Chapter 4

Results

In this section we present the results of the numerical experiments, together with the results from the data-driven methods, including neural networks and Fourier neural operators.

4.1 Numerical results

In this chapter we present the results of the numerical experiments, where we have implemented the finite volume method for solving the shallow water equations in 1D and tested it on several problems. [1] A key focus is to validate the implementation, as it will gemerate data for the data-driven methods, including neural networks and Fourier neural operators. To test the implementation, we have solved the 1D dam break problem, and the five test cases from Toro's book [1]. These problems are all discontinuous in either the water height h or the fluid velocity u. The idea is, that if the numerical solution can capture the discontinuities, it should be well-suited to handle smoother solutions as well.

4.1.1 The 1D Dam Break Problem

First we solve the 1D dam break problem, with the following initial conditions:

$$h(x,0) = \begin{cases} h_L, & \text{if } x < x_0, \\ h_R, & \text{if } x > x_0, \end{cases}$$

where $x \in [0, 50], h_L = 3.5$ m, $h_R = 1.25$ m and $x_0 = 20$ m. Since it is a dam break problem the initial fluid velocity is zero, i.e., $u(x,0) = 0$. We solve the problem starting at $t = 0$ and ending at $t = 2.5$ seconds. The numerical solution to the 1D Dam Break Problem using the FVM, together with the true solution, provided from the course [11], can be seen in Figure 4.1.

[1] Code and small animations can be found at github, visit `https://github.com/MelissaJessen/Shallow-Water-Equations`.

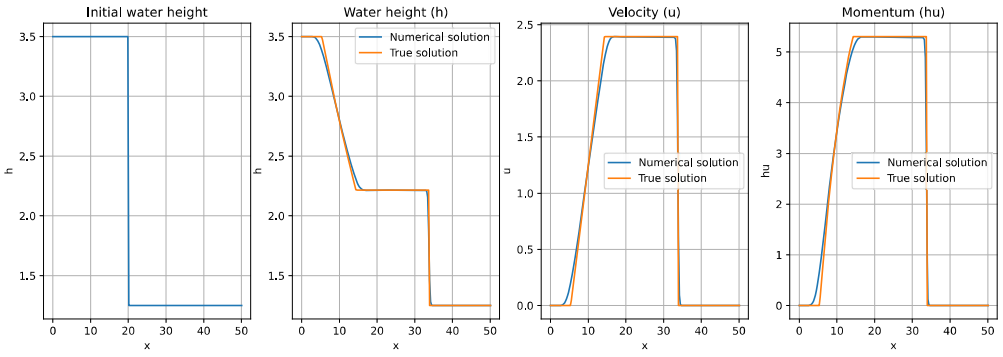


Figure 4.1: The initial water height h at $t = 0$, together with the water height, the fluid velocity u and the momentum hu after $t = 2.5$ seconds.

From Figure 4.1 we see that the numerical solution aligns well with the true solution, and successfully captures the discontinuity.

4.1.2 Toro test cases

We have tested the method on the five test cases for Riemann problems from Toros book [1]. The initial conditions for the five test cases are given in Table 4.1.

Test case	h_L	u_L	h_R	u_R	x_0	t_{end}
1	1.0	2.5	0.1	0.0	10.0	7.0
2	1.0	-5.0	1.0	5.0	25.0	2.5
3	1.0	0.0	0.0	0.0	20.0	4.0
4	0.0	0.0	1.0	0.0	30.0	4.0
5	0.1	-3.0	0.1	3.0	25.0	5.0

Table 4.1: Initial conditions for the five test cases.

The domain is $x \in [0, 50]$ for all test cases. The Riemann problems are chosen to test the method on different types of waves, such as shock waves and rarefaction waves. To solve the test cases there are used different fluxes, namely:

1. Godunov method with exact Riemann solver,
2. Lax-Friedrich flux,
3. Lax-Wendroff flux,
4. FORCE flux,
5. HLL flux,

Test case 1

The initial conditions for test case 1 are given in Figure 4.2 and the final solution after $t = 7.0$ seconds is given in Figure 4.3. In test case 1, we observe a right shock wave and a left rarefaction wave.

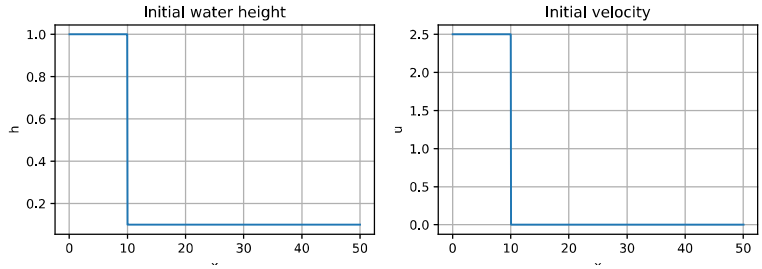


Figure 4.2: Initial conditions for the test case.

