}$$

$$\phi'_L = B_L \mathrm{e}^{kz} \mathrm{e}^{i(ks-\omega t)}, \tag{4.44c}$$

where $\hat{\eta}$, B_G, and B_L are constants which determine the amplitudes of the disturbances, and k and ω are the wavenumber and frequency of these disturbances. The wavenumber can take any value, and a sum of these functions with different wavenumbers can create an arbitrary wave form. Substituting (4.44) in (4.43) (along with $z = 0$) yields the linear system

$$\mathbf{A}\mathbf{u} = \mathbf{0}, \tag{4.45a}$$

$$\mathbf{A} = \begin{bmatrix} -i\left(u_G k - \omega\right) & -k & 0 \\ -i\left(u_L k - \omega\right) & 0 & k \\ \left(\rho_L - \rho_G\right) g & -i\rho_G (u_G k - \omega) & i\rho_L (u_L k - \omega) \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} \hat{\eta} \\ B_G \\ B_L \end{bmatrix}. \tag{4.45b}$$

This system will have multiple solutions other than $\mathbf{u} = \mathbf{0}$ if and only if the determinant is zero. This condition,

$$-\left(\rho_L - \rho_G\right) g k^2 - i^2 \rho_G (u_G k - \omega)^2 k - i^2 \rho_L (u_L k - \omega)^2 k = 0, \tag{4.46}$$

produces the dispersion relation [134]:

$$\omega = k \frac{\rho_L u_L + \rho_G u_G}{\rho_L + \rho_G} \pm k \sqrt{\frac{g}{k} \frac{\rho_L - \rho_G}{\rho_L + \rho_G} - \rho_L \rho_G \frac{\left(u_G - u_L\right)^2}{\left(\rho_L + \rho_G\right)^2}}, \tag{4.47}$$

or

$$\omega = k \frac{\rho_L u_L + \rho_G u_G \pm \sqrt{\frac{g}{k}(\rho_L + \rho_G)(\rho_L - \rho_G) - \rho_L \rho_G \left(u_G - u_L\right)^2}}{\rho_L + \rho_G}, \tag{4.48}$$

which gives the frequency $\omega/2\pi$ for a wave of wavelength $\lambda = 2\pi/k$.

We can replace g by its vertical component $g\cos(\phi)$ in the case of a tilted domain:

$$\omega = k \frac{\rho_L u_L + \rho_G u_G \pm \sqrt{\frac{g\cos(\phi)}{k}(\rho_L + \rho_G)(\rho_L - \rho_G) - \rho_L \rho_G \left(u_G - u_L\right)^2}}{\rho_L + \rho_G}. \tag{4.49}$$

The eigenvectors of $\mathbf{A}$ are the solutions of (4.45) for $\mathbf{u}$:

$$B_G = -i\left(u_G - \frac{\omega}{k}\right)\hat{\eta} \tag{4.50}$$

$$B_L = i\left(u_L - \frac{\omega}{k}\right)\hat{\eta}, \tag{4.51}$$

with η free to be chosen and (two different) ω given by (4.49). According to the notation of subsection 4.3.2 we may write $\eta = \Delta h_{\text{int}}$. The amplitudes of the horizontal velocity perturbations are then given by

$$\Delta u_G = \frac{\partial \phi'_G}{\partial s} \quad \rightarrow \quad \Delta\hat{u}_G = (ku_G - \omega)\,\mathrm{e}^{-kz}\Delta\hat{h}_{\text{int}}, \tag{4.52a}$$

$$\Delta u_L = \frac{\partial \phi'_L}{\partial s} \quad \rightarrow \quad \Delta\hat{u}_L = -(ku_L - \omega)\,\mathrm{e}^{kz}\Delta\hat{h}_{\text{int}}, \tag{4.52b}$$

which may be used to yield an analytical expression for the solution in time in the form of (4.38). The (complex) amplitudes of the vertical velocity perturbations, which are superimposed on the base flow of $w_G = w_L = 0$, are calculated in the same way:

$$\Delta w_G = \frac{\partial \phi'_G}{\partial z} \quad \rightarrow \quad \Delta\hat{w}_G = i\,(ku_G - \omega)\,\mathrm{e}^{-kz}\Delta\hat{h}_{\text{int}}, \tag{4.53a}$$

$$\Delta w_L = \frac{\partial \phi'_L}{\partial z} \quad \rightarrow \quad \Delta\hat{w}_L = i\,(ku_L - \omega)\,\mathrm{e}^{kz}\Delta\hat{h}_{\text{int}}. \tag{4.53b}$$

We may also substitute $z = h - h_{\text{int}}$.

4.4.2 Bounded domain

Milne-Thomson (1960) [136] has done an analysis similar to the one above, but with horizontal walls bounding the domain at the top and bottom, as opposed to the domain extending out infinitely in the vertical direction. This bounded geometry is the one considered in section 3.5. The only difference in the derivation with respect to the unbounded domain (i.e. Kelvin-Helmholtz) discussed above, is a changed boundary condition. For the unbounded case we take $\partial\phi/\partial z = 0$ at $z \to \pm\infty$, stemming from the demand that the disturbance dies out infinitely far from the interface. For the bounded case we take $\partial\phi/\partial z = 0$ at $z = -h_{\text{int}}$ and $z = H - h_{\text{int}}$, stemming from a demand that the vertical velocity must be zero at the lower and upper bounding walls. This means that the expressions for the perturbations change; instead of (4.44) we get

$$\eta = \hat{\eta}\mathrm{e}^{i(ks-\omega t)} \tag{4.54a}$$

$$\phi'_G = B_G \cosh\left(k(H - h_{\text{int}} - z)\right)\mathrm{e}^{i(ks-\omega t)}, \tag{4.54b}$$

$$\phi'_L = B_L \cosh\left(k(z + h_{\text{int}})\right)\mathrm{e}^{i(ks-\omega t)}. \tag{4.54c}$$

The rest of the analysis proceeds in the same manner as in the unbounded case.

The resulting dispersion relationship is similar to the dispersion found for an unbounded domain (i.e. Kelvin-Helmholtz) (4.49):

$$\omega = k\frac{\rho_L u_L + \rho_G u_G \xi_2 \pm \sqrt{\frac{\tanh(kh_{\text{int}})g\cos\phi}{k}(\rho_L + \rho_G\xi_2)(\rho_L - \rho_G) - \rho_L\rho_G\xi_2\left(u_G - u_L\right)^2}}{\rho_L + \rho_G\xi_2}, \tag{4.55}$$

with

$$\xi_2 = \frac{\tanh(kh_{\text{int}})}{\tanh(k(H - h_{\text{int}}))}.$$

This equation is equivalent to equations 2.43 and 2.80 in Montini (2011) [131].

If we take the limits $kh_{\text{int}} \to \infty$, $k(H - h_{\text{int}}) \to \infty$ (i.e. short wavelengths), the hyperbolic tangent functions approach 1, so $\xi_2 \to 1$, and the dispersion relations of unbounded and bounded domains become identical. If we take $\rho_G = 0$, we obtain the dispersion relationship for free-surface gravity waves (usually derived with $u_L = 0$):

$$\omega = ku_L \pm \sqrt{gk\tanh(kh_{\text{int}})}. \tag{4.56}$$

In the limit of long wavelengths (or shallow fluid layers),

$$\frac{\tanh(kh_{\text{int}})}{k} \to h_{\text{int}} \quad \text{for} \quad kh_{\text{int}} \to 0, \tag{4.57a}$$

$$\xi_2 = \frac{\tanh(kh_{\text{int}})}{\tanh(k(H-h_{\text{int}}))} \to \frac{h_{\text{int}}}{H-h_{\text{int}}} = \xi_1 \quad \text{for} \quad k(H-h_{\text{int}}) \to 0. \tag{4.57b}$$

In this limit the dispersion relation thus reduces to that of the 1D two-fluid model (4.14), which we repeat here for convenience:

$$\omega = k\frac{\rho_L u_L + \rho_G u_G \xi_1 \pm \sqrt{(\rho_L + \rho_G\xi_1)(\rho_L - \rho_G)h_{\text{int}} g\cos\phi - \rho_L\rho_G\xi_1 (u_G - u_L)^2}}{\rho_L + \rho_G\xi_1}, \tag{4.58}$$

The most important difference between the inviscid dispersion relation for the 2D analysis (4.55) and the inviscid dispersion relation for the 1D analysis (4.58), is the appearance of the factor $1/k$ inside the square root. This means that according to the 2D analysis, the velocity difference for which the wave is unstable is dependent on the wavelength of the wave (compare to (4.17)):

$$\Delta U^2 = (u_G - u_L)^2 > \left(\frac{\tanh(kh_{\text{int}})}{\rho_L} + \frac{\tanh(k(H-h_{\text{int}}))}{\rho_G}\right)(\rho_L - \rho_G)\frac{g\cos\phi}{k}. \tag{4.59}$$

For a given velocity difference, the system can be unstable to waves with short wavelengths (large k) while being stable for waves with long wavelengths (small k). However, taking the limit of long wavelengths was seen to remove the dependence on k (4.58). According to Montini (2011) [131], 'the logical consequence is that the averaging of the equations in order to obtain a one-dimensional system intrinsically implies the long wavelength ($k \to 0$) assumption'. The two-fluid model assumes negligible vertical velocities and a requirement for this is that the interfacial height does not vary too steeply along the horizontal direction. This is assured with the long wavelength assumption. The long wavelength assumption can be associated with the assumption $L \gg H$, discussed in subsection 2.3.1 and subsection 3.4.2.

The amplitudes of the velocity perturbations, similar to (4.52), are calculated by determining the coefficients B_G and B_L in (4.54) using equations (4.43). The coefficients are given by

$$B_G = -i\frac{ku_G - \omega}{k\sinh(k(H-h_{\text{int}}))}\hat{\eta}, \tag{4.60a}$$

$$B_L = i\frac{ku_L - \omega}{k\sinh(kh_{\text{int}})}\hat{\eta}. \tag{4.60b}$$

According to the notation of subsection 4.3.2 we may write $\eta = \Delta h_{\text{int}}$. Therefore the horizontal velocity perturbations can be calculated to be

$$\Delta u_G = \frac{\partial \phi'_G}{\partial s} \quad \to \quad \Delta\hat{u}_G = (ku_G - \omega)\frac{\cosh(k(H-h_{\text{int}}-z))}{\sinh(k(H-h_{\text{int}}))}\Delta\hat{h}_{\text{int}}, \tag{4.61a}$$

$$\Delta u_L = \frac{\partial \phi'_L}{\partial s} \quad \to \quad \Delta\hat{u}_L = -(ku_L - \omega)\frac{\cosh(k(z+h_{\text{int}}))}{\sinh(kh_{\text{int}})}\Delta\hat{h}_{\text{int}}. \tag{4.61b}$$

The vertical velocity perturbations, which are superimposed on the base flow of $w_G = w_L = 0$, are given by

$$\Delta w_G = \frac{\partial \phi'_G}{\partial z} \quad \to \quad \Delta\hat{w}_G = i(ku_G - \omega)\frac{\sinh(k(H-h_{\text{int}}-z))}{\sinh(k(H-h_{\text{int}}))}\Delta\hat{h}_{\text{int}}, \tag{4.62a}$$

$$\Delta w_L = \frac{\partial \phi'_L}{\partial z} \quad \to \quad \Delta\hat{w}_L = i(ku_L - \omega)\frac{\sinh(k(z+h_{\text{int}}))}{\sinh(kh_{\text{int}})}\Delta\hat{h}_{\text{int}}. \tag{4.62b}$$

Again, the vertical coordinate used here is related to h via $z = h - h_{\text{int}}$.

4.5 Comparison of dispersion relations

We compare the different dispersion relations for inviscid flow. We have

- for 1D flow (two-fluid model) the dispersion relation (4.14) with perturbations (4.35) and (4.36),
- for unbounded 2D flow (Kelvin-Helmholtz) the dispersion relation (4.48) with the velocity perturbations (4.52) and (4.53),
- and for bounded 2D flow the dispersion relation (4.55) and the perturbations (4.61) and (4.62).

We plot the dispersion relations in Figure 4.2, using the values in Table 4.1. Apart from the velocities, these are the test case parameters used in section 5.2. The stability limit for the two-fluid model given by (4.17) is $u_G - u_L = 7.55$ m/s.

Table 4.1: Kelvin-Helmholtz test case parameters.

u_L	ρ_L	ρ_G	g	H	ϕ	h_{int}/H
0.5 m/s	998 kg/m^3	1.2 kg/m^3	9.81 m/s^2	0.01 m	0 rad	0.3

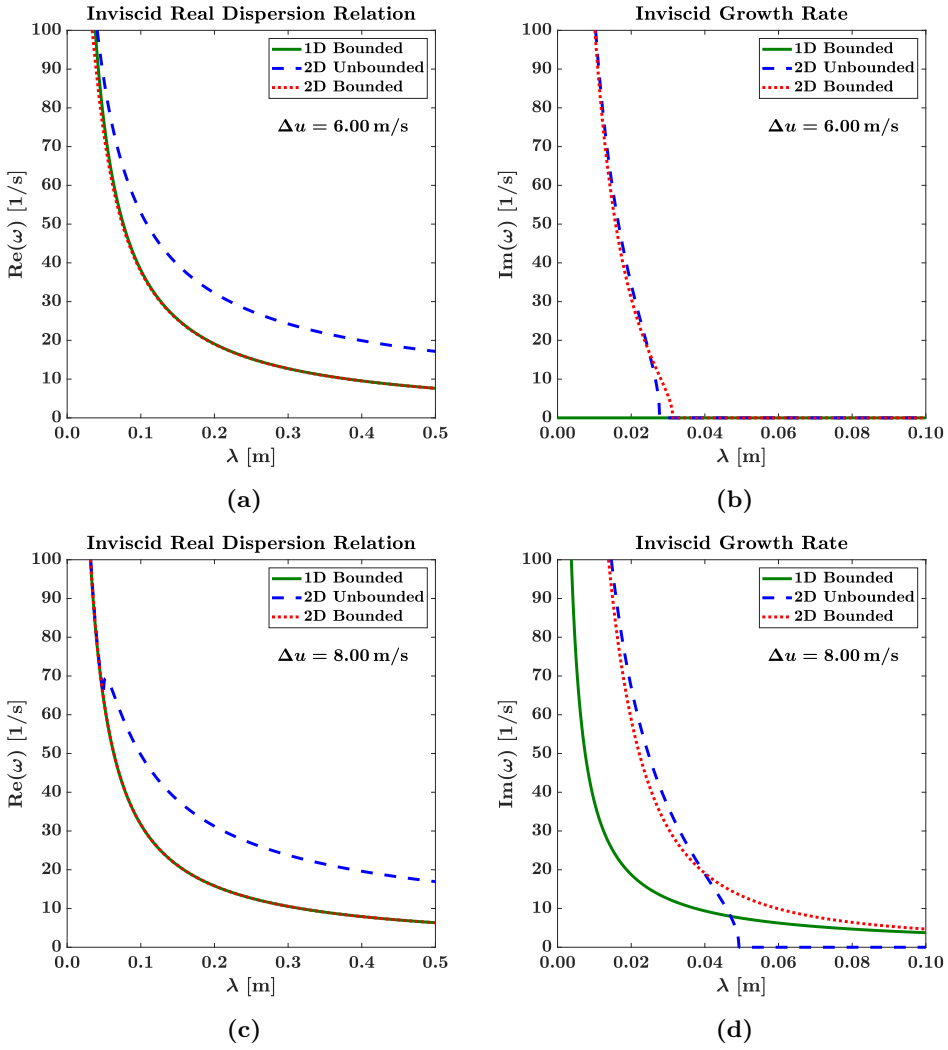


Figure 4.2: Inviscid dispersion relations, using the parameters given in Table 4.1.

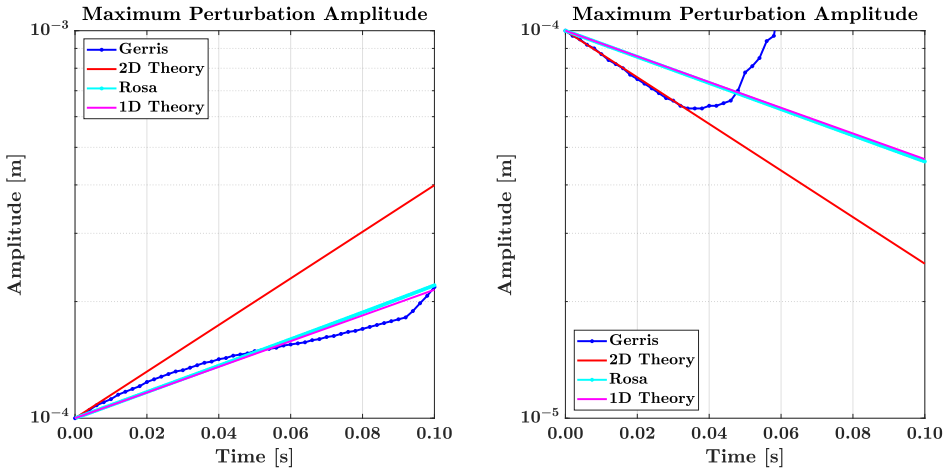
In the figure the limiting behavior of the dispersion relations is readily observed. The dispersion relations for the 1D bounded and 2D bounded cases indeed converge for long wavelengths, and the dispersion relations for the 2D unbounded and 2D bounded cases converge for short wavelengths. The 2D cases are always unstable to perturbations with the shortest wavelengths. Only if viscosity were to be added stable solutions can be reached. The instability of the inviscid 2D case does not mean that a

similar case that would include viscous terms is ill-posed, since the analysis is based on the linearized equations, and not on the full set. Moreover, the viscous terms in the 2D case contain derivatives of the velocity and thus should be included in the homogeneous part of the equations when analyzing viscous flow.

The inviscid 1D bounded model is either stable and well-posed or unstable and ill-posed for all wavelengths, depending on the velocity difference. The inviscid equations form the homogeneous part of the viscous equations, so that the viscous two-fluid model can also directly said to be ill-posed.

For a slightly differenct casein which all models have a nonzero imaginary part of ω, Figure 4.3 shows the growth of initial perturbations of a certain wavelength with time. Depending on the sign of the imaginary part of the ω to which the perturbations correspond (referring to the $\pm$ in (4.55)), the perturbation is amplified or damped. The 2D Gerris simulations match the predicted growth rate initially, but deviate after some time. This is to be expected; nonlinear effects are not taken into account in the linear stability theory.

The two-fluid Rosa simulations stick to their predicted growth rate for longer. This is to be expected since this growth rate can be deduced directly from the full set of inviscid equations, not from a linearized version as is the case for the 2D equations. For the inviscid case, the linear stability analysis performed for the two-fluid model in section 4.4 yields the same dispersion relations as the treatment in subsection 4.3.1 which is based on the full, non-linearized equations. These two-fluid simulations are fragile though, since the model equations are ill-posed, and if there is not enough numerical diffusion small wavelength perturbations might grow with an unbounded growth rate (see the discussion in subsection 4.3.3). Nevertheless, these plots indicate the correctness of the simulation codes, and our understanding of linear stability theory.



(a) Initialized with a perturbation corresponding to the *plus* sign in (4.55).

(b) Initialized with a perturbation corresponding to the *minus* sign in (4.55).

Figure 4.3: Comparison of the growth rate of the perturbations in Gerris simulations, in 2D linear stability theory (4.55), in Rosa (two-fluid) simulations, and in 1D two-fluid model analysis (4.14), all inviscid. Using most parameters given in Table 4.1 but with $\rho_G = 980\,\mathrm{kg/m^3}$ and $u_L = 0.01\,\mathrm{m/s}$, $u_G = 0.06\,\mathrm{m/s}$. The wavelength is equal to the domain height and length, 0.1 m.

4.6 Stability diagrams

A stability diagram can be made by computing the steady state of the two-fluid model for a range of liquid and gas velocities, and for each steady state computing whether the dispersion relation predicts complex angular frequencies. This is shown in Figure 4.4a and is similar to the map found in [137]. Alternatively, these steady states can be computed as a function of the given pressure gradient and hold-up - this is shown in Figure 4.4b.

The boundary between ill-posedness and well-posedness (shown in Figure 4.4 as the dividing line between the white and gray-shaded areas) is determined by the inviscid analysis which was described

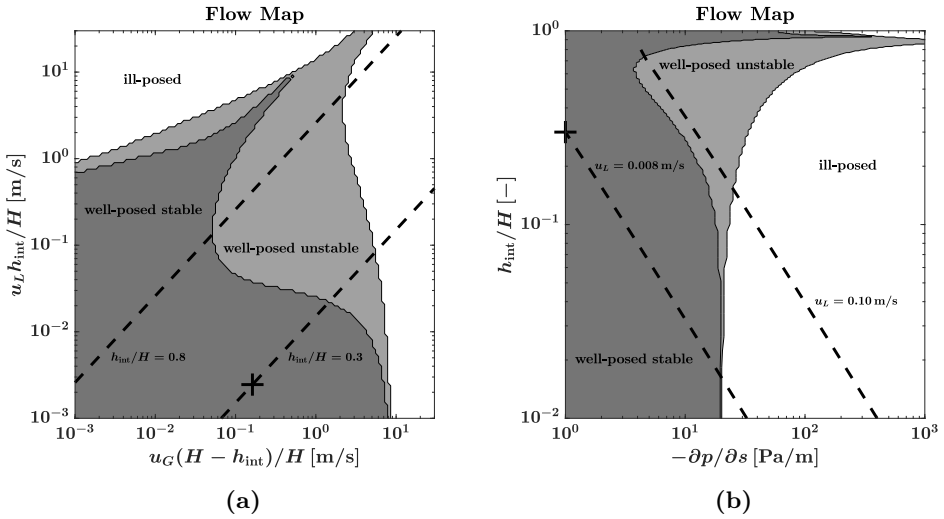


Figure 4.4: Two maps of the stability regions of the two-fluid model as described in section 3.5. The plots show whether steady states corresponding to the parameters on the axes are stable, unstable, or ill-posed. The dashed lines in **(a)** are lines of constant holdup. The dashed lines in **(b)** are lines of constant velocity.

in subsection 4.3.1. If complex eigenvalues result from the homogeneous equations via (4.14), this will lead to an unbounded growth rate (for small wavelength perturbations) and the system is thus ill-posed. This happens when the velocities, driven by the constant background pressure gradient, lie too far apart (according to (4.17)).

In the 'well-posed stable' area initial disturbances are damped. This is an area where it is safe to carry out simulations: we can have trust that our model will remain well-posed, as long as the initial disturbances are small enough.

There is an area where the dispersion relation based on the homogeneous equations, or equivalently the inviscid equations, yields real eigenvalues, but the dispersion relation based on the viscous equations yields complex eigenvalues. Here, the two-fluid model is well-posed, but the viscous two-fluid equations are unstable. This is called the 'well-posed unstable' area in Figure 4.4b. An initial disturbance will grow according to (4.38). Initial states in this area can often only be analyzed for limited amounts of time with the two-fluid model, since with time these states are likely to evolve into states which are ill-posed. On the other hand, these states might lead to the transition from wavy flow to slug flow.

4.7 Conclusion

If one wants to find a well-behaved solution to the two-fluid model equations, the equations must be well-posed. If the set is hyperbolic, with correct boundary conditions, the time-dependent problem is well-posed. The propagation without growth or damping of a waveform or the existence of a complete set characteristics with all real eigenvalues are sufficient conditions for proving hyperbolicity of a quasi-linear PDE system.

The considered two-fluid model is hyperbolic and thus well-posed when the velocity difference between the liquid and gas phase is not too large. This analysis concerns only the homogeneous part of the equations, which for the two-fluid model coincides with the inviscid equations. Linear stability analysis of the full (viscous) equations shows that the two-fluid model can be unstable and well-posed at the same time.

In our generation of training data for the neural network, we will need to take care that our two-fluid model is well-posed for the chosen parameter ranges, and stable so that there is no danger of ill-posedness arising after some time. For our two-fluid model is useless (produces unphysical results) when the model equations are ill-posed. This is a given with the current two-fluid model. Closure terms aimed at getting the two-fluid model to agree with high-fidelity simulations can do nothing to change the ill-posedness problem, since they are a source term in the model equations (4.8). They *do* however have an influence on the boundary between the well-posed stable and well-posed unstable regimes shown in Figure 4.4.

A similar linear stability analysis can be done for the incompressible Euler equations for 2D channel flow. There is a difference between the vertically bounded and vertically unbounded case, though they converge for short wavelengths. The inviscid 2D bounded dispersion relation and the inviscid two-fluid dispersion relation converge for long wavelengths. Since the inviscid dispersion relations do not depend on the closure terms, any kind of closure term will do nothing to change the discrepancy between the two-fluid model and the 2D analysis for low wavelengths. Put otherwise, the assumption $L \gg H$ is inherent to the employed two-fluid model, and cannot be remedied by appropriate closure terms.

With the analysis in this chapter, we have thus gained insight into the fundamental limitations of the two-fluid model, which we should not hope to improve upon with our closure terms learned from high-fidelity simulations.

Chapter 5

Viscous Validation

5.1 Introduction

In this chapter, we conduct viscous simulations with both our high fidelity (Gerris) and low fidelity (Rosa) models. We validate our models for the viscous case by comparison to the steady state analytical relations (3.80)–(3.83). By evaluating the convergence of the Gerris simulations to the analytical solution, we can make a well-founded choice for the resolution of the simulations. For Rosa we need to perform wavy unsteady simulations in order to make a good choice of spatial and temporal resolution. We compare Gerris and Rosa wavy unsteady simulations and analyze the results. Additionally, we discuss the complications in extracting the necessary neural network training data, in particular the stresses, from the high fidelity simulations.

We use Gerris to simulate 2D channel flow with periodic boundaries under a constant body force. For pipe flow the boundary conditions at top and bottom are no-slip. The constant body force is a combination of a constant background pressure gradient and a gravitational force arising due to inclination of the domain:

$$B_L = \left.\frac{\partial p}{\partial s}\right|_0 + \rho_L g \sin(\phi), \quad B_G = \left.\frac{\partial p}{\partial s}\right|_0 + \rho_G g \sin(\phi).$$

The body forces arising due to channel inclination are different for the liquid and the gas. However, we will limit our analysis here to cases driven by solely a background pressure gradient.

The physics behind Gerris is explained in chapter 2 and its numerical details are given in section 2.5. The 1D two-fluid model is described in chapter 3 and the numerical properties of the Rosa code are discussed in section 3.6.

5.2 Test case

As described in Figure 1.4, we first examine the steady, fully developed, flat interface case of fully developed stratified flow (step [2] in Figure 1.4 is the current validation). We take as a basic test case a case with relatively low Reynolds numbers, so as to ensure laminar flow and thus adherence to the analytical solution. Another condition for the test case is that the two-fluid model is well-posed and stable for the conditions of the test case. The linear stability of the steady states of the two-fluid model, which result for a given initial holdup and pressure gradient in a periodic domain, is discussed in section 4.6. The cross in Figure 4.4b locates our standard test case, and its parameters are given in Table 5.1.

The simulations are run with a domain length of $L = H$. The domain length does not matter very much because the simulations are started from a fully developed initial condition and the simulations remain fully developed throughout. The length of the domain is not a characteristic length scale here; the height is the relevant measure.

With the analytical solution given by [122] and [9] (discussed in subsection 3.4.4), the averaged velocities and stresses can be calculated. These are given in Table 5.2. It is clear that the chosen parameters lead to low Reynolds numbers which fall well in the laminar regime (in the literature the critical Reynolds number for single phase pipe flow ranges from 1700 to 2300 [138]).

Table 5.1: Test case parameters.

Parameter	Symbol	Value	Units
Background pressure gradient	$\partial p/\partial s$	-1	$\mathrm{kg\,m^{-2}\,s^{-2}}$
Liquid density	ρ_L	998	$\mathrm{kg\,m^{-3}}$
Gas density	ρ_G	1.2	$\mathrm{kg\,m^{-3}}$
Channel height	H	0.01	m
Initial interface height	h_{int}	$0.3H$	m
Liquid viscosity	μ_L	$1.002 \cdot 10^{-3}$	$\mathrm{kg\,m^{-1}\,s^{-1}}$
Gas viscosity	μ_G	$1.82 \cdot 10^{-5}$	$\mathrm{kg\,m^{-1}\,s^{-1}}$
Acceleration of gravity	g	9.81	$\mathrm{m\,s^{-2}}$
Pipe inclination	ϕ	0	degrees

Table 5.2: Test case solution.

Parameter	Symbol	Value	Units
Averaged liquid velocity	u_L	0.00818	$\mathrm{m\,s^{-1}}$
Averaged gas velocity	u_G	0.232	$\mathrm{m\,s^{-1}}$
Liquid wall stress	τ_L	-0.00646	$\mathrm{kg\,m^{-1}\,s^{-2}}$
Gas wall stress	τ_G	-0.00354	$\mathrm{kg\,m^{-1}\,s^{-2}}$
Interfacial stress	τ_{int}	-0.00346	$\mathrm{kg\,m^{-1}\,s^{-2}}$
Liquid Reynolds number	Re_L	48.9	-
Gas Reynolds number	Re_G	214	-
Liquid Froude number	Fr_L	0.00114	-
Gas Froude number	Fr_G	0.391	-

5.3 Flat interface validation

Figure 5.1 shows that if we run Gerris simulations for the case discussed in section 5.2, initialized either from a zero velocity field or the theoretical steady state, we indeed get the theoretical velocity profile predicted by Biberg and Halvorsen [122], from which the averaged stresses and velocities given above are calculated. The theoretical, red-dashed, line lies on top of the Gerris simulation results and its cubic spline interpolation.

Figure 5.2 shows how the simulated averaged velocity converges towards the theoretical steady state value, both with time and with grid refinement. The simulations are initialized with a zero velocity field, which is set in motion by the imposed constant pressure gradient. The results of two-fluid model simulations, performed with the Rosa code introduced in section 3.6, are plotted alongside the 2D simulations.

It is good to see that the temporal evolution of the 2D simulations converges toward the temporal evolution of the 1D simulations, upon grid refinement. Clearly the two-fluid code is very accurate for this simple case. It shows the same results (down to machine precision) for a wide range of relatively low different grid resolutions ($\Delta s = H/32$ is plotted), at constant $\Delta t/\Delta s = 3.2$. This is because for the flat interface case with periodic boundaries, the flow is uniform along s and all derivatives (except the driving pressure) in this direction are zero and can be exactly calculated to be so by the 1D code. The 2D code still needs to deal with derivatives in the vertical direction though, and needs a sufficiently fine resolution to resolve the velocity profiles well.

