Predicting the Steady State Solutions of the 1D Shallow Water Equations

Eystein Waade, Simon Huss and William Sjösten

Project in Computational Science: Report

January 2020

Abstract

This project is in collaboration with the company Schlumberger, which provides technology in oil and gas industry. Great interest lies in modeling the geological processes that are involved in the creation of oil and gas reservoirs. These processes evolve over very large time spans, making numerical simulations time-consuming. In this study one of the key processes is considered, namely water flow in channels and rivers, mathematically described by the shallow water equations. The aim is to construct an artificial neural network that can predict the steady-state solution of these equations, based on training data generated by a numerical solver using finite differences. The numerical solver is used to simulate the steady-state solutions for 2 700 different randomly generated bottom topographies and initial conditions. The result is used to train a feedforward neural network, which is then used to predict the steady-state solution given a new unseen bottom topography. The predictions, obtained from the neural network setting, are found successful, supporting the necessity to continue the work further, using even more advanced mathematical models.

List of abbreviations

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
FNN	Feedforward Neural Network
GPM	Geological Process Modelling
MSE	Mean Squared Error
ReLu	Rectified Linear Unit
SGD	Stochastic Gradient Descent
TVD	Total Variation Diminising

Contents

1 Introduction

For years geologists have been studying the ground to find clues about how the Earth ended up with with the forms and structures it has today. An important tool to help understanding various natural events on and beneath the surface of the Earth is the so-called Geological Process Modelling (GPM). The goal of GPM is to recreate these events using computer simulations that are based on the underlying physics. Schlumberger has developed their own GPM software which covers three key processes regarding the creation of oil and gas reservoirs: erosion, transport and deposition. The idea is to simulate the processes over a long period of time to investigate whether the position of a known reservoir can be recreated via numerical simulations. Ideally it is possible to use this knowledge in new simulations to locate new reservoirs. Figure 1 shows an example of a simulation of erosion, transport and deposition, using the Schlumberger GPM software.

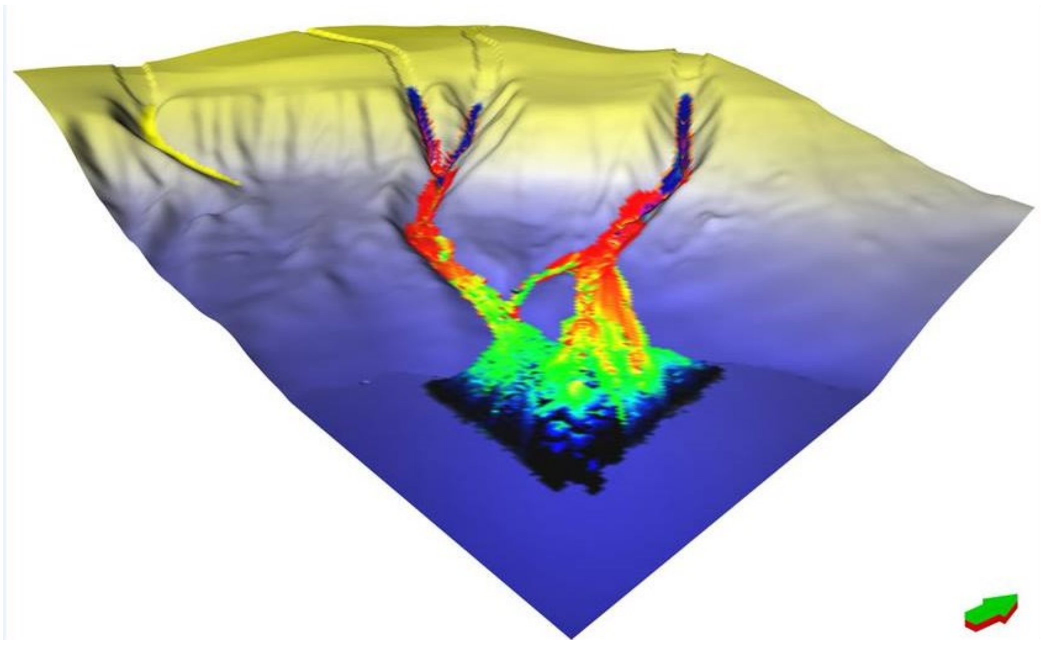


Figure 1: Example simulation from Schlumberger GPM software.

Geological processes are many, complex and often coupled. In addition they evolve over thousands or hundreds of thousands of years, making simulations computationally expensive. To save time, different processes in the GPM software are simulated separately. In this study we focus on one fundamental process in GPM, namely water flow in channels and rivers, commonly described by the shallow water equations. The river flows are integrated in the complete GPM software by simulating the flow until it reaches a steady state, at which it remains constant. New conditions are then introduced from separate simulations of erosion and deposition and the algorithm is repeated for a given time frame.

Current experience shows that the most time consuming part is the simulation of channel flow reaching its steady state. Therefore, accelerating this part of the simulations is of vital importance. The aim of this project is, as a proof of concept, to investigate the possibility of constructing an artificial neural network (ANN) that is able to predict the steady sate solutions of the 1D shallow water equations, thus, avoiding time-consuming simulations.

