

## MACHINE LEARNING FOR CLOSURE MODELS IN MULTIPHASE FLOW APPLICATIONS

**Jurriaan Buist<sup>1,2</sup>, Benjamin Sanderse<sup>1</sup>, Yous van Halder<sup>1</sup>,  
Barry Koren<sup>2</sup>, and GertJan van Heijst<sup>2</sup>**

<sup>1</sup>Centrum Wiskunde & Informatica  
Science Park 123, 1098XG Amsterdam  
j.f.h.buist@cwj.nl, b.sanderse@cwj.nl, y.van.halder@cwj.nl

<sup>2</sup>Eindhoven University of Technology  
PO Box 513, 5600 MB Eindhoven  
b.koren@tue.nl, g.j.f.v.heijst@tue.nl

**Keywords:** Machine Learning, Closure Terms, Multiphase Flow, Two-Fluid Model

**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 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. The neural network 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.*

## 1 INTRODUCTION

The simulation of multiphase flow of gas and liquid in a pipeline is a problem of interest in the oil and gas industry. The two fluids can have complex interactions leading to different flow regimes, such as smoothly stratified flow, wavy stratified flow, and slug flow. Predicting the transition from stratified flow to slug flow in dynamic simulations is a difficult problem [21], for which we consider different computational models. We restrict ourselves in this paper to incompressible 2D channel flow, as a simplified representation of 3D circular pipe flow.

A general model which can describe the flow regimes mentioned above is formed by the well-known Navier-Stokes equations. These can be solved numerically, using for example a volume-of-fluid (VOF) method [19] for the treatment of the interface. However, when many model evaluations are needed, such as in uncertainty quantification, solving the full Navier-Stokes equations is too computationally expensive.

We therefore consider a simplified model which is computationally less expensive, the so-called two-fluid model [3]. The 1D two-fluid model is obtained by averaging the Navier-Stokes equations for each fluid, over the respective cross-sections. This spatial averaging process introduces a closure problem; the shear stresses in the flow become unknowns with a priori no direct relation to the averaged quantities present in the two-fluid model. Relations between the averaged quantities and the stresses need to be postulated; these relations are called ‘closure terms’.

Conventionally, these closure terms for the two-fluid model are obtained from correlations with experimental data for steady state pipe flow. A pressure difference is applied to a section of the pipe and the resulting volumetric fluxes and liquid holdup (fraction of the total pipe cross-section occupied by the liquid) are measured. These are related to the stresses via the steady state balances for both fluids and via assumptions on the relations between the different stresses, to form the closure terms. On this principle, for example, the widely used Taitel and Dukler [40] closure terms are based.

Alizadehdakhel et al. [1] and Osgouei et al. [32] used physical experimental data to train neural networks to predict pressure drops in two-phase pipe flow. They related the superficial velocities to the spatially and temporally averaged pressure drop, like in the conventional approach, but used a neural network to construct the relation.

For bubbly flow in a vertical channel, Ma et al. [26, 27], introduced a more general approach. They conducted 3D unsteady Navier-Stokes DNS (with front tracking), the results of which can be related to the averaged quantities present in their low-fidelity 1D model, at any point along the 1D model’s spatial axis and at any point in time. A neural network was employed to learn the relation between the two. They report satisfactory results, and emphasize the general applicability of their approach: no prior knowledge is needed on the relation between known quantities and the quantities requiring closure. Ma et al. refer to earlier work by Lu et al. [24, 25], 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. Besides these references, in multiphase flow, the literature on machine learning for closure terms is sparse.

However, in the field of turbulence closure modeling, neural networks have already proved their worth, when applied to specific cases. Sargini et al. [38] 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. A similar approach was taken by Tracey et al. [42] for air flow in a data

center. Gamahara and Hattori [16] recently used DNS directly to obtain a functional relation for the SGS tensor which shows performance close to that of a Smagorinsky SGS model. Ling et al. [23] learned RANS stress tensors similarly, choosing the inputs and neural network structure such that Galilean invariance is incorporated in the expressions directly.

Motivated by the success of machine learning in the field of turbulence closure modeling, we continue the application of machine learning to multiphase flow, taking inspiration from Ma et al. [26, 27]. We make two new contributions:

- We extend the methodology to more generic neural networks.
- We study a different physical situation (stratified flow versus bubbly flow), with a different low-fidelity model, and different unclosed terms.

Compared to the conventional literature on closure terms for the 1D two-fluid model, the novelties of this work are:

- We base closure terms on the results of fully resolved unsteady 2D Navier-Stokes simulations (for channel flow), which we refer to as our high-fidelity simulations.
- We employ an artificial neural network to find the relation between the two.
- The preceding two points make it straightforward to add non-local, non-instantaneous input variables to the closure relations; in this work we have added the streamwise derivative of the interface height.

The differences between our approach and the conventional approach mean that:

- Closure terms can be constructed for specific cases (specific duct geometries or flow regimes), as long as accurate high-fidelity simulations are available.
- Unsteady behavior may be reproduced more accurately by the low-fidelity model.

With our approach, it is our aim to use high-fidelity simulations to improve the accuracy of low-fidelity simulations, with the promise to reach the accuracy of the high-fidelity model at the cost of the low-fidelity model.

The structure of the paper is as follows. The physics and numerics of the high- and low-fidelity models are discussed in section 2. This leads us to an explanation of the required closure terms, and the fundamental limitations imposed by the model averaging process, which are not fixable by improving closure terms of the considered form.

In section 3 we tune the neural network, and show that closure terms based on steady state flow are unsatisfactory for the case of wavy unsteady flow. We then describe the training of the tuned neural network on wavy unsteady flow data. Finally, section 4 presents the results of applying the trained neural networks as closure terms in the low-fidelity simulations. Here the agreement between the high-fidelity model and the enhanced low-fidelity model is evaluated.

## 2 HIGH- AND LOW-FIDELITY MODEL DESCRIPTION

Our approach is shown schematically in Figure 1. We have a 2D high-fidelity model for channel flow, with horizontal and vertical velocity components  $u$  and  $w$ , being functions of the coordinates  $s$  and  $h$ . The low-fidelity model is 1D and as such only knows velocities  $u_L$  and  $u_G$ , which are averaged over the portions of the channel containing liquid and gas respectively, so

that they are only functions of  $s$  (as is the interface height  $h_{\text{int}}$ ). From the 2D high-fidelity field results we calculate these averaged quantities and their corresponding stresses. These represent inputs and desired outputs to a neural network, respectively, between which the neural network is given the task to find a relation. The resulting functions can be fed as closure terms to the low-fidelity model.