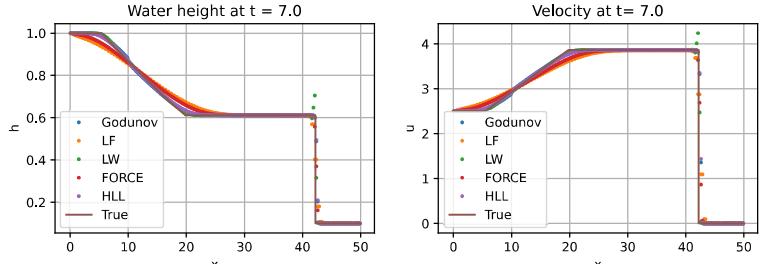


Figure 4.3: Final solution for the test case.

For this test case all the fluxes work well, but there are minor differences in the solution, which can be seen in Figure 4.3. We also see that Lax-Wendroff flux has some oscillations in the solution, which is not present in the other fluxes.

Test case 2

The initial conditions for test case 2 are illustrated in Figure 4.4.

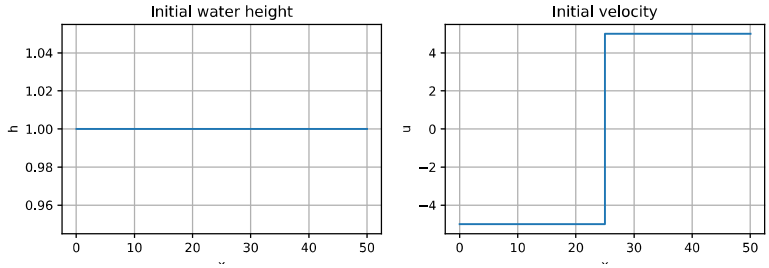


Figure 4.4: Initial conditions for the test case.

In test case 2 we have two rarefaction waves, one on the left side and one on the right side. As they are

travelling in opposite directions (away from each other), there will be created a nearly dry bed in the middle of the domain. Many methods have difficulties with this test case as they may compute a negative water height. For these experiments we were able to get close to the true solution, using Lax-Friedrich flux, FORCE flux and HLL flux. The final solution after $t = 2.5$ seconds is illustrated in Figure 4.5.

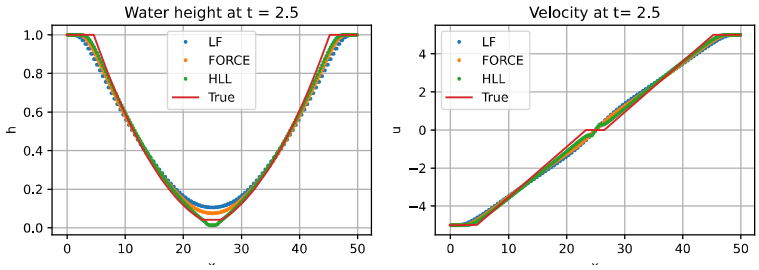


Figure 4.5: Final solution for the test case.

For the fluxes, Godunov method with exact Riemann solver, and Lax-Wendroff, it was not possible to get an acceptable solution.

Test case 3

The initial conditions for test case 3 are given in Figure 4.6, and the final solution after $t = 4.0$ seconds is given in Figure 4.7.

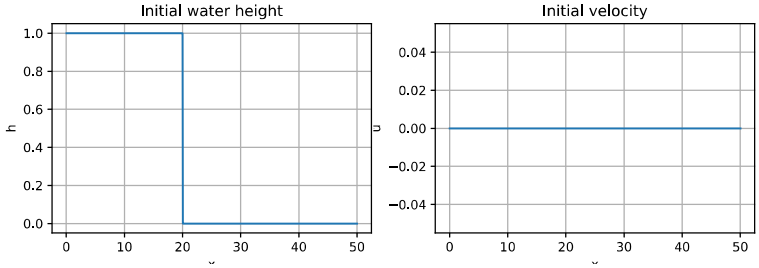


Figure 4.6: Initial conditions for the test case.

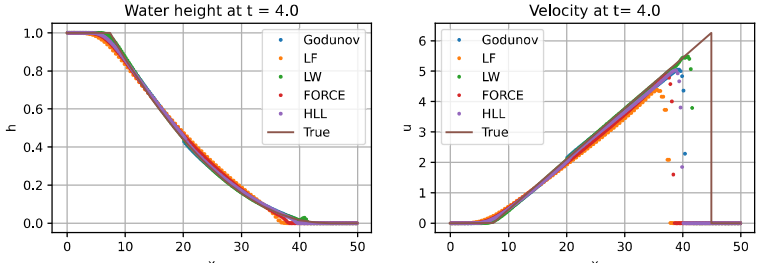


Figure 4.7: Final solution for the test case.