2 Problem formulation

The goal of the project is to construct an ANN that can predict the steady state solutions of the 1D shallow water equations, based on training data generated by a numerical solver using finite differences.

3 Theory

3.1 The Shallow Water Equations

The shallow water equations form a nonlinear system of hyperbolic partial differential equations describing the depth and average flow of a fluid. Neglecting rain, infiltration, viscous forces and the Coriolis force, the 1D shallow water equations in conservative form are given by

$$\begin{cases} h_t + (hu)_x = 0, \\ (hu)_t + \left(hu^2 + \frac{1}{2}gh^2\right)_x = -gh\left(b_x + S_f\right), \end{cases} \tag{1}$$

where h is the water depth, u is the average velocity of the water, g is the gravitational acceleration and hu is the discharge or flow. The source term in the second equation includes the bottom slope, b_x, and friction, S_f. Friction can be expressed in various ways [4]. Here a friction law from the Manning-Stickler's family is used:

$$S_f = n^2 \frac{u|u|}{h^{4/3}}, \tag{2}$$

where n is the Manning's roughness coefficient. In conservative form the shallow water equations constitute in fact a nonlinear system of conservation laws, representing conservation of mass and momentum [6].

For convenience, (1) is written in the more compact form

$$U_t + F(U)_x = R(U), \tag{3}$$

where

$$U = \begin{pmatrix} h \\ hu \end{pmatrix}, \quad F(U) = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{pmatrix} \quad \text{and} \quad R(U) = \begin{pmatrix} 0 \\ -gh\,(b_x + S_f) \end{pmatrix}. \tag{4}$$

In general, hyperbolic equations have wave-like solutions, propagating with characteristic wave speeds. For the shallow water equations these characteristics are determined by the eigenvalues of the Jacobian matrix

$$A = F'(U) = \begin{pmatrix} 0 & 1 \\ gh - u^2 & 2u \end{pmatrix}. \tag{5}$$

The eigenvalues of A are

$$\lambda_1 = u - \sqrt{gh} \quad \text{and} \quad \lambda_2 = u + \sqrt{gh}. \tag{6}$$

Thus, the waves move with non-constant wave speeds and will deform over time, possibly creating shocks [6]. This means that when solving the shallow water equations numerically, the method of choice must be able to handle the discontinuities arising from shocks.

3.2 Boundary Treatment

Another challenge lies in specifying correct boundary conditions, which is not straightforward due to the non-linearity. In linear theory the number of boundary conditions is the same as the number of eigenvalues of the Jacobian matrix. Moreover, the sign of the eigenvalues determines whether the conditions are imposed on the left or the right boundary points (in 1D). With non-constant eigenvalues the sign may differ both in time and space. To help with the boundary treatment a ratio called the *Froude number* is introduced:

$$Fr = \frac{|u|}{\sqrt{gh}}. \tag{7}$$

The Froude number corresponds to the Mach number in gas dynamics [6]. If the Froude number is less than one the flow is said to be *subcritical.* In this regime the water is generally deep and the flow is slow. Imposing the correct boundary conditions in the 1D case means specifying one condition on the left and one on the right boundary. If the Froude number is greater than one the flow is *supercritical.* Here the flow is faster than the gravity driven characteristic wave speeds, and both boundary conditions should be imposed at the same boundary.

The difficulties of the boundary treatment are associated with the fact that the Froude number is local in space. Consequently, in some situations the flow jumps between the two different regimes, depending on the bottom topography. When this happens the flow is called *transcritical.* In this study, the boundary conditions are imposed in the following way (if nothing else is specified): a non-homogeneous Dirichlet boundary condition for the discharge is always imposed at the left boundary. Next, the local Froude number at the left boundary is observed. If it is less than one then no more conditions are set at this boundary. If it is greater than one then an additional boundary condition in form of a homogeneous Neumann boundary is used for the water depth. Then the Froude number at the right boundary is observed. If it is less than one then a homogeneous Neumann condition for the discharge is imposed. Otherwise no boundary conditions are imposed.

4 Numerical Discretization Method

We discretize the one-dimensional shallow water equations, given by (1), with a MacCormack scheme. The computational domain is discretized using equidistant grid points, given by

$$x_j = (j-1)\Delta x, \quad j = 1, ..., M, \quad \Delta x = \frac{x_r - x_l}{M-1}, \tag{8}$$

where x_r and x_l are the right and left boundary points, respectively. The MacCormack method consists of two steps and when applied to (3) it takes the following form:

$$\begin{aligned} U_j^* &= U_j^n - \frac{\Delta t^n}{\Delta x}\left(F(U_{j+1}^n - F(U_j^n)\right) + \Delta t^n R(U_j^n), \\ U_j^{n+1} &= \frac{1}{2}\left(U_j^n + U_j^* - \frac{\Delta t^n}{\Delta x}\left(F(U_j^*) - F(U_{j-1}^*)\right) + \Delta t^n R(U_j^*)\right), \end{aligned} \tag{9}$$

where the subscript j denotes the discrete spatial point and the superscript n the discrete time. Furthermore, an adaptive time step Δt^n is used and set to fulfill the Courant-Friedrichs-Lewy (CFL) condition, which is a necessary condition to guarantee convergence of the discrete method. For the one-dimensional shallow water equation, the CFL condition is given by

$$C = \left\| u + \sqrt{gh} \right\|_\infty \frac{\Delta t}{\Delta x} \leq C_{\max}, \tag{10}$$

where C is the dimensionless Courant number and $C_{\max} = 2$ is the upper limit [7]. The adaptive time step is chosen as

$$\Delta t^n = C_{\text{pre}} \cdot \frac{\Delta x}{\left\| u^n + \sqrt{g h^n} \right\|_\infty}, \tag{11}$$

where C_{pre} is set to 0.5.