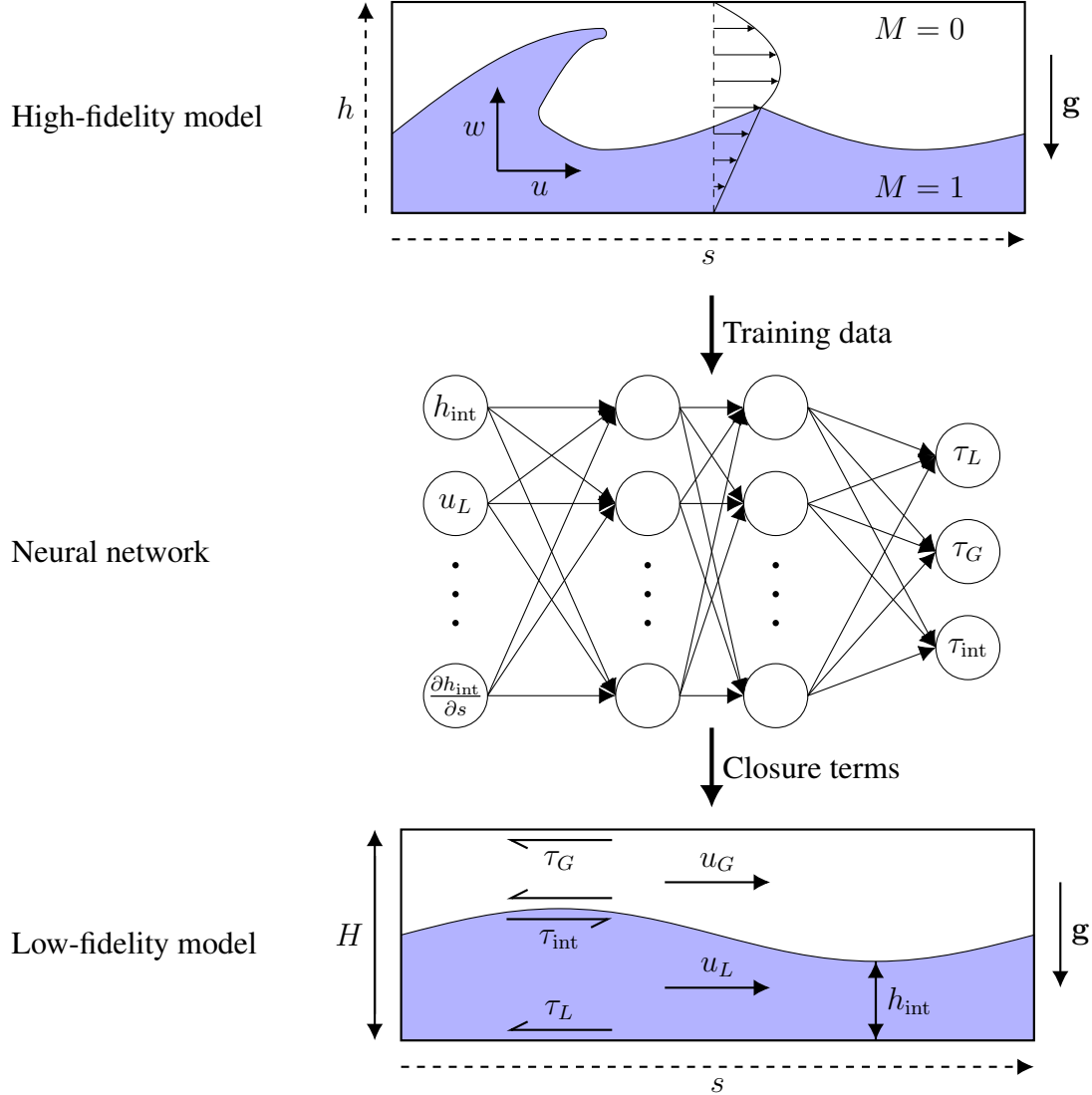


Figure 1: An outline of the approach for learning closure terms from high-fidelity simulations.

## 2.1 High-fidelity model

The high-fidelity model that we use to generate the training data is the open source code Gerris [33, 34]. It is based on the one-fluid formulation for multiphase flow. This entails the solution of the Navier-Stokes equations for incompressible flow:

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{1}{\rho} (-\nabla p + \nabla \cdot [\mu \nabla \mathbf{u} + \mu (\nabla \mathbf{u})^T]) + \mathbf{g}, \quad (2)$$

with velocity field  $\mathbf{u} = \mathbf{u}(s, h, t)$  and pressure field  $p = p(s, h, t)$  encompassing the entire domain, gravitational acceleration  $\mathbf{g}$ , density  $\rho$ , and viscosity  $\mu$  (see Figure 1).

Gerris discretizes these equations spatially with a finite volume method on a collocated grid, with central interpolation and the Van Leer generalized minmod limiter with  $\theta = 2$  for the face-centered gradient calculation. We do not make use of Gerris' capability to adaptively refine the grid at different levels.

For temporal discretization Gerris uses a second order projection method [11], in which a multilevel Gauss-Seidel iterative method is used to solve the pressure Poisson equation. The velocity advection term is discretized according to the second order unsplit upwind scheme of Bell et al. [4], and for the diffusion term a Crank-Nicolson discretization is employed.

In the one-fluid approach for multiphase flow, the density  $\rho$  and viscosity  $\mu$  are functions of the spatial coordinates, via a marker function  $M = M(s, h, t)$ :

$$\rho = \rho(M), \quad \mu = \mu(M).$$

This marker function  $M$ , typically 1 in the liquid and 0 in the gas, is advected by the velocity field. We make the *assumption of sharp interfaces* [43, p. 22] and disregard phase transition, so that the advection of the marker function can be described by

$$\frac{DM}{Dt} = \frac{\partial M}{\partial t} + \mathbf{u} \cdot \nabla M = 0. \quad (3)$$

Gerris advects the marker function numerically using the volume-of-fluid (VOF) method. In the VOF method [19], the marker function is averaged over the grid cells to define the color function

$$C_i = \frac{1}{V_i} \int_{V_i} M \, dV. \quad (4)$$

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. We use the expressions

$$\rho_i = C_i \rho_1 + (1 - C_i) \rho_0, \quad (5)$$

$$\mu_i = \left( \frac{C_i}{\mu_1} + \frac{1 - C_i}{\mu_0} \right)^{-1}. \quad (6)$$

with  $\rho_1$  and  $\mu_1$  the density and viscosity of the fluid indicated by  $M = 1$  and  $\rho_0$  and  $\mu_0$  the fluid indicated by  $M = 0$ . For the viscosity we do not use an arithmetic mean but rather the harmonic mean [14], which improves the accuracy of the velocities and stresses at a flat, horizontal interface.

Prior to the actual advection step, the interface is reconstructed from the color function using the PLIC method [47]. The color function is then advected geometrically by the velocity field.

We use most of the standard Gerris settings, except that we lower the tolerance of the projection steps from  $1 \cdot 10^{-3}$  to  $1 \cdot 10^{-6}$ . After a convergence study, the grid spacing  $\Delta s = \Delta h$  is set to  $H/64$ , and the time step is set so that the maximum value of

$$\text{CFL} = \frac{|\mathbf{u}| \Delta t}{\Delta s} \quad (7)$$

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.

We choose the one-fluid formulation for multiphase flow with the VOF interface advection method for its conservative properties, its simplicity, and its similarity to our low-fidelity model. Alternative interface advection methods for the one-fluid formulation of multiphase flow include the front tracking [46] and level-set methods [39], but these are not naturally mass conservative.

## 2.2 Low-fidelity model

Our low-fidelity model is known as the 1D two-fluid model. It is obtained by considering control volumes in a channel, separate for liquid and gas, as pictured in Figure 2. The limit  $\delta s \rightarrow 0$  is taken, while the control volumes fill the full channel height. Since we do not consider phase change, the velocities are continuous at the interface. The stresses tangential to the interface, the shear stresses, must be continuous, and since we assume hydrostatic balance (without surface tension) the pressure should be continuous along the vertical direction, as well as the stresses along the vertical direction.

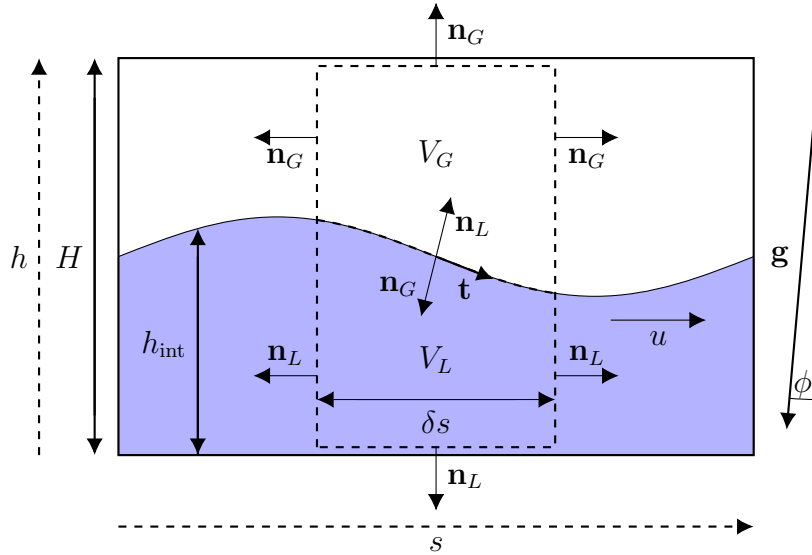


Figure 2: Two small ( $\delta s \ll H$ ) control volumes for two-phase pipe flow. At the top and bottom the control volume is bounded by impenetrable no-slip boundaries. The interface separates the two control volumes.