The corresponding stresses are plotted in Figure 5.3. The Gerris simulations indeed converge to the Ullmann analytical closure terms, evaluated at the steady state. These closure terms were used in the two-fluid model simulations, and the combination of Figure 5.2 and Figure 5.3 shows that using these closure terms the two-fluid model simulations converge to the 2D analytical steady state. This is encouraging: *for the flat interface steady state case the two-fluid model, 2D theory, and 2D simulations give the same results.*

However, initially the two-fluid model does give a deviating result: the liquid wall stress first rises to become positive, causing a force in the direction of the average flow. Only after some time the stress becomes negative, opposed to the direction of the average flow, as would be expected. The factor

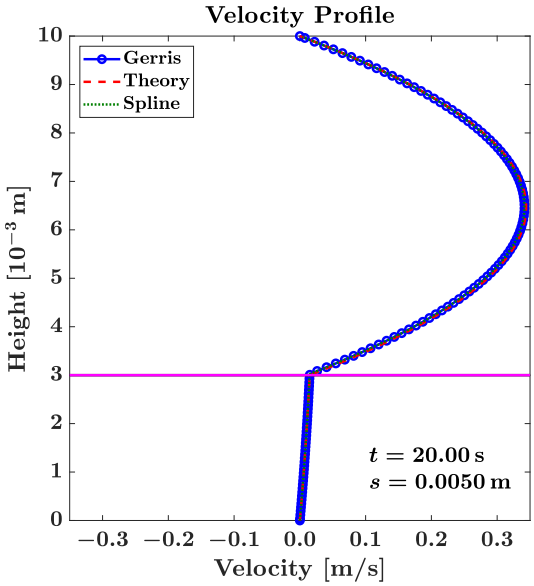


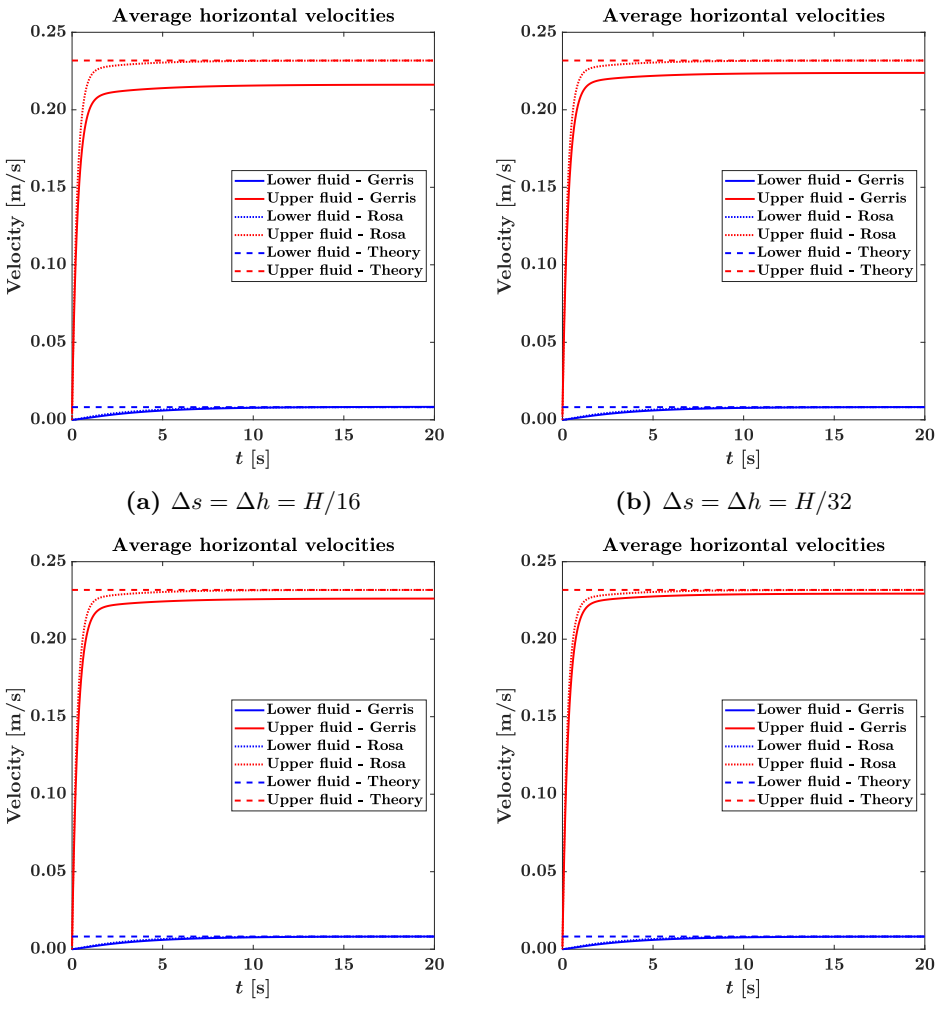
Figure 5.1: Simulated velocity profile, with its spline interpolation, compared to the theoretical profile. Grid spacing is $H/128$, uniform in both directions. The magenta line indicates the interface height.

responsible for this change of sign is the correction factor F_L^* given by (3.88). The possibility for this factor to change sign is desired in order to take account cases where the velocity profile has a change of sign: for example if there is a region of backflow at the bottom of the channel, above which the flow reverses to make the overall depth-averaged liquid velocity positive (see Figure 5.4). In this case a liquid wall stress in the direction of the flow would be physical. However such a flow profile is not observed in the plotted 2D simulations; such a flow profile would sooner be expected in a flow forced by a channel inclination.

This error can be explained by considering that the closure terms are in principle derived for steady state only. Their generalization to unsteady flow is not supported by analytical derivations, rather we expect the closure terms to generalize because we expect the depth averaged velocities (which are the main inputs for the closure terms) to have a direct relation to the velocity gradients (which actually directly determine the stresses). This may be the case when only considering steady states, but it is certainly not the case when also considering unsteady states, for which in principle infinitely many velocity fields are possible for a given averaged velocity, as long as it is divergence-free and continuous. The problem of determining the velocity profile from the averaged velocity is an inverse problem with infinitely many solutions; we call this a uniqueness problem.

From the information that is fed to a closure term there is no way to know if the flow is unsteady, and if so what the actual profile is out of infinite possibilities. We can only attempt to learn, for example with a neural network, what the most likely unsteady profiles are and return the stresses for these profiles. Which velocity profiles arise will depend on the initial conditions and on the stability of different profiles: the flow will converge to the steady state profile with time; and we will see the profiles that take the flow from the initial condition to the steady state. For now, our analytical closure term takes the (exact) profile corresponding to the steady state, which in this case is inclined flow with backflow. The conditions for liquid backflow are a large (negative) pressure gradient with an opposite inclination, leading to large u_G and low u_L as seen in Figure 5.4a, and indeed corresponding to the situation at low t in Figure 5.2, making this explanation plausible.

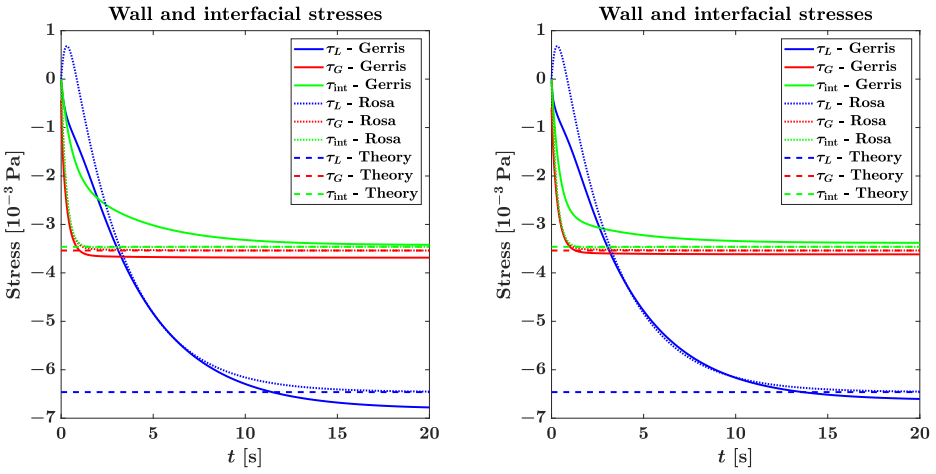
This error is a good illustration of the limitation of applying closure terms, with limited input information, to unsteady flow. Still, when the transient (and non-fully-developed) terms are relatively small the flow will be near some steady state and the approximation of assuming steady state flow, for which a one-to-one mapping between averaged velocities and profiles does exist, will be acceptable.



(a) $\Delta s = \Delta h = H/16$ **(b)** $\Delta s = \Delta h = H/32$

(c) $\Delta s = \Delta h = H/64$ **(d)** $\Delta s = \Delta h = H/128$

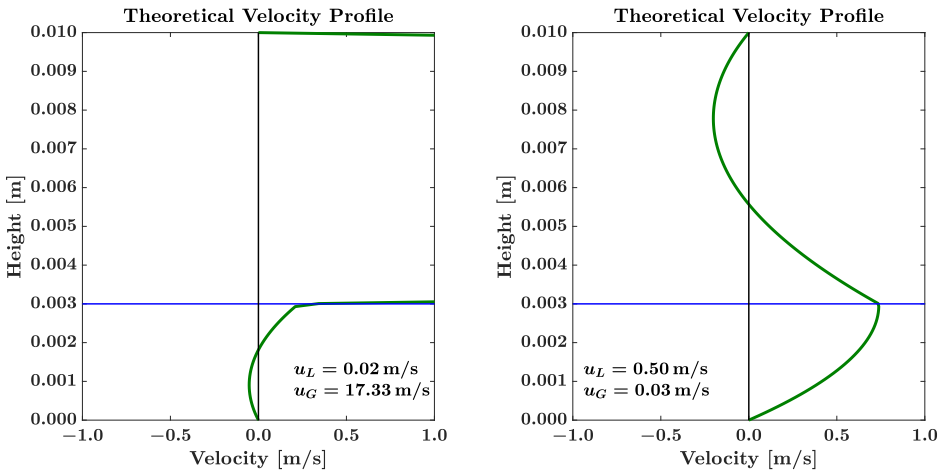
Figure 5.2: This figure shows the evolution in time of the vertically averaged velocities, which are constant along s. The subplots show the results of using different grid resolutions in the 2D Gerris simulations. The grid spacing is varied while keeping the Courant number $\sim \Delta t/\Delta s$ constant. Changing the resolution in the 1D Rosa simulations had no effect on these plots ($\Delta s = H/32$ is plotted).



(a) $\Delta s = \Delta h = H/16$ **(b)** $\Delta s = \Delta h = H/32$

(c) $\Delta s = \Delta h = H/64$ **(d)** $\Delta s = \Delta h = H/128$

Figure 5.3: This figure shows the evolution in time of the stresses, which are constant along s. The subplots show the results of using different grid resolutions in the 2D Gerris simulations. The grid spacing is varied while keeping the Courant number $\sim \Delta t/\Delta s$ constant. Changing the resolution in the 1D Rosa simulations had no effect on these plots ($\Delta s = H/32$ is plotted). Two interfacial points are excluded in the interfacial stress calculation (see section 5.4).



(a) Liquid backflow. **(b)** Gas backflow.

Figure 5.4: This figure shows steady state theoretical velocity profiles according to relations (3.80)–(3.83). Using the same parameters as in Table 5.1, but changing $\partial p/\partial s$ and ϕ, we can create regions of backflow, while the averaged velocities u_L and u_G remain positive.

5.4 Extracting stresses from 2D Gerris simulations

5.4.1 Problem description

In section 5.3, Figure 5.3, stresses extracted from 2D Gerris simulations were shown. The code does not give these directly; they must be calculated from the velocity profile. This can be done in different ways. Here, we fit a cubic spline to the velocity profile (see Figure 5.1). A cubic spline consists of piecewise polynomials set between each pair of consecutive points. Each polynomial is based on four surrounding points. At the endpoints the polynomials are based on three surrounding points and the boundary condition is set that the second derivative of the polynomial should be zero.

Using a spline has the advantage that we can easily evaluate the spline's, and its exact derivative's, value at any point (this is also used for integrating the velocity profile in the calculation of the average velocities). The stresses are calculated using the derivative of the splines, fitted separately for liquid and gas, at the endpoints. The stress can be calculated on either side of the interface, since the stress must be continuous over the interface. In order to reduce the error the average is taken of the stresses on either side.

Figure 5.5a and Figure 5.5b show a problem that is encountered when performing the spline interpolation. The figures show that the 2D Gerris steady state deviates slightly from the analytical steady state - this was already clear in Figure 5.2 and Figure 5.3. At the interface minor differences in the velocity profile lead to large differences in the velocity gradient. This in turn leads to inaccurate interfacial stresses. This error in the velocity profile is caused by the inaccurate calculation of stresses at the interface in a VOF simulation. Figure 5.6a and Figure 5.6b show that this problem is not resolved simply by increasing the resolution of the simulation. Although the difference between the analytical and the theoretical velocity profiles indeed becomes smaller, the velocity gradient is calculated over a smaller vertical distance so that the error in the interfacial stress is not reduced.

Figure 5.7 illustrates the problem with the numerical handling of the interfacial stress. We would like to see the velocity profile as illustrated in the figure, with a discontinuous velocity gradient (with a jump at the interface) but a continuous stress (discussed in subsection 2.3.1). If we want to advance a time step from this situation, we need to calculate the stress $\tau_{j+1/2}$ at the boundary of the finite volume cell corresponding to u_j. With a central discretization, this stress is given by

$$\tau_{j+1/2} = \mu_{j+1/2}\frac{u_{j+1}-u_j}{\Delta h} = \mu\left(C_{j+1/2}\right)\frac{u_{j+1}-u_j}{\Delta h}. \tag{5.1}$$

The viscosity $\mu_{j+1/2}$ at the grid cell boundary is a priori unknown and can only be based on the viscosities and color functions C of grid cells j and $j+1$. This formulation for the stress will generally carry a large error because the approximation of the velocity profile's slope is inaccurate due to its discontinuous nature. In the depicted case the calculated slope will be lower than is realistic. The final interfacial stress depends also on the viscosity. If we use the standard method, (2.56), known as the arithmetic mean

$$\mu(C) = C\mu_L + (1-C)\mu_G, \tag{5.2}$$

with a central interpolation scheme

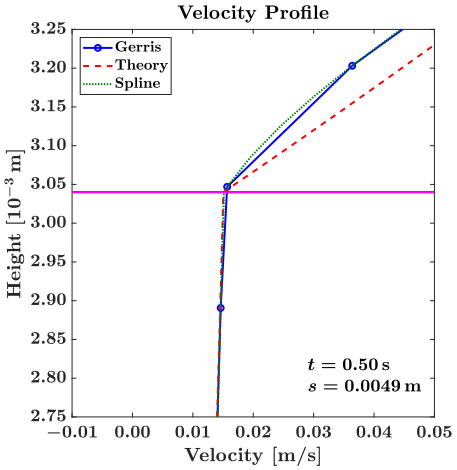
$$C_{j+1/2} = \frac{C_{j+1}+C_j}{2}, \tag{5.3}$$

the calculated viscosity at point $i+1/2$ will be orders of magnitude larger than the gas viscosity, while the point is in fact occupied by the gas. These effects combine to make the interfacial stress rather inaccurate, which then interacts with the velocity profile to take it away from the analytical solution, which in turn alters the stress again. Figure 5.11 shows that the effect is strongly dependent on where the interface is located relative to the grid points; if the interface is located approximately halfway between two grid points the accuracy is better.

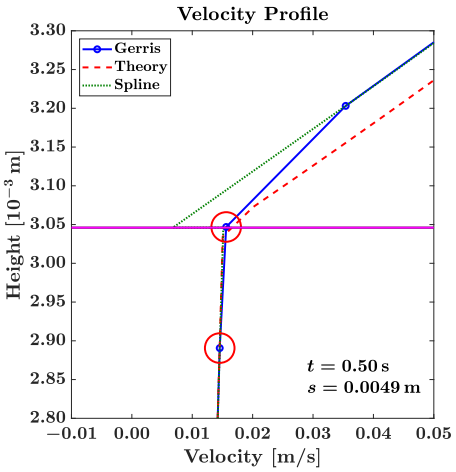
5.4.2 Mixed cell viscosity

The problem described above can be alleviated by defining the viscosity used in (5.1) differently. One can use a so-called harmonic mean for the viscosity:

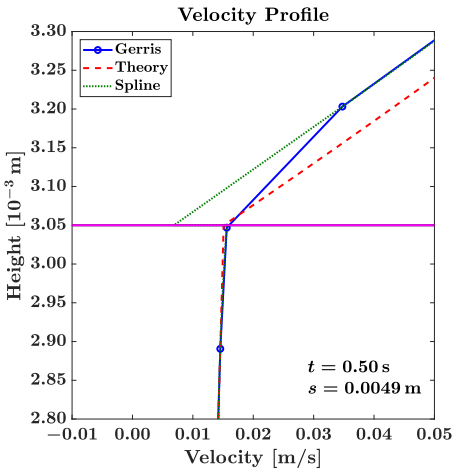
$$\mu_{j+1/2} = \left(\frac{1-C_{j+1/2}}{\mu_G} + \frac{C_{j+1/2}}{\mu_L}\right)^{-1}. \tag{5.4}$$



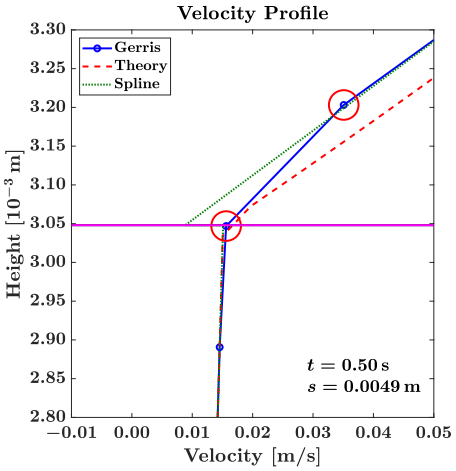
(a) Interface located just below a grid point, **no** interfacial points excluded in the velocity profile spline interpolation.



(b) Interface located just below a grid point, with the two red-circled interfacial points excluded in the velocity profile spline interpolation.



(c) Interface located just above a grid point, **no** interfacial points excluded in the velocity profile spline interpolation.

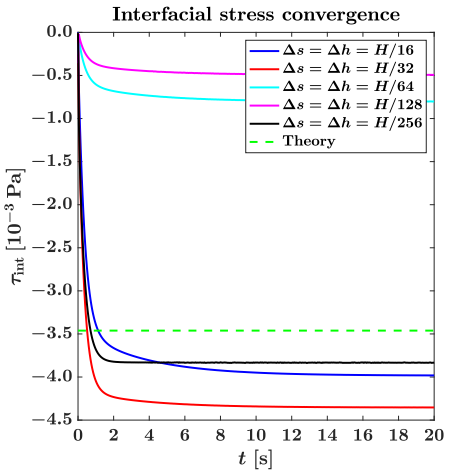


(d) Interface located just above a grid point, with the two red-circled interfacial points excluded in the velocity profile spline interpolation.

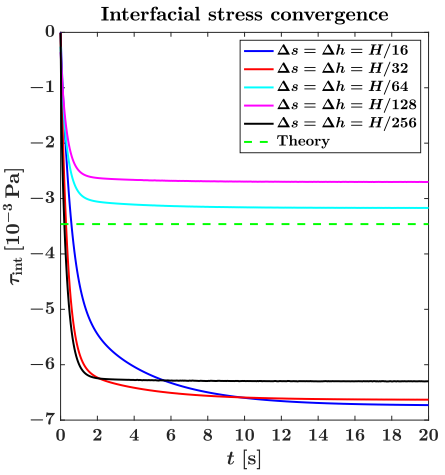
Figure 5.5: This figure shows velocity profiles, zoomed in at the interface, for flat interface steady state viscous flow. For the plots in which the grid points closest to the interface are not excluded in the calculation of the velocity profile slope, the error in the slope is large on the side which harbors the grid point closest to the interface. Excluding the two points closest to the interface helps to make the slope of the interpolated velocity profile match the theoretical profile better.

This was proposed by Coward et al. [139] for viscous shear flow. In a finite volume scheme, if the interpolation of the color function is exact, this ensures continuity of the stress across a flat, horizontal interface (see also [44, p. 62–64]). In our stratified channel flow the interface will often indeed be approximately horizontal. Figure 5.8 shows that this continuity is not observed in practice, though.

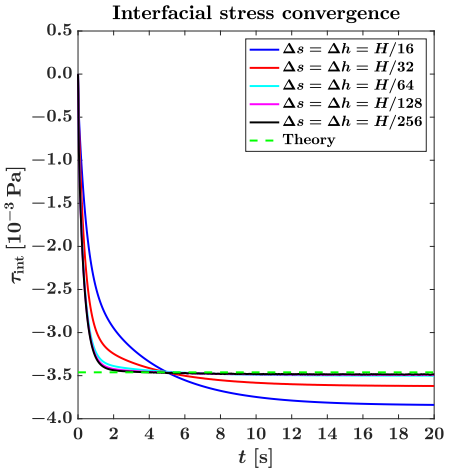
Using the harmonic mean means that the mixed cell viscosity is less 'biased' towards the higher viscosity. See Figure 5.9 for an illustration. The arithmetic mean's higher viscosity generates an artificially large viscosity region by displacing the "effective interface" from the true interface position towards the less viscous fluid [44, p. 64]. This has a stabilizing effect, but it distorts the velocity profile, bringing the effective bending point upward and reducing the gas average velocity as a result. This is shown in Figure 5.10. Overall, regarding Figure 5.6 and Figure 5.10, we can conclude that using the harmonic mean has a positive effect on the agreement between the Gerris simulations and the theoretical relations,



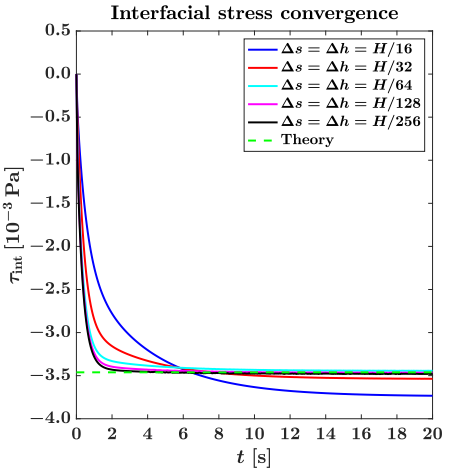
(a) With arithmetic mean for viscosity, without exclusion.



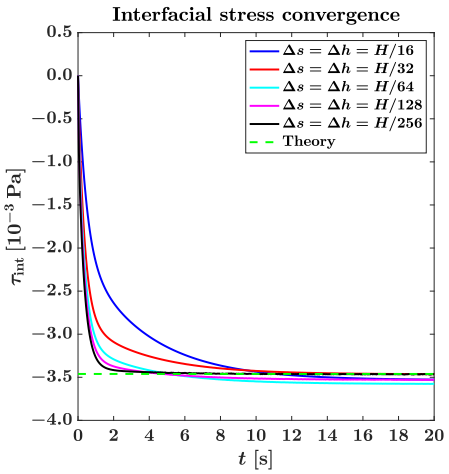
(b) With harmonic mean for viscosity, without exclusion.



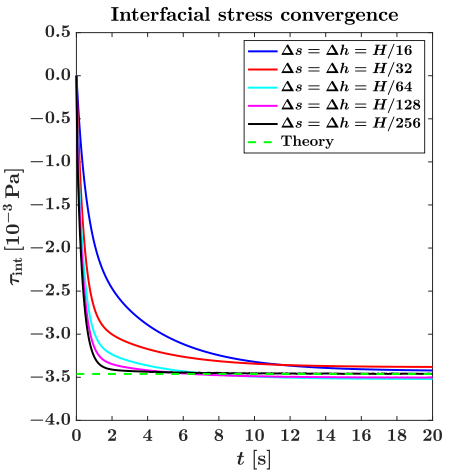
(c) With arithmetic mean for viscosity, excluding 1 interfacial point.



(d) With harmonic mean for viscosity, excluding 1 interfacial point.



(e) With arithmetic mean for viscosity, excluding 2 interfacial points.



(f) With harmonic mean for viscosity, excluding 2 interfacial points.

Figure 5.6: The convergence of the interfacial stress, with time and grid resolution. The interfacial stress is calculated with or without exclusion of interfacial points in the interpolation of the velocity profile, and with different methods for calculating the mixed cell viscosity.

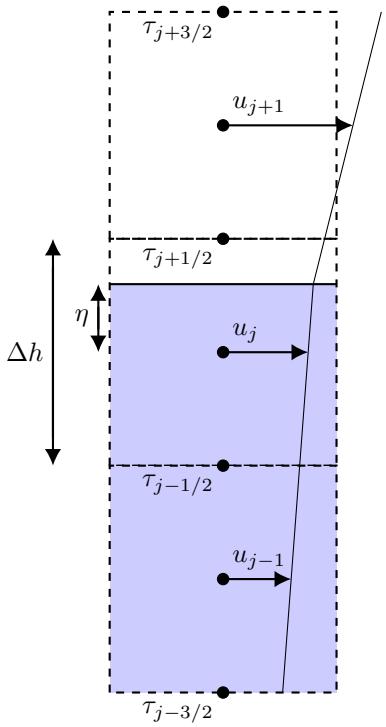


Figure 5.7: A schematic of grid cells and the desired velocity profile near a VOF interface.

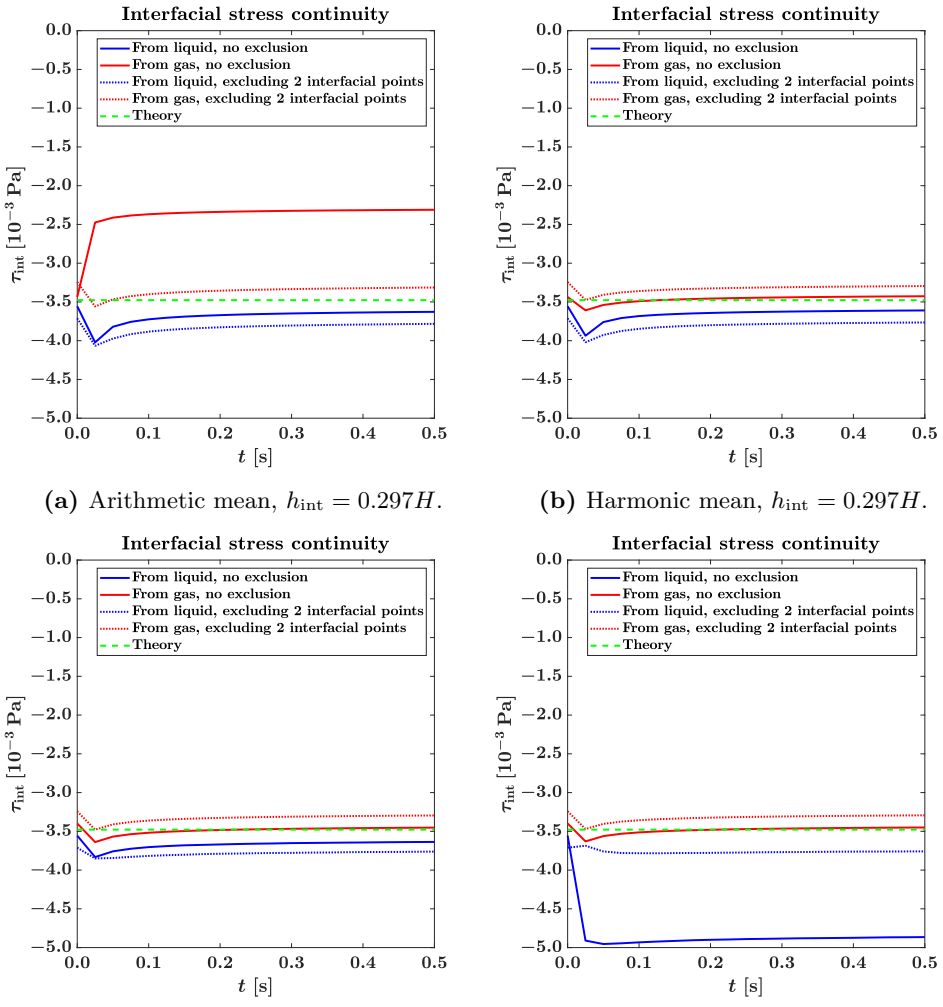
when used in conjunction with the exclusion of interfacial points, discussed in the following section.

5.4.3 Exclusion of near-interface grid points

As seen in Figure 5.6 and Figure 5.10, the harmonic mean for viscosity leads to a faster convergence of the interfacial stresses and averaged velocities. However, for the interfacial stress, a much more important improvement comes from excluding a few grid points closest to the interface from the spline interpolation. The spline interpolation based on the remainder of the grid points is simply extrapolated to the interface, assuming a zero second derivative. This simple act filters out much of the unphysical behavior near the interface. Further from the interface there are no difficulties with mixed cell viscosities and velocity field discontinuities, so we can trust the solution better in this region and base the interfacial stress on the velocity profile slope here. This principle is shown in Figure 5.5, where comparing plots **(a)** and **(b)** we can directly (graphically) see the improvement of the gas velocity profile slope at the interface.

The choice is left of how many points to leave out of the analysis. In the schematic of Figure 5.7 we have one grid point (center of a grid cell) on each side of the interface which is impacted directly by the questionable stress $\tau_{j+1/2}$. Therefore it would seem prudent to leave out the two velocities u_j and u_{j+1}, i.e. the velocities of the two grid points closest to the interface.

One empirical argument for this choice is the relatively quick convergence of the interfacial stress in Figure 5.6 (where the interfacial stress without exclusion does not seem to converge at all). Furthermore, Figure 5.11 shows that this choice leads to agreement between the stress calculated in cases where the interface position is almost exactly the same, but just on the other side of a grid point or grid cell boundary. This is at should be; the stress should not depend on the locations of grid points or cell boundaries. Additionally, Figure 5.8 shows that excluding points reduces the disagreement between the interfacial stress calculated from the liquid side and the same thing calculated from the gas side of the interface.



(a) Arithmetic mean, $h_{\text{int}} = 0.297H$.

(b) Harmonic mean, $h_{\text{int}} = 0.297H$.

(c) Arithmetic mean, $h_{\text{int}} = 0.29675H$.

(d) Harmonic mean, $h_{\text{int}} = 0.29675H$.

Figure 5.8: Shown is the stress calculated from the velocity profile just below the interface (liquid), and from the velocity profile just above the interface (gas). In a perfect simulation these should be the same, and normally they are averaged to yield a single value for the interfacial stress. Plotted are cases with the interface located just below a grid cell boundary ($h_{\text{int}} = 0.29675H$) and just above a grid cell boundary ($h_{\text{int}} = 0.297H$). The grid cell boundary is located at $h = 0.296875H$.

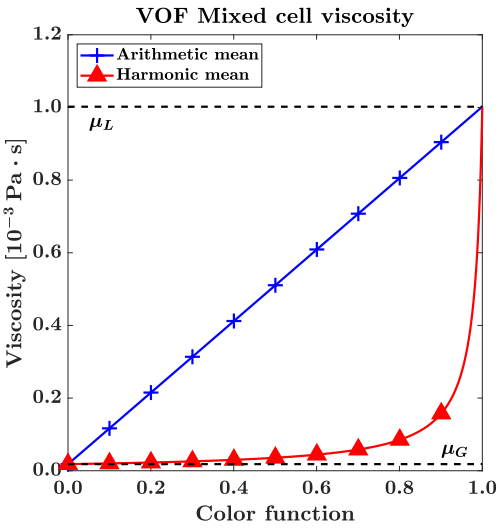
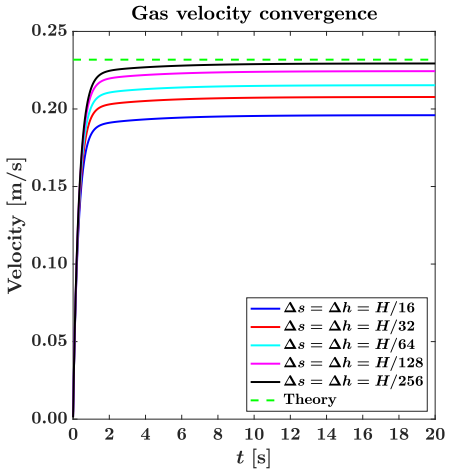
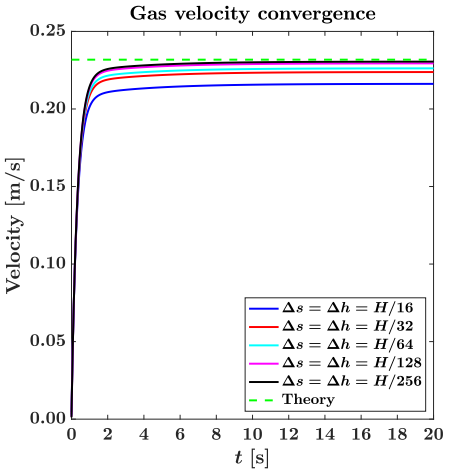


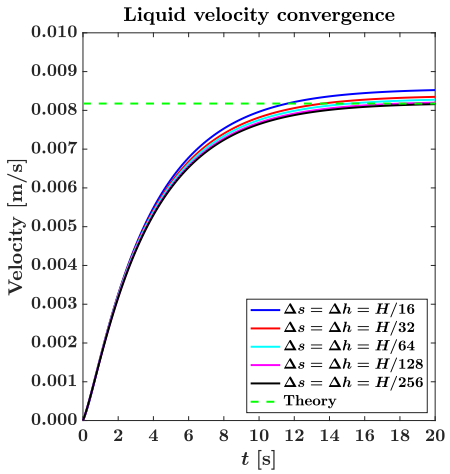
Figure 5.9: A comparison of the arithmetic and harmonic mean for the calculation of the mixed cell viscosity, with $\mu_L = 1.002 \cdot 10^{-3}$ and $\mu_G = 1.82 \cdot 10^{-5}$.



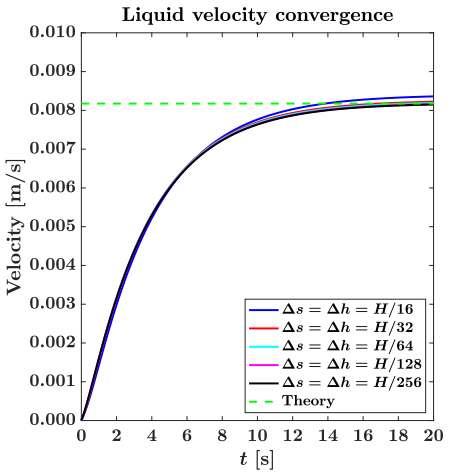
(a) With arithmetic mean for viscosity.



(b) With harmonic mean for viscosity.

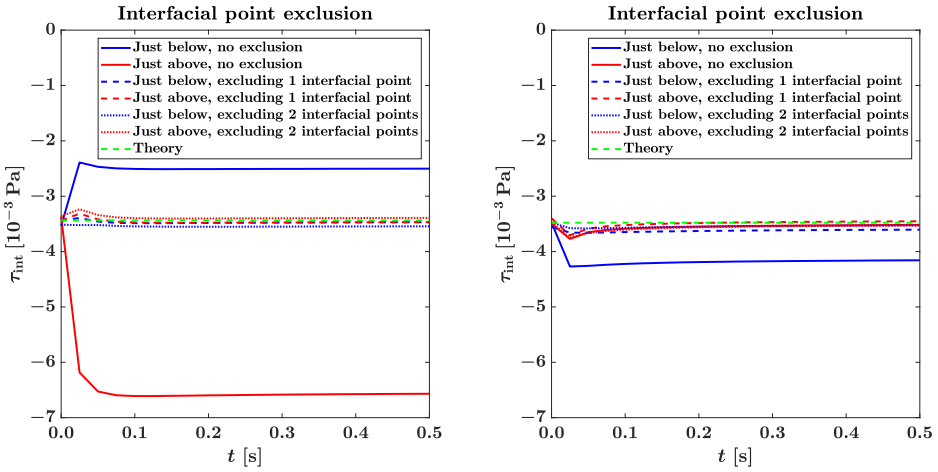


(c) With arithmetic mean for viscosity.