The first step in (9) is the so-called *predictor step*, where a temporary update of the solution is computed. The spatial derivative of the flux term F, given by (4), is here discretized using a forward difference approximation. The second step is the so-called *corrector step*, where the predicted solution is improved. Here, when approximating the spatial derivative of F, we use a backward difference formula. In this method, one could alternatively switch the backward and forward difference approximations, such that the backward difference approximation is used in the predictor step and the forward difference in the corrector step. For sufficiently smooth solutions the MacCormack method is of second order accuracy [5]. An advantage with this method compared to some other finite difference based methods, such as the Lax-Wendroff method, is that the MacCormack scheme does not include any computation of the Jacobian matrix.

A drawback of the MacCormack method is that it only gives reliable results when the solution is sufficiently smooth. Hence, spurious oscillations can arise in the solution close to discontinuities, or very sharp slopes. For general systems of nonlinear equations, it is almost impossible to obtain the stability results that are necessary to prove convergence. However, it has been found that by restricting the total variation to not grow with time increases the stability of the solution [5]. These methods are called Total Variation Diminishing (TVD) methods and limit the slope or the flux near discontinuities so that no spurious oscillations arise in the numerical solution. A modification of the standard MacCormack scheme is made by adding a TVD term, referred to as a TVD-MacCormack method. For details about the TVD term we refer to [2].

We use the standard MacCormack method and the TVD-MacCormack method for solving a test problem similar to the one presented by LeVeque [6]. We use MATLAB when implementing these numerical discretization methods and when setting up and solving the problem. The initial water height for the problem is given by $h_0(x) = 1 + 0.4e^{-5x^2}$ and is shown in Figure 2a. The velocity of the water is initially zero in the whole domain. We set the gravitational acceleration to $g = 1$ m/s^2 and the bottom to a flat surface, neglecting any friction. The initial hump evolves into a right- and a left-going wave with time. The problem is solved until the final time

$T = 3$ s. The result of the water height and discharge from the standard MacCormack method is shown in Figure 2c and 2d. It can be observed that there are two sharp slopes in the solution, which have given rise to spurious oscillations. Figures 2e and 2f show the corresponding result from the TVD-MacCormack method. Using this method the oscillations have disappeared, which indicates that the TVD-MacCormack method can successfully capture shocks.