We obtain one equation for mass balance and one for momentum balance for each fluid. In channel flow these take the following form:

$$\frac{\partial}{\partial t} (\rho_L h_{\text{int}}) + \frac{\partial}{\partial s} (\rho_L u_L h_{\text{int}}) = 0, \quad (8a)$$

$$\frac{\partial}{\partial t} (\rho_G (H - h_{\text{int}})) + \frac{\partial}{\partial s} (\rho_G u_G (H - h_{\text{int}})) = 0, \quad (8b)$$

$$\begin{aligned} \frac{\partial}{\partial t} (\rho_L u_L h_{\text{int}}) + \frac{\partial}{\partial s} (\rho_L u_L^2 h_{\text{int}}) = & - \frac{\partial p_{\text{int}}}{\partial s} h_{\text{int}} + L G_L + F_L \\ & - \rho_L h_{\text{int}} g \sin(\phi), \end{aligned} \quad (8c)$$

$$\begin{aligned} \frac{\partial}{\partial t} (\rho_G u_G (H - h_{\text{int}})) + \frac{\partial}{\partial s} (\rho_G u_G^2 (H - h_{\text{int}})) = & - \frac{\partial p_{\text{int}}}{\partial s} (H - h_{\text{int}}) + L G_G + F_G \\ & - \rho_G (H - h_{\text{int}}) g \sin(\phi), \end{aligned} \quad (8d)$$

with  $u_L$  and  $u_G$  the averaged velocities of the liquid and gas respectively,  $\rho_L$  and  $\rho_G$  likewise for the densities,  $h_{\text{int}}$  the interface height,  $p_{\text{int}}$  the interfacial pressure, and  $\phi$  the channel inclination. Here the stresses are bundled into closure terms

$$F_L = \tau_L - \tau_{\text{int}}, \quad F_G = \tau_G + \tau_{\text{int}}, \quad (9)$$

and the level gradient terms represent

$$LG_L = -\frac{\partial}{\partial s} \left[ \frac{1}{2} \rho_L g \cos(\phi) h_{\text{int}}^2 \right], \quad LG_G = \frac{\partial}{\partial s} \left[ \frac{1}{2} \rho_G g \cos(\phi) (H - h_{\text{int}})^2 \right]. \quad (10)$$

The equations are of the same form as those for pipe flow in circular cross-sections, but with different relations between the cross-sections, perimeters, and interface height (see e.g. [36]).

In this research the ‘Rosa’ code developed by Sanderse et al. [35, 36, 37] is employed for solving the incompressible form of (8).

The code discretizes the equations using a finite volume method on a staggered grid. This allows for a strong and straightforward coupling between pressure and velocity. Interpolation is needed for the convective scheme: here we employ a central interpolation, which ensures second order spatial accuracy.

After the system is discretized spatially, the time stepping is considered. We use the constraint-consistent time integration framework for the incompressible two-fluid model presented in [36], with the three-stage, third order strong-stability preserving Runge-Kutta method referenced in [37], which follows Gottlieb et al. [17].

### 2.3 Closure terms

The liquid wall stress  $\tau_L$ , gas wall stress  $\tau_G$ , and interfacial stress  $\tau_{\text{int}}$ , which appear in (9), represent the stresses acting in the streamwise direction:

$$\tau = (\boldsymbol{\tau} \cdot \mathbf{n}) \cdot \hat{\mathbf{s}}, \quad (11)$$

with  $\boldsymbol{\tau}$  the stress tensor and  $\hat{\mathbf{s}}$  the unit vector along the  $s$ -axis. Accounting for the no-slip boundary conditions, assuming hydrostatic balance and horizontal length scales far larger than the vertical length scale, they are related to the velocity profile via

$$\tau_L = -\mu_L \frac{\partial u}{\partial h} \Big|_{h \downarrow 0}, \quad \tau_G = \mu_G \frac{\partial u}{\partial h} \Big|_{h \uparrow H}, \quad \tau_{\text{int}} = -\mu_G \frac{\partial u}{\partial h} \Big|_{h \downarrow h_{\text{int}}} = \mu_L \frac{\partial u}{\partial h} \Big|_{h \uparrow h_{\text{int}}}, \quad (12)$$

in which  $x \uparrow y$  and  $x \downarrow y$  are limits from below and from above respectively (see [10] for a more detailed discussion).

These stresses are a priori unknown in the 1D two-fluid model, since in this model the velocities are not resolved in the transverse direction, meaning that the stresses cannot be calculated according to (12). Conventionally, steady state experiments are employed to correlate the stresses to the averaged quantities through the steady state balance, essentially implying a streamwise and temporally averaged description of the flow. This yields relations of the form<sup>1</sup>

$$\tau_L, \tau_G, \tau_{\text{int}} = f(h_{\text{int}}, u_L, u_G, \rho_L, \rho_G, \mu_L, \mu_G, H), \quad (13)$$

<sup>1</sup>Expressions for the stresses based on the body forces as opposed to the averaged velocities do not close the steady state equations [13], and cannot generalize to unsteady flow.

in which all the variables on the right-hand side *are* known in the 1D two-fluid model.

Many experiments are needed to obtain good relations, and therefore it may be difficult to find closure terms in the literature which generalize well to the case at hand. Furthermore, when considering wavy flow, with this method of generation of closure terms only the averaged (positive) effect of waves on the interfacial friction can be taken into account; local effects are averaged out.

For the strongly simplified case of laminar, flat interface, fully developed, steady channel flow, Ullmann et al. [45] have derived analytical solutions for the stresses of the form (13). They are given by

$$\tau_L = -\frac{1}{2}f_L\rho_L u_L|u_L|F_L^*, \quad \tau_G = -\frac{1}{2}f_G\rho_G u_G|u_G|F_G^*, \quad \tau_{\text{int}} = -\frac{1}{2}f_G\rho_G (u_G - u_L)|u_G|F_{\text{int},G}^*, \quad (14)$$

with friction factors

$$f_L = \frac{3}{2} \frac{16}{\text{Re}_L}, \quad f_G = \frac{3}{2} \frac{16}{\text{Re}_G}, \quad (15)$$

depending on Reynolds numbers

$$\text{Re}_L = \frac{\rho_L|u_L|D_L}{\mu_L}, \quad \text{Re}_G = \frac{\rho_G|u_G|D_G}{\mu_G}, \quad (16)$$

based on hydraulic diameters

$$D_L = 2h_{\text{int}}, \quad D_G = 2(H - h_{\text{int}}). \quad (17)$$

$F_L^*$  and  $F_G^*$  are the two-phase correction factors for the wall friction:

$$F_L^* = \frac{1 + \frac{1}{2} \frac{u_G}{u_L} \left[ \frac{\mu_L u_L}{\mu_G u_G} \frac{H-h_{\text{int}}}{h_{\text{int}}} - 1 \right]}{1 + \frac{\mu_L}{\mu_G} \frac{H-h_{\text{int}}}{h_{\text{int}}}}, \quad F_G^* = \frac{1 + \frac{1}{2} \frac{u_L}{u_G} \left[ \frac{\mu_G u_G}{\mu_L u_L} \frac{h_{\text{int}}}{H-h_{\text{int}}} - 1 \right]}{1 + \frac{\mu_G}{\mu_L} \frac{h_{\text{int}}}{H-h_{\text{int}}}}, \quad (18)$$

and  $F_{\text{int},G}^*$  that for the interfacial friction:

$$F_{\text{int},G}^* = \frac{1}{1 + \frac{\mu_G}{\mu_L} \frac{h_{\text{int}}}{H-h_{\text{int}}}}. \quad (19)$$

These closure terms form a reference, with which we benchmark our solvers for the case of steady flow, and to which we compare the new neural network closure terms.

## 2.4 Closure term limitations

By using closure relations of the form (13), we introduce a fundamental limitation. The averaged velocities cannot be translated back uniquely to velocity profiles, the slopes of which determine the stresses; information is lost in the averaging process. The consequence of this uniqueness problem is that for collections of very different velocity profiles, with the same averaged velocities, the closure relations will predict the same stresses, while the actual stresses can in reality be very different. In most of the literature [2, 5, 12, 15, 20, 40, 44] the analysis is therefore limited to fully developed steady state flow, for which the stresses *are* uniquely related to the averaged velocities.

There is a second limitation to the degree to which the low-fidelity model can be made to reproduce results of the high-fidelity model in this framework. Closure terms of the form (13)



are introduced as source terms in the low-fidelity model, and cannot be expected to resolve the entire difference in dynamics introduced by the averaging process and the associated assumptions. The difference between the dynamics of the high- and low-fidelity models, in absence of friction (equivalent to taking the homogeneous part of the equations for the two-fluid model at zero inclination), can be illustrated by performing a linear stability analysis of the two models. Results of this are shown in Figure 3, based on the parameters from Table 1.