(d) With harmonic mean for viscosity.

Figure 5.10: This figure shows the convergence of the cross-sectionally averaged velocities, with time and grid resolution.



(a) 'Just below' and 'Just above' refer to the position of the interface relative to nearest grid *point*.

(b) 'Just below' and 'Just above' refer to the position of the interface relative to the nearest grid *cell boundary*.

Figure 5.11: The interfacial stress error is dependent on where the interface is located relative to the grid points. Shown in **(a)** are cases with the interface located just below a grid point ($h_{\text{int}} = 0.3046H$) and just above a grid point ($h_{\text{int}} = 0.3048H$). This grid point is located at $h = 0.3046875H$; the resolution is $\Delta s = \Delta h = H/64$. Shown in **(b)** are cases with the interface located just below a grid cell boundary ($h_{\text{int}} = 0.29675H$) and just above a grid cell boundary ($h_{\text{int}} = 0.297H$). The grid cell boundary is located at $h = 0.296875H$.

5.5 Wavy interface validation

We have validated Gerris for the flat interface case, by using theoretical solutions for its steady state. For the wavy case we do not have a theoretical solution. Although a linearized evolution in time of the flow may be derived for the case of infinitesimal perturbations using the Orr-Sommerfeld equation, as performed for example by Bagué et al. [69]. This is outside the scope of this thesis. Instead, we compare the results of Gerris and Rosa simulations.

5.5.1 Short and long wavelengths

We conduct simulations with the same parameters as used previously, given in Table 5.1. But we superimpose a perturbation on the interface of $\Delta h_{\text{int}} = 0.01H$. We also initialize the simulations in the same way, starting from a zero velocity field. The theoretical closure terms (3.84) are again used as closure terms in the Rosa simulations. We initially perform simulations with the same dimensions for the domain: $L = H$. Figure 5.12 shows the interface perturbation in the whole domain, at different time steps.

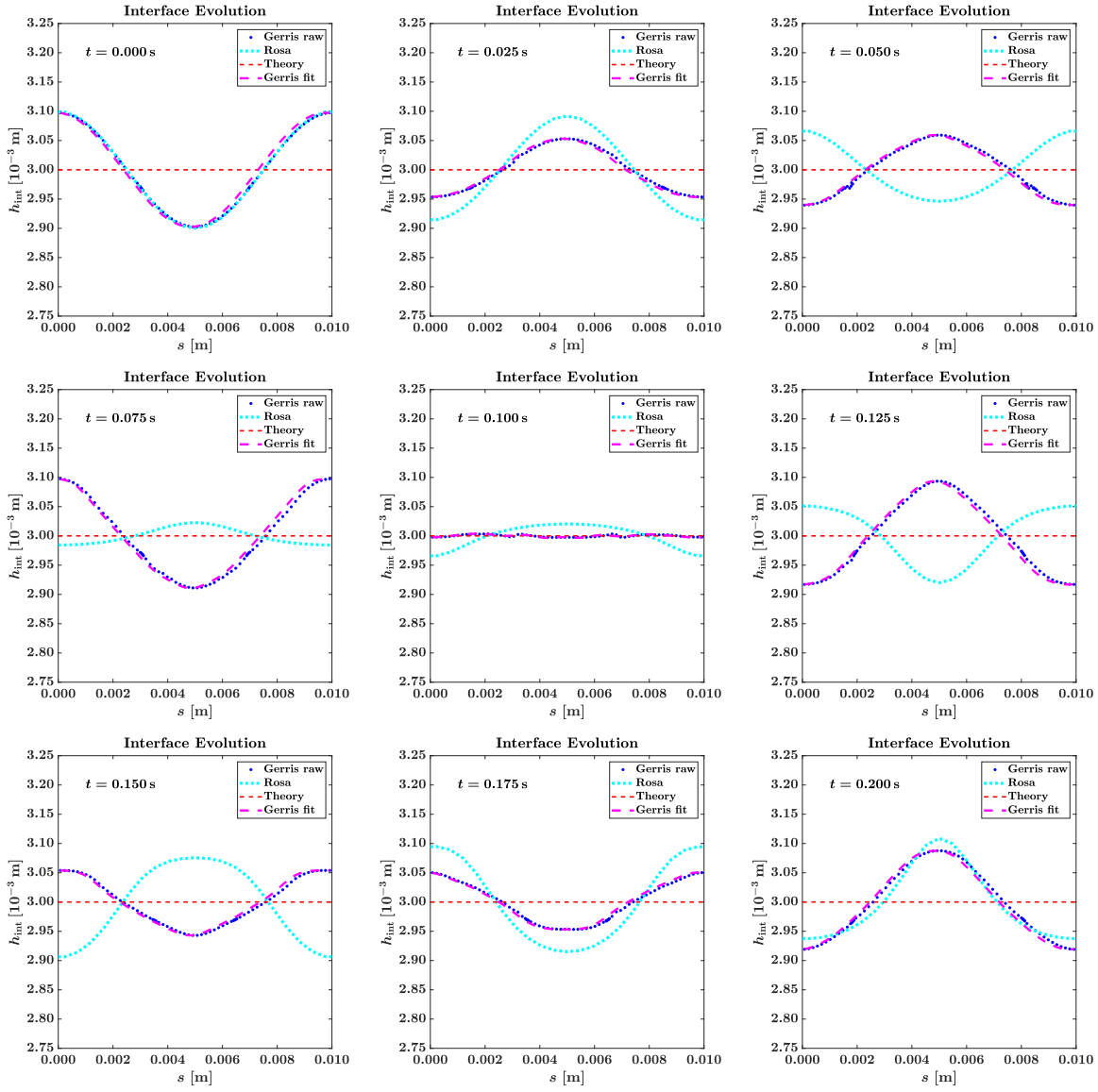


Figure 5.12: Evolution in time of the interface between liquid and gas throughout the domain, zoomed in ($H = 0.01\,\text{m}$). Each plot shows a different time step. Gerris resolution is $\Delta s = \Delta h = H/64$, Rosa resolution and time step are $\Delta s = H/21$ and $\Delta t = 0.001\,\text{s}$ respectively. The domain length is $L = H$.

The wave clearly oscillates at a much higher frequency in the Rosa simulations. This can be explained using the inviscid stability analysis discussed in section 4.4 and section 4.5, as a first approximation, valid when viscous effects are small. Viscous effects are not necessarily small in our low Reynolds number flow, but it still gives an impression. Figure 5.13 provides plots of the dispersion relations analogous to Figure 4.2, but with the parameters of the current test case. Here we see that at small wavelengths $\sim 0.01\,\mathrm{m}$, the two-fluid model equations ('1D Bounded') yield real angular frequencies larger in magnitude than those of the '2D Bounded' analysis, thus causing the Rosa wave to lead the Gerris wave in Figure 5.12.

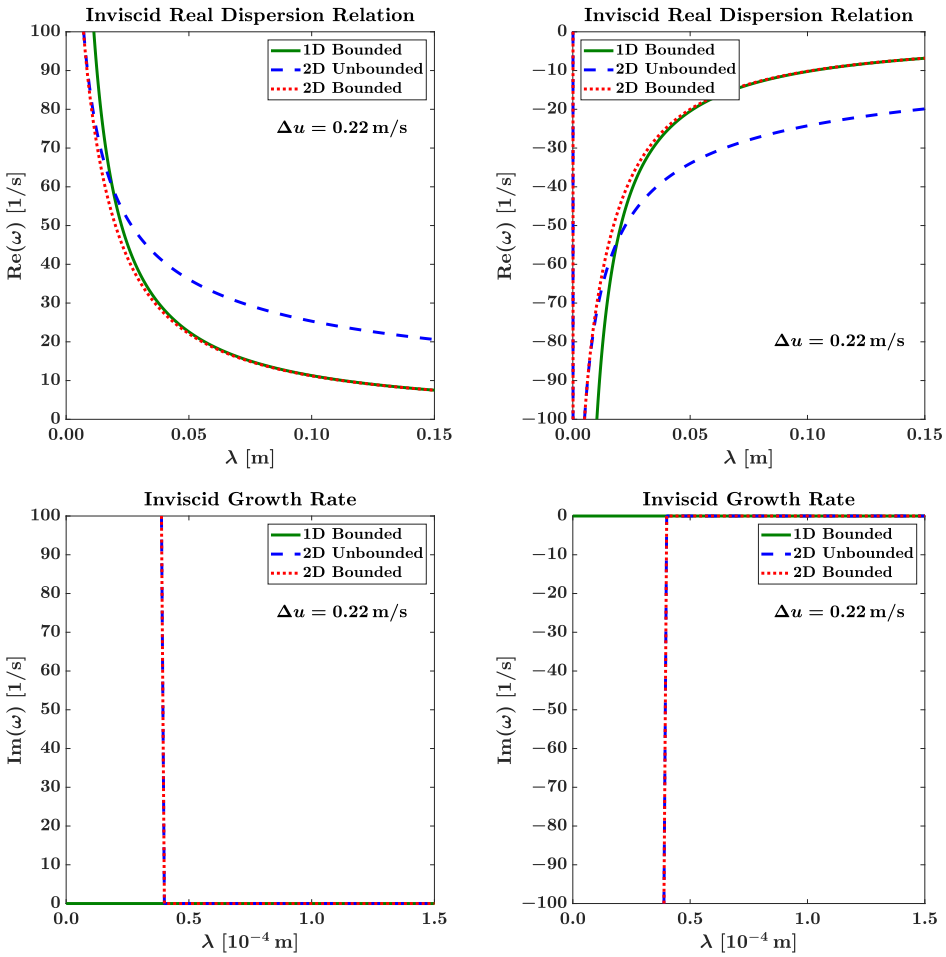


Figure 5.13: Inviscid dispersion relations, using the standard test case parameters of Table 5.1 and uniform base velocities corresponding to the average velocities of Table 5.2. The column on the left shows the angular frequency for one of the two possible waves, the column on the right for the other.

This difference in angular frequencies is also visible in the plot of the interface height at the middle of the domain in Figure 5.14. The stresses oscillate at the same angular frequency as the interface. The Rosa solution eventually breaks apart into two waves, while Gerris continues with just one.

It should be expected that the Gerris and Rosa solutions do not match for the described case. For it was noted in subsection 4.4.2 that there is a long-wavelength assumption (kh_{int}, $kH \to 0$) inherent to the two-fluid model. The two-fluid model should be used to describe flow in long pipelines, with spatial variation occurring across similarly large horizontal length scales. If the wavelength is too short, the angular frequency will be very high and the vertical velocities will be too high[1], which the two-fluid model cannot describe due to the assumption of hydrostatic balance. The vertical velocities needs to be small compared to the horizontal velocities, and the same applies to its derivatives to s, h, and t[2]. Thus the two-fluid model can only describe dynamics along the horizontal direction, not along the vertical direction.

[1]Since the material derivative of the density (2.9) is zero, no particles traverse the interface and there must be particles with the same velocity as the interface. Recall also the interface condition (2.24).

[2]This is necessary to be allowed to write (3.14) and to substitute (3.15) and (3.16) into (3.17) and (3.18).

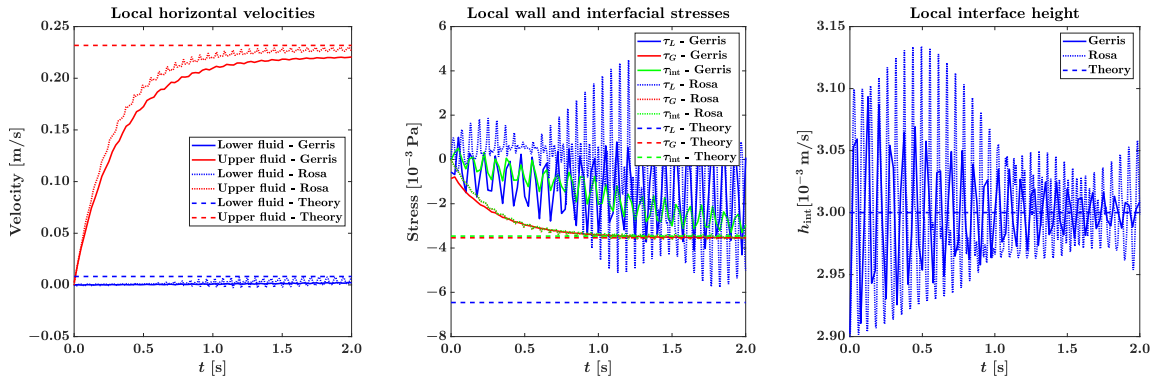


Figure 5.14: Simulations with the parameters of Table 5.1, started from a zero velocity field, but with a perturbation on the interface. The Rosa simulations use the theoretical closure terms. Gerris resolution is $\Delta s = \Delta h = H/64$, Rosa resolution and time step are $\Delta s = H/21$ and $\Delta t = 0.001\,\mathrm{s}$ respectively. The length of the domain, and the wavelength, is equal to the height: $L = H$.

It is not a fundamental problem that the horizontal velocity varies along the vertical direction[3] (as can be seen in Figure 5.1), as long as its averaged value is much larger than the vertical velocity. Increasing the wavelength changes nothing for the averaged velocity, but does ensure that the vertical velocities vanish.

We therefore turn our attention to perturbations with longer wavelengths. Figure 5.15 shows frames of a simulation with a domain length of $L = 12H$ and an equal wavelength. The remaining parameters are the same. It is directly seen that the Rosa results remain in tune with the Gerris results for much longer. The Rosa wave still has a higher frequency than the Gerris wave, but the difference is much smaller, on the time scale of an oscillation: the Rosa wave remains close to the Gerris wave for two full oscillation periods.

Figure 5.16 shows the results for u_L, u_G, h_{int}, τ_L, τ_G, τ_{int} at the middle of the domain, for these simulations with $L = 12H$. The agreement between Rosa and Gerris is much better than in Figure 5.14. The period of oscillation clearly matches, the same steady states are reached[4], and the evolution to this steady state follows similar patterns. The discrepancies that are present (such as again a backflow related liquid stress as discussed in section 5.3) are similar to those of the flat interface case, which is intuitive if we see the flat interface case as a special case of the wavy case, with infinite wavelength. What is odd is that in Gerris the interface perturbation amplitude decreases monotonically, while in Rosa there is an upswing around $t = 7$. This behavior is strongly dependent on the closure term, as we will see in chapter 7.

5.5.2 Wave dynamics

In the frames of Figure 5.15, we see a standing wave. This is a result of initializing the interface perturbation without velocity profile perturbations corresponding to eigenvectors of this perturbation.

This can again be explained using the inviscid linear stability analysis of subsection 4.4.2. The 2D dispersion relation (in the long wavelength limit equal to the 1D dispersion relation) is (4.55), repeated here:

$$\omega = k\frac{\rho_L u_L + \rho_G u_G \xi_2 \pm \sqrt{\frac{\tanh(kh_{\mathrm{int}})g\cos\phi}{k}(\rho_L + \rho_G\xi_2)(\rho_L - \rho_G) - \rho_L\rho_G\xi_2\left(u_G - u_L\right)^2}}{\rho_L + \rho_G\xi_2}, \tag{5.5}$$

with

$$\xi_2 = \frac{\tanh(kh_{\mathrm{int}})}{\tanh(k(H - h_{\mathrm{int}}))}.$$

If the velocities are zero (which is how we initialize the simulations), the first part of the dispersion relation is zero. Only the second part is nonzero. Since we are in the well-posed regime, the second part

[3] The effect of the variation of the horizontal velocity along h is the existence of stresses and is modeled by the closure terms.

[4] Save for the inaccuracy in Gerris' gas velocity due to the limited resolution: the constraint $\Delta s = \Delta h$ means that these simulations are more expensive than the $L = H$ simulations, at constant Δh.

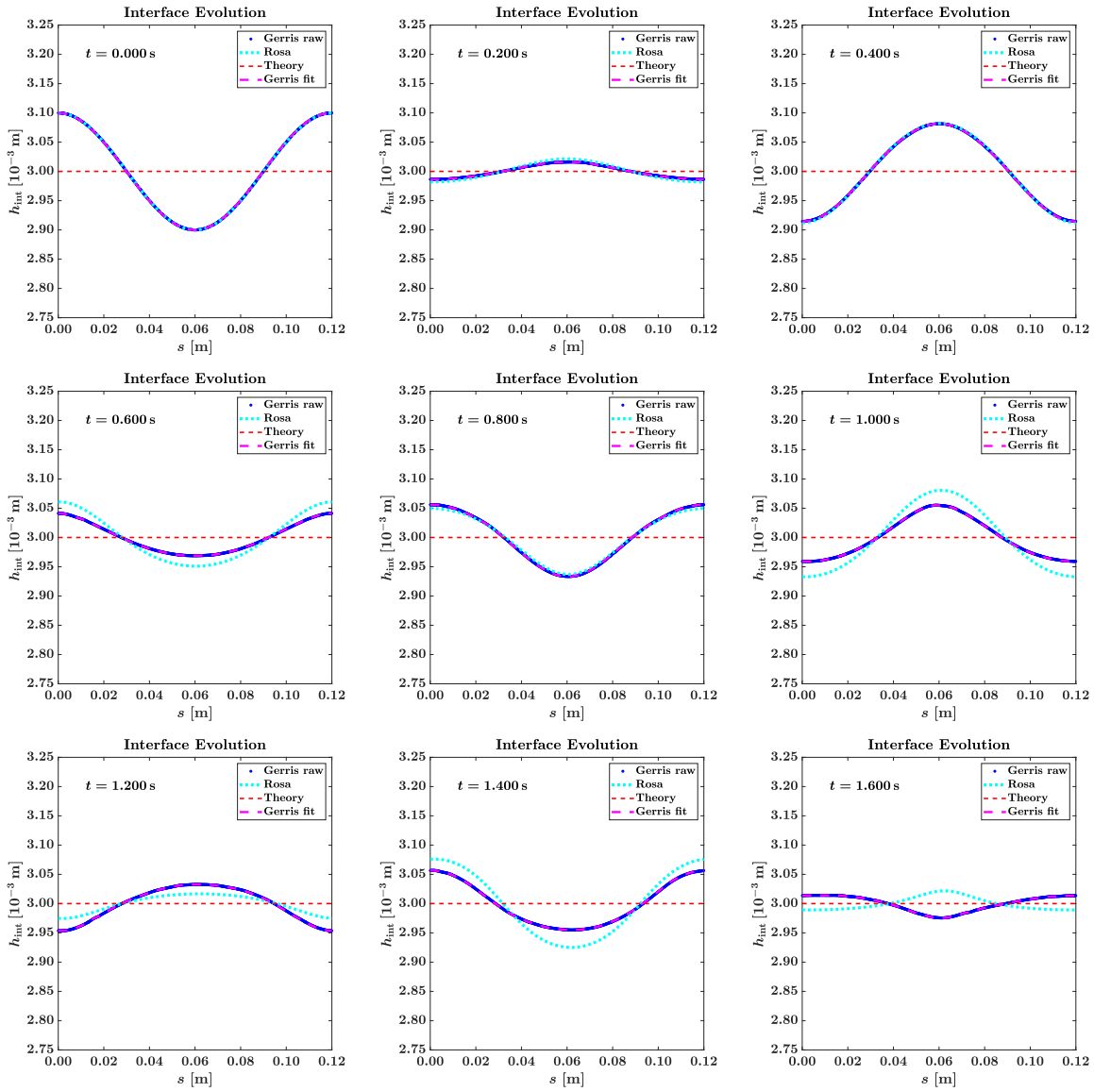


Figure 5.15: Evolution in time of the interface between liquid and gas throughout the domain, zoomed in ($H = 0.01\,\mathrm{m}$). Each plot shows a different time step. Gerris resolution is $\Delta s = \Delta h = H/64$, Rosa resolution and time step are $\Delta s = H/21$ and $\Delta t = 0.001\,\mathrm{s}$ respectively. The domain length is $L = 12H$.

is real. We get a summation of the two waves $\pm$. This forms a standing wave. The disturbance vector (including the perturbations in the interface height and velocities) can be written as (4.38), repeated here:

$$\boldsymbol{\Delta w} = |\boldsymbol{\Delta \hat{w}}_1|\, e^{\mathrm{Im}\{\omega_1\}t} \cos\left(ks - \mathrm{Re}\left\{\omega_1\right\} t + \boldsymbol{\theta}_1\right) + |\boldsymbol{\Delta \hat{w}}_2|\, e^{\mathrm{Im}\{\omega_2\}t} \cos\left(ks - \mathrm{Re}\left\{\omega_2\right\} t + \boldsymbol{\theta}_2\right). \tag{5.6}$$

Since the imaginary parts of ω and $\Delta\hat{h}_{\mathrm{int}}$ are zero, and as a result the imaginary parts of the velocity amplitudes ((4.35) for 1D, (4.61) for 2D) and thus the angles θ_1 and θ_2 are zero if we start from a cosine shaped interface disturbance, this is simplified to

$$\boldsymbol{\Delta w} = \boldsymbol{\Delta \hat{w}}_1 \cos\left(ks - \omega_1 t\right) + \boldsymbol{\Delta \hat{w}}_2 \cos\left(ks - \omega_2 t\right). \tag{5.7}$$

And since $\omega_1 = -\omega_2 = \omega$

$$\boldsymbol{\Delta w} = \boldsymbol{\Delta \hat{w}}_1 \cos\left(ks - \omega t\right) + \boldsymbol{\Delta \hat{w}}_2 \cos\left(ks + \omega t\right). \tag{5.8}$$

We can define $\boldsymbol{\Delta \hat{w}}_1 = \boldsymbol{\Delta \hat{w}}_2$, by splitting $\Delta\hat{h}_{\mathrm{int}}$ into to equal parts ($\Delta\hat{h}_{\mathrm{int},1} = \Delta\hat{h}_{\mathrm{int},2} = \Delta\hat{h}_{\mathrm{int}}/2$). This is possible because the initial velocity and pressure perturbations are zero, and we only have one height

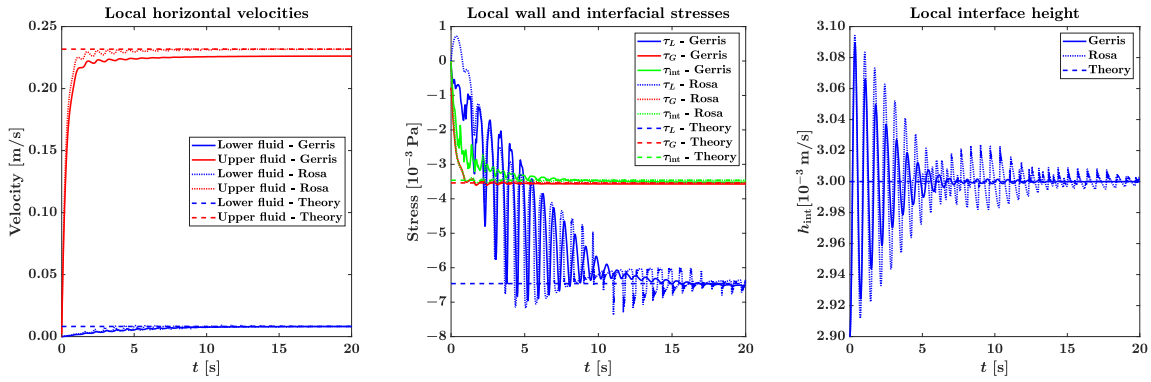


Figure 5.16: Simulations with the parameters of Table 5.1, started from a zero velocity field, but with a perturbation on the interface. The Rosa simulations use the theoretical closure terms. Gerris resolution is $\Delta s = \Delta h = H/64$, Rosa resolution and time step are $\Delta s = H/21$ and $\Delta t = 0.001\,\mathrm{s}$ respectively. The length of the domain, and the wavelength, is equal to twelve times the height: $L = 12H$.

perturbation, which might just as well result in one or the other wave. In this case the above simplifies to

$$\begin{aligned}\boldsymbol{\Delta}\mathbf{w} &= \boldsymbol{\Delta}\hat{\mathbf{w}}\left[\cos\left(ks-\omega t\right)+\cos\left(ks+\omega t\right)\right] &(5.9)\\ &= \boldsymbol{\Delta}\hat{\mathbf{w}}\left[\frac{\exp\left(iks\right)\exp\left(-i\omega t\right)+\exp\left(-iks\right)\exp\left(i\omega t\right)}{2}+\frac{\exp\left(iks\right)\exp\left(i\omega t\right)+\exp\left(-iks\right)\exp\left(-i\omega t\right)}{2}\right]\\ &= 2\boldsymbol{\Delta}\hat{\mathbf{w}}\cos(ks)\cos(\omega t). &(5.10)\end{aligned}$$

This is a wave with a spatial structure which does not change from the beginning cosine. The amplitude is amplified or damped by the time dependent $\cos(\omega t)$.

If there are small liquid and gas velocities so that the first part of the dispersion relation (5.5) is nonzero, the two ω will not be the same in magnitude but will differ slightly:

$$\omega_1 = \omega_0 + \delta\omega, \quad \omega_2 = -\omega_0 + \delta\omega,$$

where ω_0 is the dominant part, corresponding to the square root term in (5.5). In this case the result is a bit different:

$$\boldsymbol{\Delta}\mathbf{w} = 2\boldsymbol{\Delta}\hat{\mathbf{w}}\cos(ks-\delta\omega t)\cos(\omega_0 t). \tag{5.11}$$

This is a traveling wave with wave speed $c = \delta\omega/k$ in which $\delta\omega$ is half the difference between ω_1 and ω_2. So the traveling speed is slow compared to the amplitude modulation by $\cos(\omega_0 t)$. The spatial structure does not change though, the wavelength should remain $2\pi/k$. This behavior is indeed what we see in Gerris, after the velocities have had some time to grow, or when initializing the simulations from a state with nonzero velocities.

In practical pipeline applications, we will usually have higher pipe diameters and higher velocities than those given in Table 5.1 and Table 5.2. Via a larger liquid Froude number (very low for our test case) this will result in a higher $\delta\omega$ than for our test case, and thus waves with larger traveling velocities $\delta\omega/k$, compared to their 'standing wave frequency' ω_0. Furthermore, if the simulation is initialized with velocity perturbations corresponding to either ω_1 or ω_2, we will get only one of the two waves, which will not oscillate but only travel.

If we somehow had two standing (or slowly traveling) waves of the form (5.10), with slightly different ω:

$$\begin{aligned}\boldsymbol{\Delta}\mathbf{w} &= \boldsymbol{\Delta}\hat{\mathbf{w}}\cos(ks)\cos(\omega_1 t)+\boldsymbol{\Delta}\hat{\mathbf{w}}\cos(ks)\cos(\omega_2 t) &(5.12)\\ &= 2\boldsymbol{\Delta}\hat{\mathbf{w}}\cos(ks)\left[\frac{\exp(i\omega_1 t)+\exp(-i\omega_1 t)}{2}+\frac{\exp(i\omega_2 t)+\exp(-i\omega_2 t)}{2}\right]\\ &= 2\boldsymbol{\Delta}\hat{\mathbf{w}}\cos(ks)\cos\left(\frac{\omega_1+\omega_2}{2}t\right)\cos\left(\frac{\omega_1-\omega_2}{2}t\right), &(5.13)\end{aligned}$$

we get the phenomenon referred to as a 'beat' (in time) with beat angular frequency $\frac{\omega_1-\omega_2}{2}$. This might explain the resurgence of the interface oscillation in the Rosa simulations seen in Figure 5.16 from $t = 7$.

Of course the inviscid stability analysis cannot explain everything in these viscous simulations. One aspect that it cannot explain is the damping of the perturbations, despite the absence of imaginary components in the angular frequencies. This is an obvious result of viscosity in the simulations, along with the forming of the velocity profiles and steady states.

5.5.3 Interfacial stress problems

Despite the methodology developed for extracting stresses from Gerris simulations accurately (discussed in section 5.4), problems still appear when conducting wavy simulations. The three stresses as a function of s are shown in Figure 5.17, and the main problem is shown in Figure 5.18. At certain points in the domain, there are sharp discontinuities in the interfacial stress. This is likely to be at points where the interfaces traverses a grid cell boundary or grid cell center point, as we have seen in Figure 5.11 that significantly different values for the interfacial stress are obtained depending on which side of such a location the interface finds itself on. The effect was reduced by excluding interfacial points in the spline interpolation, but not eradicated. Luckily, such extreme situations are rare (as can be seen in Figure 5.17), and it is our hope that a neural network might filter out this 'noise' and discover the pattern with physical meaning.

Even without these interfacial stress discontinuities, the agreement between the Gerris and Rosa simulations is poor. However this might be expected, since in the given simulations Rosa employs closure terms derived for the fully developed steady state. It is our hope that a neural network might learn the stresses for these unsteady states and thus improve upon the steady state closure terms.

5.5.4 Grid resolution

In section 5.3, we have conducted Gerris simulations with a flat interface, with different spatial resolutions. From the results we can select a vertical spatial resolution Δh, which sufficiently resolves the velocity profiles and interface (after which Gerris automatically selects a time step to keep the Courant number below 0.8). A good resolution is $\Delta h = H/128$, a sufficient resolution when computational resources are limited is $\Delta h = H/64$. The grid cells in 2D Gerris are square; $\Delta s = \Delta h$, and so our resolution in both directions is determined.

However, for Rosa the flat interface case does not help us to select a resolution: as we have noted the results are identical for different spatial resolutions (keeping the Courant number constant), down to machine precision. This is because of the nature of the equations Rosa solves; there are no derivatives with respect to the vertical coordinate, only with respect to the horizontal coordinate. When the flow starts out fully developed, and the boundary conditions are periodic, there is no source of variation along s. This could only arise from instability of a numerical perturbation, but this is not observed (and not predicted by linear stability analysis). Because the flow is constantly fully developed, there are no spatial derivatives, and any resolution will be sufficient to exactly compute these zero derivatives.

Therefore for evaluation of the convergence of Rosa, we consider the wavy case, in which horizontal spatial derivatives do exist. A convergence study of Rosa has been conducted with the long domain case ($L = 12H$), starting from a zero velocity field. We have analyzed the convergence at constant Courant number $\sim \Delta t/\Delta s$ of our variables of interest h_{int}, u_L, u_G, τ_L, τ_G, τ_{int}, at the center of the domain ($s = 0.06$), at time $t = 5$. Some results are shown in Figure 5.19.

Figure 5.19a seems to show significant error for simulations with fewer time steps per second. However, from Figure 5.19b we can deduce that this is solely the result of the increasing spatial resolution associated with increasing the number of time steps at constant Courant number. For in Figure 5.19b, we see almost no difference between simulations with the same number of cells per diameter (i.e. H) and different Courant numbers, i.e. number of time steps. This indicates that at the analyzed temporal resolutions the simulations are easily converged with time. We may expect this as we are using a third-order Runge-Kutta method (see section 3.6).