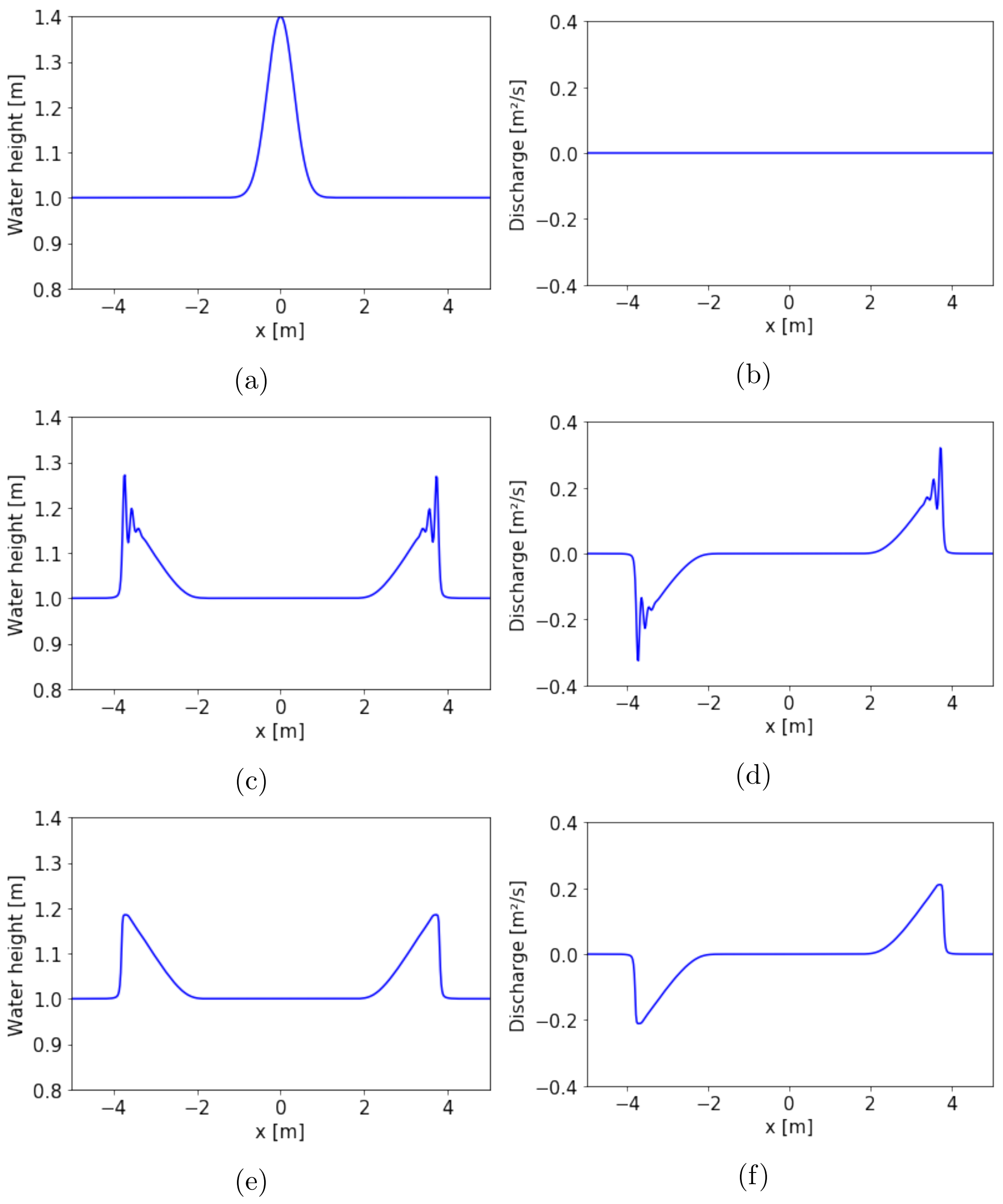


Figure 2: A test problem with an initial hump in the water height that evolves into a right- and left-going wave. The initial water height and discharge is shown in (a) and (b), respectively. The numerical solution with the standard MacCormack method after 3 s is presented in (c) and (d) and the corresponding result using the TVD-MacCormack method is shown in (e) and (f).

5 Data Generation

In order to learn an ANN, we need training data. In this study we generate numerous training data sets as follows. The training data consists of different randomized one-dimensional bottom topographies and initial conditions, together with the corresponding steady-state solutions of the water depth and discharge. In total, a set of 2 700 different bottom topographies are generated by the following principle: the domain is chosen to be a channel of length $L = 100$ m and divided into ten sub-domains. The bottom elevation at the left boundary is fixed to $b_l = 0.5$ m. At each sub-domain a random slope is introduced with probability $p_1 = 0.9$. Let v_i be a sample from an uniform distribution in the interval $(0, 1)$. A slope $a_k, k = 1, ..., 10$, is chosen as $a_k = -v_k \cdot b_l/L$, thus preventing the bottom elevation from being negative. Additionally, at each sub-interval, except for the left- and right-most sub-interval, a hump is generated with probability $p_2 = 0.8$. The height of the hump is v_q/l^2, $q = 2, ..., 9$, where l is the length of the sub-interval.

The initial state is always chosen as a surface at rest, that is $h_0 + b =$ constant, where the depth is randomized according to $h_0 = 0.8 + v_i$. The initial discharge is set to zero everywhere except for at the left boundary, where the inflow is chosen as $(hu)_0 = 0.8 + 2v_j$. The Manning's roughness coefficient is fixed to $n = 0.033$ throughout all tests.

Since there are no analytic solutions available to these randomized conditions, the corresponding steady states are simulated using the TVD-MacCormack method, up to the final time $T = 150$ s. Ultimately, the result of the procedure described above is 2 700 varying bottom topographies with the corresponding steady state solutions of the water depth and discharge.

6 Artificial Neural Network

6.1 Network Architecture

A feedforward neural network (FNN) is designed in PyTorch to handle data from three variables with 201 data points each. This is to match the 201 grid points in the numerical scheme when generating the training data. The motivation behind using 201 grid points in the spacial discretization is that it was found to be sufficiently accurate and fast. Additional accuracy could be achieved by increasing the number of grid points above 201, but was irrelevant in our one-dimensional case. The input for the network consists of the slope of the bottom topography, the initial water depth and initial discharge, i.e. b_x, h_0 and $(hu)_0$, as one vector of 603 values. The output of the network contains two variables with 201 data points each, corresponding to the steady state water depth and discharge. Thus, the output vector has a length of 402 data points. When the number of grid points in the numerical scheme were to be changed, a new network would need to be trained. For m grid points, the network architecture would have an input of length $3 \times m$ and an output of $2 \times m$.