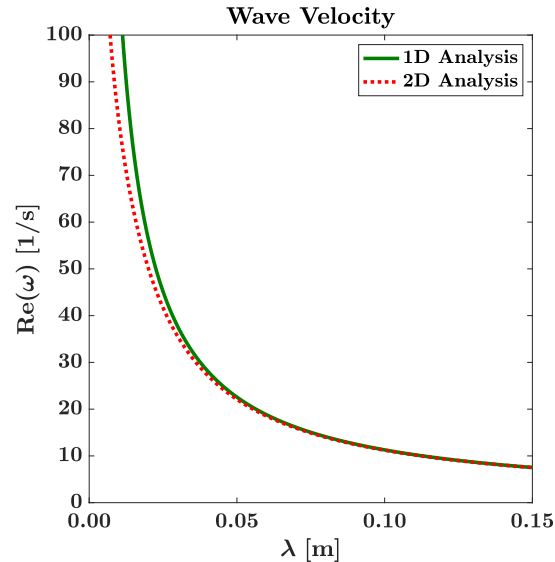


Figure 3: Dispersion relations  $\omega(k)$  with  $k = 2\pi/\lambda$ , for the 1D two-fluid model without friction terms [22], and for inviscid 2D potential flow [29]. The parameters are given in Table 1, with the steady state solution given in Table 2.

Table 1: Test case parameters.

Parameter	Symbol	Value	Units
Background pressure gradient	$\partial p/\partial s$	-1	$\text{kg m}^{-2} \text{s}^{-2}$
Liquid density	$\rho_L$	998	$\text{kg m}^{-3}$
Gas density	$\rho_G$	1.2	$\text{kg m}^{-3}$
Channel height	$H$	0.01	m
Initial interface height	$h_{\text{int}}$	$0.3H$	m
Liquid viscosity	$\mu_L$	$1.002 \cdot 10^{-3}$	$\text{kg m}^{-1} \text{s}^{-1}$
Gas viscosity	$\mu_G$	$1.82 \cdot 10^{-5}$	$\text{kg m}^{-1} \text{s}^{-1}$
Acceleration of gravity	$g$	9.81	$\text{m s}^{-2}$
Pipe inclination	$\phi$	0	degrees

With the analytical solution given by [6] and [45], the corresponding averaged velocities and stresses (for the steady state) can be calculated. These are given in Table 2.

It is observed that the dispersion relation of the 1D model only converges to the dispersion relation of the 2D model at large wavelengths. The cross-sectional averaging of the equations, and the associated assumption of hydrostatic balance, implicitly implies the long wavelength assumption [30].

Table 2: Test case steady state solution, for the parameters given in Table 1.

Parameter	Symbol	Value	Units
Averaged liquid velocity	$u_L$	0.00818	$\text{m s}^{-1}$
Averaged gas velocity	$u_G$	0.232	$\text{m s}^{-1}$
Liquid wall stress	$\tau_L$	-0.00646	$\text{kg m}^{-1} \text{s}^{-2}$
Gas wall stress	$\tau_G$	-0.00354	$\text{kg m}^{-1} \text{s}^{-2}$
Interfacial stress	$\tau_{\text{int}}$	-0.00346	$\text{kg m}^{-1} \text{s}^{-2}$
Liquid Reynolds number	$\text{Re}_L$	48.9	-
Gas Reynolds number	$\text{Re}_G$	214	-
Liquid Froude number	$\text{Fr}_L$	0.00114	-
Gas Froude number	$\text{Fr}_G$	0.391	-

## 2.5 Stress extraction

By extracting stresses from high-fidelity simulations via (12), it is possible to consider local and unsteady effects. We can take any position along the  $s$ -axis in the simulations and calculate the stresses, and the corresponding averaged variables  $u_L$ ,  $u_G$ ,  $h_{\text{int}}$ , at that point. We can calculate additional, local, quantities, which are not defined in a streamwise averaged description — the streamwise derivatives of the averaged variables — and relate these to the stresses as well. Since we can extract the averaged variables and stresses at any point in time in the unsteady simulations, the same holds for temporal derivatives. Because we use a neural network, such new inputs can easily be added to the closure relations, without prior information on the complex relation between them and the stresses.

The stresses are determined practically by fitting cubic splines to the velocity profiles of the liquid and gas separately and taking their analytical derivatives. For the final determination of the interfacial stress, the stresses at the interface as calculated from the liquid and gas profiles are averaged. For more details, see [10].

## 3 NEURAL NETWORKS

A neural network is used to construct a relation of the form (13), using the high-fidelity model data. For laminar steady state flow the analytical solution can be used to train the neural network instead of the high-fidelity solution (since they are equal). This simple case, for which conventional closure terms are exact, is used to tune the neural network hyper-parameters (in subsection 3.1 and subsection 3.2). Afterwards (in subsection 3.3 and subsection 3.4), the tuned neural network is applied to the more difficult case of unsteady, wavy flow.

### 3.1 Neural network settings

The network is a multilayer perceptron network (MLP), implemented in the MATLAB Deep Learning Toolbox [41]. It is tuned mainly by comparison of training data error and validation data error as measured by a mean squared error cost function

$$C = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (20)$$

where  $y_i$  is the data for a set of input variables  $i$  and  $\hat{y}_i$  is the model prediction for these inputs. The final value of the cost function is the average of (20) over the three stresses  $\tau_L$ ,  $\tau_G$ ,  $\tau_{\text{int}}$ .

The result is a network with 4 hidden layers with 18 nodes each, each with a hyperbolic tangent activation function, and no regularization term in the cost function (given the size of our training data set). The network is trained using the Levenberg-Marquardt training algorithm [18], an efficient algorithm for smaller networks. A small percentage of the training data (15%) is taken apart and not used for the training; the optimization is stopped if the error on this validation data does not decrease. The training inputs and outputs are mapped to the range  $[-1, 1]$  (the same translation and multiplication is later applied to unseen data).

We tested the effect of random initialization via the Nguyen-Widrow algorithm [31] and found that different random initializations yield very similar final values of the validation data error. We also verified the convergence of the training and validation errors with increasing amounts of data. These results are available in [10].

### 3.2 Performance of networks trained on steady state data

Training the network with steady state data is useful for the network tuning. However, Figure 4 shows that networks trained on steady state data have little predictive capacity for stresses found in wavy unsteady simulations. On the horizontal axis stress values observed in high fidelity simulations are set out, and on the vertical axis the neural network predictions for the same  $h_{\text{int}}$ ,  $u_L$ ,  $u_G$ , ... are given. We show the squared correlation coefficient

$$R^2 = 1 - \frac{\sum_{i=1}^N (\hat{y}_i - y_{l,i})^2}{\sum_{i=1}^N \left( \hat{y}_i - (1/N) \sum_{i=1}^N \hat{y}_i \right)^2}, \quad (21)$$

with a range between 0 and 1. In this definition,  $\hat{y}_i$  is the model prediction. We construct a linear fit of the model prediction  $\hat{y}_i$  as a function of the data and call it  $y_{l,i}$ . The value  $y_{l,i}$  is the value of the linear fit at the data point  $y_i$ , corresponding to prediction  $\hat{y}_i$ .

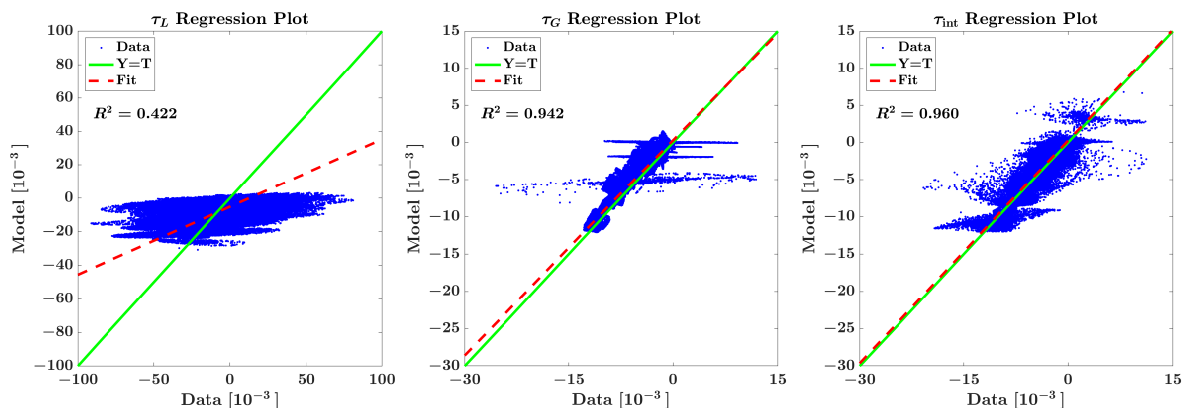


Figure 4: Regression plots for  $\tau_L$ ,  $\tau_G$ , and  $\tau_{\text{int}}$  with a neural network trained on steady state data as the model, tested on the wavy unsteady high-fidelity simulation data.