Only the line with the highest Courant number in Figure 5.19b starts to diverge slightly from the others. On this basis we choose the highest shown ratio $\Delta t/\Delta s = 2.13$ for which this does not occur, so that the number of time steps is not unnecessarily high but the simulations are still strongly converged with respect to Δt. A supporting argument for this ratio is obtained by computing the Courant number using the steady state gas velocity (much higher than the liquid velocity) given in Table 5.2, which yields

$$C = \frac{u_G \Delta t}{\Delta s} = 0.49, \tag{5.14}$$

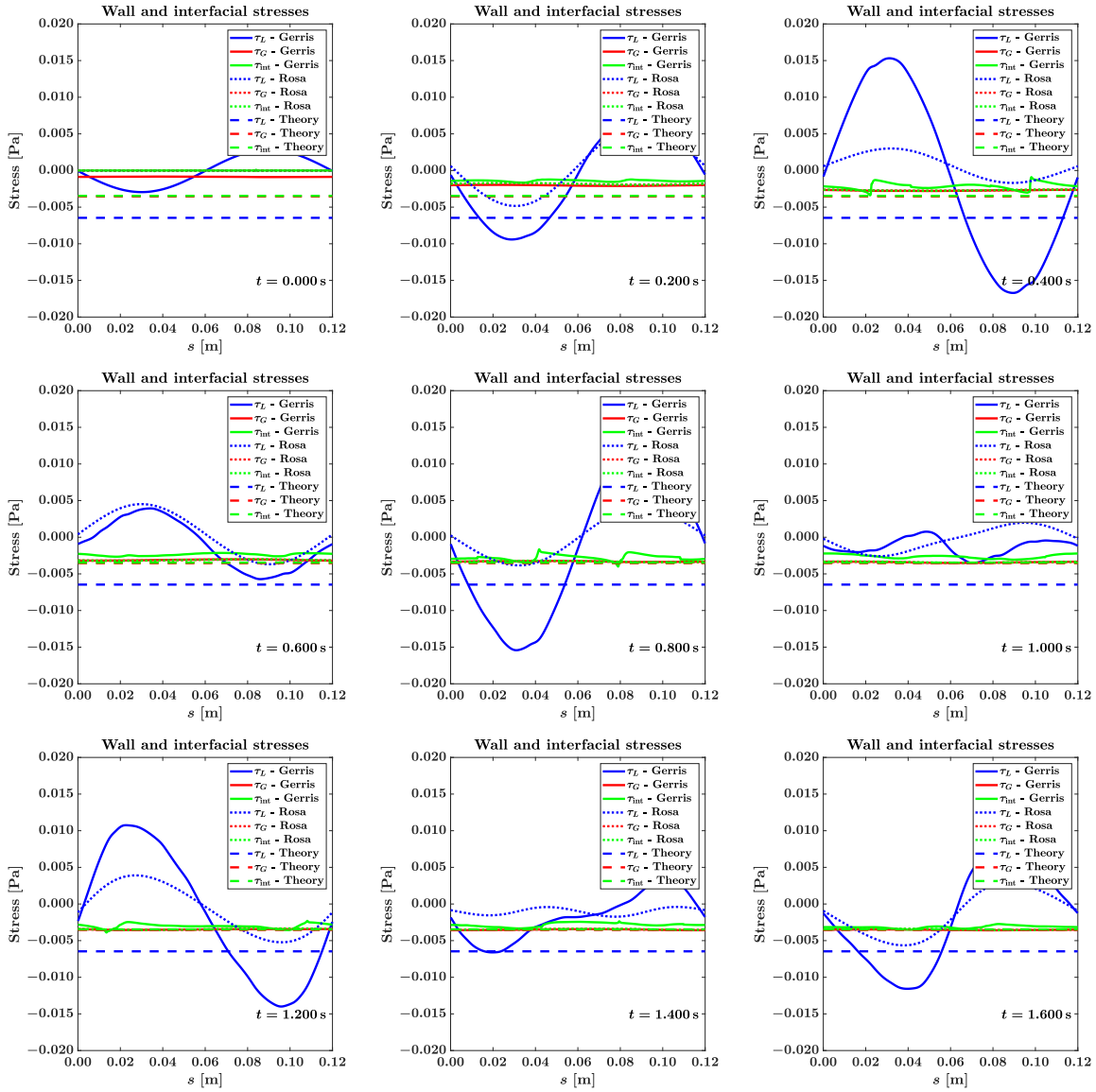


Figure 5.17: Evolution in time of the stresses throughout the domain, for the same simulations as Figure 5.15 and Figure 5.16 ($L = 12H$). Each plot shows a different time step.

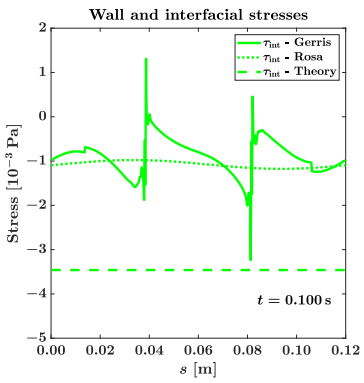


Figure 5.18: Interfacial stress throughout the domain, for the same simulations as Figure 5.17. At this time instant there are particularly sharp discontinuities in the Gerris interfacial stress.

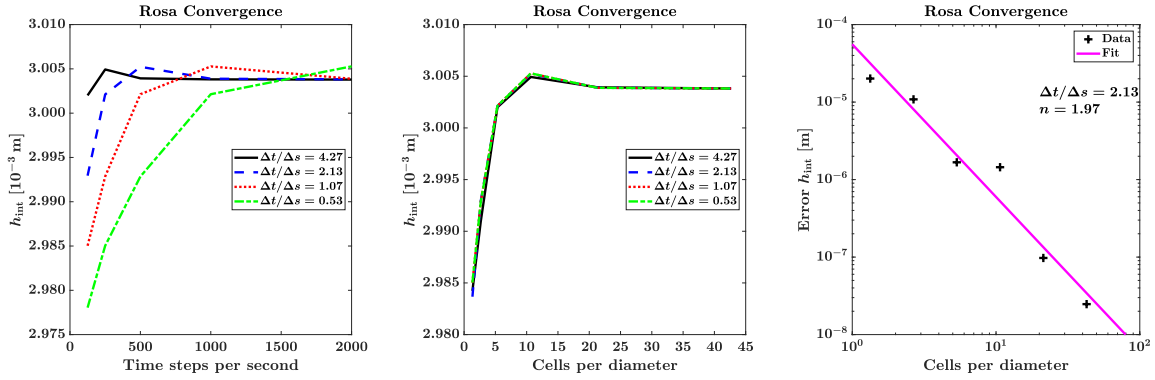


Figure 5.19: The convergence of h_{int} at $s = 0.06$, $t = 5$, for wavy Rosa simulations. Convergence is shown with respect to spatial and temporal resolution, at constant Courant numbers (proportional to $\Delta t/\Delta s$). In **(c)** the absolute error is shown with respect to the simulation with the finest resolution (which itself does not appear in this plot). The line is a fit (on the log-log scale) of Error $= C\Delta s^n$, with n the convergence order.

a number safely below 1 but not unnecessarily low.

For the spatial resolution we choose a point which lies in the flat-looking part on the linear scale of Figure 5.19b and closely below the approximately second order convergence line of Figure 5.19c (to be expected using a finite volume scheme with central interpolation of which equivalence to a central finite difference scheme can be shown). At this point the resolution is $\Delta s = H/21.33 = L/256$, and the time step is $\Delta t = 0.001$s; 1000 time steps per second.

5.6 Conclusion

We have validated the Gerris simulation code by observing the evolution of 2D two-phase channel flow starting from rest to a laminar fully developed steady state which is predicted analytically. The parameters of the simulations lie well in the stable and well-posed regime of the two-fluid model. The agreement with respect to the exact analytical solution converges with time and with grid resolution. The agreement between Rosa and Gerris for these simulations is good for both the steady state, and for the evolution in time of the averaged velocities towards this steady state. On the other hand, the evolution of the stresses proceeds differently for Rosa and Gerris. This exposed a fundamental limitation of closure terms using the conventional cross-sectionally-averaged inputs: there is only a direct relation between the averaged velocities and the velocity profile (and by extension the stresses) when steady states are considered. Therefore these closure terms do not have enough information to determine the correct stresses for unsteady states in general.

Generally, in a simulation we do not expect just any velocity profile to occur, but only certain velocity profiles which lie between the initial condition and the steady state (if perturbations are damped). A neural network might be able to find the pattern of velocity profiles which do occur, given specific simulation conditions. Additionally, closure terms which incorporate more inputs than the conventional closure terms might have enough information to form a direct relation between averaged velocities and velocity profiles (and by extension the stresses). Additional input features are relatively easy to incorporate in a neural network. See chapter 6 and chapter 7.

Extracting the wall stresses from the employed VOF code (Gerris) is rather straightforward, but extracting the interfacial stresses is not. We base the stresses on spline interpolations of the velocity profile. In a VOF code the interface is not sharp; rather there exist cells with intermediate values of the viscosity and velocity gradient. We employ the harmonic mean for the viscosity to sharpen the interface, and disregard one cell on each side of the interface in our velocity profile interpolation, favoring cells without intermediate viscosities and velocity gradients. This improves results, but we do not reach full agreement between the interfacial stress predicted from the liquid or gas velocity profiles, or from interface placements just below or above a grid cell boundary or grid cell center, indicating that numerical problems remain.

We compare unsteady Gerris and Rosa simulations with wavy interfaces, and confirm that the two-fluid model is not suitable for cases with short-wavelength perturbations. In this case the dispersion relations significantly differ, and this case violates the assumption of hydrostatic balance. The shortcoming of the two-fluid model that it cannot model dynamics along the vertical direction cannot be remedied simply by closure terms. The hydrostatic balance is a fundamental assumption of the two-fluid model. Addition of a more advanced vertical momentum equation (with additional closure terms) is probably necessary to resolve dynamics in the vertical direction. Closure terms of the form considered can only serve to model the profile of the horizontal velocity and the associated horizontal stresses.

With a longer wavelength perturbation, the agreement between Gerris and Rosa improves significantly. We observe slowly traveling 'standing' waves which are formed as a superposition of two opposing waves, each with one of two possible angular frequencies. In wavy simulations, the extremes of the deficiencies of our calculation of the interfacial stress are visible (e.g. in Figure 5.18). But these extremes are rare and may be filtered out by a regularizing neural network, that will be studied in the following chapter. The computed stresses in our 2D simulations differ from the stresses predicted for the same conditions by closure terms based on the steady state. Thus closure terms learned for unsteady states have the potential to improve low fidelity model results.

Based on Gerris and Rosa results computed at different grid resolutions and time steps, and considering the available computational resources, we make a choice of resolution of $\Delta s = \Delta h = H/64$ for Gerris with a Courant number of 0.8, and $\Delta s = H/21.33$ with 1000 time steps per second for Rosa.

With these results we can have some confidence in our simulations and analytical calculations and proceed to use them to train neural networks in the coming chapters.

Chapter 6

Neural Networks

6.1 Introduction

Artificial neural networks are called neural networks because they are initially based on a model for neurons in the brain, proposed by Rosenblatt (1958) [16]. Such a neuron can be represented as a basic computational unit, depicted in Figure 6.1. It takes an arbitrary number of inputs x_i, multiplies them by their weights w_i, and sums them together. Usually also a neuron-specific bias b is added to the sum of the inputs. A nonlinear activation function is applied to the output (a common choice is $\tanh(x)$). The output is then passed on, as an input to other neurons.

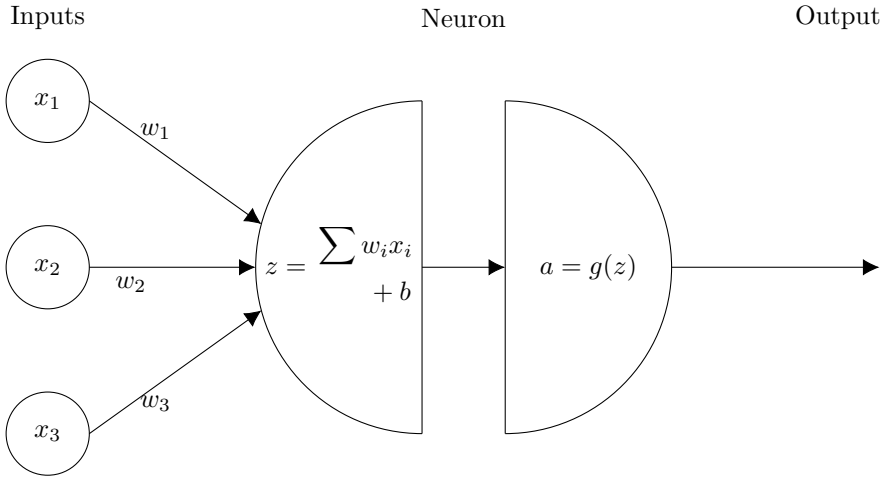


Figure 6.1: A schematic of a single perceptron, or neuron.

The power and versatility of neural networks stem from the possibility of large and diverse combinations of neurons into networks. Taken apart, the neuron is a very simple model, but taken together larger networks can form complex relations between inputs and outputs. A typical structure of a neural network is shown in Figure 6.2. However, this is just one possibility, and the choices available to the analyst in the structure of the network and choice of activation functions mean that the framework can be adjusted to the needs of different types of problems (e.g. convolutional neural networks for image classification problems). Due to these freedoms the neural network framework can be shown to be a generalization of a number of different machine learning methods [18]. It is this power to make predictions (or choices) for virtually any kind of problem, together with the iterative data-based learning algorithm, which justifies the comparison of this computational graph to the human brain.

In this chapter we will discuss various aspects of neural networks and the associated choices to be made in the definition and training of a neural network. We will evaluate the effect of these choices by training networks on wall and interfacial stress data for the case of flat interface, steady, fully developed, laminar 2D channel flow. This corresponds to the connections [3] and [4] in the project flow chart Figure 1.4 shown in the introduction. However, since it has been established in chapter 5 that the Gerris simulations

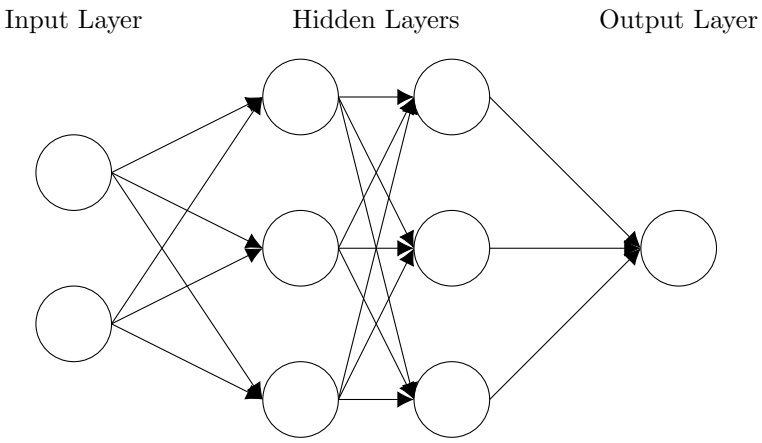


Figure 6.2: A schematic of a multi-layer perceptron neural network. Its nodes are fully connected, in a feed-forward fashion.

converge to the theoretical steady state (equations (3.80)–(3.83)), we can use analytical steady state data to train the networks, instead of Gerris simulation data. This is advantageous because this saves costly simulation time.

In this project we use the MATLAB shallow neural network implementation, included in the Deep Learning Toolbox [140].

6.2 Learning algorithm

The goal of the neural network is to use known data to create a predictor (or merely an interpolator) for unseen cases. In supervised training this is done by getting the neural network to predict the known data well. The measure of the difference between the neural network prediction and the known output, for a given set of inputs, is called the cost function (or loss- or performance function). Here, we use the mean squared error cost function

$$C = \frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2, \tag{6.1}$$

where y_i is the data for a set of input variables i and $\widehat{y}_i$ is the model prediction for these inputs. This cost function is nice and smooth and likens our optimization objective to that of linear regression. In this chapter we will often refer to the value of this cost function for a given data set and set of network predictions as an 'error'.

The training objective of a neural network can be formulated as minimizing the cost function over the training data. This is done by adjusting the weights and biases in the network. They are adjusted iteratively; the basic algorithm for this is gradient descent. One step in the optimization consists of a modification to the weight and bias vector $\mathbf{w}$ of

$$\mathbf{w} \leftarrow \mathbf{w} - \lambda\frac{\partial C}{\partial \mathbf{w}}, \tag{6.2}$$

where λ is a parameter called the learning rate which dictates the step size of the optimization. For this step we need the derivatives of the cost function to all the weights and biases in the network. These can be calculated efficiently using the backpropagation algorithm [17], which basically iteratively calculates the derivative of the cost function to activations further and further from the end of network, using the derivative to the activation one step closer to the end, and the chain rule.

The backpropagation algorithm runs the risk of encountering the vanishing or exploding gradient problem. Because the gradient of the cost function to weights earlier in the network is calculated using the multiplication of the gradients further in the network, it might 'vanish' if all these gradients are smaller than 1 or 'explode' if they are all larger than 1. This means that the training becomes respectively insensitive or oversensitive to weights earlier in the network, which in turn makes the network hard to train. This problem depends on the choice of activation functions; see section 6.4.

In this work, we use the Levenberg-Marquardt neural network training algorithm [20], [141]. This is a second-order algorithm, meaning that it makes use not only of the first derivative, but also of an approximation to the second derivative of the cost function to the weights. It combines the Gauss-Newton method (basically an approximation to Newton's method) and the gradient descent method. Initially, the algorithm takes a step similar close to the step gradient descent would take. As the optimization moves closer towards a minimum or saddle point in the loss landscape the algorithm takes steps closer to that of the Gauss-Newton method. This helps the algorithm to move into the minimum much more quickly than a purely first order method would. The ratio between the gradient descent step and the Gauss-Newton step is determined by a parameter μ, similar to the learning rate in (6.2). Only an initial value needs to be set, the algorithm handles the subsequent adaptation of this important parameter.

The Levenberg-Marquardt method is particularly efficient for smaller networks, where its additional computational complexity is not inhibitive.

The training is concluded when either

- A goal for the cost function is reached.
- The norm of the gradient of of the cost function to the weights falls below a specified value.
- The maximum number of epochs is reached.
- The algorithm, adjusting μ up to some maximum value, fails to perform a step which reduces the value of the cost function.

For the hyper-parameters involved, we use the MATLAB defaults [142].

6.3 Network structure

The starting point for a choice of neural network structure is the fully connected, feed-forward structure of Figure 6.2. This structure is very general: it encompasses many different network structures. If more is known about the desired structure of the relation between input and output, the network structure can be altered to force this structure. For example, connections can be removed and the weights of certain connections can be forced to match in order to form a convolutional neural network [143], [144]. In this way a convolutional neural network incorporates spatial structure in the data.

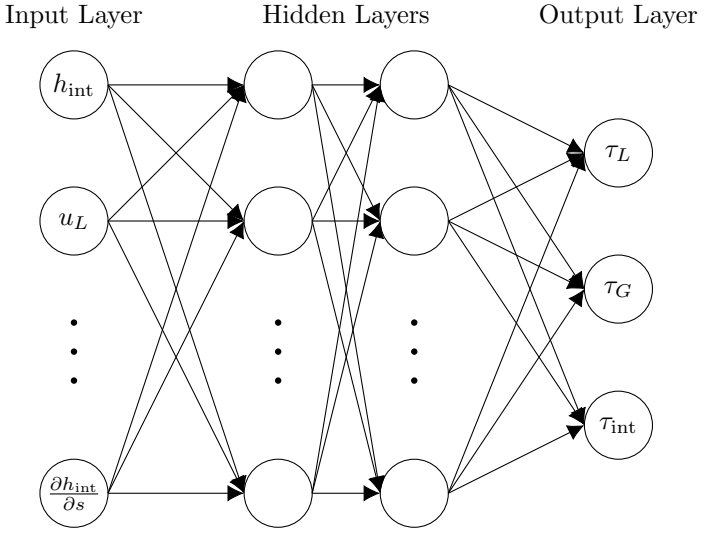


Figure 6.3: A schematic of the neural network structure as applied in this work. The number of hidden nodes and layers are still to be specified. The inputs and outputs are as given in (6.3).

As a first pass we do not specialize the structure of the network as it is a priori unclear what kind of special structure would be beneficial for our application. We are fitting functions,

$$\tau_L, \tau_G, \tau_{\text{int}} = f(h_{\text{int}}, u_L, u_G, \rho_L, \rho_G, \mu_L, \mu_G, H, \frac{\partial h_{\text{int}}}{\partial s}), \tag{6.3}$$

without regard for spatial or temporal correlations[1]. The stresses are only functions of the averaged variables at a single location, and are independent of the forcing ($\partial p/\partial s$, g) as is customary for friction closure relations and is necessary for generalization to flows with different types of forcing and boundary conditions (this was discussed in subsection 3.4.3 and subsection 3.4.4 and e.g. [145]). For this kind of function fitting, where the inputs represent very different things, and input and output have a range which is in principle unbounded, this straightforward structure may be the most appropriate.

We do break somewhat from the structure of conventional closure terms by including the slope of the interface as an input. However this term can also be defined locally and is not directly related to forcing or boundary conditions. This is necessary in order to take into account the influence of wavy interfaces on the interfacial stress, discussed in literature and in subsection 3.4.4. This term will not come into play in this chapter where we only consider steady, fully developed flow. In chapter 7 it will make an important difference.

The interfacial slope can be added easily to the neural network as an input without any knowledge of the relation between it and the stresses. This term is not possible to include in conventional closure terms which are calculated for the fully developed steady state, since it is a locally defined variable. If fully developed flow is assumed, or similarly the effect of the wavy interface is averaged out over a length of pipe (as is done by e.g. Andritosos and Hanratty [107]), the average interface slope will be zero (for a flow with a wavy perturbation) and cannot be used to differentiate stresses at different phases of the wavy perturbation.

We *can* conclude from our knowledge of conventional closure terms (subsection 3.4.4) that the different stresses $\tau_L, \tau_G, \tau_{\text{int}}$ typically depend on similar factors. This suggests that we can use a single network with $\tau_L, \tau_G, \tau_{\text{int}}$ as the three different outputs, shown schematically in Figure 6.3. In this way they share the largest part of the network, with the exception of the connections from the last hidden layer to the output layer. However, due to the training the situation may arise that nodes in the later layers are connected to only one of the three stresses.

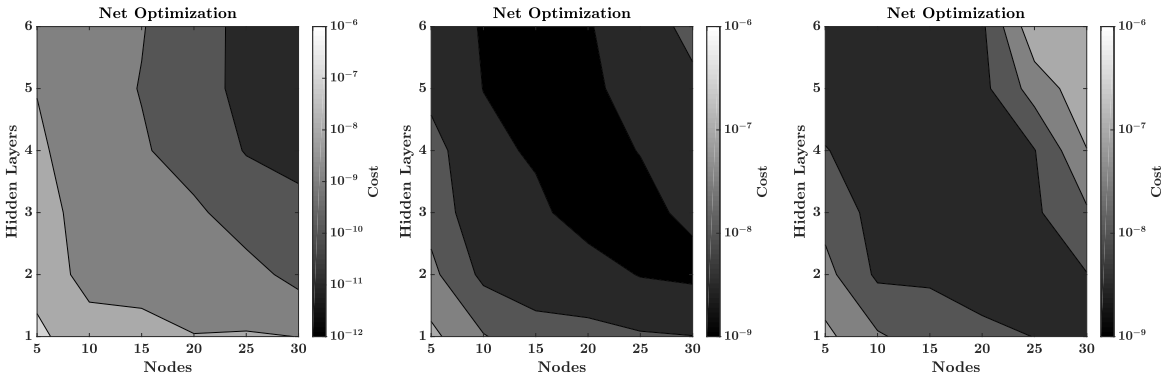
This means that the first part of the network can be trained on more data and more diverse data, from all three stresses, so that the network will have a larger chance to indeed learn these common factors, with more generally applicability, instead of just over-fitting the data. Training a single network in this way is also more efficient than training three different networks. Moreover, using a single network saves time when it is applied in the low-fidelity model for the calculation of closure terms.

The question remains of how many hidden nodes and layers to include in the network. This is determined by a search of this parameter space. The results are shown in Figure 6.4. Results for the final cost function value (defined in (6.1)), measured on data excluded from the training (so-called 'validation data'), are shown for different network configurations. Three different plots are shown, of which two on the right show results when adding noise to τ_L, τ_G, and τ_{int} in the training data. This is done in order to let the analytically calculated data better represent actual simulated data. Furthermore, adding noise to the training data can also be a strategy for regularization, though this mostly concerns the input variables [146, p. 242].

Without added noise the network performance improves monotonically with the number of degrees of freedom in the network. This is probably just because the data can be overfit more and more with the increase in degrees of freedom. The validation data without noise here is not distinct enough from the training data (see section 6.7 and section 6.8). Adding noise to the data gives rise to an optimal band in the plots of Figure 6.4. We make the remaining trade-off between number of nodes and number of layers on the basis of the following considerations:

- Increasing the amount of layers at the expense of the amount of nodes per layer (i.e. with a fixed total number of nodes) increases the ability of the network to incorporate patterns in the data, as opposed to just 'memorizing' the data plainly and overfitting, and allows networks to approximate the function better with fewer degrees of freedom [147], [148] (as long as the function is complex enough).
- Deeper networks are harder to train [149], [150]. This is because the structure of the *loss landscape*, the value of the cost function as a function of the network parameters, becomes more complex and less convex [151], [152]. This makes it harder to find a good (near-global) minimum during the network training. An other important contributor to the difficulty in training is the exacerbation of

[1]Though in future research it would be interesting to incorporate velocities and interface heights at surrounding grid points and preceding time steps as inputs. This would give the network extra information, particularly on the unsteadiness and waviness of the flow. If this were to be done it would be a good idea to include convolutional structures.



(a) No noise in the data. **(b)** 5% Noise. **(c)** 10% Noise.

Figure 6.4: Plots of the validation cost (6.1) after training of nets with different numbers of nodes and layers. The second and third plots have Gaussian noise added to τ_L, τ_G, and τ_{int} in the training data, with the given percentage of the original data value as the noise standard deviation.

the vanishing gradient problem with increasing network depth (discussed in section 6.2).

We settle at a balance of 4 hidden layers with 18 nodes each, located in the middle of the optimal bands of Figure 6.4 **(b)** and **(c)**.

6.4 Activation function

The choice of activation function is also influenced by the problem at hand. In order to be able to produce the outputs $\tau_L, \tau_G, \tau_{\text{int}}$ with unbounded ranges, the output layer neurons have linear activation functions. The hidden layers carry hyperbolic tangent $g(z) = \tanh(z)$ activation functions. These provide a good deal of nonlinearity, and have a range of $[-1, 1]$, appropriate for our regression problem (see Figure 6.5). They suffer less from the 'vanishing gradient' problem (see section 6.2) than the traditional sigmoid activation function.

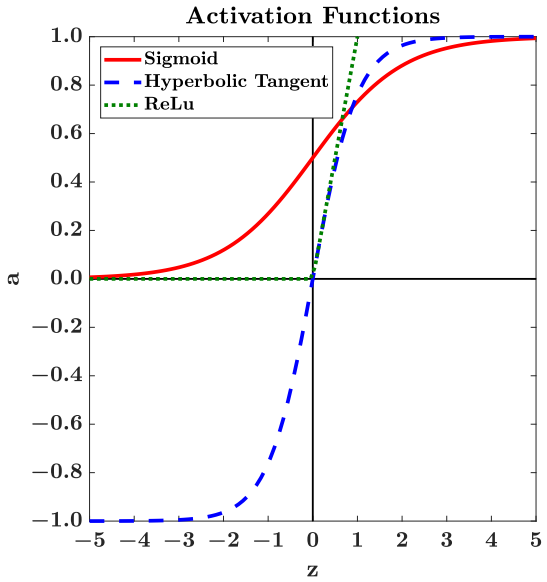


Figure 6.5: Different activation functions.

In recent years, the ReLu activation function has become very popular. This activation function is

given by

$$g(z) = \begin{cases} 0, & z < 0 \\ z, & z \geq 0. \end{cases} \tag{6.4}$$

It does not suffer from the vanishing or exploding gradient problems, and its gradient is very easy to compute. Moreover, in a trained network some nodes can be set to give a zero output for the entire input space, thus pruning unnecessary nodes from the network. These advantages are all only significant for very deep networks though, which we will not be using. The last advantage can also be a disadvantage; if neurons reach this inactive state during the training their weights will no longer be adjusted because the gradient of the activation function is zero here. The neurons are said to 'die'; they remain in this inactive state even though they might be advantageous to a fully trained network.

The sigmoid activation function is more appropriate for e.g. classification problems where a probability between 0 and 1 is required. Its maximum gradient is lower than that of the hyperbolic tangent, leading to more vanishing gradient problems.

6.5 Regularization

A great concern when training neural networks is the danger of *overfitting* the data. The high number of degrees of freedom in neural networks can enable them to approximate the training data very well, without having any generalizing capability to points located between or outside of the training data points. These points, unseen by the network during the training, are used as *validation data*, to test how well the trained network generalizes. Overfitting a large neural network is akin to fitting to data a polynomial with degree equal to the number of data points minus 1. See Figure 6.6 for an illustration.

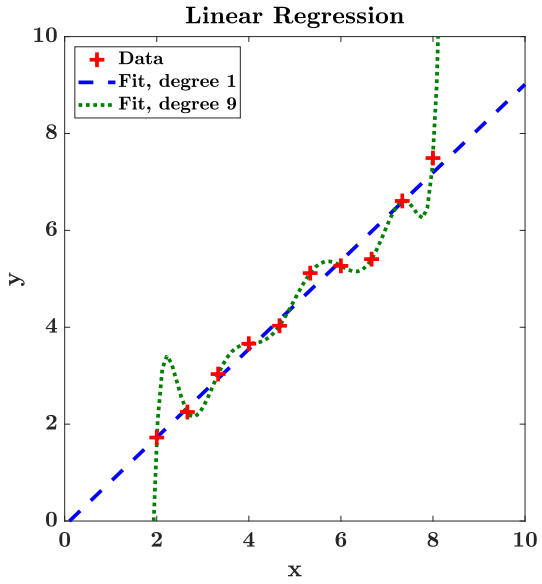


Figure 6.6: A demonstration of overfitting with a polynomial of a degree equal to the number of data points minus 1. The data is generated by setting $y = 0.9x$, and adding random noise. The high degree polynomial has a much lower value of the cost function on the training data (theoretically zero), but the low degree polynomial interpolates and extrapolates better. This means it will have a lower cost function value when measured on distinct validation data.

One possible solution to this problem is adding a regularization term to the cost function. In L2 regularization, the cost function (6.1) is modified as follows:

$$C = \frac{1-\alpha}{N} \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2 + \frac{\alpha}{n} \sum_{i=1}^{n} w_i^2, \tag{6.5}$$

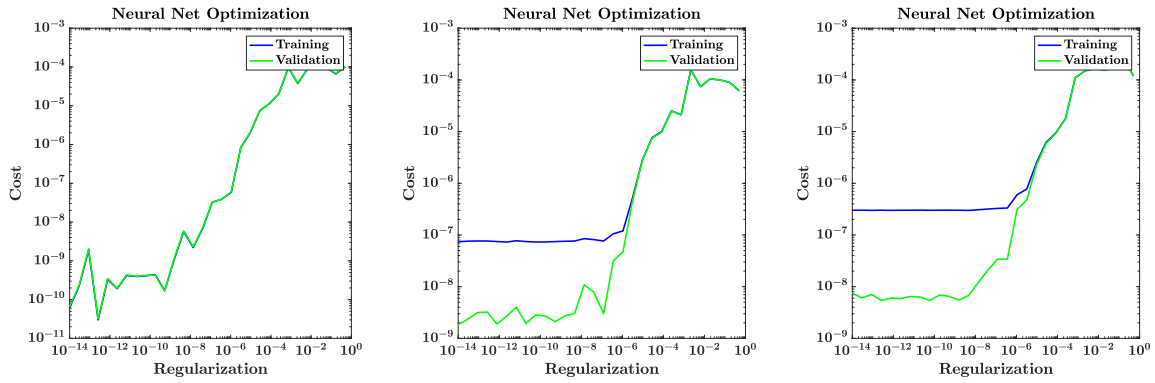
with α the parameter determining the strength of the regularization, and w_i all the weights and biases of the network. Adding this term penalizes high weights and biases, reverting neurons with the hyperbolic

tangent activation function back towards the point where the activation is zero and the derivative of the activation to the input is 1 (see Figure 6.5). This means that the neuron will pass the input through very weakly, and linearly.

The incoming weights and biases of a neuron are pushed towards this passive state if the reduction in cost function due to increasing weights and biases to values which fit the data better is not worth the increase in the cost function due to the regularization term. Thus neurons only divert from their passive state if that would contribute to getting a significantly better fit on a significant part of the data. This has the effect of pushing the network to discover patterns in the data (which yield a large decrease in the cost function), which allows the network to generalize better.

In Figure 6.7 we test the effect of regularization for our data set (see section 6.7 for details on the data). The plots, surprisingly, do not show a minimum in validation error depending on the regularization parameter α from (6.5). They do show a range in regularization below which the validation error is low and quite constant. Moreover, the validation error is lower than the training error, which can be explained from the fact that the training error contains noise. Apparently the network learns to filter out the noise from the training data and find some underlying structure which predicts the clean data well; better than the random noise can be fit.

Since this takes place at low values of the regularization parameter, it might be concluded that our specific network does not need regularization to prevent overfitting, with the properties and size of this data set. Regularization here only inhibits the degree to which the network is allowed to learn the patterns in the noisy training data which predict the clean validation data well. Therefore we decide not to use this form of regularization.



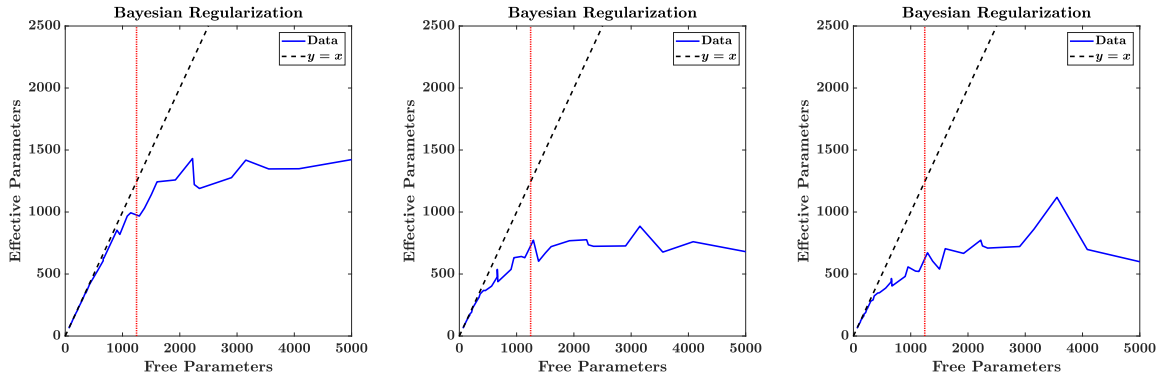
(a) No noise in the data. **(b)** 5% Noise. **(c)** 10% Noise.