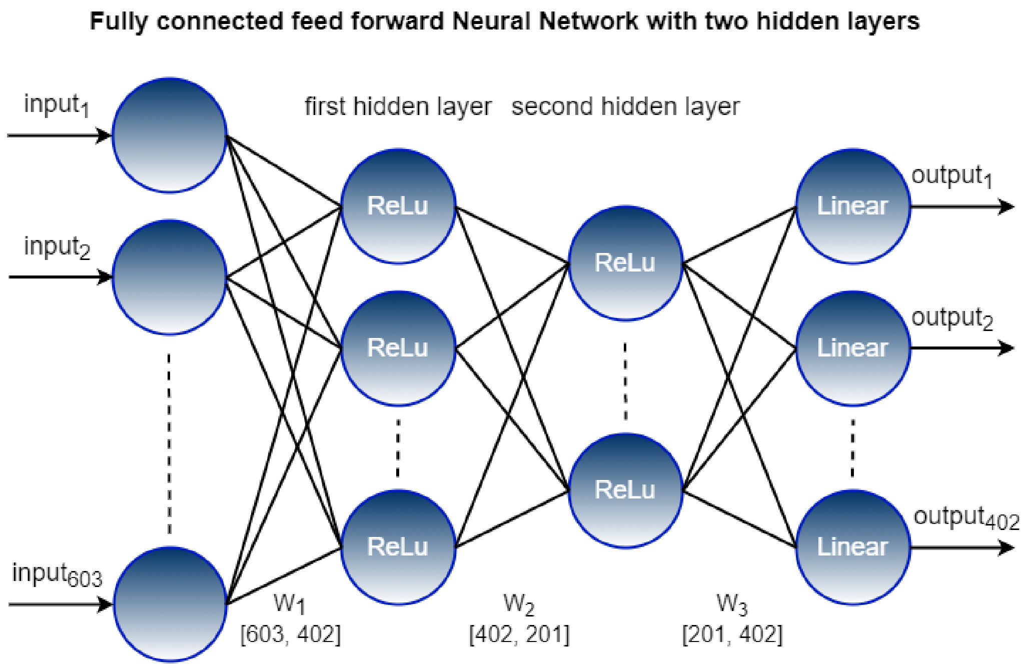


Figure 3: Visualisation of the feedforward neural network. Two hidden layers are used.

6.2 Activation Function

The activation function used in both hidden layers of the FNN, chosen in this study, is the Rectified Linear Unit, more commonly known as *ReLu*. In the context of neural networks, the ReLu activation function is a function defined as the positive part of its argument.

$$f_{\text{ReLu}}(x) = x^+ = \begin{cases} x & x \geq 0 \text{ m}, \\ 0 & \text{otherwise}. \end{cases} \tag{12}$$

6.3 Optimizer

The optimizer of choice is the Adam optimizer, an alternative to the "classical" stochastic gradient descent (SGD) and the weights are updated through backpropagation. The name *Adam* is derived from *adaptive moment estimation* [1]. The main difference between Adam and SGD is that Adam maintains a learning rate for each network weight when the SGD has a static learning rate for the whole network.

6.4 Loss function

Mean squared error (MSE) is used as loss function to calculate the loss at each iteration. The mean squared error penalizes outliers due to its quadratic nature.

7 Results

7.1 Convergence Analysis of the Discretization Scheme

The TVD-MacCormack scheme, presented in Section 4, is first verified on two test problems with known analytical steady-state solutions, presented in [4]. The computational domain is the same in both test problems and consists of a one-dimensional channel of length 25 m. The bottom topography is a single hump without friction, given by

$$b(x) = \begin{cases} 0.2 - 0.05(x-10)^2 & 8 \text{ m} < x < 12 \text{ m}, \\ 0 & \text{otherwise.} \end{cases} \tag{13}$$

The water surface level and the water discharge are given as initial conditions and set to $h(x) + b(x) = h_I$, where h_I is 2 m in the first test problem and 0.33 m in the second problem. The initial discharge is set to zero everywhere except for at the left boundary. The boundary conditions are imposed as

$$\begin{cases} (hu)(x) = q_0 & x = 0 \text{ m}, \\ h(x) = h_I & x = 25 \text{ m}, \end{cases} \tag{14}$$

where q_0 is set to 4.42 m/s^2 and 0.18 m/s^2 in the first and second problem, respectively. The first test problem is an example of subcritical channel flow and the second is an example of transcritical flow.

The numerical solutions are compared to known analytical solutions, obtained from the FullSWOF package [3]. FullSWOF is developed in C++ and contains a limited amount of analytical solutions to the shallow water equations. Figure 4 shows the numerical and analytical steady-state solution for the two test problems. The numerical solution is updated until time $T = 250$

s and 801 grid points are used. The numerical solution reaches the analytical steady-state solution in both cases, up to a certain accuracy.

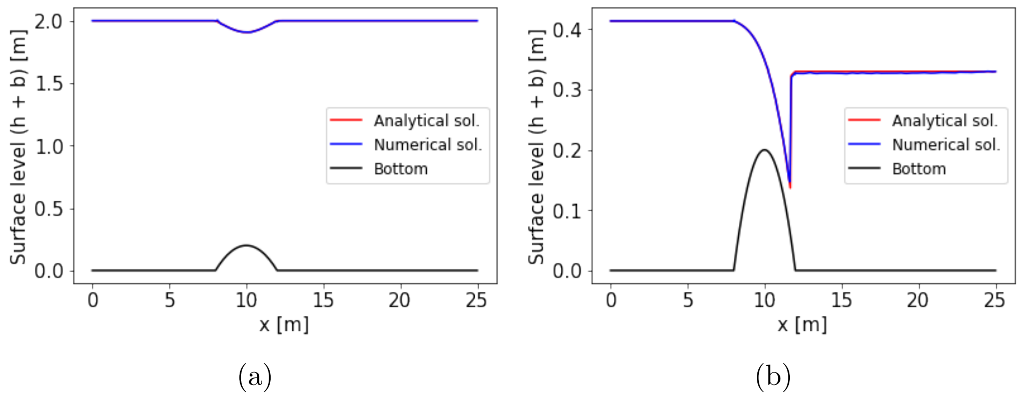


Figure 4: Water surface level at steady state for two test problems. The numerical solutions coincide with the analytical solutions. (a) Sub-critical flow. (b) Trans-critical flow.