To solve case 3, with the FVM we must add a small amount to h_R, since the code does not handle $h_R = 0$ well. We set $h_R = 0.00005$ to solve it numerically, but the true solution is for $h_R = 0$. By running experiments with

different values of h_R, we see that the solution converges to the true solution as h_R approaches 0. The solution consists of a left rarefaction wave. From Figure 4.7 we see that when it comes to predicting the velocity, there are some differences in how the different fluxes perform.

Test case 4

The initial conditions for test case 4 are given in Figure 4.8, and the final solution after $t = 4.0$ seconds is given in Figure 4.9.

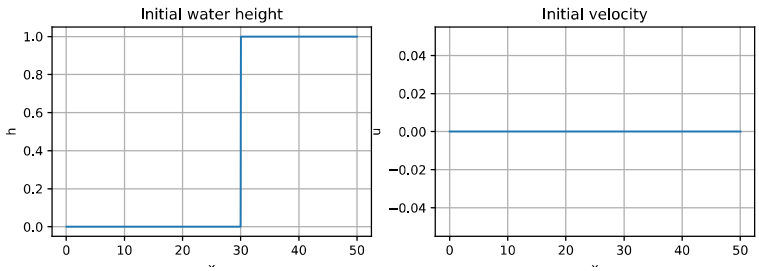


Figure 4.8: Initial conditions for the test case.

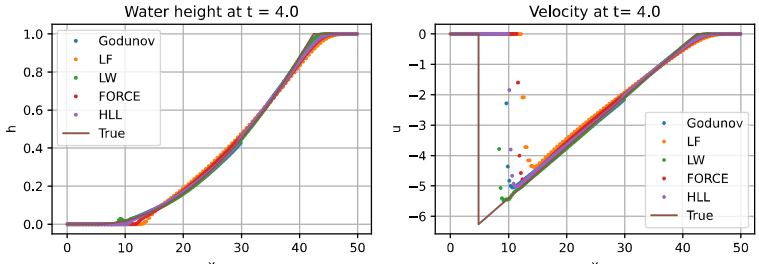


Figure 4.9: Final solution for the test case.

In case 4 we face the same challenges as in case 3. We set $h_L = 0.00005$, and the solution converges to the true solution as h_L approaches 0. This test case is symmetric to test case 3, and the solution consists of a right rarefaction wave. The case is included to test if the results are as expected. As in test case 4, we observe differences in the fluxes performance.

Test case 5

The initial conditions for test case 5 are given in Figure 4.10, and the final solution after $t = 5.0$ seconds is given in Figure 4.11.

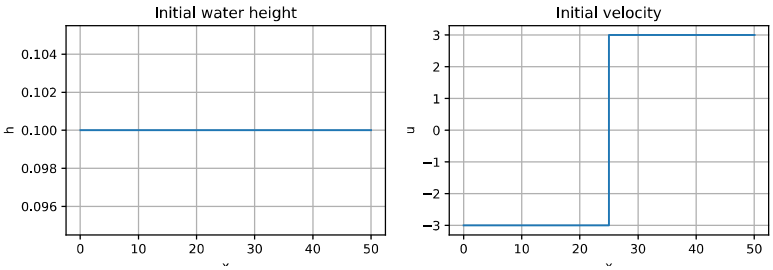


Figure 4.10: Initial conditions for the test case.

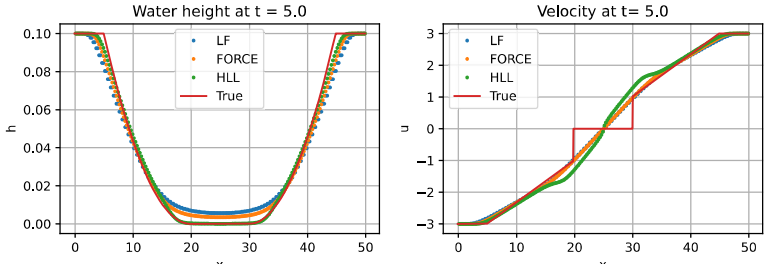


Figure 4.11: Final solution for the test case.

From Figure 4.11 we see that the numerical solutions for the velocity v at $t = 5.0$ are smooth, where the true solution is discontinuous. In this test case there are also challenges with some of the fluxes due to the generation of a dry-bed region. The fluxes that are not able to solve this case are Godunov method with exact Riemann solver and Lax-Wendroff flux, the same as in test case 2. The solution consists of two rarefaction waves, one on the left side and one on the right side, and a dry-bed region in the middle.