Figure 6.7: Plots of the training and validation cost function value (sometimes referred to as 'error') after training nets with different regularization parameters. The second and third plots have Gaussian noise added to τ_L, τ_G, and τ_{int} in the training data, with the given percentage of the original data value as the noise standard deviation.

If we were to use regularization, we might want to choose a regularization parameter automatically. To this end an algorithm has been devised to choose the regularization parameter on the basis of Bayesian probability theory. The Bayesian regularization method was devised by MacKay [153], [154] and combined with the Levenberg-Marquardt algorithm by Foresee and Hagan [155]. In this method, a Gaussian probability distribution for the weights given the network and the data is formulated as a combination of the likelihood of the data given the weights, the prior probability for the weights, and the likelihood of the data given the model. From this, the maximum of the probability distributions for the regularization parameters α and $1-\alpha$ (independent in this framework) is derived. A useful intermediate result is the number of effective parameters γ, a measure for the amount of weights and biases (degrees of freedom of the network) that contribute significantly (i.e. without overfitting) to fitting the data well.

Our choice of network structure, discussed in section 6.3, is supported by this metric, displayed for different network structures, with differing degrees of freedom, in Figure 6.8. Initially, all the degrees of freedom of the network can be used effectively to fit the data. However, after a certain network size, the amount of effective parameters increases less with network size and the amount of unnecessary parameters starts growing. There are large oscillations in the plot because it makes a difference if we increase the amount of nodes or layers, even if such an increase carries the same amount of extra free parameters.

However, there does seem to be a trend of convergence (where the number of effective parameters no longer grows with the number of free parameters), which is underway for our chosen network structure with 4 hidden layers with 18 nodes each and 1245 degrees of freedom.



(a) No noise in the data. **(b)** 5% Noise. **(c)** 10% Noise.

Figure 6.8: Plots of the amount of effective parameters γ found via Bayesian regularization, for the different network structures also mapped out in Figure 6.4. Increasing the amount of layers or nodes increases the amount of free parameters in the network, but the growth in the amount of these parameters that can be effectively used falls off after a certain network size. The red-dotted vertical line shows the amount of free parameters in a network with 4 hidden layers with 18 nodes each.

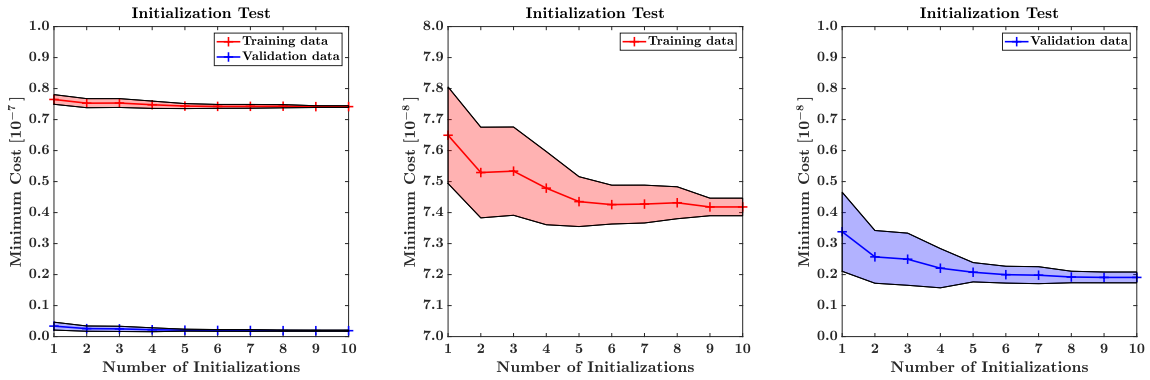
6.6 Initialization

Before training, the weights and biases of the neural network must be initialized to some value. It is important to consider how to initialize the network, because for some sets of weights and biases the network may train very poorly. For example, if using ReLu activation functions (see section 6.4), the network may be initialized in such a way that all the neurons are 'dead' from the beginning. Similarly, with our hyperbolic tangent activation function, the network may exhibit the vanishing gradient problem from the start and training will be slow.

A certain degree of randomization is desired in the initialization, because we do not know a priori what the weights and biases should be and we do not want the trained network weights to be strongly dependent on our baseless guess. Furthermore, the different weights and biases should differ from one another. Otherwise they would all be modified in exactly the same way during the training and differences between the neurons together in a layer would never arise (so that the network is effectively reduced to a network with 1 node per hidden layer).

Nguyen-Widrow initialization [156] is used as the method of random initialization. In this method of initialization, the input space of a layer is divided over all of the neurons in that layer. The neurons are assigned weights and biases such that they all have their linear domain (see Figure 6.5) in different parts of the input space, and that together their linear domains cover the whole input space. The linear domains are allowed to overlap slightly and there is a random element to the initialization. With this initialization we know that the whole input space will be utilized in the training from the beginning, and that no neurons are 'near death' (where their gradient is very small, see again Figure 6.5).

In Figure 6.9 we investigate the effect of multiple random initializations on our problem. We want to know if our network runs a high risk of not being able to reach a good optimum, when its initialization is unlucky. Therefore we initialize the network a specified amount of times, and record the validation error of the best of the trained networks. In Figure 6.9 we plot this validation error, for different numbers of initializations. By averaging the results over a number of runs, we get the expectation and standard deviation of the error. We see that the expected validation error hardly decreases with increasing amounts of initializations, on the scale which we are interested in (orders of magnitude like in Figure 6.4 and Figure 6.7). It can therefore be concluded that multiple random initializations are not necessary.



(a) **(b)** Detail of **(a)**. **(c)** Detail of **(a)**.

Figure 6.9: Plots of the training and validation error (using 5% noisy training data and clean validation data) when taking the best net (lowest validation error) out of a specified number of initializations. The results are averaged over ten runs; the shaded bands show the standard deviation.

6.7 Training data

The data used for the networks in this chapter is uniformly sampled around the parameters of the standard steady state case discussed in section 5.2. The precise ranges are given in Table 6.1. Every possible combination of the different uniformly sampled parameters is taken. This sampling results in 24576 data points. For each data point the theoretical steady state averaged velocities and stresses are calculated via equations (3.80)–(3.83).

A downside to this sampling method, is that many data points share a lot of the same parameter values, with only one or two parameters differing between the data points. This means that multiple data points are likely to end up with similar averaged velocities and stresses. The data points will thus be relatively clustered; they will not fill the parameter space well. Also, when randomly splitting the data set into training and validation data sets, the validation data set will contain points very similar to points in the training data set. Therefore, it may not be distinct enough to be good test of the generalizing behavior of the network. We have seen this in Figure 6.4 and Figure 6.7, when we use training data without added noise. We eliminate this problem by using Latin Hypercube Sampling (LHS) [157] in chapter 7.

Table 6.1: The ranges for the analytical steady state data used in this chapter for tuning the neural network parameters.

Parameter	Symbol	Range	Units	Number of samples
Background pressure gradient	$\partial p/\partial s$	$[0, -3]$	$\mathrm{kg\,m^{-2}\,s^{-2}}$	4
Liquid density	ρ_L	$[500, 1500]$	$\mathrm{kg\,m^{-3}}$	4
Gas density	ρ_G	$[0.6, 1.8]$	$\mathrm{kg\,m^{-3}}$	4
Channel height	H	$[0.005, 0.015]$	m	4
Initial interface height	h_{int}	$[0.02, 0.98]H$	m	6
Liquid viscosity	μ_L	$[0.5, 1.5] \cdot 10^{-3}$	$\mathrm{kg\,m^{-1}\,s^{-1}}$	4
Gas viscosity	μ_G	$[0.9, 2.7] \cdot 10^{-5}$	$\mathrm{kg\,m^{-1}\,s^{-1}}$	4
Acceleration of gravity	g	9.81	$\mathrm{m\,s^{-2}}$	1
Pipe inclination	ϕ	0	degrees	1

As stated above, we add Gaussian noise to the output variables τ_L, τ_G, and τ_{int}, with a standard deviation equal to 5% of the original value, so that the training data mimics measured data better. Another form of preprocessing is performed on the data. The data is mapped to the range $[-1, 1]$, not coincidentally the range of the hyperbolic tangent activation function and its approximate linear domain (see Figure 6.5). This means that the weights need not be very large or very small in order to get the data in the active range of the neuron. Furthermore, the biases can a priori be centered around zero. This helps the numerical stability of the network. When using the network (after training), input data that is

fed to it is stretched and translated in the same way as the input training data was, and the output is stretched and translated opposite to the way the output training data was.

In an equation:

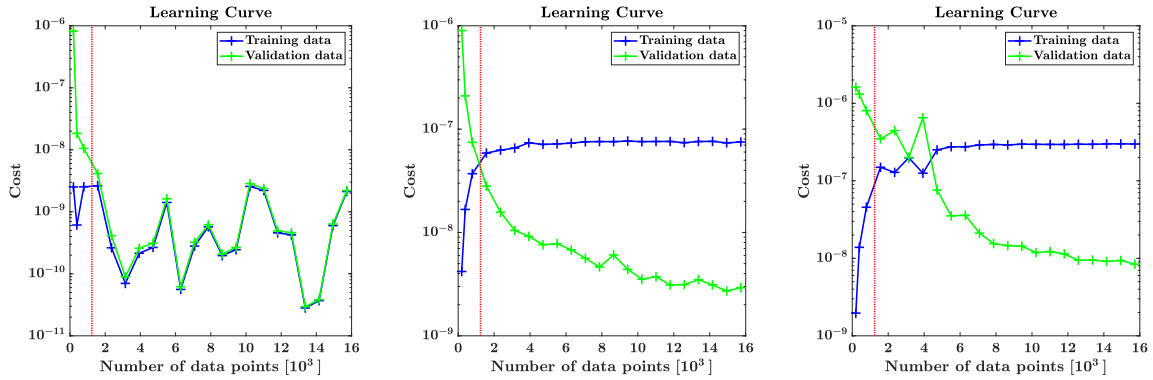
$$\widehat{x}_i \to \frac{\widehat{x}_i - (1/N)\sum_{i=1}^{N} x_i}{\max|x_i|}, \tag{6.6}$$

where $\widehat{x}_i$ is some input variable, and $(1/N)\sum x_i$ and $\max|x_i|$ are determined from the training data set. The inverse is done for the output:

$$\widehat{y}_i \to \widehat{y}_i \max|y_i| + (1/N)\sum_{i=1}^{N} y_i, \tag{6.7}$$

where $\widehat{y}_i$ is a network prediction and $\max|y_i|$ and $(1/N)\sum y_i$ are determined from the training data set.

It is important to know if we have enough data to train our network with its many degrees of freedom. If we have less data points than degrees of freedom and we apply no regularization, we can expect overfitting to occur like when fitting a polynomial of overly high degree (discussed in section 6.5). Figure 6.10 shows network performance after training our network with different amounts of data, randomly sampled from the total data set of 24576 points.



(a) No noise in the data. **(b)** 5% Noise. **(c)** 10% Noise.

Figure 6.10: Plots of the training and validation error after training with different amounts of data (the validation data set remains constant). The red dotted line shows the amount of degrees of freedom in the tested network with 4 hidden layers with 18 nodes each. The second and third plots have Gaussian noise added to τ_L, τ_G, and τ_{int} in the training data, with the given percentage of the original data value as the noise standard deviation.

We do indeed observe high validation errors with low amounts of data, decreasing with increasing amounts of data until convergence, for the networks trained on noisy data. With noise added to the data, we observe the training error growing with increasing amounts of data, before convergence. The training error grows because the network can no longer overfit a small collection of data points, but must generalize to find the pattern behind the noise, which works decently for all data points. The validation error drops while this is happening. At some point the training error is dominated by the noise in the training data and it does not grow anymore, while the network can still improve its performance on the validation data by perfecting its learned pattern with the help of more training data.

A common test for overfitting is to see if the validation error converges to the training error (if they do not match we have overfitting), however this is not applicable when adding noise to the data. It is in principle applicable to plot **(a)** of Figure 6.10, where no noise was applied to the training data. Though in this case, the network is probably still overfitting even though the training error and validation error match. The validation and training data sets used here are probably not distinct enough in this case, since their performance shows the same erratic behavior.

From plot **(b)** of Figure 6.10 we *can* conclude that when using the 5% noisy training data our amount of data is sufficient; the validation error will not be significantly reduced by adding more training data.

6.8 Results

After all the hyper-parameters have been set according to the discussion in the previous sections of this chapter, we need some metrics to evaluate the network's final performance, other than a cost function value which does not provide enough information. The scale of the cost function is determined partially by the scale of the network outputs. The single cost function value that we show in this chapter is the mean of the cost for the three different stresses. We thus need a metric which is more differentiated and which has a meaningful scale. For this we derive inspiration from Ma et al. [11] and make regression and error distribution plots.

A regression plot as meant here, plots the value of a stress as given in the data for a certain set of inputs on one axis, and the value that our model (the neural network) predicts for this same set of inputs on the other axis. Ideally, the model would predict the data perfectly, and the blue points in the regression plots of Figure 6.11 would all lie on the line $Y = T$ (prediction = target). The R^2 value

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(\widehat{y}_i - y_{l,i}\right)^2}{\sum_{i=1}^{N} \left(\widehat{y}_i - (1/N)\sum_{i=1}^{N} \widehat{y}_i\right)^2} \tag{6.8}$$

provides a metric to indicate to what degree this is the case (the maximum value is 1, the minimum value is 0). In this definition, $\widehat{y}_i$ is the model prediction. We construct a linear fit of the model prediction $\widehat{y}_i$ as a function of the data and call it y_l. The value $y_{l,i}$ is the value of the linear fit at the data point y_i, corresponding to prediction $\widehat{y}_i$. With definition (6.8), R^2 is the square of the correlation coefficient between data and predictions.

We see in Figure 6.11 that the network predicts the theoretical steady state validation data well. The other plots show the distribution of the normalized error, defined here as the difference between the model output and the (validation) data:

$$e_i = \frac{y_i - \widehat{y}_i}{y_i}. \tag{6.9}$$

It would ideally be centered around zero and remain close to zero. For the theoretical steady state validation data this is indeed the case in Figure 6.11, for all three stresses; the liquid wall stress τ_L, the gas wall stress τ_G, and the interfacial stress τ_{int}.

In chapter 7, we will perform 2D CFD simulations of wavy flow to gather neural network training data, but to make this feasible we vary only three parameters: h_{int}, $\partial p/\partial s$, and the interface amplitude Δh_{int}. Figure 6.12 shows the performance of the network trained on the Table 6.1 data, on the analytically calculated steady states of the sets of parameters corresponding to the data used in the following chapter. These steady states fall within the range of the training data, and are generated in the same way. The correlation is reasonable, but still it is significantly worse than the correlation for the validation data set with the same range and distribution as the training data set.

The problem could be that the data used in this chapter is not distributed randomly; we improve on this in the following chapter. It may also simply be that the data used in this chapter is very sparse, due to the high dimensionality of the data (we vary 7 parameters). The network performs well in the areas around clusters of data points (of which points are likely present in both the training and validation data, due to our sampling method), but bad in empty regions. New data sampled in a different way has a high chance of being in these empty regions. This is likely the case for the clusters of points far from the centerline in the regression plots of Figure 6.12, which correspond to the outliers in the error histogram.

When we evaluate our network for the actual unsteady simulation data of chapter 7, we get some more remarkable results, which are shown in Figure 6.13. Particularly for the liquid stress, our network does not predict these simulation results well. This is because the oscillations in the liquid stress are very large, while the gas and interfacial stresses remain relatively close to their steady state values. In Figure 5.17 it is seen that the stresses oscillate with a greater amplitude than steady state closure terms based on approximately the same velocity and interface height values would predict (the Rosa simulations use these closure terms and show their values). This causes the model to predict identical stresses for many different stress values actually appearing in the data. Our network does not predict this unsteady behavior at all and this is well visible in the liquid stress regression plot, with the fit through the data deviating strongly from the line $Y = T$. But even for the gas and interfacial stress we see large deviations from the center line in the regression plots. The 'shocks' in the interfacial stress depicted in Figure 5.18 will also contribute to this, because here the simulations (unphysically) give widely different interfacial stress values for exactly the same velocity profile.

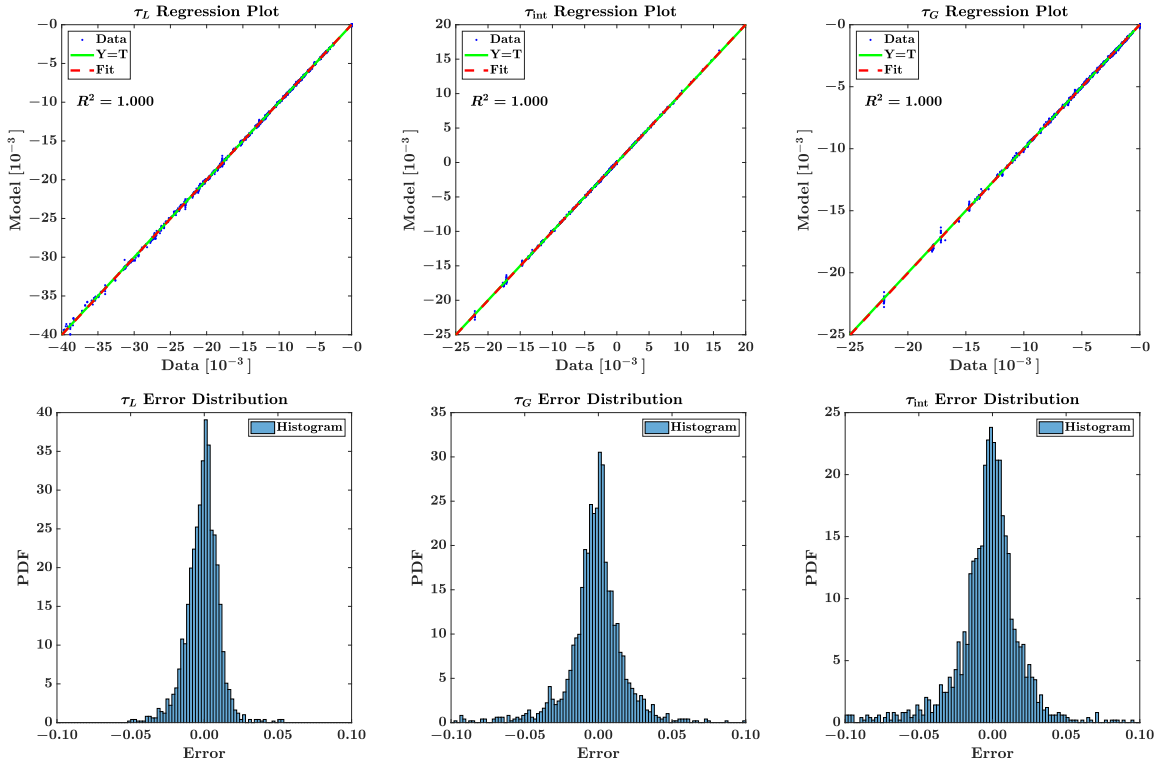


Figure 6.11: Regression and normalized error $((y_i - \widehat{y}_i)/y_i)$ distribution plots for a neural network trained on the 5% noisy data from Table 6.1, tested on clean analytical validation data corresponding to the wavy simulations done in chapter 7.

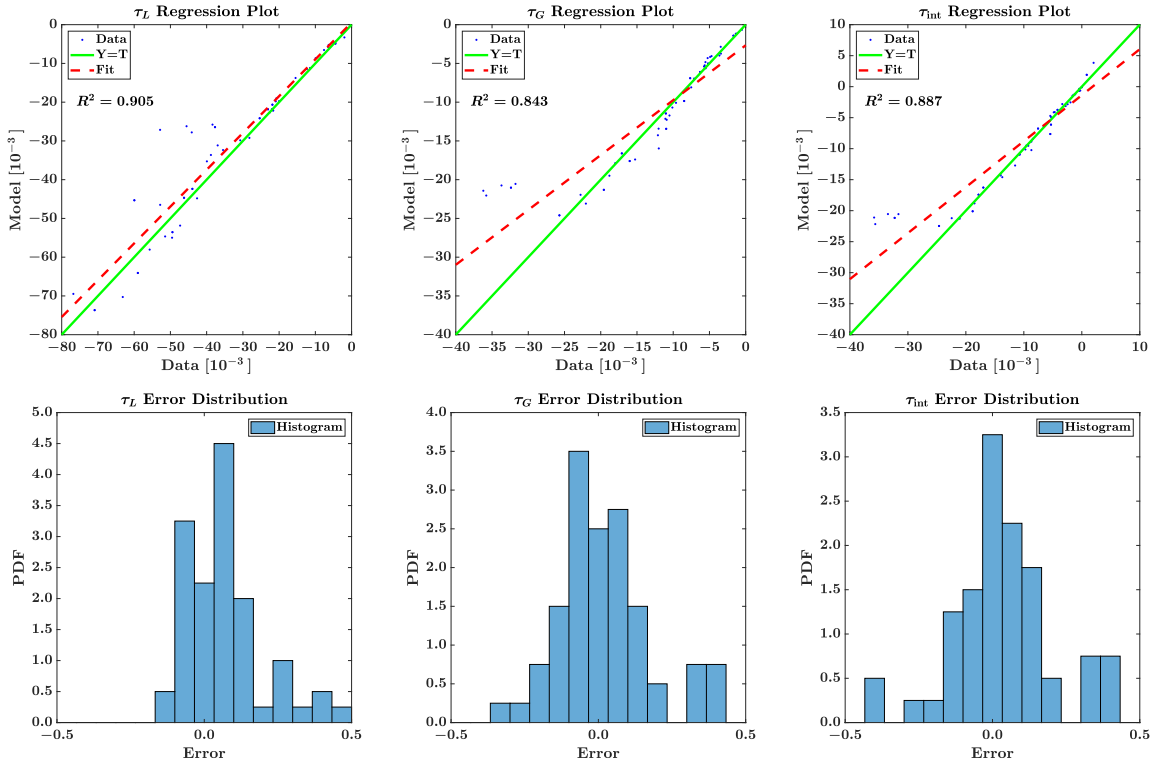


Figure 6.12: Regression and normalized error $((y_i - \widehat{y}_i)/y_i)$ distribution plots for a neural network trained on the 5% noisy data from Table 6.1, tested on analytical steady state data corresponding to the parameters of the data set used in chapter 7.

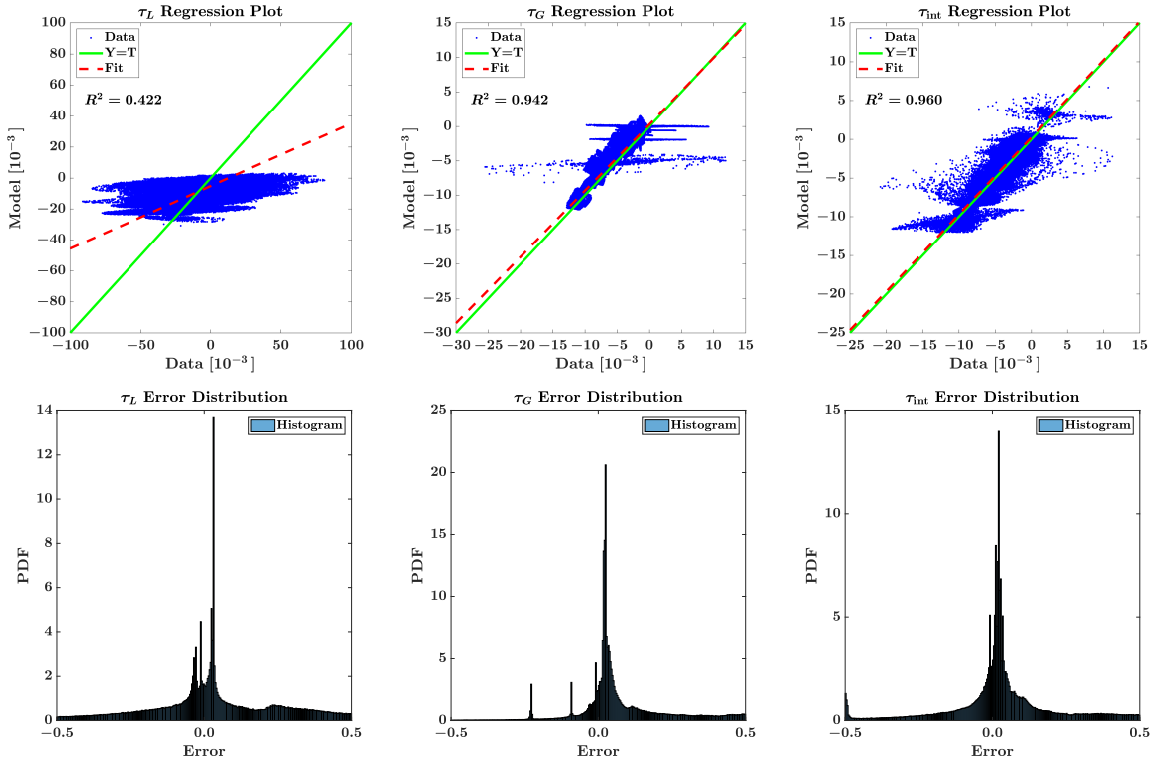


Figure 6.13: Regression and error $((y_i - \widehat{y}_i)/y_i)$ distribution plots for a neural network trained on the 5% noisy data from Table 6.1, tested on the wavy unsteady simulation data used in chapter 7.

The same poor results as seen in Figure 6.13 are obtained when the analytical closure relations are tested (as the model) on the wavy unsteady simulation data directly, so without passing through a neural network first (as the training data). What this tells us is that we need to train our network on unsteady simulation data in order to obtain accurate closure terms for the unsteady case. This may seem obvious, but the conventional closure terms discussed in subsection 3.4.4 are all based on fully developed steady state flow, and will thus likely correlate similarly poorly with our 2D wavy unsteady simulation data. Training a neural network on the unsteady simulation data, which we will do in chapter 7, thus has the potential to yield new dynamic closure terms which are much better suited to dynamic simulations.

In section 5.3 we had already discussed the uniqueness issue with calculating stresses, implicitly via velocity profiles, from averaged velocities. This is a problem inherent to forming a relation of the form (6.3), and is not alleviated by training a network on unsteady simulation data. However, adding extra inputs, such as the interface slope, does alleviate the issue, and this is easy to do when using our method. Notably, this specific extra parameter helps to distinguish between unsteady and steady states.

The point of training on unsteady data itself stands somewhat separated. This will in principle improve the performance of the closure terms for unsteady flow at the expense of its performance for steady state flow. Though there need not be a negative effect for the steady state closure terms if these are also included in the training data and the structure behind them is similar to that of the unsteady closure terms.

6.9 Conclusion

We have looked into a number of the numerous choices to be made in the formulation and training of a neural network. In the process we have discussed a number of aspects of neural networks and metrics to monitor. We have performed experiments with analytical steady state data to inform these choices. With these choices we have come to a network which performs reliably ($R^2 = 1$) on the validation data corresponding to its training data, and a bit worse ($R^2 = 0.84$–0.91) on analytical steady states sampled very differently. This performance can likely be improved by changing the data sampling method, and we

will do so in the following chapter.

Nonetheless, we have found a satisfactory network architecture using steady state data and validated it with analytical data, corresponding to a (modified) step [3] and [4] in our project flow chart Figure 1.4. The architecture is a fully connected multilayer perceptron network (MLP), a basic first choice for function fitting. With the chosen input-output relation, more advanced architectures are hard to justify. The chosen neural network should have just enough degrees of freedom to produce this relation, and not too many. It was shown that multiple initializations are unnecessary, and that the tested data set containing 24000 points was large enough, so that regularization via the cost function is unnecessary.

On wavy, unsteady, simulation data, the network performance is unsatisfactory. Particularly for the most unsteady output variable, the liquid stress, the model prediction is poorly correlated to the data. This may seem obvious, but conventional closure terms (subsection 3.4.4) are all based on steady state flow. Seeing as our network can reproduce steady state closure terms relatively well, it seems here that closure terms based on steady state flow are not sufficient for application to unsteady flow.

This justifies the research of the next chapter, where we will use unsteady high fidelity simulation data to learn closure terms which should hopefully perform better for the unsteady case.

Chapter 7

Closure Terms for Unsteady Simulations

7.1 Introduction

In this chapter, we train neural networks using unsteady, wavy, Gerris simulation data. This corresponds to step [5] in our flow chart Figure 1.4. The simulations are similar to those analyzed in section 5.5. In chapter 5 we saw that Gerris simulations and Rosa simulations using the analytical steady state closure terms show good agreement at steady state, but less during the unsteady phase of the simulations. Furthermore, in chapter 6 we saw that neural networks trained using theoretical steady state relations showed poor agreement (especially for the liquid wall stress) with unsteady wavy simulation data. In chapter 5 it was noted that no direct relation exists between the current closure term inputs (listed in (6.3)) and the velocity profile (and thus the stresses), when all possible unsteady states are included. However, it may be possible to find which unsteady velocity profiles are common, using our high fidelity code, when starting from certain types of initial conditions. A neural network might learn the commonly occurring velocity profiles and the associated stresses in an automated fashion, and in this way it might be possible to find closure terms which improve the agreement between our high fidelity and low fidelity codes, for the wavy unsteady regime.

In order to judge the performance of closure terms based on unsteady data we insert our neural network into the low-fidelity code to serve as its closure terms, in place of the analytical closure terms or any of the empirical correlations discussed in subsection 3.4.4. This is step [7] in Figure 1.4. For step [8] in Figure 1.4, we quantify the difference between the Rosa results that follow and the Gerris results, and compare this to the difference found when using analytical closure terms. We also compare physical aspects of the Rosa solutions using the two different sets of closure terms to the Gerris simulation results.

7.2 Training data

Our aim is to learn closure terms for a limited parameter space encasing the standard test case described in section 5.2. This test case is plotted as a cross in Figure 4.4b, where it is shown to be well in the well-posed stable regime of the two-fluid model. This means that we can safely apply perturbations to this case, with confidence that the waves will damp out with time, so that our simulations will remain limited to stratified, small perturbation flow. In the long wavelength limit, 2D inviscid stability analysis gives a stability boundary equal to the ill-posedness boundary for the two-fluid model (also shown in Figure 4.4b). Therefore we can be confident that our 2D simulations will be similarly stable. Because our standard parameter set is deep in the well-posed stable area, we can safely conduct simulations with a parameter range around this point.

Our parameter ranges are shown in Table 7.1. Two of the three varied parameters are axes of the stability map shown in Figure 4.4b, the third parameter Δh_{int} has no influence on this plot. We choose to vary only three parameters as opposed to the seven varied in chapter 6, because now we are using space- and time-dependent simulation data as opposed to analytical steady state data, so that we require much more computation time per parameter set. This makes it infeasible to sufficiently fill out a

Table 7.1: The ranges for the parameters of our wavy unsteady high fidelity simulations, used as training data. The remaining parameters are kept constant and equal to the standard parameters of Table 5.1. We start simulations with two different types of initial conditions; one where the fluids are completely at rest, and one where at each position along s the theoretical steady state is calculated, corresponding to the local interface height h_{int}.

Initialization	$h_{\text{int}}\,[H]$	$\partial p/\partial s\,[\text{Pa/m}]$	$\Delta h_{\text{int}}\,[H]$	N
zero wavy	$[0.05, 0.95]$	$[0, -3]$	$[0.00, 0.04]$	30
developed wavy	$[0.05, 0.95]$	$[0, -3]$	$[0.00, 0.04]$	30

seven-dimensional parameter space with the available computational resources, while a three-dimensional space remains manageable.

The parameters of Table 7.1 are sampled using Latin Hypercube Sampling (LHS) [157], with $N = 30$ samples per initial condition. With this sampling method, the input space of each variable parameter is divided into N sections. Then a random sample is taken within each of these sections, for each variable. The random samples taken per parameter are randomly coupled with the random samples of the other variables. With this method we ensure that the input space of all parameters is representatively sampled, and that values of the parameters are never repeated over multiple samples. As is noted in the original article by McKay et al. [157], this is an advantage when the output of the model is dominated by only one or a few of the varying parameters, which was hypothesized to be a problem with the uniform sampling method described in section 6.7.

Out of all the parameters that we can vary, we choose to vary h_{int}, $\partial p/\partial s$, and Δh_{int}, with regard for practical application. If the fluids and pipe geometry are set, these parameters are all that remain, besides perhaps the wavelength of the wave. Having a closure model trained on data in which these parameters are varied allows dynamic low fidelity simulation of flow in a fixed pipe with fixed fluids, without running much risk (determined only by the chosen ranges) that the model might enter a regime for which the closure model has not been trained.

We limit the parameter ranges as to not consider very thin layer flow (at low and high h_{int}), which is difficult to resolve with our 2D code, leads to different effects in our 1D code (see top of Figure 4.4b), and which nears single-phase flow, the transition to which our codes cannot handle.

The dynamics of our test case were studied in subsection 5.5.2. For both initial conditions we get 'standing' waves which travel at low speeds, and damp out after some time. For the developed wavy case the waves travel a bit faster from the beginning. The standing wave behavior arises because two waves traveling in the opposite direction are generated from the initial perturbation. In practical pipeline applications, the velocities will be higher and the traveling behavior will be more prominent than the oscillating behavior. However, our test case does include both behaviors and serves as an example of low Reynolds and low Froude number behavior.