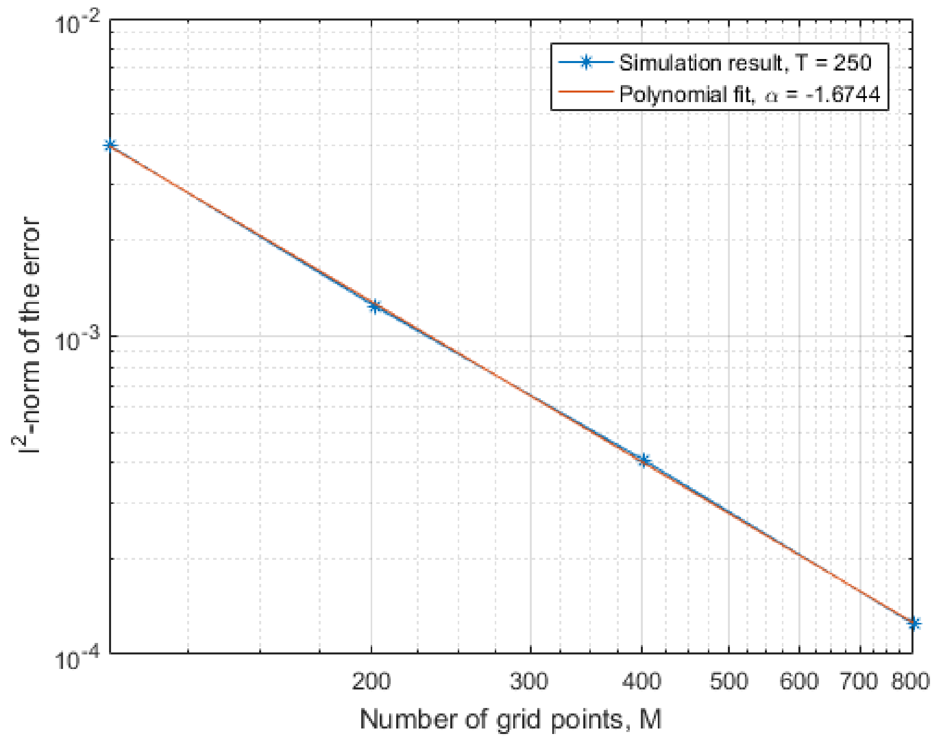


Figure 5: l^2-norm of the error as a function of number of grid points for the first test problem. A polynomial fit to the data is also made, which gives the convergence rate (α) of the TVD-MacCormack method for this test problem.

We make a convergence rate analysis for the first test problem (see Figure 4a). Four different grid-sizes are used, $m = 101$, $m = 201$, $m = 401$ and $m = 801$ points. The result in seen in Figure 5, where the l^2-norm of the numerical error is plotted against the grid size. The plot is in logarithmic scales. The convergence rate of the TVD-MacCormack method is estimated by using linear regression with the MATLAB-function `polyfit`. The rate is estimated to be 1.6744, and the slope is plotted as the red line.

7.2 FNN performance

Ten samples are removed from the training data set and are used for validation. The rest of the data is divided into training and test data with an 89% to 11% ratio. Figure 6 shows the training and test loss in the training process of the FNN. The network is trained for 1 000 epochs with a batch size of 100 and a learning rate of 0.01. The training and test losses approach zero with increasing number of iterations.