The poor performance shown in Figure 4, particularly for the strongly oscillatory liquid stress, could have been expected, as the analytical stresses with which we train the network are derived for steady state flow, while the actual flow is wavy unsteady. Similar poor results are observed when using the analytical stresses as the model directly. This motivates training on unsteady data.

### 3.3 Generation of wavy unsteady data

We consider 2D channel flow with periodic boundaries left and right under a constant body force in the form of a background pressure gradient  $\partial p/\partial s$ . No-slip boundary conditions are applied at the top and bottom walls. A sine wave perturbation with wavelength  $\lambda$  and amplitude  $\Delta\hat{h}_{\text{int}}$  is applied to the interface between liquid and gas.

We generate data by running the high fidelity code Gerris 60 times, with varying input parameters randomly selected from the ranges given in Table 3. The parameters are selected from these ranges using Latin Hypercube Sampling [28], ensuring a space-filling sampling, without repetition of parameter values. The material properties, channel height, and the channel inclination are kept at the values given in Table 1. This limits the required amount of (costly) simulations, while allowing practical application to an unsteady flow in a specified pipe and with specified fluids.

The wavelength of the perturbation is fixed at  $\lambda = L = 0.12$  m, where  $L$  is the length of the domain.

Table 3: The ranges of the parameters of the unsteady high fidelity simulations used as training data.

Initialization	$h_{\text{int}} [H]$	$\partial p/\partial s [\text{Pa/m}]$	$\Delta\hat{h}_{\text{int}} [H]$	Number of simulations
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

The simulations are initialized from two different initial conditions:

- ‘zero wavy’: the velocities in the entire domain are zero,
- ‘developed wavy’: the velocities are initialized at their (flat interface) steady state values (determined analytically [6]),

and then run from  $t = 0$  to  $t = 10$  seconds. With these initial conditions and the given parameters, we get slowly traveling standing waves (see Figure 9), initially approximated by

$$\Delta h_{\text{int}}(s, t) = 2\Delta\hat{h}_{\text{int}} \cos(ks - \delta\omega t) \cos(\omega_0 t). \quad (22)$$

These waves are formed as the superposition of two waves traveling in the opposite direction with wave velocities

$$c_1 = \frac{\omega_0 + \delta\omega}{k}, \quad c_2 = \frac{-\omega_0 + \delta\omega}{k}, \quad (23)$$

with  $\delta\omega \ll \omega_0$  due to the small Froude numbers (see Table 2). Later, nonlinear and damping effects become important; in most cases the waves are largely damped out at  $t = 10$ .

Note that the parameter ranges in Table 3 are chosen such that the perturbations damp out in time, avoiding a transition to (near-) slug flow and problems of ill-posedness in the 1D two-fluid model (see e.g. [7]).

### 3.4 Neural network training

At each time step and at each grid point along  $s$  the quantities given in (13) are extracted from the Gerris simulations, yielding many data points, which may not all provide distinct information. The networks are therefore trained on small random subsets of the data.

Neural networks are trained on different portions of the data:

- ‘zero wavy net’: networks trained on data with the ‘zero wavy’ initial condition.
- ‘developed wavy net’: networks trained on data with the ‘developed wavy’ initial condition.
- ‘zero + developed wavy net’: networks trained on a combination of the data with the ‘zero wavy’ initial condition and with the ‘developed wavy’ initial condition.

Per item in the above list, we sample the data with replacement to get five different data sets, each a small percentage of the total data set<sup>2</sup>. We train (randomly initialized) networks on each of these subsets of the data. The final prediction for the stresses is obtained by averaging the predictions of each of the five networks, for a given set of inputs. This averaging procedure is called ‘bagging’ and has been shown to improve accuracy for learning algorithms sensitive to changes in the training data [9]. This technique was also employed by Ma et al. [27].

An extra input is added compared to those given by equation (13): the interface slope  $\partial h_{\text{int}}/\partial s$ . The interfacial slope can easily be added to the neural network as an input. This input does not fit in conventional closure terms which are calculated for the fully developed steady state, since it is a locally defined variable<sup>3</sup>. 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. Andritsos and Hanratty [2]), 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.

With this addition, and the given selection of variable parameters, the neural network takes the form shown in Figure 5.

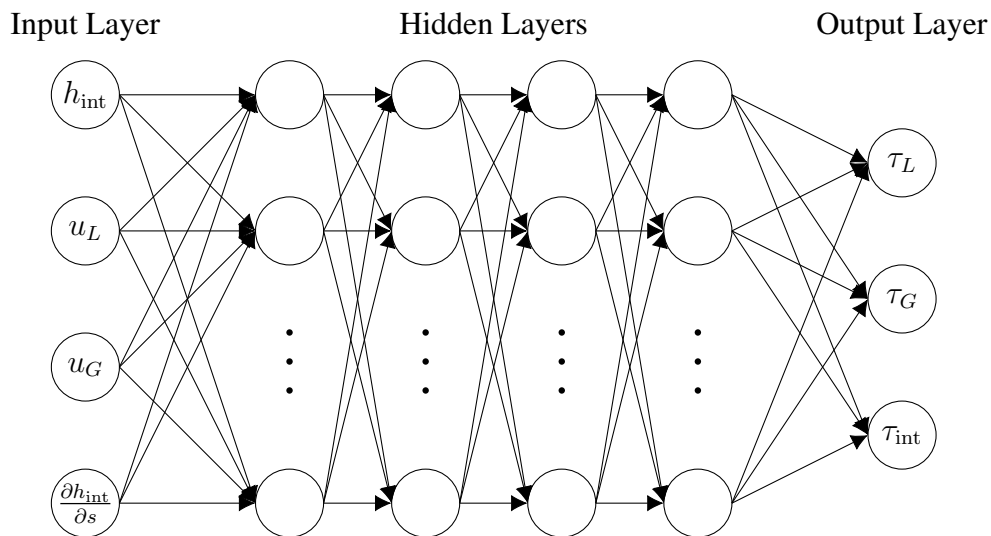


Figure 5: A schematic of the neural network trained in subsection 3.4, and tested in section 4, with four variable inputs, four hidden layers (with 18 nodes per layer), and three outputs.

<sup>2</sup>The percentages of the data sets taken per individual training are 5% for the ‘zero wavy net’ and ‘developed wavy net’, and 10% for the ‘zero + developed wavy net’

<sup>3</sup>An exception is Brauner and Moalem Maron [8], who modified conventional closure terms (based on Taitel and Dukler [40]) to add a dependency of the interfacial stress on the interfacial slope, by matching experimentally observed and theoretically calculated stability boundaries (the latter of which depends on the closure terms). The advantage of our method is that the interfacial slope is included in the correlation from the beginning, in the same straightforward manner as the other inputs.

Resulting regression plots for a ‘zero + developed wavy net’ are shown in Figure 6. The correlation is satisfactory, considering the variation in flow patterns found in the data. The influence of the extra input parameter  $\partial h_{\text{int}}/\partial s$  is illustrated by comparison of Figure 6 to Figure 7, where in the latter figure results are shown if this parameter is left out. Apparently the interfacial slope is an important piece of information for the determination of the stresses. This parameter allows the stress prediction to vary based on the wave’s amplitude and local phase, and allows distinction between fully developed and unsteady flow, alleviating the uniqueness issue discussed in subsection 2.4.