Varying the parameters as given in Table 7.1 will give rise to many different velocity profiles, and stresses. Performing unsteady simulations means we will acquire many data points per parameter set, equal to the number of grid cells along the s-axis multiplied by the number of snapshots that we take, as opposed to just one steady state per parameter set as in chapter 6. Though we can increase the amount of data by increasing the spatial and temporal resolution, if the resolutions are high the data points might be nearly identical and not actually add extra information. This also depends on the state of the flow; in the initial unsteady phase of the simulations different points will contain more information than in the eventual steady and fully developed phase. Therefore we take the approach to take many snapshots (40 per second, over 10 second simulations) and take data from each horizontal position (spatial resolution from chapter 5 is $H/64$, with $L = 12H$ making 768 points along the s-axis). We aggregate the ~ 300000 data points per simulation to form our data set, resulting in over 9 million points per initial condition, and then randomly sample a small portion (5–10%) of it.

In this chapter, we train networks on the initial conditions of Table 7.1 separately, and we train networks on a data set in which the data for the two initial conditions is combined. We refer to these networks as

- 'zero wavy net',
- 'developed wavy net',
- 'zero + developed wavy net'.

The first two nets are trained on 5% of their respective data sets. The third net is trained on 10% of the data combined for both initial conditions.

7.3 Training of the network

We initialize a neural network with the structure discussed in chapter 6. However, because we only vary h_{int}, $\partial p/\partial s$, and Δh_{int}, the only network inputs of those listed in (6.3) that remain variable are h_{int}, u_L, u_G, and $\partial h_{\text{int}}/\partial s$, so that our network will only learn the relation between these variables and the stresses, at constant values for the other parameters. This could have some influence on the optimal choice of network structure, but we assume that a network which performs well with all the parameters as inputs will perform adequately when only a subset of them is varied. Technically, when the inputs are limited to this the structure of the network changes from Figure 6.3 to Figure 7.1.

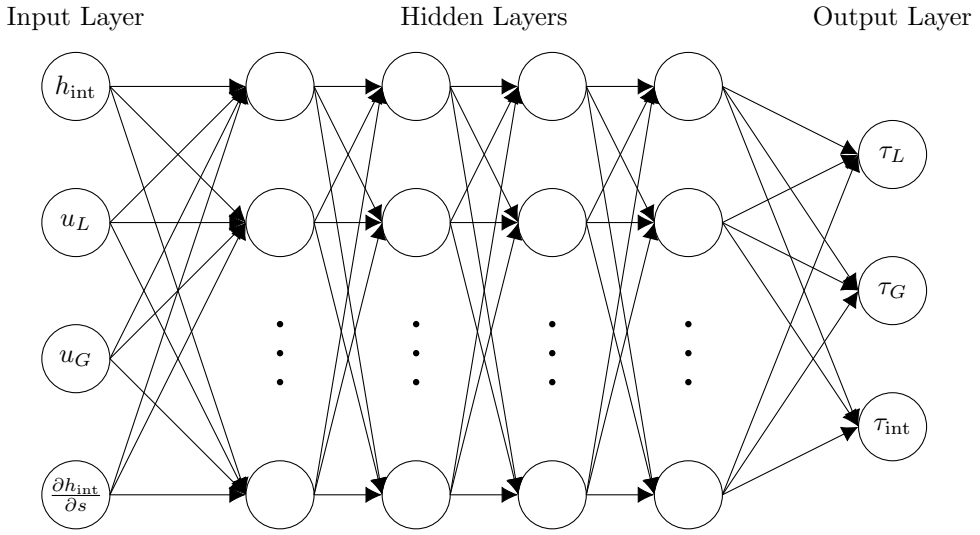


Figure 7.1: A schematic of the neural network structure as applied in this chapter. We have only four variable inputs, and three outputs. There are four hidden layers with 18 nodes per hidden layer.

We train the network using the Levenberg-Marquardt algorithm discussed in section 6.2. In order to prevent unnecessarily long training, and improve generalization, we apply validation stopping. The network is trained on training data which is a random subset of the total data set. After each optimization iteration, also called an epoch, the performance on a different subset of the data set, the validation data, is tested. If the performance on the validation set fails to improve a certain number of epochs in a row, the training is halted. The progression of the training is shown in Figure 7.2.

During the initial phase of the training, the cost function decreases rapidly. Afterwards the progress is slow. This may indicate a structure of the loss landscape as described by Choromanska et al. [151] for larger networks, where many local minima of high quality are located in a narrow band in terms of the cost function. In the high-dimensional loss landscape spanned by the weights and biases, there is a group of local minima with similar cost function values. It is relatively easy to get into this band, but finding the best minimum in this band is hard, though in practice it does not matter much where exactly in this band we end up (so we might as well end our training earlier).

The training shown above was done on 10% of the combined data for both initializations, and took in the order of a couple of days to complete (on a single CPU). For each epoch, the training goes through the entire selected training data set. The computational cost and generalization capacity may be improved in the future by using a stochastic gradient descent algorithm [158], which is not implemented in the current version of the used MATLAB toolbox.

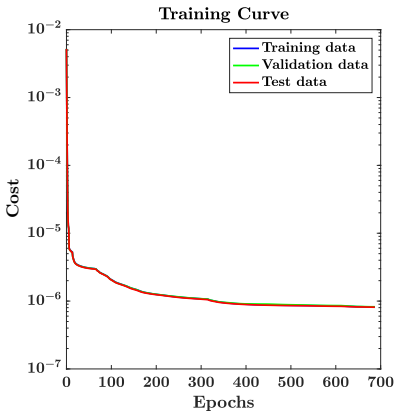


Figure 7.2: Training curve for a network trained on wavy unsteady Gerris data for both initial conditions given in Table 7.1. The cost values for the training, validation, and test data largely overlap. The training stops when the cost measured on the validation data no longer diminishes.

7.4 Quality of approximation of data

In Figure 7.3 we show the regression plots and error distributions for our neural network trained on the combined Gerris data with both initializations: the 'zero + developed wavy net'. It shows the performance of the network on validation data from the same dataset as the training data. The plots in the figure show that our data is predicted sufficiently well by our neural network. The prediction is less precise than the perfect score of our net trained on noisy analytical data in Figure 6.11, but this is to be expected since in actual simulation data there will be artifacts (such as the shocks in the interfacial stress shown in Figure 5.18), and noise. Moreover, the simulations exhibit complex wavy unsteady behavior, as opposed to the simple steady states considered in chapter 6.

The validation data used in Figure 7.3 is the same data which is compared to our analytically trained net in Figure 6.13. With the current network being trained on the unsteady data, it succeeds in reproducing the data much better. The current good performance is a confirmation of our network architecture and training. The network seems to be able to capture the unsteady stresses, when given unsteady training data.

In Figure 7.4 we can see the results if we leave out the interface slope as an input for the neural network. The performance deteriorates significantly, particularly for the liquid stress (seen in the reduction of R^2 from 0.978 to 0.732). Apparently the interfacial slope is an essential piece of information for the determination of the stresses. It seems to be a good measure of the interface waviness, and of the local phase of the wave. The liquid stress was already seen in section 5.5 to oscillate strongly for our test case, at the wave frequency. Thus to determine if the liquid stress should locally be at a positive or negative extreme, or in between, it is essential to know the local phase of the wave, and this information is provided by the interface slope.

The deterioration in the approximation of the gas and interface stress seems insignificant in comparison. However, the relative strength of the oscillation in the liquid stress compared to that of the gas and interfacial stresses is likely particular to the current test case. For in this test case the liquid velocity is low, and much lower than the wave velocity. This is indicated by the very low Froude number in Table 5.2; the flow is deep in the subcritical regime. Thus the varying liquid velocity induced by the waves is large compared to the mean flow; and the same holds true for the liquid stress. The gas Froude number is much larger and thus the oscillations are weaker compared to main flow.

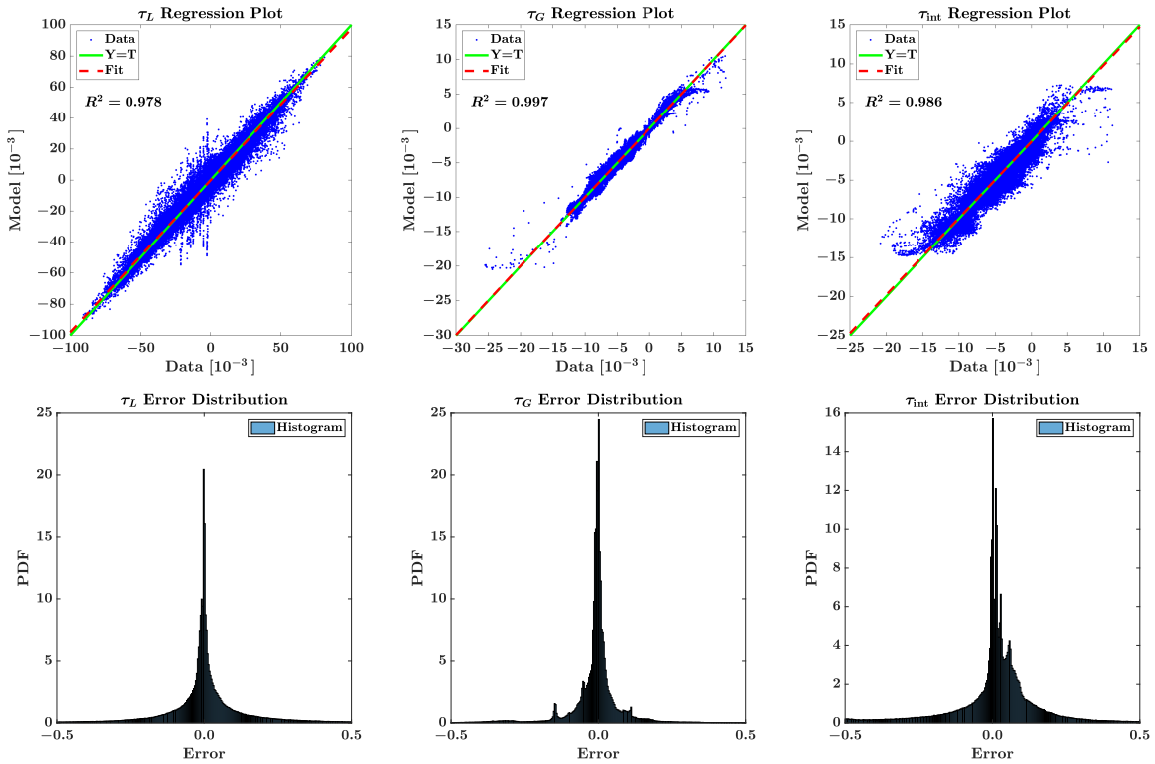


Figure 7.3: Regression and relative error $((y_i - \widehat{y}_i)/y_i)$ distribution plots for a neural network trained on the data combined for both initial conditions of Table 7.1.

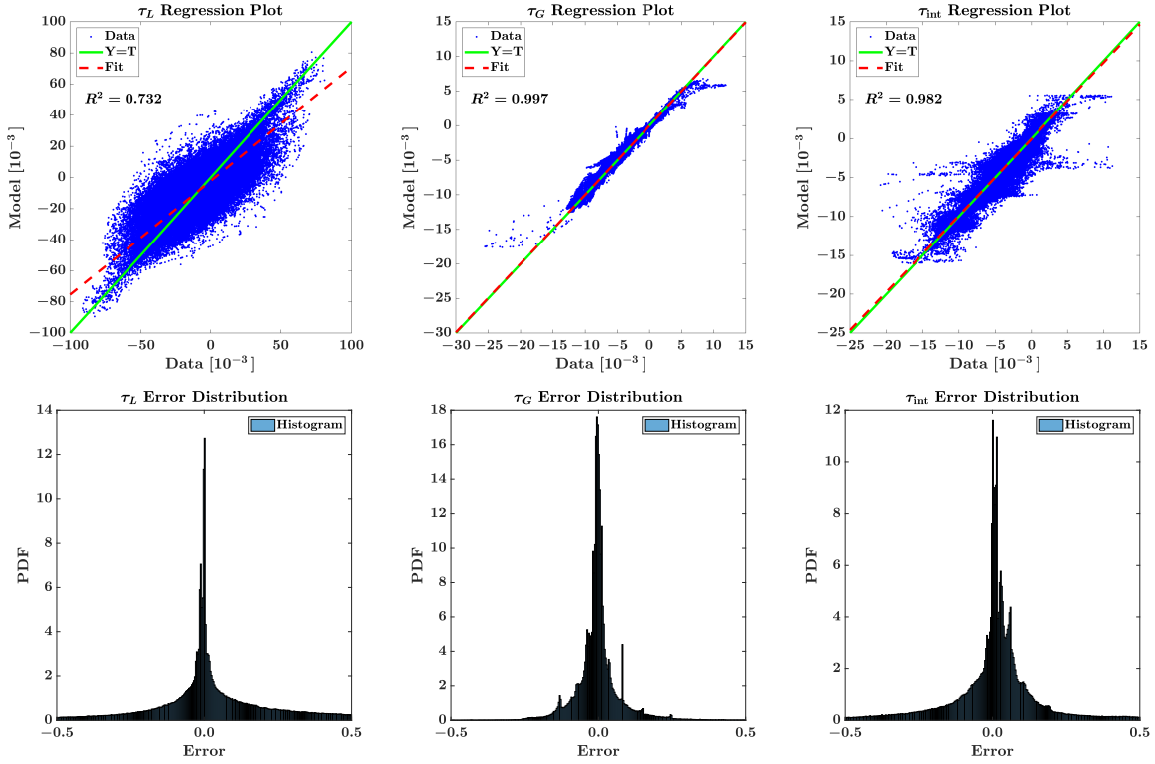


Figure 7.4: Regression and relative error $((y_i - \widehat{y}_i)/y_i)$ distribution plots for a neural network trained on the data combined for both initial conditions of Table 7.1. However, in this network the interface slope is not included as an input. This leads to a clear deterioration in performance, particularly for the liquid stress.

7.5 Two-fluid code results

7.5.1 Overview

We sample the data with replacement to get five different data sets, each a small percentage of the total data set. We train (randomly initialized) networks on each of these subsets of the data, and load each of them into the Rosa code. This corresponds to step [7] in Figure 1.4. At each stage in the Runge-Kutta time integration scheme, the variables as given in (6.3) are fed to the trained neural networks to arrive at values for the stresses τ_L, τ_G, and τ_{int}. The final prediction for the stresses is obtained by averaging the predictions of each of the five networks. This averaging procedure is called 'bagging' and was shown to improve accuracy for learning algorithms sensitive to changes in the training data [159]. This technique was also employed by Ma et al. [11].

Unfortunately, with the current MATLAB shallow neural network implementation, using these networks significantly slows down the Rosa code simulations. Just using one neural network, instead of the analytical closure terms, the simulation time is increased by a factor of five. Bagging five neural networks as described above, the simulation time increases by almost a factor of twenty. This is currently a large drawback, but it might be improved by implementing the networks more efficiently; as they are not very deep and would not be expected to be so costly to calculate.

We evaluate our results mainly by comparison to the high fidelity simulation 'ground truth'. This refers to step [8] in the project flow chart Figure 1.4. An important assumption in our procedure is that the high fidelity data represents a good ground truth.

The Gerris and Rosa simulations use different grid resolutions. In order to be able to compare Gerris and Rosa results quantitatively, cubic splines of the variables of interest are constructed, along the horizontal axis. The s-dependent Gerris result at $t = t_i$ is y_i, with $\widehat{y}_i$ the corresponding Rosa result. We compute characteristic values y_c for each variable of interest, based on analytical solutions for laminar single phase flow. Table 7.2 shows the following relative error measure for the difference between Gerris and Rosa results, termed the 'normalized averaged error' (NAE):

$$\text{NAE} = \frac{1}{N_T} \sum_{i=1}^{N_T} \sqrt{\frac{1}{L} \int_{s=0}^{L} \left(\frac{y_i - \widehat{y}_i}{y_c} \right)^2 \mathrm{d}s}. \tag{7.1}$$

The parameter N_T is the total number of time steps and L is the length of the domain. This can be regarded as the root-mean-square-error (RMSE) of the normalized variables along the horizontal axis, averaged over time.

This error is shown for simulations initialized from different initial conditions, and using different closure terms. Analytical closure terms (3.84) are tested alongside closure terms learned from the wavy unsteady data of Table 7.1, using neural networks. Where the neural network assisted error is smaller than the analytical closure error, the error value is highlighted green in Table 7.2.

We calculate the factor by which the error is changed, using neural network instead of analytical closure terms, for each variable of interest. This factor is averaged over all the variables of interest and displayed in the last column of Table 7.2.

Table 7.2: Normalized and averaged errors between high-fidelity and low-fidelity simulations, according to (7.1), for different variables of interest. The results are given for Gerris and Rosa simulations starting with different initial conditions, and with the Rosa simulations using either analytical or learned closure terms. The final column shows the value of the neural network closure error divided by the analytical closure error, for each variable separately, and then averaged.

Case		Normalized Averaged Error $[10^{-3}]$						Relative
Initialization	Closure	h_{int}	u_L	u_G	τ_L	τ_G	τ_{int}	
zero wavy	analytical	1.05	84.6	13.9	212	7.33	26.4	–
zero wavy	zero wavy net	0.31	754	4.22	283	26.2	38.5	2.65
zero wavy	zero + developed wavy net	0.52	193	15.2	233	10.8	28.5	1.25
developed wavy	analytical	1.09	75.5	12.2	215	7.56	18.2	–
developed wavy	developed wavy net	0.42	385	3.52	215	5.07	18.7	1.41
developed wavy	zero + developed wavy net	0.51	112	12.6	173	8.20	26.6	1.06

The table shows that the neural network assisted error is smaller than the analytical closure error, for some of the variables of interest. The performance is improved most for the interfacial height, and least for the liquid velocity. Looking at the final column in Table 7.2, on average the learned closure terms do *not* seem to improve the performance, though the zero + developed wavy net comes close. However, the error metric displayed in Table 7.2 is just one metric, which does not measure the similarity in dynamic behavior between the high and low fidelity simulations.

The best results overall are obtained for the developed wavy initialization, using networks trained on both initial conditions (the zero + developed wavy net). We show the results graphically side by side with the results obtained using the analytical closure terms, in Figure 7.5. The figure shows the velocities, stresses, and interface height at the center of the domain, as a function of time.

In Figure 7.6 we see a comparison of the Rosa and Gerris results for the interface height throughout the domain, for the same developed wavy case. Here, the Rosa simulations also use the zero + developed wavy net closure terms. This figure can be compared directly to Figure 5.15, where analytical closure terms are used in the Rosa simulations. From this comparison we can see that the Rosa simulations using neural network closure terms succeed in following the Gerris simulations better.

7.5.2 Wave damping behavior

The figures Figure 7.5 and Figure 7.6 show how the scale and form of the oscillations is captured better by the neural network closure than by the analytical closure. Particularly at later times the analytical closure causes different wave behavior. This difference in the replication of the wave behavior is seen in the velocities, the stresses, and the interface height. The problem with the analytical closure terms is shown well in Figure 7.7, in which the same simulation results are shown for later time instants (with the analytical closure). The waves acquire a sharp wavefront, and in its wake small spurious waves are formed. This seems to be an example of numerical dispersion.

The simulations using the neural network closure use the same numerical scheme, but do not show this problem. The neural networks, with unsteady wavy data as training data and the interface slope as an input, learn to replicate the damping behavior. Rosa simulations with the neural networks as closure terms show the same damping behavior as the high fidelity simulations, and do not form near-shocks or spurious oscillations.

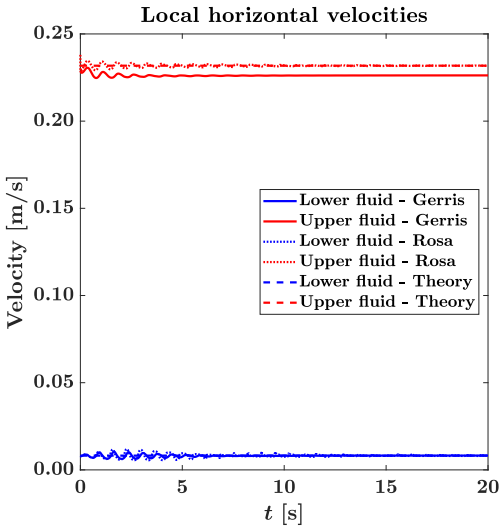
An explanation is that the stresses should damp the oscillations. Inviscid linear stability analysis predicts that the waves should propagate without growth or damping, for this test case (and the inviscid 1D two-fluid model was shown to agree with this analysis to a good degree in section 4.5). In these viscous simulations, the waves are damped.

Both the analytical closure terms and the neural network have the averaged velocities and interface heights as inputs, which vary along the length of the wave. This allows stresses to vary along the length of the wave, enabling damping. However, these alone, considered locally, offer no information on the wavelength or phase of the wave: the analytical closure terms just consider the steady state corresponding to these inputs.

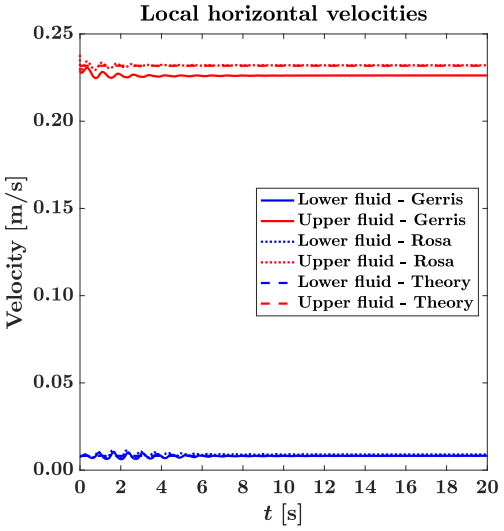
A part of the oscillations of the stresses can likely not be fully explained by the changes in velocity. This is because the velocity profiles are shaped differently during different phases of the wave, so that a given averaged velocity at some point along the wave gives a different velocity profile than if the same averaged velocity were found at another point along the wave. An explicit dependence on the waviness and local wave phase may be necessary. This can be provided, for example, by the interfacial slope. Adding an input parameter to the closure relation alleviates, in this way, the uniqueness problem discussed in section 5.3.

For the neural network the interfacial slope is easily added as an extra input, since the neural network is based on direct calculation of the stresses from high fidelity data as opposed to steady state balances (as discussed in section 6.3). This extra parameter makes a significant difference, given that we have also tested a neural network without this parameter as an input (seen earlier in Figure 7.4); this results in extreme oscillations.

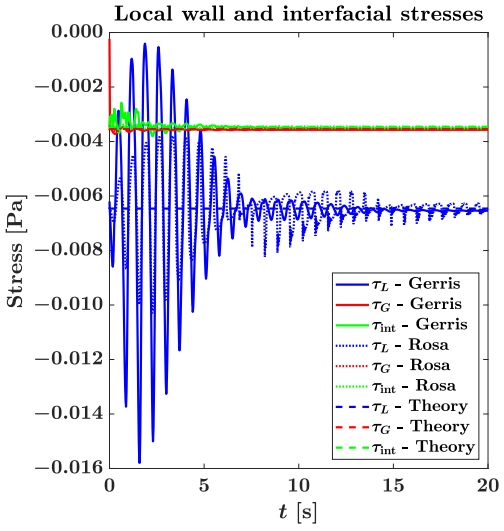
The neural networks learn the relation between the averaged velocities, the interface height, the slope in the interface height, and the stresses. On the other hand, the analytical closure terms are based on the flat interface steady state and are not informed on the stresses in wavy flow. They freely allow the growth of interface height discontinuities and small wavelength perturbations (which may arise due to numerical error).



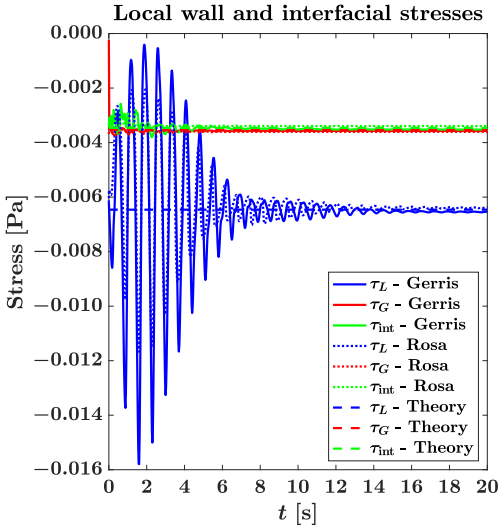
(a) Analytical closure.



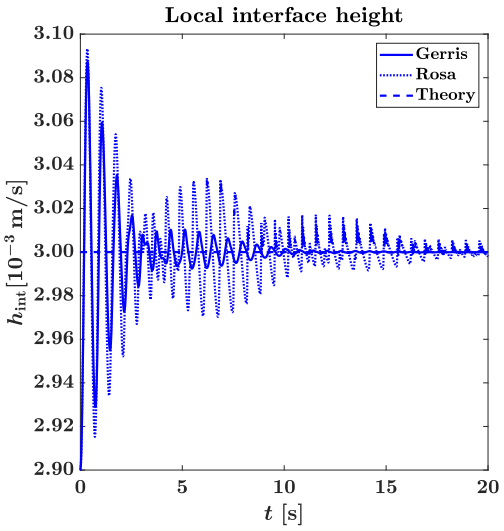
(b) Zero + developed wavy net closure.



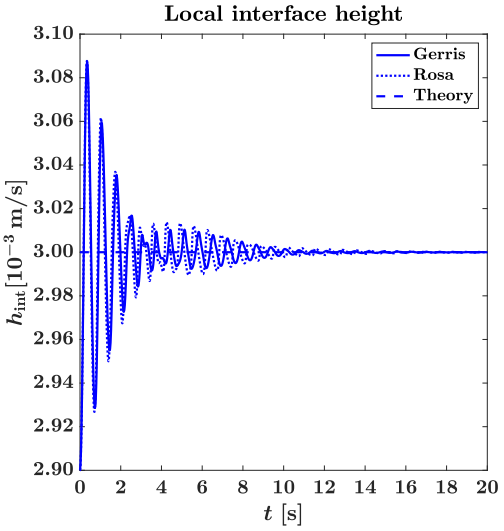
(c) Analytical closure.



(d) Zero + developed wavy net closure.



(e) Analytical closure.



(f) Zero + developed wavy net closure.

Figure 7.5: Evolution in time of the velocities, stresses and interface height at the center of the domain. Initialized with the 'developed wavy' initial condition.

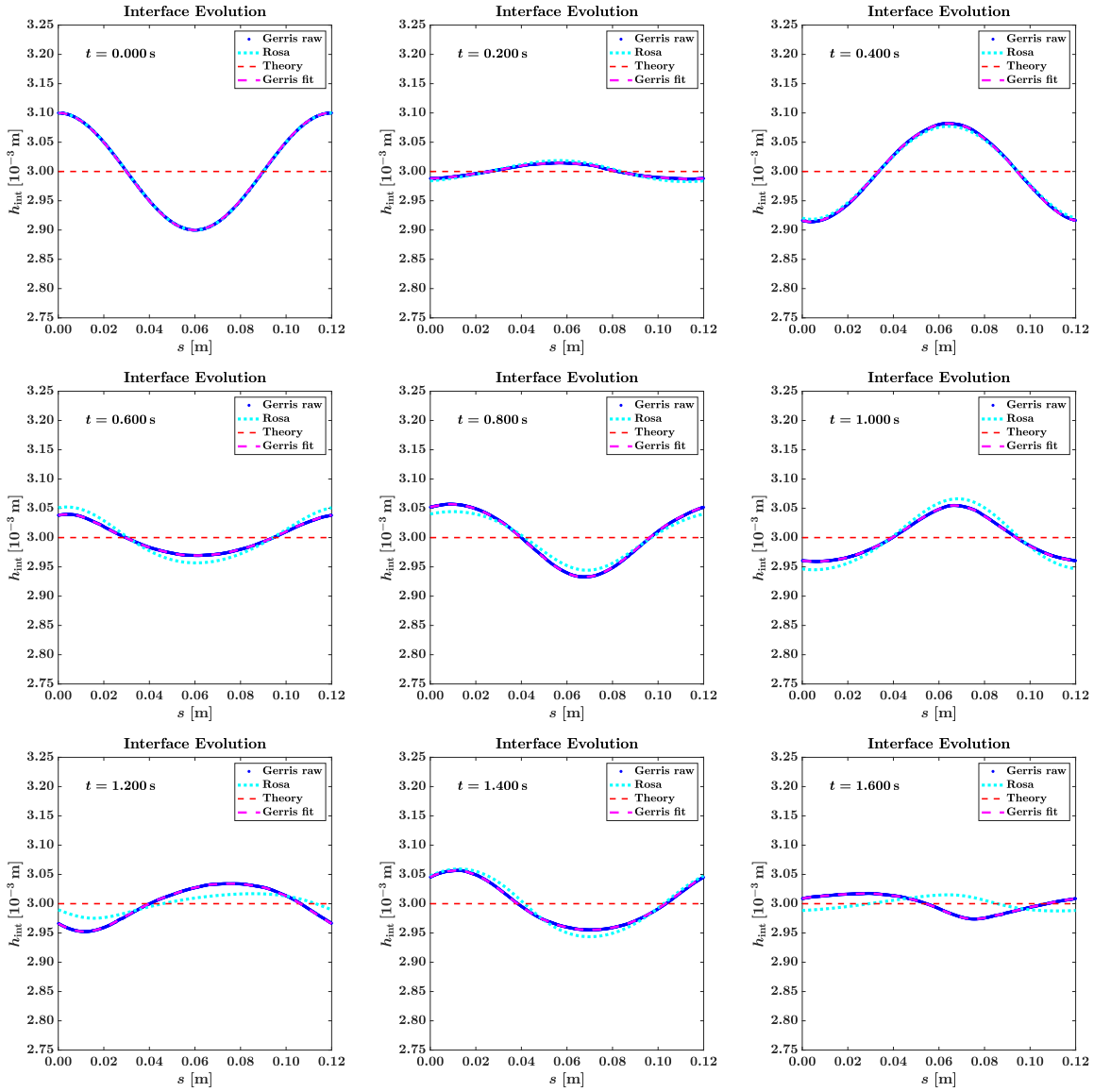


Figure 7.6: Evolution in time of the interface between liquid and gas throughout the domain, zoomed in ($H = 0.01\,\mathrm{m}$). We see Rosa results with a 'developed wavy' initialization and 'zero + developed wavy net' closure. This figure can be compared directly to Figure 5.15, where analytical closure terms are used in the Rosa simulations.

We note that closure terms for wavy flow exist (see e.g. Andritsos and Hanratty (1987) [107] or Cheremisinoff and Davis (1979) [109]). However, even these wavy closure terms are based on the fully developed steady state. This means that the waves are not explicitly modeled with the interface height as a function of the horizontal coordinate. Rather, the averaged effect of the wavy interface in a section of pipe is considered. Because of this, it is not possible to include the interfacial slope as an input for the closure relations; its average value is zero.

Our methodology allows for direct calculation of the stresses and any known quantity in the 1D averaged model, at any time and location in the high fidelity simulations, and automatically infers a relation between them, so that we can have closure terms which differentiate between different phases of the wave.

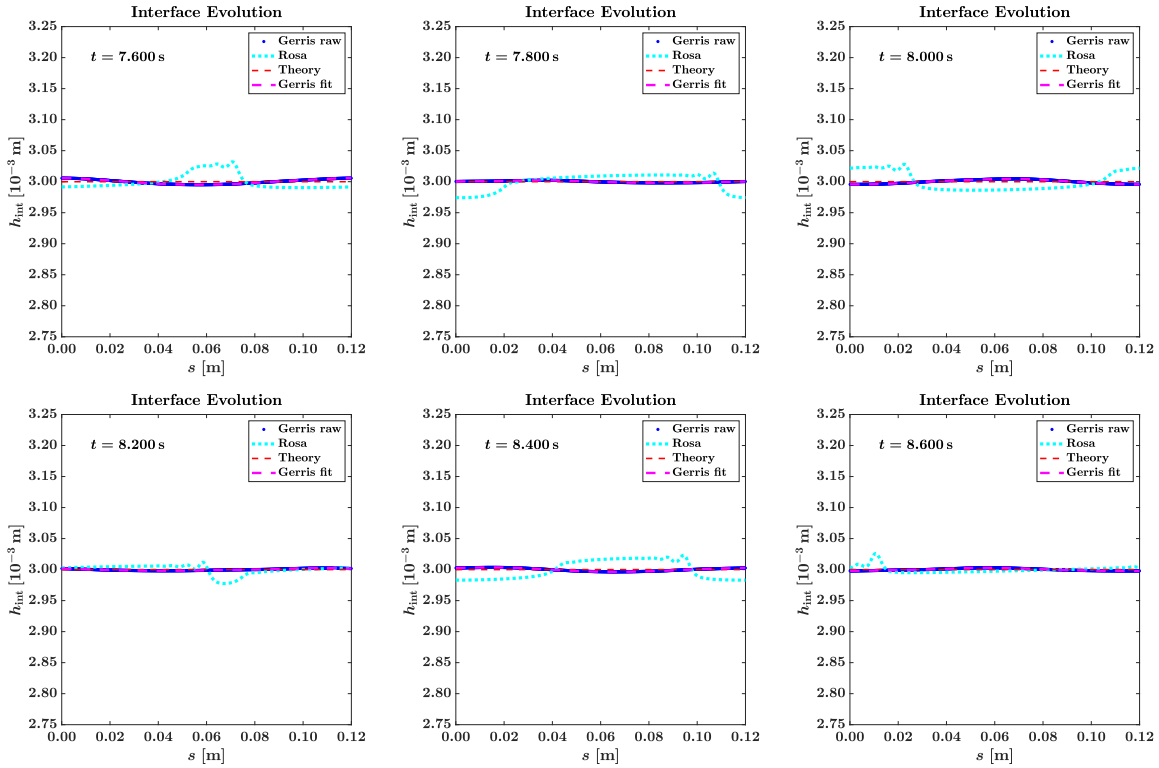


Figure 7.7: Evolution in time of the interface between liquid and gas throughout the domain, zoomed in ($H = 0.01\,\mathrm{m}$). We see Rosa results with a 'developed wavy' initialization and analytical closure.