To get an overview of which fluxes that were able to solve the problems, consider the table below.

Test case	**Godunov**	**LF**	**LW**	**FORCE**	**HLL**
1	✓	✓	✓	✓	✓
2	×	✓	×	✓	✓
3	✓	✓	✓	✓	✓
4	✓	✓	✓	✓	✓
5	×	✓	×	✓	✓

Table 4.2: Overview of which fluxes that were able to solve the test cases.

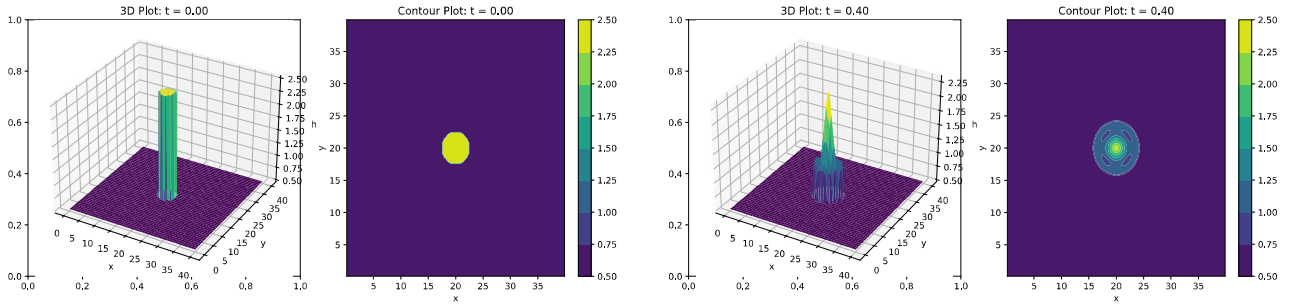
4.1.3 2D idealised Circular Dam Break Problem

We now proceed to the 2D case, focusing on an idealised circular dam break problem over a horizontal bottom. This problem is also from Toro's book [3]. We assume there is an infinitely thin circular wall at radius $R = 2.5$ m

is a square domain of size 40×40 with centre at $(x_c, y_c) = (20, 20)$. The initial conditions are

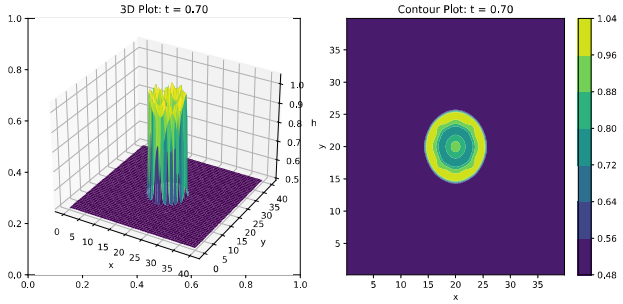
$$h(x, y, 0) = \begin{cases} 2.5\ m, & \text{if } \sqrt{(x - x_c)^2 + (y - y_c)^2} \leq R, \\ 0.5\ m, & \text{otherwise,} \end{cases}$$
$$u(x, y, 0) = 0,$$
$$v(x, y, 0) = 0.$$

We use a mesh of size 200×200. The results after $t = 0.0, 0.4, 0.7$ and 1.4 seconds are given in Figure 4.12.

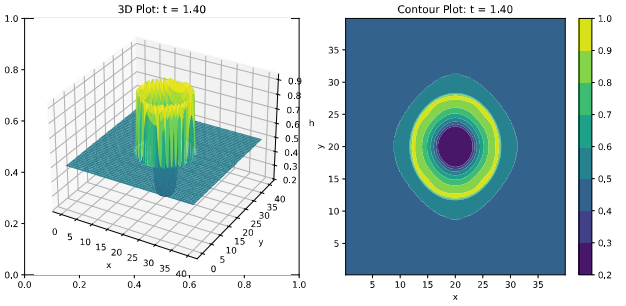


(a) 2D dam break problem after $t = 0$.

(b) 2D dam break problem after $t = 0.4$.



(c) 2D dam break problem after $t = 0.7$.



(d) 2D dam break problem after $t = 1.4$.

Figure 4.12: Snapshots of the 2D dam break problem at different times.

By comparing Figure 4.12 with the results from the book by Toro [3], and see that the numerical solution aligns well with the true solution from the book.