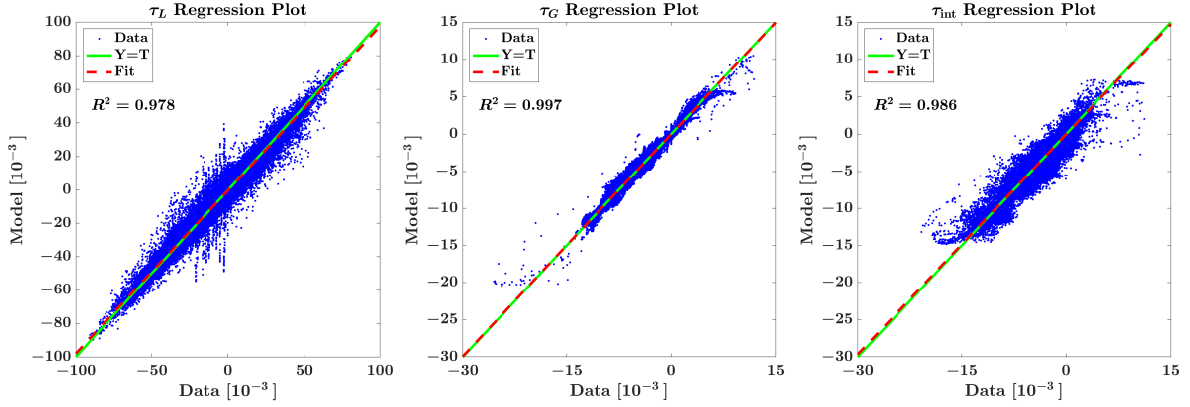


Figure 6: Regression plots for a neural network trained on the data combined for both initial conditions of Table 3 (‘zero + developed wavy net’).

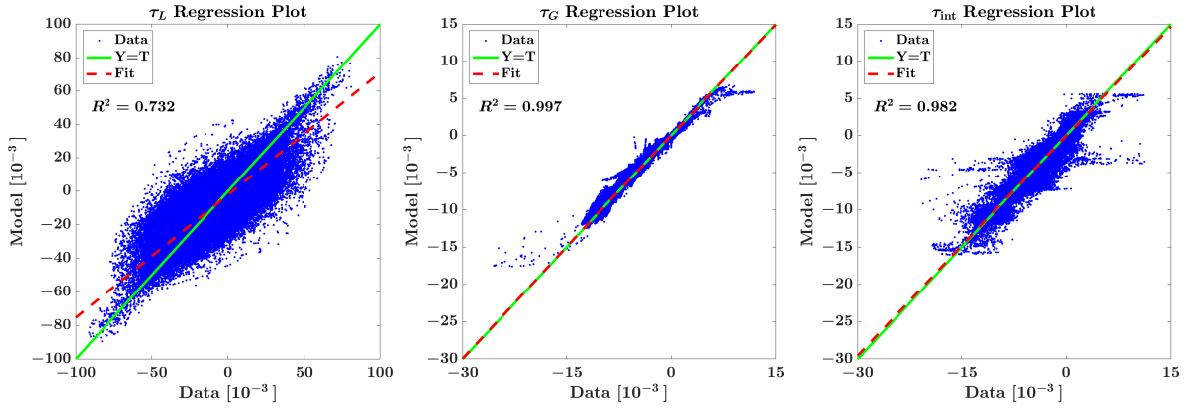


Figure 7: Regression plots for a neural network trained on the data combined for both initial conditions of Table 3 (‘zero + developed wavy net’) excluding the interface slope  $\partial h_{\text{int}}/\partial s$  as an input.

## 4 RESULTS

The true test of the learned closure terms lies in their application to the low-fidelity model Rosa, and comparison of the resulting predictions to high-fidelity model Gerris predictions (presumed to be the truth). At each stage in the Runge-Kutta time integration scheme, the variables as given in (5) are fed to the trained neural networks to arrive at values for the stresses  $\tau_L$ ,  $\tau_G$ , and  $\tau_{\text{int}}$ . The current MATLAB shallow neural network implementation significantly slows down the Rosa code simulations, compared to when the analytical closure relations (14) are used.

In order to be able to compare Gerris and Rosa results (with different grid resolutions) quantitatively, cubic splines of the variables of interest are constructed, along the horizontal axis. The resulting  $s$ -dependent Gerris result at  $t = t_i$  is  $y_i = y_i(s)$ , with  $\hat{y}_i = \hat{y}_i(s)$  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 4 shows the following relative error measure for the difference between Gerris and Rosa results, termed the ‘normalized averaged error’ (NAE):

$$\text{NAE} = \frac{1}{N_T} \sum_{i=1}^{N_T} \sqrt{\frac{1}{L} \int_{s=0}^L \left( \frac{y_i - \hat{y}_i}{y_c} \right)^2 ds}. \quad (24)$$

The parameter  $N_T$  is the total number of time steps and  $L$  is the length of the domain.

This error is shown for simulations initialized from different initial conditions, and using different closure terms. Analytical closure terms (14) are tested alongside closure terms learned from the wavy unsteady data of Table 3, using neural networks. Where the neural network assisted error is smaller than the analytical closure error, the error value is highlighted green in Table 4.

Table 4: Normalized averaged errors (24) between Gerris and Rosa simulations, for different variables of interest. Results are given for Gerris and Rosa simulations starting from different initial conditions, with the Rosa simulations using either analytical or neural network closure terms. Where the neural network closure terms outperform the analytical closure terms (for the same initialization), the result is highlighted in green.

Case		Normalized Averaged Error [ $10^{-3}$ ]					
Initialization	Closure	$h_{\text{int}}$	$u_L$	$u_G$	$\tau_L$	$\tau_G$	$\tau_{\text{int}}$
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
zero wavy	zero + developed wavy net	0.52	193	15.2	233	10.8	28.5
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
developed wavy	zero + developed wavy net	0.51	112	12.6	173	8.20	26.6

Overall, with error measure (24), the results with neural network closure terms do not show a significant improvement, except perhaps for the interface height. However, this error measure is crude and does not show how well the wave dynamics are reproduced.

We therefore study the values of  $h_{\text{int}}$ ,  $u_L$ ,  $u_G$ ,  $\tau_L$ ,  $\tau_G$ ,  $\tau_{\text{int}}$  as a function of time in Rosa simulations for the test case given by Table 1, at a point at the center of the domain ( $s = 0.06$  m) (see Figure 8). The scale and form of the oscillations are captured better when using the neural network closure terms; the wave damping behavior corresponds better to the high-fidelity simulations. The interface height in the entire domain is shown in Figure 9 for a number of time instants, with the shown Rosa simulations employing the neural network closure terms.

The problem with the analytical closure terms is highlighted in Figure 10, in which the same simulation results are shown for later time instants (with the analytical closure). The waves acquire a sharp wavefront, in the wake of which small spurious waves are formed. These effects are unphysical and are not observed in the Gerris simulations. The neural network closure terms do not suffer from these spurious effects, probably due to their better damping behavior.