7.5.3 Wave speed

An aspect where the Rosa simulations using neural network closure show a difference with the high-fidelity simulations in Figure 7.5, is the wave speed. The wave speed of the Rosa simulations is slightly larger than that of the Gerris simulations, so that the two slowly drift out of phase. They seem to be 180 degrees out of phase after about 12 oscillations. This effect becomes unnoticeable due to the damping of the wave, after which the simulations come to agree roughly on the steady state.

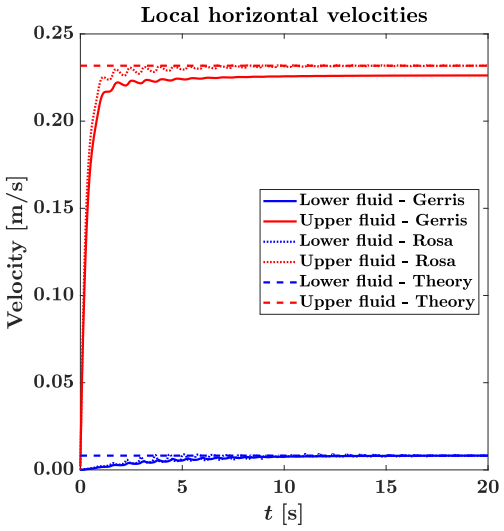
This difference in wave speed causes the large differences between the Gerris and Rosa codes as calculated according to (7.1) (shown in Table 7.2) to persist, even though the behavior of the waves is otherwise similar.

The difference in wave speed between Gerris and Rosa simulations is unsurprising, for it was discussed in chapter 4 and section 5.5 that the models differ fundamentally in this aspect due to the form of the model equations, already without consideration of the closure terms. The inviscid dispersion relations for the test case, plotted in Figure 5.13, indeed show a higher wave speed for the 1D model than for the 2D model. This wave speed vanishes at increasing wavelengths, but apparently is still visible at the wavelength $\lambda = 12H$ considered here.

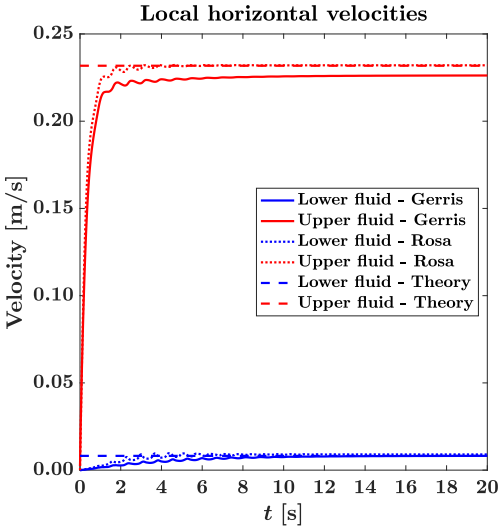
7.5.4 Other networks

The Rosa simulations using the other learned closure terms, listed in Table 7.2, have roughly the same characteristics as the results shown here. The difference for simulations initialized with the 'zero wavy' initial condition is of course the evolution of the averaged velocities to the steady state. The same plots as given in Figure 7.5 are shown in Figure 7.8 for this initial condition.

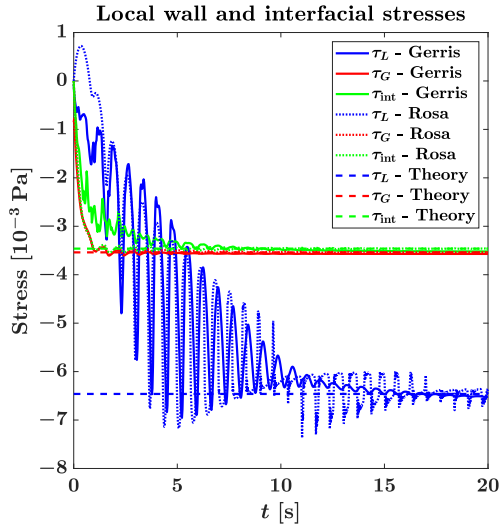
The agreement is once again improved, although in these simulations the liquid stress prediction is off at the beginning of the simulations. This can be attributed to odd behavior of the liquid stress in the initial period of the high fidelity simulations. The liquid stress spikes quite sharply and erratically here, particularly when simulations with low interface height are considered. This might be due to a fault in the high fidelity simulations, or in the calculation of the stresses from the simulations (as discussed in section 5.4 for the interfacial stress).



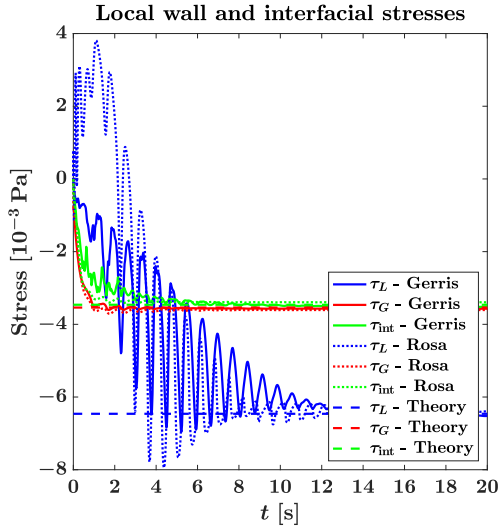
(a) Analytical closure.



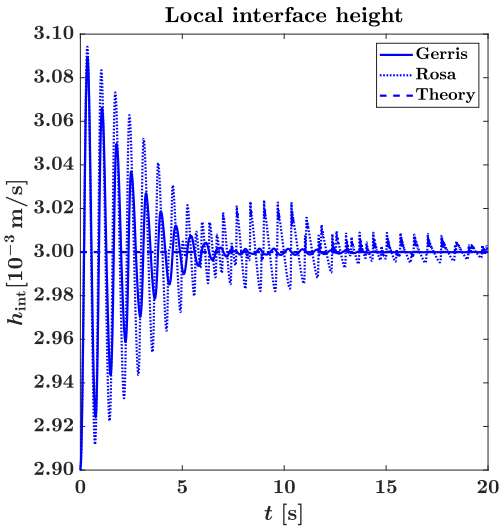
(b) Zero + developed wavy net closure.



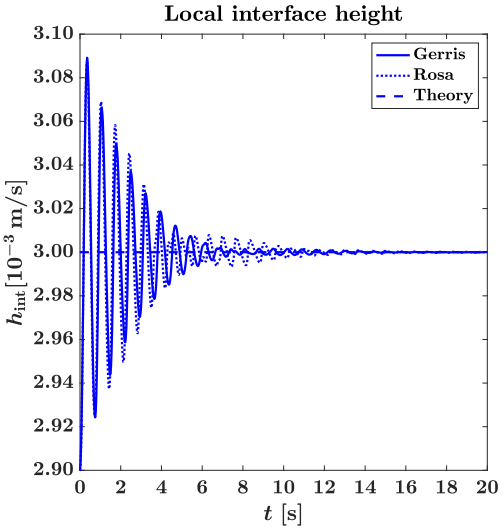
(c) Analytical closure.



(d) Zero + developed wavy net closure.



(e) Analytical closure.



(f) Zero + developed wavy net closure.

Figure 7.8: Evolution in time of the velocities, stresses and interface height at the center of the domain. Initialized with the 'zero wavy' initial condition.

For the network trained solely on training data with the 'zero wavy' initial condition (not pictured), the final liquid velocity is badly missed however: it is predicted to be near zero. Also for the developed wavy case, the network trained on both initial conditions (the 'zero + developed wavy nets') seems to outperform the network trained on only the single initial condition. One factor in this will undoubtedly be the higher amount of data that networks trained on both initial conditions have seen.

Additionally, a regularizing effect may be caused by training on these different flows, with possibly similar patterns connecting the averaged variables and the stresses, and differing noise. This is not at all obvious; one might equally well expect different patterns for the different initial conditions, which would mean that networks would have to be trained for the different initial conditions separately. The finding is advantageous however: it means that our neural network closure can be applied to a wider range of flows.

It should be noted here that neural networks trained on data of only one of the initial conditions show poor correlation coefficients when tested on data of the other initial condition. This effect is particularly strong for the liquid stress. This indicates that our neural network training does not extrapolate well; it only interpolates.

We have also applied neural network closure without bagging (averaging the predictions of multiple networks trained on different subsets of the data). The performance without bagging is significantly worse than the performance with bagging. Small and sharp oscillations are not damped sufficiently.

7.6 Conclusion

We have trained neural networks on unsteady wavy DNS data, with a parameter range ensuring damped wavy flow, starting from two different types of initial conditions. Our networks were observed to correlate well with validation data, indicating that our architecture is suitable. However, leaving the interfacial slope out as a neural network feature was seen to significantly deteriorate the neural network performance. Apparently in this case the network misses the information needed to have a unique relation between the known inputs and the stresses.

We have used only the interfacial slope as an extra input, compared to conventional closure terms. However, we might relatively easily include more inputs, so that with more information the network might make even better predictions. Example extra inputs are velocities and interface heights at surrounding grid points and at preceding time steps.

The neural networks were applied as closure terms in our low fidelity 1D two-fluid model code, Rosa. Using these new closure terms significantly improved agreement with the high fidelity simulations on which they were trained. Particularly the damping behavior of the waves was captured better, when compared to Rosa simulations using conventional analytical closure terms. This is enabled by the closure terms being dependent on the local phase of the wave (via the interface slope). This is different to the conventional wavy interface closure terms of e.g. Andritsos and Hanratty [107], which consider only the spatially averaged effect of a wavy interface on a fully developed flow.

The steady state behavior was also captured satisfactorily. The main difference that remains is the difference in wave speed, which is caused by the differences between the 1D model and 2D model that do not involve the closure terms.

Networks trained on data sets including high fidelity simulations initialized from both initial conditions were found to perform well. This leads us to believe that the relation between averaged variables and stresses has a structure common to both initial conditions, which is beneficial for the generalizing capability of our neural network closure.

Furthermore, averaging the predictions of multiple networks was found to increase the accuracy, and improve the damping behavior.

What is lacking at the moment is an efficient implementation of the neural network forward propagation which allows quick predictions, so that the low-fidelity code remains computationally efficient.

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

In this thesis, we have trained neural networks on high fidelity simulation data to learn closure terms for the wall and interfacial stresses in our low fidelity model; the 1D two-fluid model for stratified channel flow. We used a 2D volume of fluid code as our high fidelity simulation code. The aim of learning closure terms in this way was to get our low fidelity model to give predictions which match the predictions of the high fidelity model better.

Closure terms of the conventional form suffer from a uniqueness problem, which is a result of the cross-sectional averaging of the two-fluid model equations. This is masked by assuming fully developed steady state flow, for which it *is* possible to formulate a unique relation between local averaged quantities and stresses. In this work, the 'conventional' closure terms of Ullmann et al. [9] are shown to not generalize well to unsteady wavy flow.

Another problem with conventional closure terms is their great variety, from which it can be hard to make a choice. Particularly for non-standard geometries, or for wavy interface flow, the choice is not straightforward. For two-phase flow, closure terms based on fully developed flow need to contain assumptions on the form of at least one of the stresses or the relations between them, because the three stresses cannot be determined uniquely from the two steady state balances.

Therefore our approach is to conduct 2D transient simulations of two-phase channel flow with a perturbed interface, and use a neural network to find a relation between the averaged quantities and the stresses. In these simulations, the stresses and the averaged quantities can be calculated locally, and at each instant in time, also for unsteady flow, in a direct manner. As a result, no assumptions are needed, and local effects can be taken into account. In conjunction with the neural network, this allows the addition of extra inputs to the closure relations, alleviating the uniqueness problem; in this work we add the interfacial slope.

Our method can in principle be applied to any geometry and flow conditions for which good high fidelity simulation data is available, without much knowledge or an experimental setup. This provides independence from the closure terms in literature, which might not generalize to the problem at hand.

We tuned and validated our network architecture by experimenting on analytical steady state data. We made a choice for a single network to predict all three stresses; since in literature they are all based on similar dimensionless groups, and the network will be more likely to find these groups if tasked with the prediction of three different stresses. Additionally, using a single network for the three stresses reduces the computation time when applying the network in the low fidelity model.

Training on wavy unsteady simulation data, we have succeeded in generating closure terms for the wall and interfacial stresses in the 1D two-fluid model, which perform better, for wavy unsteady flow, than the reference conventional closure terms of Ullmann et al. [9]. We considered laminar flows with damped waves, initialized from a zero velocity field or with velocities near the steady state velocities. The performance is evaluated by measuring the difference between results of low fidelity simulations using the learned closure terms, and high fidelity simulations. For some of the variables of interest, the space- and time-averaged error is smaller when using the learned closure terms than when using the conventional closure terms.

The learned closure terms excel particularly in reproducing the damping behavior of a wavy perturba-

tion. This is likely largely enabled by the addition of the interfacial slope as an input parameter to the closure relations. This addition is an important novelty in our method, not present in conventional closure terms. It allows the closure terms to have a dependence on the phase of the wave, which goes beyond only a dependence on the averaged velocities and interface height varying over the phase of the wave.

The wave velocity remains different for the 1D and 2D models, as expected from the stability analysis. This explains, for a large part, the significant remaining error between the neural network enhanced low fidelity simulations and the high fidelity simulations, even though the damping behavior is replicated well. The difference in wave velocity is the result of a limit to the degree to which improved expressions for the stresses can be used to improve the agreement between the models, a limit which is shared by our approach and the conventional approach. The difference between the dynamics of the 1D and 2D models is not wholly attributable to deficiencies in the stresses. Other approaches are possible, e.g. finding closure terms such that the difference between low and high fidelity simulations is minimal.

'Bagging' the neural networks, i.e. averaging the predictions of networks trained on different subsets of the training data, was found to be an important factor in improving their performance.

The network trained on both considered initial conditions performed well for simulations starting from both initial conditions. This bodes well for the more general application of neural networks as closure terms for the 1D two-fluid model.

8.2 Recommendations

With our approach validated for a relatively simple case, with low Reynolds numbers, and a smoothly damping wave, we should move on to more challenging cases. In order to be able to use a neural network practically, it needs to be robust to different flow conditions. With the current applicability of the neural network to flow with two different types of initial conditions, it would be interesting to see if the network can generalize to flow with multiple wavelengths, inclined flow, and flows with growing waves (but with the 1D two-fluid model remaining well-posed, and stratified).

It may also be more interesting to consider a traveling wave, rather than the near-standing wave we consider here. For this it would be good to initialize simulations with waves with accompanying velocity perturbations computed via Orr-Sommerfeld viscous stability analysis. This could provide some more insight in the physical behavior of our simulations and the simulations with neural network closure.

Even better results may be obtained by incorporating extra inputs in the closure relations. The success of adding the interfacial slope motivates this. An option is adding averaged velocities at surrounding grid points and preceding time steps. This would be a good application for convolutional layers in the neural network. Similarly, we might add appropriate combinations of the averaged quantities as inputs.

The structure of the learned closure terms should be investigated further. We might evaluate the response of the network directly, when varying input parameters or meaningful ratios and products of them, and compare this to conventional closure relations.

Going further, there is room for improvement in terms of the neural network architecture. For example, by specialization of the network architecture to the current application. This might be possible if we were to train a neural network to reproduce the velocity profile instead of the stresses, and then calculate the stresses from the profile. One could include knowledge about wall boundary conditions and interface conditions in the neural network training step.

Good training data is essential for the capability of a neural network, and we have encountered some problems generating it. In the employed one-fluid model with volume of fluid interface advection, the interfacial stress is a quantity which does not converge with grid resolution and is not continuous across the interface. The interface is not sharp but spans a grid cell. Fortunately, with smart post-processing, good results could still be achieved with the neural network. In the future, even better results might be obtained with for example front tracking methods with a sharp interface.

In the longer term, it would be interesting to see how the method performs for 3D turbulent pipe flow; which is a problem of greater practical interest. The methodology can in principle be applied similarly, if good high fidelity simulations are available (which is not easy for multiphase flow). The difference for a 3D geometry is that the stresses need to be computed all along the wall in the high fidelity simulations and the perimeter average needs to be taken to obtain the stresses needed in the 1D two-fluid model. The performance of the network might be reduced if the interface height is not well-defined (e.g. with an overhanging wave) or if turbulent fluctuations are too large.

Bibliography

[1] G. T. Oud, 'A dual interface method in cylindrical coordinates for two-phase pipe flows', PhD thesis, Delft University of Technology, 2017.

[2] T. Bunnik and A. Veldman, 'Modelling the effect of sloshing on ship motions', *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE*, vol. 1, pp. 279–286, 2010. DOI: 10.1115/OMAE2010-20458.

[3] Royal Dutch Shell. (2018). Shell LNG Outlook 2018, [Online]. Available: https://www.shell.com/energy-and-innovation/natural-gas/liquefied-natural-gas-lng/lng-outlook.html (visited on 04/06/2018).

[4] M. J. Van Twillert, 'The effect of sloshing in partially filled spherical LNG tanks on ship motions', Delft University of Technology, 2015.

[5] R. H. M. Huijsmans, G. Tritschler, G. Gaillarde and R. P. D. Dallinga, 'Sloshing of partially filled LNG carriers', 2004, pp. 424–432.

[6] S. Popinet, 'Gerris: A tree-based adaptive solver for the incompressible Euler equations in complex geometries', *Journal of Computational Physics*, vol. 190, pp. 572–600, 2 2003.

[7] A. J. C. Crespo, J. M. Domínguez, B. D. Rogers, M. Gómez-Gesteira, S. Longshaw, R. Canelas, R. Vacondio, A. Barreiro and O. García-Feal, 'DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH)', *Computer Physics Communications*, vol. 187, pp. 204–216, 2015.

[8] D. Xiu, *Numerical methods for stochastic computations: a spectral method approach*. Princeton, N.J: Princeton University Press, 2010.

[9] A. Ullmann, A. Goldstein, M. Zamir and N. Brauner, 'Closure relations for the shear stresses in two-fluid models for laminar stratified flow', *International Journal of Multiphase Flow*, vol. 30, no. 7, pp. 877–900, 2004. DOI: 10.1016/j.ijmultiphaseflow.2004.03.008.

[10] M. Ma, J. Lu and G. Tryggvason, 'Using statistical learning to close two-fluid multiphase flow equations for a simple bubbly system', *Physics of Fluids*, vol. 27, no. 9, 2015. DOI: 10.1063/L4930004.

[11] ——, 'Using statistical learning to close two-fluid multiphase flow equations for bubbly flows in vertical channels', *International Journal of Multiphase Flow*, vol. 85, pp. 336–347, 2016. DOI: 10.1016/j.ijmultiphaseflow.2016.06.021.

[12] A. Krizhevsky, I. Sutskever and G. E. Hinton, 'ImageNet classification with deep convolutional neural networks', in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12, USA: Curran Associates Inc., 2012, pp. 1097–1105.

[13] J. Weston, S. Chopra and K. Adams, '#TagSpace: Semantic embeddings from hashtags', in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1822–1827. DOI: 10.3115/v1/D14-1194.

[14] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel and D. Hassabis, 'Mastering the game of Go with deep neural networks and tree search', *Nature*, vol. 529, no. 7587, pp. 484–489, 2016. DOI: 10.1038/nature16961.

[15] K. Hornik, M. Stinchcombe and H. White, 'Multilayer feedforward networks are universal approximators', *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989. DOI: 10.1016/0893-6080(89)90020-8.

[16] F. Rosenblatt, 'The perceptron: A probabilistic model for information storage and organization in the brain.', *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958. DOI: 10.1037/h0042519.

[17] D. E. Rumelhart, G. E. Hinton and R. J. Williams, 'Learning representations by back-propagating errors', *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. DOI: 10.1038/323533a0.

[18] C. C. Aggarwal, *Neural Networks and Deep Learning: A Textbook*. Yorktown Heights, NY, USA: Springer International Publishing, 2018. DOI: 10.1007/978-3-319-94463-0.

[19] C. F. Higham and D. J. Higham, 'Deep learning: An introduction for applied mathematicians', *arXiv:1801.05894 [cs, math, stat]*, 2018. arXiv: 1801.05894.

[20] M. T. Hagan, H. B. Demuth, M. H. Beale and O. De Jésus, *Neural Network Design*, 2nd. Martin Hagan, 2014.

[21] F. Sarghini, G. de Felice and S. Santini, 'Neural networks based subgrid scale modeling in large eddy simulations', *Computers & Fluids*, vol. 32, no. 1, pp. 97–108, 2003. DOI: 10.1016/S0045-7930(01)00098-6.

[22] A. Moreau, O. Teytaud and J. P. Bertoglio, 'Optimal estimation for large-eddy simulation of turbulence and application to the analysis of subgrid models', *Physics of Fluids*, vol. 18, no. 10, p. 105 101, 2006. DOI: 10.1063/1.2357974.

[23] I. Esau, 'On application of artificial neural network methods in large-eddy simulations with unresolved urban surfaces', *Modern Applied Science*, vol. 4, no. 8, pp. 3–11, 2010. DOI: 10.5539/mas.v4n8p3.

[24] S. Yarlanki, B. Rajendran and H. Hamann, 'Estimation of turbulence closure coefficients for data centers using machine learning algorithms', in *13th InterSociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, 2012, pp. 38–42. DOI: 10.1109/ITHERM.2012.6231411.

[25] B. Tracey, K. Duraisamy and J. Alonso, 'A machine learning strategy to assist turbulence model development', 2015. DOI: 10.2514/6.2015-1287.

[26] M. Gamahara and Y. Hattori, 'Searching for turbulence models by artificial neural network', *Physical Review Fluids*, vol. 2, no. 5, 2017. DOI: 10.1103/PhysRevFluids.2.054604.

[27] C. Lu, 'Artificial neural network for behavior learning from meso-scale simulations, application to multi-scale multimaterial flows', Master's thesis, University of Iowa, 2010.

[28] C. Lu, S. Sambasivan, A. Kapahi and H. Udaykumar, 'Multi-scale modeling of shock interaction with a cloud of particles using an artificial neural network for model representation', *Procedia IUTAM*, vol. 3, pp. 25–52, 2012. DOI: 10.1016/j.piutam.2012.03.003.

[29] F. Gibou, D. Hyde and R. Fedkiw, 'Sharp interface approaches and deep learning techniques for multiphase flows', *Preprint*, 2018.

[30] W. H. Shayya and S. S. Sablani, 'An artificial neural network for non-iterative calculation of the friction factor in pipeline flow', *Computers and Electronics in Agriculture*, vol. 21, no. 3, pp. 219–228, 1998. DOI: 10.1016/S0168-1699(98)00032-5.

[31] S. S. Sablani and W. H. Shayya, 'Neural network based non-iterative calculation of the friction factor for power law fluids', *Journal of Food Engineering*, vol. 57, no. 4, pp. 327–335, 2003. DOI: 10.1016/S0260-8774(02)00347-3.

[32] S. S. Sablani, W. H. Shayya and A. Kacimov, 'Explicit calculation of the friction factor in pipeline flow of Bingham plastic fluids: A neural network approach', *Chemical Engineering Science*, vol. 58, no. 1, pp. 99–106, 2003. DOI: 10.1016/S0009-2509(02)00440-2.

[33] D. Fadare and U. Ofidhe, 'Artificial neural network model for prediction of friction factor in pipe flow', *Journal of Applied Sciences Research*, vol. 5, no. 6, pp. 662–670, 2009.

[34] Z. Yuhong and H. Wenxin, 'Application of artificial neural network to predict the friction factor of open channel flow', *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 5, pp. 2373–2378, 2009. DOI: 10.1016/j.cnsns.2008.06.020.

[35] F. Salmasi, R. Khatibi and M. Ghorbani, 'A study of friction factor formulation in pipes using artificial intelligence techniques and explicit equations', *Turkish Journal of Engineering and Environmental Sciences*, vol. 36, no. 2, pp. 121–138, 2012. DOI: 10.3906/muh-1008-30.

[36] S. Samadianfard, M. T. Sattari, O. Kisi and H. Kazemi, 'Determining flow friction factor in irrigation pipes using data mining and artificial intelligence approaches', *Applied Artificial Intelligence*, vol. 28, no. 8, pp. 793–813, 2014. DOI: 10.1080/08839514.2014.952923.

[37] S. Besarati, P. Myers, D. Covey and A. Jamali, 'Modeling friction factor in pipeline flow using a GMDH-type neural network', *Cogent Engineering*, vol. 2, no. 1, 2015. DOI: 10.1080/23311916.2015.1056929.

[38] D. Brkić and Z. Ćojbasić, 'Intelligent flow friction estimation', *Computational Intelligence and Neuroscience*, vol. 2016, 2016. DOI: 10.1155/2016/5242596.

[39] S. Haaland, 'Simple and explicit formulas for the friction factor in turbulent pipe flow', *Journal of Fluids Engineering, Transactions of the ASME*, vol. 105, no. 1, pp. 89–90, 1983. DOI: 10.1115/1.3240948.

[40] J. Ling, R. Jones and J. Templeton, 'Machine learning strategies for systems with invariance properties', *Journal of Computational Physics*, vol. 318, pp. 22–35, 2016. DOI: 10.1016/j.jcp.2016.05.003.

[41] S. Popinet, 'An accurate adaptive solver for surface-tension-driven interfacial flows', *Journal of Computational Physics*, vol. 228, no. 16, pp. 5838–5866, 2009. DOI: 10.1016/j.jcp.2009.04.042.

[42] P. Wesseling, *Principles of Computational Fluid Dynamics*. Berlin Heidelberg: Springer-Verlag, 2001.

[43] P. K. Kundu, I. M. Cohen and D. R. Dowling, *Fluid Mechanics*, ser. Academic Press. Academic Press, 2015.

[44] G. Tryggvason, R. Scardovelli and S. Zaleski, *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge University Press, 2011.

[45] V. V. Ranade, 'Numerical solution of model equations', in *Computational Flow Modeling for Chemical Reactor Engineering*, ser. Process Systems Engineering. New York: Academic Press, 2002, vol. 5, pp. 151–189. DOI: 10.1016/S1874-5970(02)80007-8.

[46] H. F. Meier, J. J. N. Alves and M. Mori, 'Comparison between staggered and collocated grids in the finite-volume method performance for single and multi-phase flows', *Computers & Chemical Engineering*, vol. 23, no. 3, pp. 247–262, 1999. DOI: 10.1016/S0098-1354(98)00270-1.

[47] A. Chorin, 'Numerical solution of the Navier-Stokes equations', *Mathematics of Computation*, vol. 22, no. 104, pp. 745–762, 1968. DOI: 10.1090/S0025-5718-1968-0242392-2.

[48] J. Kim and P. Moin, 'Application of a fractional-step method to incompressible Navier-Stokes equations', *Journal of Computational Physics*, vol. 59, no. 2, pp. 308–323, 1985. DOI: 10.1016/0021-9991(85)90148-2.

[49] R. I. Issa, 'Solution of the implicitly discretised fluid flow equations by operator-splitting', *Journal of Computational Physics*, vol. 62, no. 1, pp. 40–65, 1985. DOI: 10.1016/0021-9991(86)90099-9.

[50] C. Hirt and B. Nichols, 'Volume of fluid (VOF) method for the dynamics of free boundaries', *Journal of Computational Physics*, vol. 39, no. 1, pp. 201–225, 1981. DOI: 10.1016/0021-9991(81)90145-5.

[51] D. L. Youngs, 'Time-dependent multi-material flow with large fluid distortion', in *Numerical Methods for Fluid Dynamics*, K. M. Morton and M. J. Baines, Eds., New York: Academic Press, 1982, pp. 27–39.

[52] M. Rudman, 'Volume-tracking methods for interfacial flow calculations', *International Journal for Numerical Methods in Fluids*, vol. 24, no. 7, pp. 671–691, 1997. DOI: 10.1002/(SICI)1097-0363(19970415)24:7<671::AID-FLD508>3.0.CO;2-9.

[53] W. J. Rider and D. B. Kothe, 'Reconstructing volume tracking', *Journal of Computational Physics*, vol. 141, no. 2, pp. 112–152, 1998. DOI: 10.1006/jcph.1998.5906.

[54] S. O. Unverdi and G. Tryggvason, 'A front-tracking method for viscous, incompressible, multi-fluid flows', *Journal of Computational Physics*, vol. 100, no. 1, pp. 25–37, 1992. DOI: 10.1016/0021-9991(92)90307-K.

[55] S. Osher and J. A. Sethian, 'Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations', *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988. DOI: 10.1016/0021-9991(88)90002-2.

[56] M. Sussman, P. Smereka and S. Osher, 'A level set approach for computing solutions to incompressible two-phase flow', *Journal of Computational Physics*, vol. 114, no. 1, pp. 146–159, 1994. DOI: 10.1006/jcph.1994.1155.

[57] Y. C. Chang, T. Y. Hou, B. Merriman and S. Osher, 'A level set formulation of Eulerian interface capturing methods for incompressible fluid flows', *Journal of Computational Physics*, vol. 124, no. 2, pp. 449–464, 1996. DOI: 10.1006/jcph.1996.0072.

[58] J. J. Monaghan, 'Smoothed particle hydrodynamics', *Reports on Progress in Physics*, vol. 68, no. 8, pp. 1703–1759, 2005. DOI: 10.1088/0034-4885/68/8/R01.

[59] M. B. Liu and G. R. Liu, 'Smoothed particle hydrodynamics (SPH): An overview and recent developments', *Archives of Computational Methods in Engineering*, vol. 17, no. 1, pp. 25–76, 2010. DOI: 10.1007/s11831-010-9040-7.

[60] B. Stahl, B. Chopard and J. Latt, 'Measurements of wall shear stress with the lattice Boltzmann method and staircase approximation of boundaries', *Computers & Fluids*, vol. 39, no. 9, pp. 1625–1633, 2010. DOI: 10.1016/j.compfluid.2010.05.015.

[61] FlowKit Ltd. (2012). Palabos home page, [Online]. Available: http://www.palabos.org/ (visited on 14/05/2018).

[62] ANSYS, Inc., *ANSYS Fluent Tutorial Guide*, version 18.0, 2600 ANSYS Drive, Canonsburg, PA 15317, 2017.

[63] OpenCFD Ltd (ESI Group). (2016). OpenFOAM standard solvers, [Online]. Available: https://www.openfoam.com/documentation/user-guide/standard-solvers.php (visited on 14/05/2018).

[64] B. van Leer, 'Towards the ultimate conservative difference scheme. v. a second-order sequel to Godunov's method', *Journal of Computational Physics*, vol. 32, no. 1, pp. 101–136, 1979. DOI: 10.1016/0021-9991(79)90145-1.

[65] A. J. Chorin, 'On the convergence of discrete approximations to the Navier-Stokes equations', *Mathematics of Computation*, vol. 23, no. 106, pp. 341–353, 1969. DOI: 10.1090/S0025-5718-1969-0242393-5.

[66] J. B. Bell, P. Colella and H. M. Glaz, 'A second-order projection method for the incompressible Navier-Stokes equations', *Journal of Computational Physics*, vol. 85, no. 2, pp. 257–283, 1989. DOI: 10.1016/0021-9991(89)90151-4.

[67] A. Almgren, J. Bell and W. Crutchfield, 'Approximate projection methods: Part I. inviscid analysis', *SIAM Journal on Scientific Computing*, vol. 22, no. 4, pp. 1139–1159, 2000. DOI: 10.1137/S1064827599357024.

[68] D. Fuster, A. Bagué, T. Boeck, L. Le Moyne, A. Leboissetier, S. Popinet, P. Ray, R. Scardovelli and S. Zaleski, 'Simulation of primary atomization with an octree adaptive mesh refinement and VOF method', *International Journal of Multiphase Flow*, vol. 35, no. 6, pp. 550–565, 2009. DOI: 10.1016/j.ijmultiphaseflow.2009.02.014.

[69] A. Bagué, D. Fuster, S. Popinet, R. Scardovelli and S. Zaleski, 'Instability growth rate of two-phase mixing layers from a linear eigenvalue problem and an initial-value problem', *Physics of Fluids*, vol. 22, no. 9, p. 092 104, 2010. DOI: 10.1063/1.3483206.