4.2 Data-driven results

In this section we present the results of the data-driven models. We start by showing the numerical solution of the shallow water equations, then we present the predictions of the FNN and FNO models. Until now, we have mostly considered discountinuous initial conditions, but we will also consider smooth initial conditions in this section. We solve the SWE with the following initial conditions:

$$h(x, 0) = h_0 \exp\left(\frac{-(x - \mu)^2}{2\sigma^2}\right), \tag{4.2.1}$$
$$u(x, 0) = 0,$$

where $h_0 = 1, \mu = 0.5, \sigma = 0.1$. The domain is $x \in [0, 1]$ with $N = 200$ points and the final time is $t = 1.0$. We use a CFL number of 0.9 and variable time steps. The numerical solution is shown in Figure 4.13, in both a contour plot and a 3D plot.

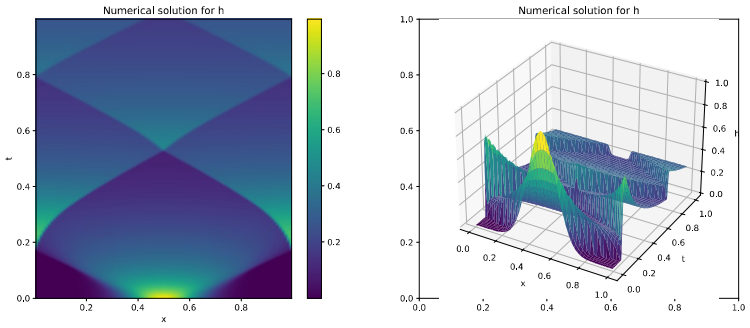


Figure 4.13: Numerical solution of the shallow water equations.

4.2.1 FNN

In the feedforward neural network, we train the model using the data generated by the numerical solution of the shallow water equations. The neural network consists of the following layers: a input dense layer, three hidden dense layers with ReLU activation functions, a batch normalization layer for stability, a dropout layer to prevent overfitting, and an output dense layer. We have also included L2 regularization in the hidden layers to prevent overfitting. Themodel has been trained using the Adam optimizer with a learning rate of 0.01, a batch size of 16 and a total of 3000 epochs. We train the model on the data from $t = 0$ to $t = 0.8$, and test it on the data from $t = 0.8$ to $t = 1.0$. The predictions of the model are shown in Figure 4.14.

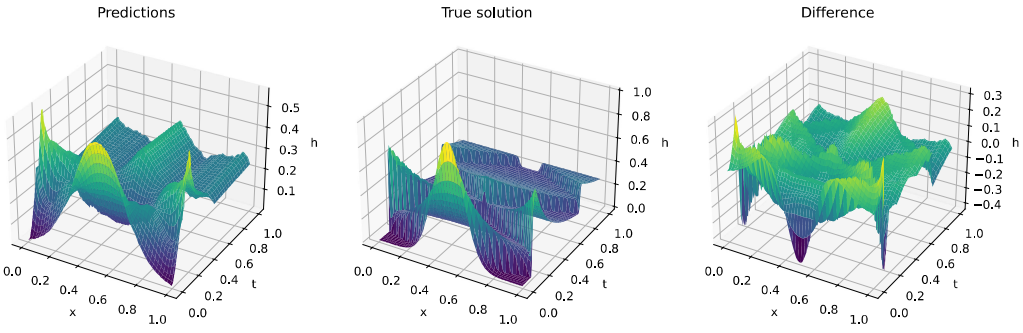


Figure 4.14: Predictions.

From Figure 4.14 we see that the neural network, to some extent, is able to learn the dynamics of the solution, but it is not accurate enough to be used as a solver. We also see that the further we get in time, the worse the

predictions become. This is also somehow expected. To understand the performance of the model, we consider the predictions for some given time steps, shown in Figure 4.15.

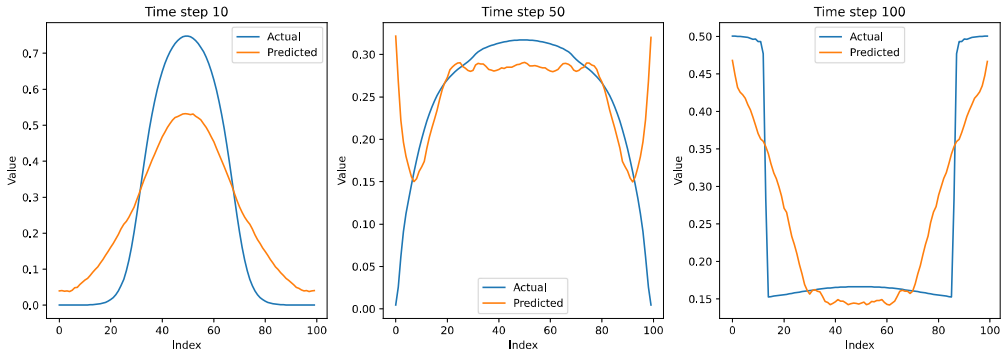


Figure 4.15: Predictions for some given time steps.