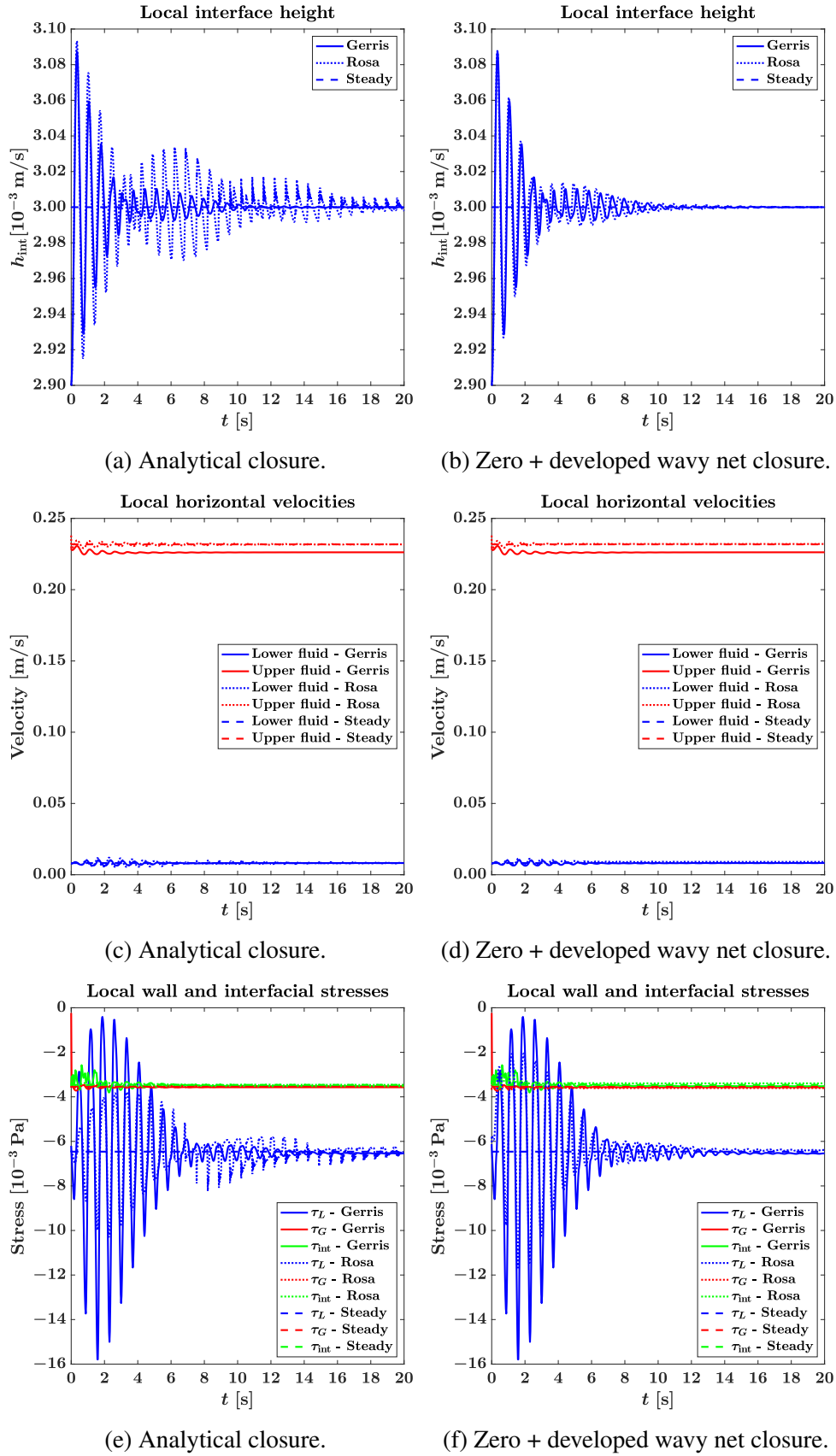


Figure 8: Evolution in time of the velocities, stresses and interface height at the center of the domain. Initialized with the ‘developed wavy’ initial condition.



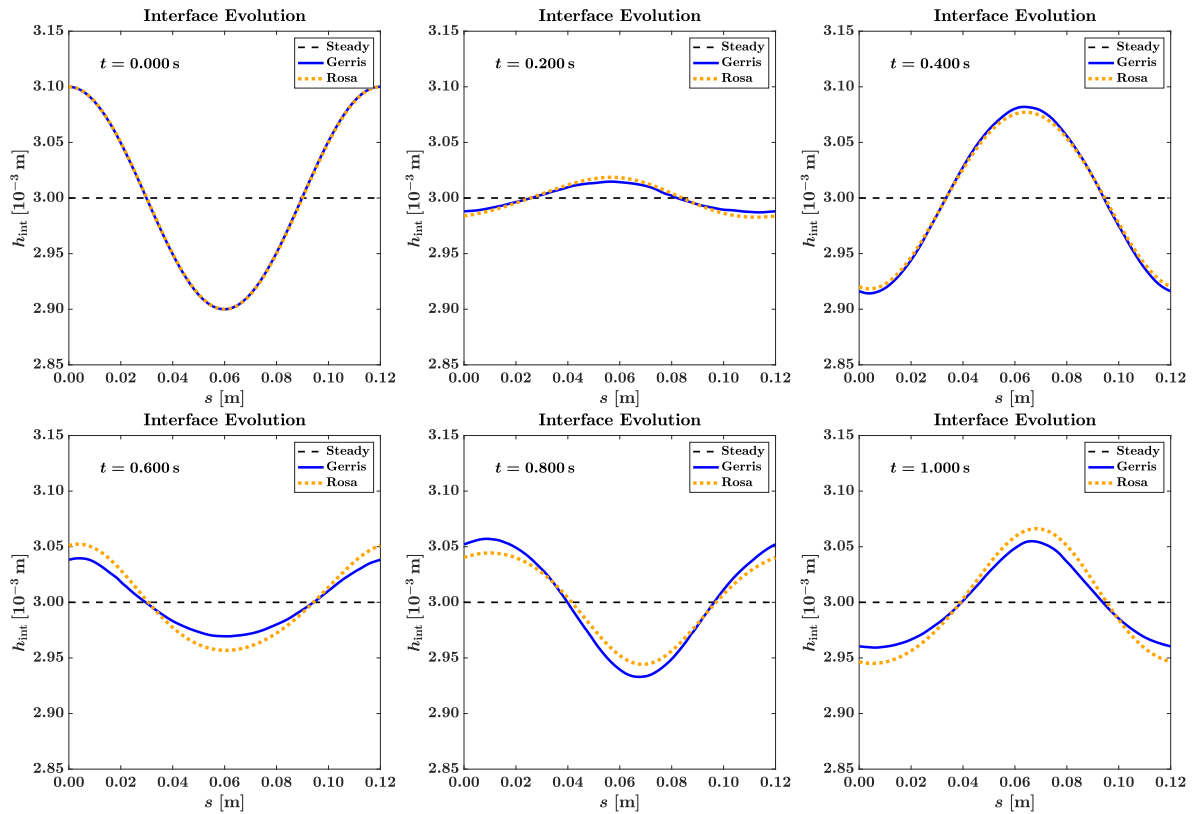


Figure 9: Evolution in time of the interface between liquid and gas throughout the domain, zoomed in at the interface ( $H = 0.01$  m). Rosa results with a ‘developed wavy’ initialization and ‘zero + developed wavy net’ closure.

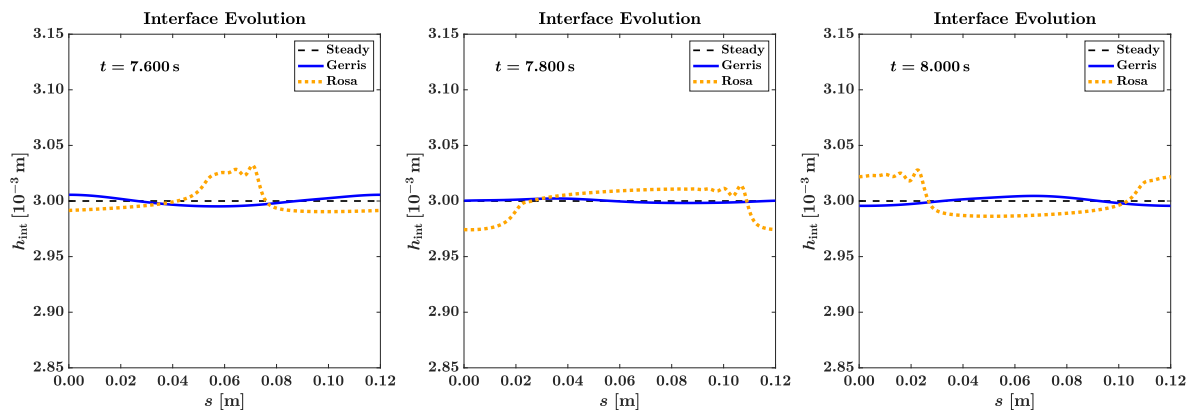


Figure 10: Evolution in time of the interface between liquid and gas throughout the domain, zoomed in at the interface ( $H = 0.01$  m). Rosa results with a ‘developed wavy’ initialization and analytical closure.

Training the neural network on unsteady simulation data allows the closure terms to capture the unsteady (damping) behavior, differentiating them from conventional steady state closure terms (including those closure terms that consider the streamwise averaged effect of a wavy interface). The addition of the extra closure input parameter  $\partial h_{\text{int}}/\partial s$ , allows the closure terms to apply the learned differences between steady state and unsteady flow patterns. By providing information on the wave amplitude and local phase, this input parameter enables distinction between steady state flow and increasingly unsteady flow during application in Rosa. This allows the closure terms to provide different results for different phases of the wave damping process. Similarly, the closure terms can produce different stresses at different points along the wave (beyond the distinction made possible by the small differences in interface height and averaged velocities).

One of the problems still visible in Figure 8 is a discrepancy in the steady state gas velocity; this can be solved by further grid refinement of the Gerris simulations.

The remaining main difference between Gerris simulations and Rosa simulations using neural network closure terms is a discrepancy in the wave speed. The wave speed of the Rosa simulations is slightly higher than that of the Gerris simulations, so that the two slowly drift out of phase. This difference in wave speed between Gerris and Rosa simulations can be explained by the fact that a discrepancy between the models remains that cannot be solved via modeling the closure terms (see subsection 2.4). The inviscid dispersion relations for the test case, plotted in Figure 3, indeed show a higher wave speed for the 1D model than for the 2D model.