[70] H. Li, J. Li, Z. Zong and Z. Chen, 'Numerical studies on sloshing in rectangular tanks using a tree-based adaptive solver and experimental validation', *Ocean Engineering*, vol. 82, pp. 20–31, 2014. DOI: 10.1016/j.oceaneng.2014.02.011.

[71] X. Yang and A. Turan, 'Simulation of liquid jet atomization coupled with forced perturbation', *Physics of Fluids*, vol. 29, no. 2, p. 022 103, 2017. DOI: 10.1063/1.4976621.

[72] P. A. Wroniszewski, J. C. G. Verschaeve and G. K. Pedersen, 'Benchmarking of Navier-Stokes codes for free surface simulations by means of a solitary wave', *Coastal Engineering*, vol. 91, pp. 1–17, 2014. DOI: 10.1016/j.coastaleng.2014.04.012.

[73] G. Wallis, *One-dimensional two-phase flow*. McGraw-Hill, 1969.

[74] M. Ishii and K. Mishima, 'Two-fluid model and hydrodynamic constitutive relations', *Nuclear Engineering and Design*, vol. 82, no. 2-3, pp. 107–126, 1984. DOI: 10.1016/0029-5493(84)90207-3.

[75] B. Sanderse, 'Development of a two-fluid model for dynamic multiphase pipeline simulations', Shell Global Solutions International B.V., Amsterdam, Tech. Rep. SR.16.10592, 2016.

[76] ——, 'Governing equations of single- and multi-phase flow in one-dimensional pipelines', Shell Global Solutions International B.V., Amsterdam, Tech. Rep. SR.13.13992, 2013.

[77] M. L. De Bertodano, W. Fullmer, A. Clausse and V. H. Ransom, *Two-Fluid Model Stability, Simulation and Chaos*. Springer International Publishing, 2017. DOI: 10.1007/978-3-319-44968-5.

[78] D. Biberg, 'An explicit approximation for the wetted angle in two-phase stratified pipe flow', *Canadian Journal of Chemical Engineering*, vol. 77, no. 6, pp. 1221–1224, 1999. DOI: 10.1002/cjce.5450770619.

[79] R. Bird, W. Stewart and E. Lightfoot, *Transport Phenomena*. Wiley, 2002.

[80] J. R. Welty, G. L. Rorrer and D. G. Foster, *Fundamentals of Momentum, Heat and Mass Transfer*, 6th. Wiley, 2015.

[81] R. J. Cornish, 'Flow in a pipe of rectangular cross section', *Proceedings Royal Society of London*, vol. 120, pp. 691–700, 1928. DOI: 10.1098/rspa.1928.0175.

[82] B. H. Bradford, 'Fluid flow in pipes of rectangular cross sections', Master's thesis, Missouri University of Science and Technology, 1966.

[83] H. Blasius, 'Das Ähnlichkeitsgesetz bei Reibungsvorgängen in Flüssigkeiten.', *Forschg. Arb. Ing.-Wes.*, 131 1913.

[84] B. McKeon, M. Zagarola and A. Smits, 'A new friction factor relationship for fully developed pipe flow', *Journal of Fluid Mechanics*, vol. 538, pp. 429–443, 2005. DOI: 10.1017/S0022112005005501.

[85] C. F. Colebrook and C. M. White, 'Experiments with fluid friction in roughened pipes', *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 161, no. 906, pp. 367–381, 1937. DOI: 10.1098/rspa.1937.0150.

[86] C. F. Colebrook, 'Turbulent flow in pipes, with particular reference to the transition region between the smooth and rough pipe laws', *Journal of the Institution of Civil Engineers*, vol. 11, no. 4, pp. 133–156, 1939. DOI: 10.1680/ijoti.1939.13150.

[87] J. Nikuradse, 'Strömungsgesetze in rauhen rohren', *Zeitschrift des Vereines deutscher Ingenieure*, 361 1933.

[88] L. Prandtl, 'Neuere ergebnisse der turbulenzforschung', *Zeitschrift des Vereines deutscher Ingenieure*, pp. 105–114, 77 1933.

[89] H. Schlichting and K. Gersten, *Boundary-Layer Theory*, 9th ed., ser. Physic and astronomy. Springer-Verlag Berlin Heidelberg, 2017. DOI: 10.1007/978-3-662-52919-5.

[90] S. W. Churchill, 'Friction-factor equation spans all fluid-flow regimes', vol. 84, pp. 91–92, 1977.

[91] C. S. W. and U. R., 'A general expression for the correlation of rates of transfer and other phenomena', *AIChE Journal*, vol. 18, no. 6, pp. 1121–1128, 1972. DOI: 10.1002/aic.690180606.

[92] C. S. W., 'Empirical expressions for the shear stress in turbulent flow in commercial pipe', *AIChE Journal*, vol. 19, no. 2, pp. 375–376, 1973. DOI: 10.1002/aic.690190228.

[93] N. Wilson and R. Azad, 'A continuous prediction method for fully developed laminar, transitional, and turbulent flows in pipes', *Journal of Applied Mechanics, Transactions ASME*, vol. 42, no. 1, pp. 51–54, 1975. DOI: 10.1115/1.3423552.

[94] J. Nikuradse, 'Untersuchungen über turbulente strömungen in nicht kreisförmigen rohren', *Ingenieur-Archiv*, vol. 1, no. 3, pp. 306–332, 1930. DOI: 10.1007/BF02079937.

[95] J. K. Vennard, *Elementary Fluid Mechanics*. Wiley, 1940.

[96] J. Hartnett, J. Koh and S. McComas, 'A comparison of predicted and measured friction factors for turbulent flow through rectangular ducts', *Journal of Heat Transfer*, vol. 84, no. 1, pp. 82–88, 1962. DOI: 10.1115/1.3684299.

[97] O. C. Jones, 'An improvement in the calculation of turbulent friction in rectangular ducts', *Journal of Fluids Engineering*, vol. 98, 2 1976. DOI: 10.1115/1.3448250.

[98] K. Rehme, 'Simple method of predicting friction factors of turbulent flow in non-circular channels', *International Journal of Heat and Mass Transfer*, vol. 16, no. 5, pp. 933–950, 1973. DOI: 10.1016/0017-9310(73)90033-1.

[99] K. Mishima, T. Hibiki and H. Nishihara, 'Some characteristics of gas-liquid flow in narrow rectangular ducts', *International Journal of Multiphase Flow*, vol. 19, no. 1, pp. 115–124, 1993. DOI: 10.1016/0301-9322(93)90027-R.

[100] Z. Duan, M. Yovanovich and Y. Muzychka, 'Pressure drop for fully developed turbulent flow in circular and noncircular ducts', *Journal of Fluids Engineering, Transactions of the ASME*, vol. 134, no. 6, 2012. DOI: 10.1115/1.4006861.

[101] M. Sadatomi, Y. Sato and S. Saruwatari, 'Two-phase flow in vertical noncircular channels', *International Journal of Multiphase Flow*, vol. 8, no. 6, pp. 641–655, 1982. DOI: 10.1016/0301-9322(82)90068-4.

[102] S. He and J. Gotts, 'Calculation of friction coefficients for noncircular channels', *Journal of Fluids Engineering, Transactions of the ASME*, vol. 126, no. 6, pp. 1033–1038, 2004. DOI: 10.1115/1.1845479.

[103] Y. Taitel and A. E. Dukler, 'A model for predicting flow regime transitions in horizontal and near horizontal gas-liquid flow', *AIChE Journal*, vol. 22, no. 1, pp. 47–55, 1976. DOI: 10.1002/aic.690220105.

[104] W. McAdams, *Heat transmission*, ser. McGraw-Hill series in chemical engineering. McGraw-Hill, 1954.

[105] S. Agrawal, G. Gregory and G. Govier, 'An analysis of horizontal stratified two phase flow in pipes', *The Canadian Journal of Chemical Engineering*, vol. 51, no. 3, pp. 280–286, 1973. DOI: 10.1002/cjce.5450510303.

[106] C. Gazley, 'Interfacial shear and stability in two-phase flow', PhD thesis, Univ. Del., Newark, 1949.

[107] N. Andritsos and T. Hanratty, 'Influence of interfacial waves in stratified gas-liquid flows', *AIChE Journal*, vol. 33, no. 3, pp. 444–454, 1987. DOI: 10.1002/aic.690330310.

[108] A. Johnston, 'An investigation into the interfacial shear stress contribution in two-phase stratified flow', *International Journal of Multiphase Flow*, vol. 10, no. 3, pp. 371–383, 1984. DOI: 10.1016/0301-9322(84)90027-2.

[109] N. Cheremisinoff and E. Davis, 'Stratified turbulent-turbulent gas-liquid flow', *AIChE Journal*, vol. 25, no. 1, pp. 48–56, 1979. DOI: 10.1002/aic.690250106.

[110] L. Cohen and T. Hanratty, 'Effect of waves at a gas-liquid interface on a turbulent air flow', *Journal of Fluid Mechanics*, vol. 31, no. 3, pp. 467–479, 1968. DOI: 10.1017/S0022112068000285.

[111] D. Barnea and Y. Taitel, 'Interfacial and structural stability of separated flow', *International Journal of Multiphase Flow*, vol. 20, no. SUPPL. 1, pp. 387–414, 1994. DOI: 10.1016/0301-9322(94)90078-7.

[112] J. Liao, R. Mei and J. Klausner, 'A study on the numerical stability of the two-fluid model near ill-posedness', *International Journal of Multiphase Flow*, vol. 34, no. 11, pp. 1067–1087, 2008. DOI: 10.1016/j.ijmultiphaseflow.2008.02.010.

[113] B. Sanderse, I. E. Smith and M. H. Hendrix, 'Analysis of time integration methods for the compressible two-fluid model for pipe flow simulations', *International Journal of Multiphase Flow*, vol. 95, pp. 155–174, 2017. DOI: 10.1016/j.ijmultiphaseflow.2017.05.002.

[114] J. Kowalski, 'Wall and interfacial shear stress in stratified flow in a horizontal pipe', *AIChE Journal*, vol. 33, no. 2, pp. 274–281, 1987. DOI: 10.1002/aic.690330214.

[115] N. Brauner and D. M. Maron, 'The role of interfacial shear modelling in predicting the stability of stratified two-phase flow', *Chemical Engineering Science*, vol. 48, no. 16, pp. 2867–2879, 1993. DOI: 10.1016/0009-2509(93)80033-M.

[116] D. Biberg, 'A mathematical model for two-phase stratified turbulent duct flow', *Multiphase Science and Technology*, vol. 19, no. 1, pp. 1–48, 2007. DOI: 10.1615/MultScienTechn.v19.i1.10.

[117] M. Espedal, 'An experimental investigation of stratified two-phase pipe flow at small inclinations', PhD thesis, Norwegian University of Science and Technology, 1998.

[118] T. Hanratty, 'Separated flow modelling and interfacial transport phenomena', *Applied Scientific Research*, vol. 48, no. 3-4, pp. 353–390, 1991. DOI: 10.1007/BF02008206.

[119] C. Omgba-Essama, 'Numerical modelling of transient gas-liquid flows (application to stratified & slug flow regimes)', PhD thesis, Cranfield University, 2004.

[120] M. Bentwich, 'Two-phase viscous axial flow in a pipe', *Journal of Fluids Engineering, Transactions of the ASME*, vol. 86, no. 4, pp. 669–672, 1964. DOI: 10.1115/1.3655918.

[121] N. Brauner, J. Rovinsky and D. M. Maron, 'Analytical solution for laminar-laminar two-phase stratified flow in circular conduits', *Chemical Engineering Communications*, vol. 141-142, no. 1, pp. 103–143, 1996. DOI: 10.1080/00986449608936412.

[122] D. Biberg and G. Halvorsen, 'Wall and interfacial shear stress in pressure driven two-phase laminar stratified pipe flow', *International Journal of Multiphase Flow*, vol. 26, no. 10, pp. 1645–1673, 2000. DOI: 10.1016/S0301-9322(99)00109-3.

[123] A. Ullmann, M. Zamir, Z. Ludmer and N. Brauner, 'Stratified laminar countercurrent flow of two liquid phases in inclined tubes', *International Journal of Multiphase Flow*, vol. 29, no. 10, pp. 1583–1604, 2003. DOI: 10.1016/S0301-9322(03)00144-7.

[124] A. Ullmann and N. Brauner, 'Closure relations for two-fluid models for two-phase stratified smooth and stratified wavy flows', *International Journal of Multiphase Flow*, vol. 32, no. 1, pp. 82–105, 2006. DOI: 10.1016/j.ijmultiphaseflow.2005.08.005.

[125] H. Yu and E. Sparrow, 'Experiments on two-component stratified flow in a horizontal duct', *Journal of Heat Transfer*, vol. 91, no. 1, pp. 51–58, 1969. DOI: 10.1115/1.3580119.

[126] B. Sanderse and A. E. P. Veldman, 'Constraint-consistent Runge-Kutta methods for one-dimensional incompressible multiphase flow', *ArXiv e-prints*, 2018. arXiv: 1809.06114 [math.NA].

[127] B. Sanderse, S. Misra, S. Dubinkina, R. A. W. M. Henkes and C. W. Oosterlee, 'Numerical simulation of roll waves in pipelines using the two-fluid model', in *11th North American Conference on Multiphase Production Technology*, Banff, Canada, 2018, pp. 373–386.

[128] S. Gottlieb, C.-W. Shu and E. Tadmor, 'Strong stability-preserving high-order time discretization methods', *SIAM Review*, vol. 43, no. 1, pp. 89–112, 2001. DOI: 10.1137/S003614450036757X.

[129] R. M. M. Mattheij, S. W. Rienstra and J. H. M. Ten Thije Boonkkamp, *Partial Differential Equations, Modeling, Analysis, Computation*. Philadelphia: Society for Industrial and Applied Mathematics (SIAM), 2005.

[130] C. Hirsch, *Numerical Computation of Internal & External Flows*, 2nd ed. Butterworth-Heinemann, 2007, vol. 1.

[131] M. Montini, 'Closure relations of the one-dimensional two-fluid model for the simulation of slug flows', PhD thesis, Imperial College London, 2011.

[132] W. D. Fullmer, V. H. Ransom and M. A. Lopez De Bertodano, 'Linear and nonlinear analysis of an unstable, but well-posed, one-dimensional two-fluid model for two-phase flow based on the inviscid Kelvin-Helmholtz instability', *Nuclear Engineering and Design*, vol. 268, pp. 173–184, 2014.

[133] N. Brauner and D. Maron, 'Stability analysis of stratified liquid-liquid flow', *International Journal of Multiphase Flow*, vol. 18, pp. 103–121, 1 1992. DOI: 10.1016/0301-9322(92)90009-6.

[134] P. G. Drazin and W. H. Reid, *Hydrodynamic Stability*, 2nd ed., ser. Cambridge Mathematical Library. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511616938.

[135] F. T. M. Nieuwstadt, *Turbulentie*, 4th ed. Amsterdam: Epsilon Uitgaven, 2016.

[136] L. M. Milne-Thomson, *Theoretical Hydrodynamics*, 4th ed. Macmillan Company Ltd., 1960.

[137] D. Barnea and Y. Taitel, 'Kelvin-Helmholtz stability criteria for stratified flow: Viscous versus non-viscous (inviscid) approaches', *International Journal of Multiphase Flow*, vol. 19, no. 4, pp. 639–649, 1993. DOI: https://doi.org/10.1016/0301-9322(93)90092-9.

[138] K. Avila, D. Moxey, A. de Lozar, M. Avila, D. Barkley and B. Hof, 'The onset of turbulence in pipe flow', *Science*, vol. 333, no. 6039, pp. 192–196, 2011. DOI: 10.1126/science.1203223.

[139] A. V. Coward, Y. Y. Renardy, M. Renardy and J. R. Richards, 'Temporal evolution of periodic disturbances in two-layer Couette flow', *Journal of Computational Physics*, vol. 132, no. 2, pp. 346–361, 1997. DOI: 10.1006/jcph.1996.5640.

[140] The MathWorks, Inc. (2019). MATLAB Documentation - Deep Learning Toolbox, [Online]. Available: https://nl.mathworks.com/help/deeplearning/index.html (visited on 03/02/2019).

[141] M. Hagan and M. Menhaj, 'Training feedforward networks with the Marquardt algorithm', *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994. DOI: 10.1109/72.329697.

[142] The MathWorks, Inc. (2019). MATLAB Documentation - trainlm, [Online]. Available: https://nl.mathworks.com/help/deeplearning/ref/trainlm.html (visited on 05/02/2019).

[143] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, 'Gradient-based learning applied to document recognition', *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. DOI: 10.1109/5.726791.

[144] Y. LeCun, Fu Jie Huang and L. Bottou, 'Learning methods for generic object recognition with invariance to pose and lighting', in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 2, Washington, DC, USA: IEEE, 2004, pp. 97–104. DOI: 10.1109/CVPR.2004.1315150.

[145] N. Coutris, J. Delhaye and R. Nakach, 'Two-phase flow modelling: The closure issue for a two-layer flow', *International Journal of Multiphase Flow*, vol. 15, no. 6, pp. 977–983, 1989. DOI: 10.1016/0301-9322(89)90025-6.

[146] I. Goodfellow, Y. Bengio and A. Courville, *Deep Learning*. MIT Press, 2016.

[147] Y. Bengio and Y. LeCun, 'Scaling learning algorithms toward AI', in *Large-Scale Kernel Machines*, MITP, 2007.

[148] Y. Bengio, 'Learning deep architectures for AI', *Foundations and Trends in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009. DOI: 10.1561/2200000006.

[149] R. K. Srivastava, K. Greff and J. Schmidhuber, 'Training very deep networks', in *Advances in neural information processing systems 28*, 2015, pp. 2377–2385.

[150] K. He, X. Zhang, S. Ren and J. Sun, 'Deep residual learning for image recognition', in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[151] A. Choromanska, M. Henaff, M. Mathieu, G. Arous and Y. LeCun, 'The loss surfaces of multilayer networks', *Journal of Machine Learning Research*, vol. 38, pp. 192–204, 2015.

[152] Z. Li, H. Zhang, S. C. Bailey, J. B. Hoagg and A. Martin, 'A data-driven adaptive Reynolds-averaged Navier–Stokes k–ω model for turbulent flow', *Journal of Computational Physics*, vol. 345, pp. 111–131, 2017. DOI: 10.1016/j.jcp.2017.05.009.

[153] D. J. C. MacKay, 'Bayesian interpolation', *Neural Computation*, vol. 4, no. 3, pp. 415–447, 1992. DOI: 10.1162/neco.1992.4.3.415.

[154] ——, 'A practical Bayesian framework for backpropagation networks', *Neural Computation*, vol. 4, no. 3, pp. 448–472, 1992. DOI: 10.1162/neco.1992.4.3.448.

[155] F. D. Foresee and M. T. Hagan, 'Gauss-Newton approximation to Bayesian learning', in *Proceedings of International Conference on Neural Networks (ICNN'97)*, vol. 3, 1997, pp. 1930–1935.

[156] D. Nguyen and B. Widrow, 'Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights', 1990, pp. 21–26.

[157] M. D. McKay, R. J. Beckman and W. J. Conover, 'A comparison of three methods for selecting values of input variables in the analysis of output from a computer code', *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979. DOI: 10.2307/1268522.

[158] L. Bottou and O. Bousquet, 'The tradeoffs of large scale learning', in *Advances in Neural Information Processing Systems 20 (NIPS 2007)*, J. Platt, D. Koller, Y. Singer and S. Roweis, Eds., NIPS Foundation (http://books.nips.cc), 2008, pp. 161–168.

[159] L. Breiman, 'Bagging predictors', *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996. DOI: 10.1007/BF00058655.

[160] B. Cushman-Roisin, *Environmental Fluid Mechanics*. Wiley, 2014.

Appendix A

Shallow Water Equations

The two-fluid model system (3.94) may also be used to consider a free surface flow, in which the gas exerts no force on the liquid. We can thus leave out the equations for the motion of the gas. In a free-surface flow, the interface pressure p_{int} is just the constant atmospheric pressure, the term containing its streamwise derivative thus drops out. If we furthermore assume constant density for the liquid, the continuity equation (3.94a) reduces to

$$\frac{\partial h_{\text{int}}}{\partial t} + \frac{\partial u_L h_{\text{int}}}{\partial s} = 0. \tag{A.1}$$

We write the level gradient term for the liquid (found in (3.94c)) as

$$LG_L^* = -\rho_L g \cos\phi h_{\text{int}} \frac{\partial h_{\text{int}}}{\partial s}. \tag{A.2}$$

Then, neglecting the interfacial friction, the momentum balance (3.94c) divided by a constant ρ_L becomes

$$\frac{\partial u_L}{\partial t} h_{\text{int}} + u_L \frac{\partial h_{\text{int}}}{\partial t} + u_L \frac{\partial u_L h_{\text{int}}}{\partial s} + \frac{\partial u_L}{\partial s} u_L h_{\text{int}} = -h_{\text{int}} \frac{\partial h_{\text{int}}}{\partial s} g \cos\phi + \frac{\tau_L}{\rho_L} - h_{\text{int}} g \sin\phi. \tag{A.3}$$

By substituting (A.1) and dividing by h_{int} this is simplified to

$$\frac{\partial u_L}{\partial t} + u_L \frac{\partial u_L}{\partial s} = -\frac{\partial h_{\text{int}}}{\partial s} g \cos\phi + \frac{1}{h_{\text{int}}} \frac{\tau_L}{\rho_L} - g \sin\phi. \tag{A.4}$$

With the closure term (3.71) substituted, equations (A.1) and (A.4) together form the nonlinear shallow water equations in one dimension. These are also known as the Saint-Venant equations (for 2D channel flow).

The system can be generalized to an arbitrary open channel geometry with any $h_{\text{int}} = h_{\text{int}}(A_L)$ and $P_L = P_L(A_L)$. To do this follow the same derivation as described above, but starting with (3.10) and (3.17) and using (3.23) for the level gradient. The resulting equations are [160]

$$\frac{\partial A_L}{\partial t} + \frac{\partial u_L A_L}{\partial s} = 0, \tag{A.5a}$$

$$\frac{\partial u_L}{\partial t} + u_L \frac{\partial u_L}{\partial s} = -\frac{\partial h_{\text{int}}}{\partial s} g \cos\phi + \frac{P_L}{A_L} \frac{\tau_L}{\rho_L} - g \sin\phi. \tag{A.5b}$$

Appendix B

Classification of PDEs

B.1 Introduction

In section 4.2 we discussed the well-posedness of the 1D two-fluid model. A sufficient condition for well-posedness of the system is that the system is hyperbolic. Here we demonstrate two methods for the classification of hyperbolic PDEs, and the determination of their characteristics. Both methods for classification are shown to reduce to finding eigenvalues c

$$\det|\mathbf{B} - c\mathbf{A}| = 0. \tag{B.1}$$

of the system

$$\mathbf{A}\frac{\partial}{\partial t}\mathbf{u} + \mathbf{B}\frac{\partial}{\partial x}\mathbf{u} = 0. \tag{B.2}$$

An example linear partial differential equation which is easy to classify is the wave equation

$$\frac{\partial^2\phi}{\partial t^2} = a\frac{\partial^2\phi}{\partial x^2}. \tag{B.3}$$

It (along with many quasi-linear second order PDEs [130, p. 116]) can be rewritten as a system of first order PDE's

$$\frac{\partial u}{\partial t} - a\frac{\partial v}{\partial x} = 0 \tag{B.4a}$$

$$\frac{\partial u}{\partial x} - \frac{\partial v}{\partial t} = 0 \tag{B.4b}$$

using the substitutions $u = \partial\phi/\partial t$ and $v = \partial\phi/\partial x$. In matrix form this system reads

$$\mathbf{A}\frac{\partial}{\partial t}\mathbf{u} + \mathbf{B}\frac{\partial}{\partial x}\mathbf{u} = 0, \tag{B.5}$$

with

$$\mathbf{A} = \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix}0 & -a\\ 1 & 0\end{bmatrix}, \quad \text{and} \quad \mathbf{u} = \begin{bmatrix}u\\ v\end{bmatrix}.$$

B.2 Classification based on the wave form

We substitute the trial solution

$$\mathbf{u} = \hat{\mathbf{u}}\exp\left[i\left(kx - \omega t\right)\right], \tag{B.6}$$

with k the wavenumber and ω the angular frequency, which dictates the change of the solution with time (and is thus of interest in determining whether the system is hyperbolic). This yields

$$\left[-i\omega\mathbf{A} + ik\mathbf{B}\right]\hat{\mathbf{u}} = \mathbf{0}, \tag{B.7}$$

which has non-trivial solutions for $\hat{\mathbf{u}}$ (that is, solutions other than $\hat{\mathbf{u}} = \mathbf{0}$) if the determinant of $-\omega\mathbf{A} + k\mathbf{B}$ is zero. This condition can be written as the generalized eigenvalue problem

$$\det|\mathbf{B} - c\mathbf{A}| = 0, \tag{B.8}$$

with $c = \omega/k$ and yields

$$c = \pm\sqrt{a}. \tag{B.9}$$

This can also be written as the dispersion relationship

$$\omega = \pm\sqrt{a}k, \tag{B.10}$$

so that ω and c are real if $a > 0$.

If $a < 0$, ω will have an imaginary component, leading to growth of the wave-like solution (B.6), so that the system is not hyperbolic according to the aforementioned definition by [130]. Moreover, the growth will tend to infinity for wavelengths tending to zero, since for these wavelengths the wavenumber k will tend to infinity. Thus the problem will be ill-posed. If $a > 0$ the wave-like solution will simply propagate without growth or damping and the system is hyperbolic.

The dispersion relationship determines the characteristics; it states that the phase speed of the two possible waves is $\omega/k = \pm\sqrt{a}$. This means that a wave crest will move at a speed $\pm\sqrt{a}$, so that the wave function can be said to be constant along characteristic lines

$$\left[\frac{\mathrm{d}x}{\mathrm{d}t}\right]_1 = \sqrt{a} \quad \rightarrow \quad x - \sqrt{a}t = C \tag{B.11}$$

and

$$\left[\frac{\mathrm{d}x}{\mathrm{d}t}\right]_2 = -\sqrt{a} \quad \rightarrow \quad x + \sqrt{a}t = C \tag{B.12}$$

with C an arbitrary constant. The functions which are constant along these characteristic lines are the solutions $\mathbf{u}$, because we have taken the trial solution (B.6). We must still find the solution for $\hat{\mathbf{u}}$, i.e. the generalized right eigenvectors of $\mathbf{B} - c\mathbf{A}$. These can easily be determined to be

$$c_1 = \sqrt{a} \quad \rightarrow \quad \hat{\mathbf{u}} = \begin{bmatrix} -\sqrt{a} \\ 1 \end{bmatrix} \tag{B.13}$$

and

$$c_2 = -\sqrt{a} \quad \rightarrow \quad \hat{\mathbf{u}} = \begin{bmatrix} \sqrt{a} \\ 1 \end{bmatrix}. \tag{B.14}$$

B.3 Classification based on characteristics

An alternative definition of a hyperbolic system is that the system can be rewritten to a system of ordinary differential equations (ODEs) with real independent variables only. For the wave equation this may be found by adding (B.4a) and (B.4b) multiplied by a constant c to yield

$$\left(\frac{\partial}{\partial t} + c\frac{\partial}{\partial x}\right)(u) + \left(\frac{\partial}{\partial t} + \frac{a}{c}\frac{\partial}{\partial x}\right)(-cv) = 0. \tag{B.15}$$

This is an ODE if u and $-cv$ are differentiated in the same, real, direction. They are differentiated in the same direction if

$$c^2 = a, \tag{B.16}$$

and $c = \pm\sqrt{a}$ and the two corresponding directions are real if $a > 0$. This yields the system of ODE's

$$c_1 = \sqrt{a} \quad \rightarrow \quad \frac{\partial}{\partial s_1}(u - \sqrt{a}v) = 0, \quad \frac{\partial}{\partial s_1} = \left(\frac{\partial}{\partial t} + \sqrt{a}\frac{\partial}{\partial x}\right), \tag{B.17a}$$

$$c_2 = -\sqrt{a} \quad \rightarrow \quad \frac{\partial}{\partial s_2}(u + \sqrt{a}v) = 0, \quad \frac{\partial}{\partial s_2} = \left(\frac{\partial}{\partial t} - \sqrt{a}\frac{\partial}{\partial x}\right). \tag{B.17b}$$

The lines along which the so-called Riemann invariants $u + \sqrt{a}v$ and $u - \sqrt{a}v$ are constant are called characteristics. The first can be found by demanding this for the first Riemann invariant:

$$\frac{\mathrm{d}}{\mathrm{d}t}(u - \sqrt{a}v) = \frac{\partial}{\partial t}(u - \sqrt{a}v) + \frac{\partial x}{\partial t}\frac{\partial}{\partial x}(u - \sqrt{a}v) = 0, \tag{B.18}$$

which by comparison with (B.17a) yields

$$\frac{\partial x}{\partial t} = \sqrt{a} \quad \rightarrow \quad x - \sqrt{a}t = C, \tag{B.19}$$

with C an arbitrary constant. The second characteristic is found similarly and is given by

$$\frac{\partial x}{\partial t} = -\sqrt{a} \quad \rightarrow \quad x + \sqrt{a}t = C. \tag{B.20}$$

The system (B.17a), (B.17b) can be found in a more general way by considering that a general linear combination of (B.4a) and (B.4b) can be written as (using (B.5))

$$\mathbf{l}_i \cdot \left(\mathbf{A}\frac{\partial}{\partial t}\mathbf{u} + \mathbf{B}\frac{\partial}{\partial x}\mathbf{u} \right) = 0. \tag{B.21}$$

We hope to find combinations such that

$$\mathbf{l}_i \cdot \left(\mathbf{A}\frac{\partial}{\partial t}\mathbf{u} + \mathbf{B}\frac{\partial}{\partial x}\mathbf{u} \right) = \mathbf{l}_i \cdot \mathbf{C}\frac{\partial}{\partial s_i}\mathbf{u} = \frac{\partial w_i}{\partial s_i} = 0, \tag{B.22}$$

where $\mathbf{w}$ are the Riemann invariants and the index i specifies one of two valid linear transformations (and corresponding Riemann invariants and characteristic directions).

These combinations may be found by performing the coordinate transformation

$$\frac{\partial}{\partial s_i} = \left[\frac{\partial t}{\partial s}\right]_i \left(\frac{\partial}{\partial t} + \left[\frac{\partial x}{\partial t}\right]_i \frac{\partial}{\partial x} \right) = \left(\frac{\partial}{\partial t} + c_i \frac{\partial}{\partial x} \right), \tag{B.23}$$

where we take the freedom to fix $[\partial t/\partial s]_i = 1$ and define $c_i = [\partial x/\partial t]_i$. Then (B.21) can be written as

$$\mathbf{l}_i \cdot \left(\mathbf{A}\frac{\partial}{\partial t}\mathbf{u} + \mathbf{B}\frac{\partial}{\partial x}\mathbf{u} \right) = \mathbf{l}_i \cdot \left(\mathbf{A}\left(\frac{\partial}{\partial s_i} - c_i\frac{\partial}{\partial x} \right)\mathbf{u} + \mathbf{B}\frac{\partial}{\partial x}\mathbf{u} \right), \tag{B.24}$$

which is an ODE to s_i if

$$\mathbf{l}_i \cdot (\mathbf{B} - c_i\mathbf{A})\frac{\partial}{\partial x}\mathbf{u} = 0. \tag{B.25}$$

The $\partial\mathbf{u}/\partial x$ will generally not be zero, so for this system to hold generally we must have

$$\mathbf{l}_i \cdot (\mathbf{B} - c_i\mathbf{A}) = 0, \tag{B.26}$$

for which non-trivial solutions will exist if and only if the determinant of $\mathbf{B} - c_i\mathbf{A}$ is zero. This yields the eigenvalues

$$c_1 = \sqrt{a}, \qquad c_2 = -\sqrt{a}, \tag{B.27}$$

which are real if $a > 0$. The solutions for the left eigenvectors which satisfy the generalized eigenvalue problem (B.26) can then be found to be

$$\mathbf{l}_1 = \begin{bmatrix} 1 & \sqrt{a} \end{bmatrix}, \qquad \mathbf{l}_2 = \begin{bmatrix} 1 & -\sqrt{a} \end{bmatrix}. \tag{B.28}$$

Substituting these eigenvectors into (B.21) and keeping in mind the definition (B.23) we find exactly the system of ODE's (B.17a), (B.17b). Alternatively we could recognize by substitution of (B.25) into (B.24) and subsequent comparison to (B.22) that $\mathbf{C} = \mathbf{A}$. Then, since $\mathbf{l}_i$ and $\mathbf{A}$ contain only constants we can find the Riemann invariants w_1 and w_2 by computing

$$w_1 = \mathbf{l}_1\mathbf{A}\mathbf{u} = u - \sqrt{a}v, \qquad w_2 = \mathbf{l}_2\mathbf{A}\mathbf{u} = u + \sqrt{a}v, \tag{B.29}$$

so that the system can be written as

$$\frac{\partial w_1}{\partial s_1} = 0, \qquad \frac{\partial w_2}{\partial s_2} = 0. \tag{B.30}$$

The characteristic lines can be found in the same way as before or by directly identifying $c_i = [\partial x/\partial t]_i$ (from the definition in (B.23)).