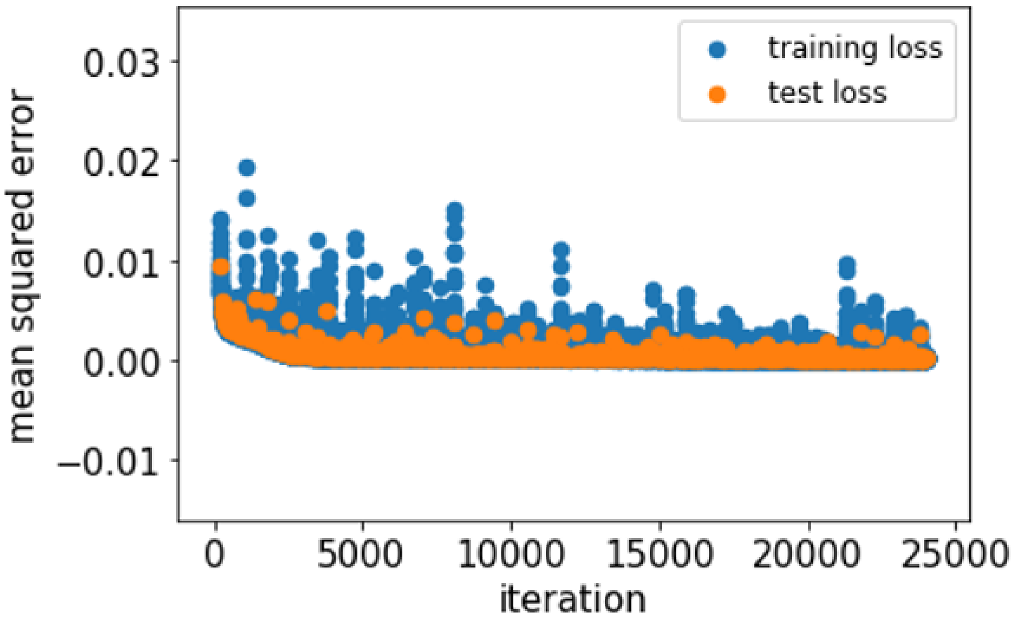
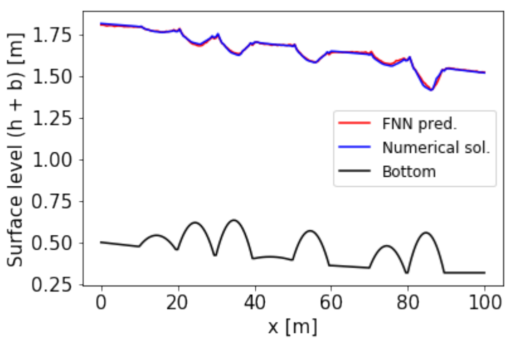
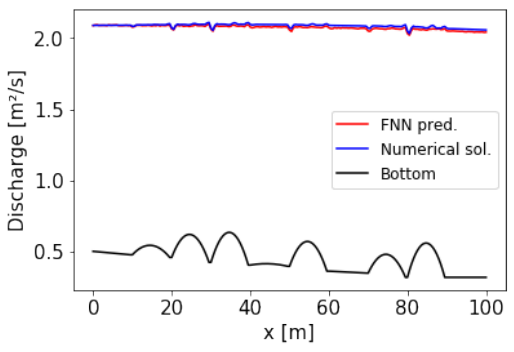


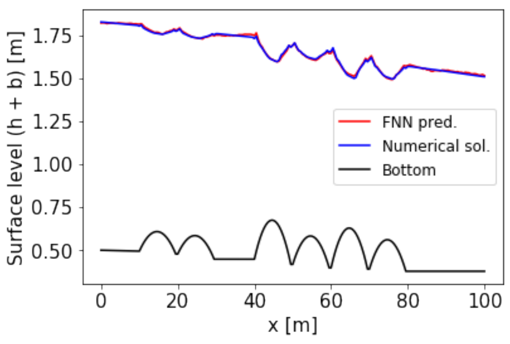
Figure 6: Training and test loss for the FNN over the training process.



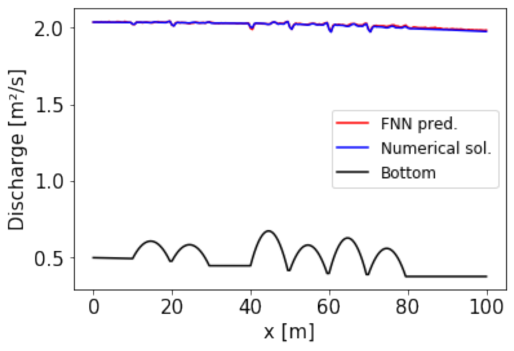
(a)



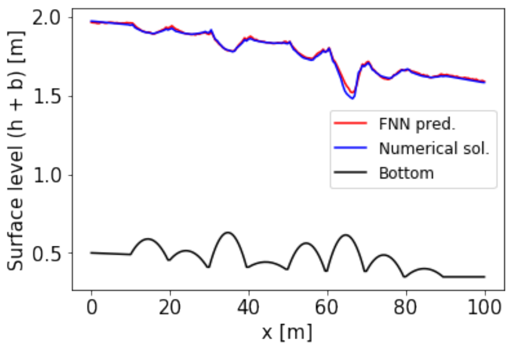
(b)



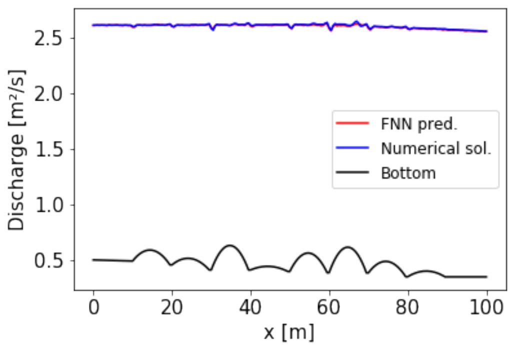
(c)



(d)



(e)



(f)

Figure 7: Three examples of a numerical steady-state solution and predicted solution from the FNN. Both the water surface level (left) and the related water discharge (right) are shown.

We test the performance of the FNN on the validation data set consisting of ten different bottom topographies and the corresponding numerical steady state solutions. Figure 7 shows the FNN predictions of the steady-state solutions for three of these. The numerical solutions are also plotted in the figures as references for the "true" solutions. The figures in the left column show the water surface level and the figures in the right column show the water discharge. It can be seen that the predicted steady-state solutions from the FNN almost coincide with the numerical solution in all three cases.