Again, we see that the model finds some of the dynamics, but in practise, it can not be used as a solver. The training and validation loss are shown in Figure 4.16.

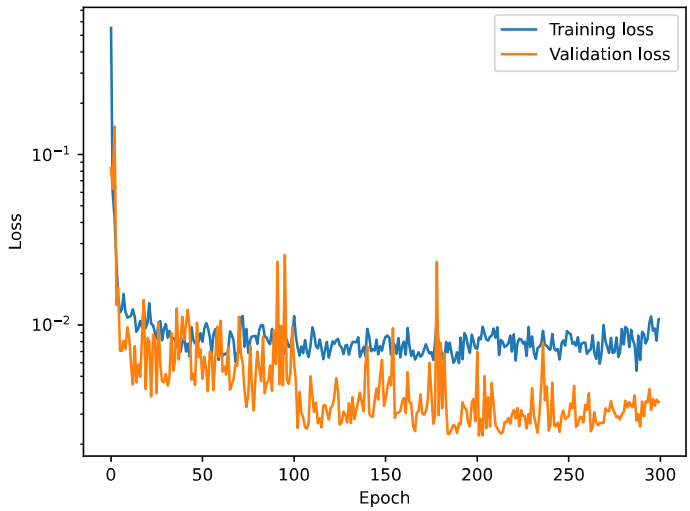


Figure 4.16: Training and validation loss.

From Figure 4.16, we see that a higher number of epochs, would probably not be beneficial, as the validation loss is increasing after a certain number of epochs.

We have also trained a LSTM model, but the results are not shown here, as the model was not able to learn the dynamics of the solution.

4.2.2 FNO

One of the main goals in this thesis is to use Fourier Neural Operators to solve the shallow water equations. We define a FNO model, which consists of an input channel, 64 hidden channels and an output channel. We use a Fourier basis with 16 modes and a batch size of 32. The model is trained using the Adam optimizer with a learning rate of 0.001, a total of 100 epochs and the critera is to minimize the mean squared error (MSE). The results of the model are shown in Figure 4.17.

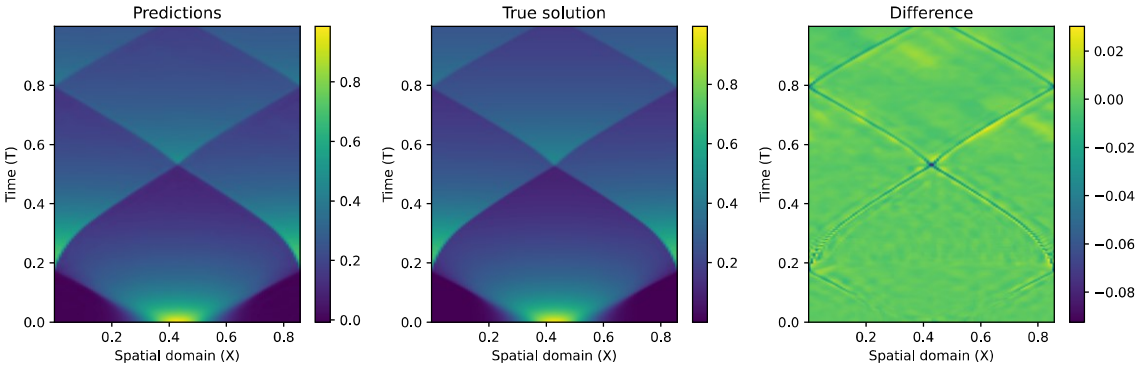


Figure 4.17: FNO predictions.

From Figure 4.17 we see that the FNO model is able to learn the dynamics of the solution, and it is able to predict the solution with high accuracy. From the figure we also see that the error follows the edges of the solution, which is expected, as the solution tends to be discontinuous. The loss for each iteration, which is epochs $\times$ batches, is shown in Figure 4.18.

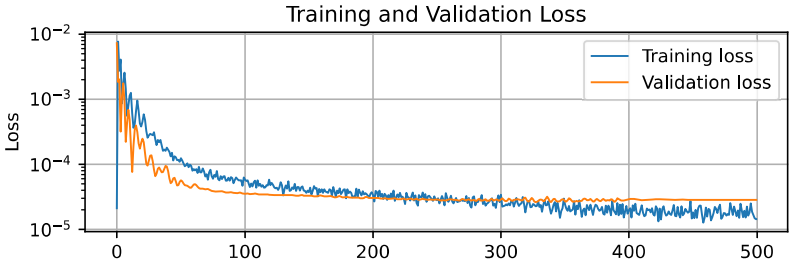


Figure 4.18: FNO loss.