## 5 CONCLUSION & OUTLOOK

In this work, we have explored a new approach based on neural networks to solve the long-standing closure problem for stratified multiphase flow in channels. We have trained neural networks on high fidelity simulation data to learn closure terms for the wall and interfacial stresses in a low fidelity model; the 1D two-fluid model for stratified channel flow. An important novelty in our work is the inclusion of the streamwise derivative of the interface height as a feature in the neural network. With this addition, the dynamic wave-damping behavior of high-fidelity simulations was reproduced better than with the conventional (steady state) set of closure terms available in literature [45].

With the proposed framework, closure terms can be constructed for specific flow regimes and duct geometries, as long as high-fidelity simulations are available. The addition of extra inputs to the closure relations, which is straightforward in this framework, alleviates their inherent uniqueness problem. An example of possible extra inputs, besides the interface slope, are the spatial and temporal derivatives of the velocities.

We note that, even with a highly accurate closure model for the stresses, the 1D model will generally not exactly reproduce the 2D results, because the stresses are not the only source of discrepancy between the 1D and 2D model. In principle, it might be possible to eliminate these discrepancies by modeling the difference between high- and low-fidelity model predictions directly, and adjusting the low-fidelity model accordingly. But this approach would be less physical, so that it might not generalize as well.

In the future we aim to improve the framework through closer inspection of the structure of the learned closure terms, and possibly through the inclusion of physical constraints in the network structure. This will open the door to more challenging cases, such as the prediction of slug flow.

## REFERENCES

- [1] A. Alizadehdakhel, M. Rahimi, J. Sanjari, and A. A. Alsairafi. CFD and artificial neural network modeling of two-phase flow pressure drop. *International Communications in Heat and Mass Transfer*, 36:850–856, 2009.
- [2] N. Andritsos and T. J. Hanratty. Influence of interfacial waves in stratified gas-liquid flows. *AIChE Journal*, 33:444–454, 1987.
- [3] D. Barnea and Y. Taitel. Interfacial and structural stability of separated flow. *International Journal of Multiphase Flow*, 20:387–414, 1994.
- [4] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 85:257–283, 1989.
- [5] D. Biberg. A mathematical model for two-phase stratified turbulent duct flow. *Multiphase Science and Technology*, 19, 2007.
- [6] D. Biberg and G. Halvorsen. Wall and interfacial shear stress in pressure driven two-phase laminar stratified pipe flow. *International Journal of Multiphase Flow*, 26:1645–1673, 2000.
- [7] N. Brauner and D. Moalem Maron. Stability analysis of stratified liquid-liquid flow. *International Journal of Multiphase Flow*, 18:103–121, 1992.
- [8] N. Brauner and D. Moalem Maron. The role of interfacial shear modelling in predicting the stability of stratified two-phase flow. *Chemical Engineering Science*, 48:2867–2879, 1993.
- [9] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.
- [10] J. F. H. Buist. *Machine Learning for Closure Models in Multiphase-Flow Applications*. Master’s thesis, Eindhoven University of Technology, 2019.
- [11] A. J. Chorin. On the convergence of discrete approximations to the Navier-Stokes equations. *Mathematics of Computation*, 23:341–353, 1969.
- [12] S. W. Churchill. Friction factor equation spans all fluid flow regimes. *Chemical Engineering*, 84:91–92, 1977.
- [13] N. Coutris, J. M. Delhaye, and R. Nakach. Two-phase flow modelling: the closure issue for a two-layer flow. *International Journal of Multiphase Flow*, 15:977–983, 1989.
- [14] 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*, 132:346–361, 1997.
- [15] M. Espedal. *An Experimental Investigation of Stratified Two-Phase Pipe Flow at Small Inclinations*. PhD thesis, Norwegian University of Science and Technology, 1998.
- [16] M. Gamahara and Y. Hattori. Searching for turbulence models by artificial neural network. *Physical Review Fluids*, 2:054604, 2017.
- [17] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43:89–112, 2001.

- [18] M. T. Hagan and M. B. Menhaj. Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5:989–993, 1994.
- [19] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:201–225, 1981.
- [20] J. E. Kowalski. Wall and interfacial shear stress in stratified flow in a horizontal pipe. *AIChE Journal*, 33:274–281, 1987.
- [21] B. I. Krasnopolsky and A. A. Lukyanov. A conservative fully implicit algorithm for predicting slug flows. *Journal of Computational Physics*, 355:597–619, 2018.
- [22] J. Liao, R. Mei, and J. F. Klausner. A study on the numerical stability of the two-fluid model near ill-posedness. *International Journal of Multiphase Flow*, 34:1067–1087, 2008.
- [23] J. Ling, A. Kurzawski, and J. Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.
- [24] C. Lu, S. Sambasivan, A. Kapahi, and H. S. Udaykumar. Multi-scale modeling of shock interaction with a cloud of particles using an artificial neural network for model representation. *Procedia IUTAM*, 3:25–52, 2012.
- [25] 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.
- [26] 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*, 27:092101, 2015.
- [27] M. Ma, J. Lu, and G. Tryggvason. Using statistical learning to close two-fluid multiphase flow equations for bubbly flows in vertical channels. *International Journal of Multiphase Flow*, 85:336–347, 2016.
- [28] 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*, 21:239–245, 1979.
- [29] L. M. Milne-Thomson. *Theoretical Hydrodynamics*. Macmillan, London, 4th edition, 1962.
- [30] M. Montini. *Closure Relations of the One-Dimensional Two-Fluid Model for the Simulation of Slug Flows*. PhD thesis, Imperial College London, 2011.
- [31] D. Nguyen and B. Widrow. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *Proceedings of the 1990 IJCNN International Joint Conference on Neural Networks*, volume 3, pages 21–26, San Diego, CA, USA. IEEE, 1990.
- [32] R. E. Osgouei, A. M. Ozbayoglu, E. M. Ozbayoglu, E. Yuksel, and A. Eresen. Pressure drop estimation in horizontal annuli for liquid–gas 2 phase flow: Comparison of mechanistic models and computational intelligence techniques. *Computers & Fluids*, 112:108–115, 2015.

- [33] S. Popinet. An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics*, 228:5838–5866, 2009.
- [34] S. Popinet. Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *Journal of Computational Physics*, 190:572–600, 2003.
- [35] B. Sanderse, S. Misra, and S. Dubinkina. Numerical simulation of roll waves in pipelines using the two-fluid model. In *Proceedings of the 11th North American Conference on Multiphase Production Technology*, pages 373–386, Banff, Canada, 2018.
- [36] B. Sanderse and A. E. P. Veldman. Constraint-consistent Runge–Kutta methods for one-dimensional incompressible multiphase flow. *Journal of Computational Physics*, 384:170–199, 2019.
- [37] B. Sanderse, I. E. Smith, and M. H. W. Hendrix. Analysis of time integration methods for the compressible two-fluid model for pipe flow simulations. *International Journal of Multiphase Flow*, 95:155–174, 2017.
- [38] F. Sarghini, G. de Felice, and S. Santini. Neural networks based subgrid scale modeling in large eddy simulations. *Computers & Fluids*, 32:97–108, 2003.
- [39] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114:146–159, 1994.
- [40] Y. Taitel and A. E. Dukler. A model for predicting flow regime transitions in horizontal and near horizontal gas-liquid flow. *AIChE Journal*, 22:47–55, 1976.
- [41] The MathWorks, Inc. MATLAB Documentation - Deep Learning Toolbox. 2019. URL: <https://nl.mathworks.com/help/deeplearning/index.html> (visited on 02/03/2019).
- [42] B. D. Tracey, K. Duraisamy, and J. J. Alonso. A machine learning strategy to assist turbulence model development. In *Proceedings of the 53rd AIAA Aerospace Sciences Meeting*, Kissimmee, Florida. American Institute of Aeronautics and Astronautics, 2015.
- [43] G. Tryggvason, R. Scardovelli, and S. Zaleski. *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge University Press, 2011.
- [44] 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*, 32:82–105, 2006.
- [45] 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*, 30:877–900, 2004.
- [46] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100:25–37, 1992.
- [47] D. L. Youngs. Time-dependent multi-material flow with large fluid distortion. In K. W. Morton and M. J. Baines, editors, *Numerical Methods for Fluid Dynamics*. Academic Press, 1982.