8 Discussion

In this study a feedforward neural network is trained to predict the steady-state solution of the 1D shallow water equations, given a specific bottom topography and initial state. The training data is based on numerical solutions using a TVD-MacCormack scheme with shock-capturing properties. The numerical discretization method is verified on a simple test problem, where analytical solutions of the steady states indeed are available. An analysis shows that the convergence rate is close to second order. This is what you expect since the standard MacCormack scheme is of second order and adding the TVD term comes with the cost of a slight decrease in accuracy. The scheme is found to be stable for the randomized bottom topographies and initial states used in this work.

The steady-state predictions from the FNN are of relatively good precision. Unfortunately, nothing can be said about the error of a prediction a priori. Standard numerical discretization methods have the advantage of a priori estimates that ensures that the error is bounded and decreasing with higher resolution. The only way to know the error of predicted steady states is to compare them to simulation results, which counterfeits the whole purpose of the project. It is also important to emphasize that the high performance of the neural network is related to a very specific setting being targeted. Even though randomness is introduced in the bottom topographies and initial states the conditions are still similar (see Section 5). Also the Manning's roughness coefficient is not randomized. The network is not tested on a completely different bottom topography, and doing so might lead to a less satisfactory accuracy. This problem can be solved by providing the network with a wider range of bottom topographies and allowing more variety in the initial states. The downside is that a larger training data set is more time-consuming to produce and the time to train the network will also increase. We recall that the system of equations is currently solved almost 2 700 times just to predict the steady-state for a few problems. Then again, once the training data is available and the network is trained, steady state predictions are obtained almost instantaneously. This trade-off must be considered.

The next step of this work would be to expand the model to two or three spatial dimensions. A tolerance should be implemented so that the final time solution is ensured to be the steady state solution. Only then meaningful run time comparisons between the numerical solver and the neural net approach can be made. Right now the final time is just chosen by observation. An idea is to compute the l^2-norm of the difference between the current solution and the previous time step and check if it is less than the given tolerance. Preferably, a wetting/drying algorithm should also be added to the scheme, though care must be taken so that the conservation laws still hold. Furthermore, we suggest to investigate the use of a convolutional neural network (CNN), due to its ability to deal with high dimensional input data. In two dimensions, the number of grid points is practically squared compared to one dimension. A relatively simple feedforward neural network is used in this project and it still performs well. Nevertheless, it would be interesting to compare to the performance and training time of a CNN.

Higher dimensions enable more variety of the bottom topographies, which is needed to generalise the training data. This is not done within the current study due to time restrictions. Obviously the 1D model is not realistic for most real-world problems. The major challenge of expanding to higher dimensions lies in how to handle problem dependent grids. In 1D the very same grid could be used for every different bottom topography and this problem was completely avoided.

9 Conclusions

The result shows that the feedforward neural network can be successful in reproducing the solutions of a standard numerical discretization method. This implies that a potential time save can be made by avoiding long simulations and instead predicting the steady state solutions instantly, with just information about the bottom topography and the initial state. A challenge lies in generalizing the model, and doing so will require a very large amount of data to be generated. Since the numerical discretization method is already reasonably fast in solving the shallow water equations in 1D, we believe that our approach is more viable in 2D or 3D. In applications where error estimates are essential it is better to go with a regular numerical discretization method. In order to compute posterior estimates, simulation results must be available as reference solutions. However, if simulated steady states are available then the perdictions from the FNN are worthless, since they will be of worse quality.

It is possible to develop a program generating more sophisticated bottom topographies than what is presented in our work. Such program should also be capable of producing steady state solutions, provided that the numerical discretization scheme is stable. With strong CPU's, large training data sets could be quickly generated. This makes our approach very interesting and there is a seemingly great potential in this field.

References

[1] Jason Brownlee. *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning.* https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/, July 2017 (November 14, 2019).

[2] Dongfang Liang, Roger A. Falconer, Binliang Lin. Comparison between TVD-MacCormack and ADI-type solvers of the shallow water equations. *Advances in Water Resources*, 29(12):1833 – 1845, December 2006.

[3] Oliver Delestre, Frédéric Darboux, François James, Carine Lucas, Christian Laguerre, Stephane Cordier. FullSWOF: Full Shallow-Water equations for Overland Flow. *The Journal of Open Source Software*, page 448, December 2017.

[4] Olivier Delestre, Carine Lucas, Pierre-Antoine Ksinant, Frédéric Darboux, Christian Laguerre, et al. SWASHES: a compilation of Shallow Water Analytic Solutions for Hydraulic and Environmental Studies. *International Journal for Numerical Methods in Fluids*, 73(3):269 – 300, 2013.

[5] Randall J. LeVeque. *Numerical Methods for Conservation Laws.* Birkhäuser, 2nd edition, 1992.

[6] Randall J. LeVeque. *Finite-Volume Methods for Hyperbolic Problems.* Cambridge University Press, 2004.

[7] Reinaldo Garcia, Rene A. Kahawita. Numerical solution of the St. Venant equations with the MacCormack finite-difference scheme. *International Journal for Numerical Methods in Fluids*, 6(5):259 – 274, 1986.