From Figure 4.18 we see that the loss is decreasing for each iteration, and the model is able to learn the dynamics of the solution. To get a deeper understanding of the performance of the model, we consider the predictions for some given time steps, shown in Figure 4.19.

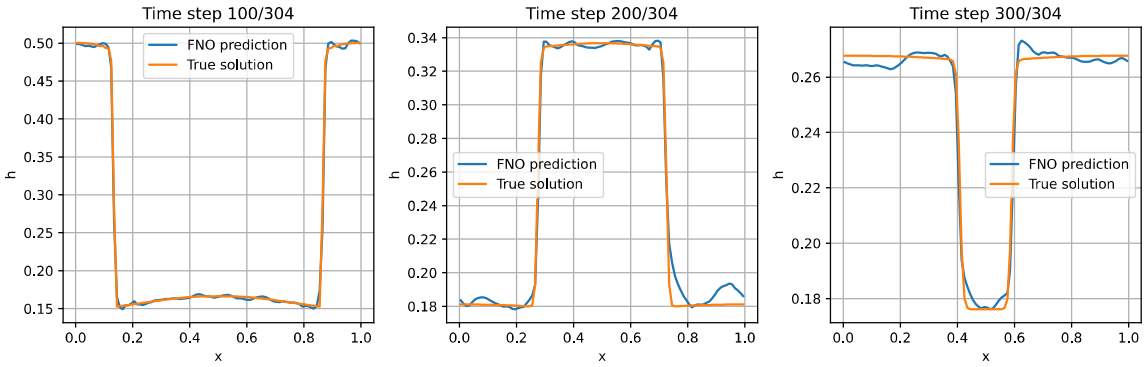


Figure 4.19: FNO predictions timesteps.

Figure 4.19 demonstrates that the predictions effectively capture the solution's dynamics, but the model has challenges when capturing discontinuities. The RMSE for the predictions is shown in Figure 4.20.

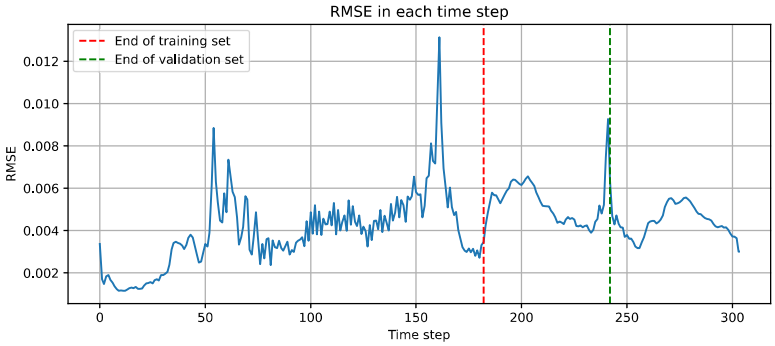


Figure 4.20: FNO predictions RMSE.

From Figure 4.20 we see that the RMSE is not increasing for the predictions, which is a good sign.

FNO Toro test 1

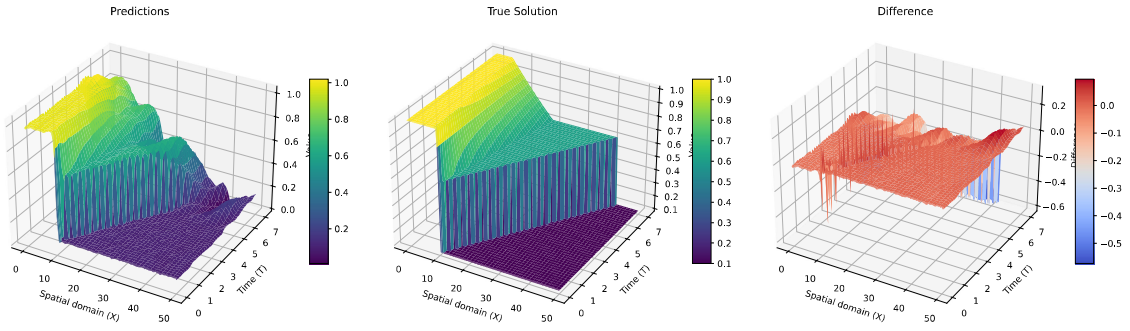


Figure 4.21: FNO Toro test 1 predictions in 3d.

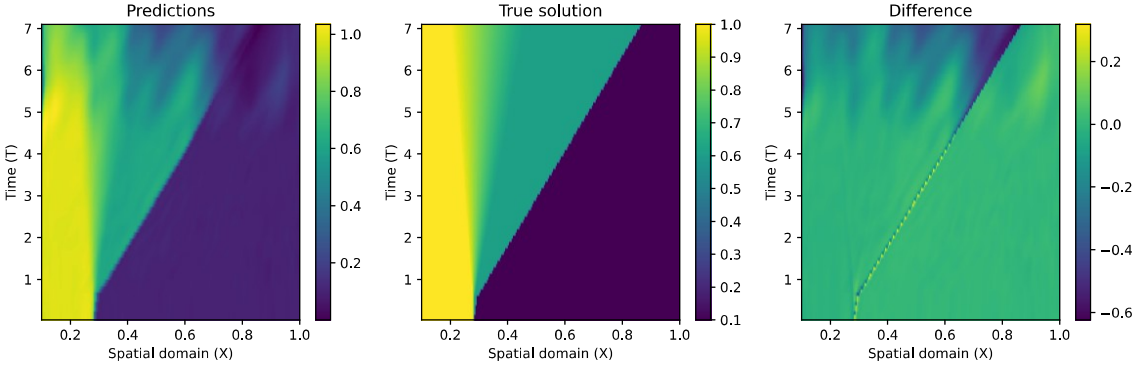


Figure 4.22: FNO Toro test 1 predictions.

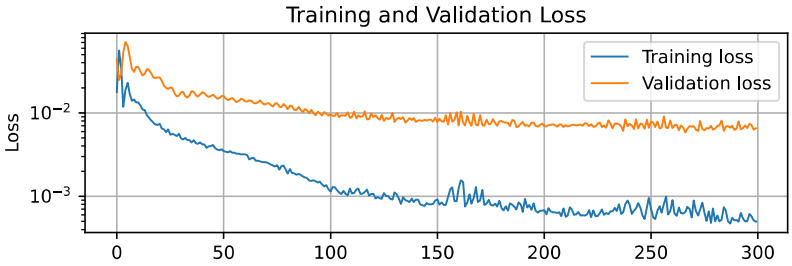


Figure 4.23: FNO Toro test 1 loss.

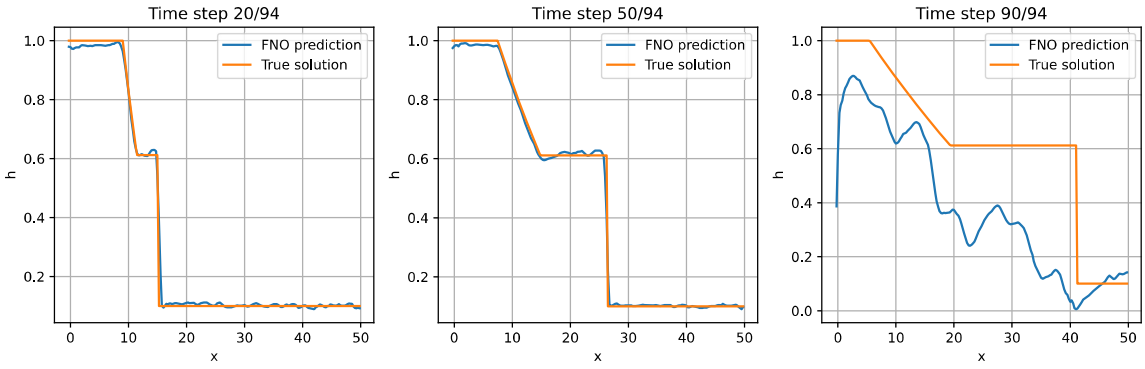


Figure 4.24: FNO Toro test 1 predictions timesteps.

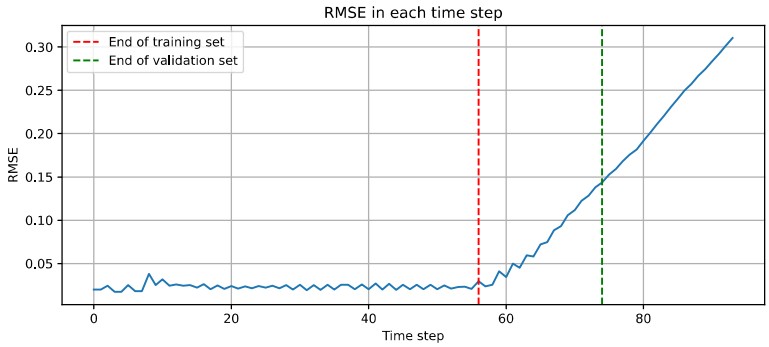


Figure 4.25: FNO Toro test 1 predictions RMSE.