

Surrogate Model for Porous Dams

E. Holzbecher

German Univ. of Technology in Oman, Muscat, Oman

Freie Univ. Berlin, Berlin, Germany

Abstract

The earth dam problem, in its simplest form, is a classic example, as it has been studied for over a century using various modeling approaches: experimental, analytical, numerical, and statistical. One reason for this surely is its importance for real embankments, which play a crucial role in flood protection around the world. Another reason is that despite its simplicity at first sight, difficulties must be overcome to set up a model for a real site. Hydraulic parameters of the unsaturated soil are usually unknown, or if they are known, models are highly sensitive to their values. Neglect of the unsaturated zone leads to a free surface problem, demanding more sophisticated numerics. A model was set up that includes an inner part of the dam, which usually has a lower permeability than the outer parts, and drainage on the outflow side. It is assumed that under unsaturated conditions the permeability decreases exponentially with the suction head. For a normalized standard geometry, I present the derivation of a surrogate model using deep neural networks. The surrogate model delivers the maximum outflow across the downstream surface of the dam in dependence on two dam design parameters: the permeability contrast between the inner and outer parts of the facility and the extension of the drainage. The model can deliver information concerning the construction of an earth dam that can be important already in the design phase of a construction project.

Keywords: Earth Dam, Finite Elements, Surrogate Model, Deep Neural Networks

Introduction

Dams, levees, and dikes play a critical role in flood prevention. Earth dams can be simple earthen embankments or engineered designs with low permeable cores and/or drainage facilities. Unlike cement dams, water flows through earthen embankments. The flow of water through an earth dam is of crucial practical importance for the stability of the structure. Of particular interest is the outflow across the downstream surface in case of high floods. When the outflow velocity increases from its zero value in normal operation, it also increases the risk of dam breaking.

Models are an important tool for the estimation of flow through a porous embankment and, in particular, for the outflow at the downstream boundary as an important indicator for dam stability. Models can be important already in the design phase of a dam project. They can also be used for real-time simulations using the varying water table in the upstream reservoir as input. It is important to find a method that provides reliable results with minimal computational resources. That's where surrogate models come into play.

The problem of flow through an earth dam is a classic topic in the modeling of porous media flow. Starting in the 19th century, mathematicians and engineers developed analytical solutions for idealized 2D cross-sections concerning the geometry and geology [1,2,3,4]. Analytical solutions depending on geometrical and hydrogeological parameters can provide valuable insights into dam stability.

As a general classification of the approaches, one can distinguish between analytical, numerical, and statistical methods [5]. Both analytical and numerical methods are deterministic. They are based on formulations in terms of partial differential equations (pdes). In models of porous media, these are derived from Darcy's Law and the law of mass conservation. Statistical methods, which are now primarily associated with Artificial Intelligence (AI) and Machine Learning (ML) take a completely different approach [6]. Derived from statistics, they require a database. There is a distinction if the underlying database is obtained by field numerical models [7,8] or by field measurements [9].

Numerical Embankment Model

The numerical model simulates a standard normalized design for a 2D vertical cross-section through the dam. The model geometry is sketched in Fig. 1. It is a trapezoid shape with a slope of 2:1 on both sides. There is an inner core within the dam that also is of trapezoid shape with 1:2 slopes. At the downstream side drainage of variable length can be included at the bottom of the model region as the major outlet for the through flow. The water table on the upstream side is set to unity 1, normalizing all other geometry parameters.

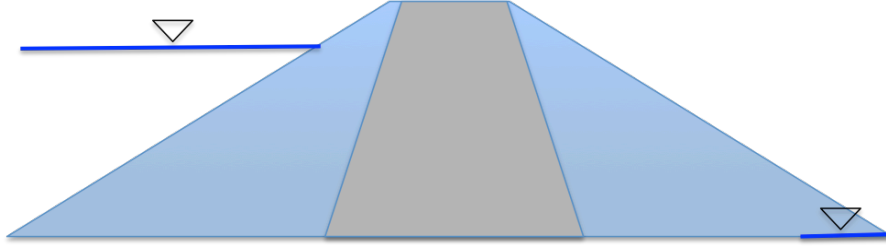


Figure 1. Dam design with core and drainage, numerical results for the saturated zone

Flow through the porous medium is governed by Darcy' Law, which states that the water flux is proportional to the hydraulic pressure. The proportionality factor is a characteristic of the porous medium, which is constant in saturated conditions but changes with pressure (i.e. suction) if the medium is unsaturated. Taking the principle of mass conservation into account and neglecting effects of variable density and viscosity, one can derive the differential equation for pressure p :

$$K \left(\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial z^2} \right) = 0$$

where x denotes the horizontal spatial coordinate, z the vertical spatial coordinate, K the hydraulic conductivity, ρ the water density and g the acceleration due to gravity. In our model K takes a small value in the reinforced core and a higher value in the outer parts of the dam, with conductivity ratio $\lambda = K_{core}/K_{earth}$. The conductivities in the unsaturated zone depend on the pressure: $K_{unsat} = K_{sat} \min(\exp(p/1000, 1)$. The differential equation can be simplified for hydraulic head $h = p - \rho g z$:

$$K \left(\frac{\partial^2 h}{\partial x^2} + \frac{\partial^2 h}{\partial z^2} \right) = 0$$

In the normalized model there are Dirichlet boundary conditions $h=1$ at the inlet and $h=0$ at the outlet. On the downstream slope the pervious layer condition is required:

$$u_n = \begin{cases} (h - z) / c & \text{for } h > z \\ 0 & \text{for } h \leq z \end{cases}$$

where n denotes the normal direction. On all other boundaries the no-flow boundary condition $dh/dn=0$ is set. For the numerical model we use the COMSOL Multiphysics® software (see [10]). The mesh discretization is extra fine, with two local refinements: along the downstream slope with maximum element size 0.025, and at the extreme downstream point with maximum element size 0.005 units.

Deep Neural Network

All networks have two inputs λ in the interval $[1, 10]$ and length of the drainage l in the interval $[0, 0.5]$. The single output is the maximum positive velocity across the downstream slope. In between there are several hidden layers. A simple network was set up as a reference. It has two hidden layers with 4 and 2 neurons. That leads to 25 network parameters that combine as follows: there is a bias parameter for each neuron, including the output layer (here: $4+2+1=7$) and 16 network parameters for the connections of the hidden layers (here $2*4=8$ for the connection of the input layer with the first hidden layer, $4*2=8$ between both hidden layers, and $2*1$ for the connection of the second hidden layer with the output layer). The activation function for all nodes is tanh. There is scaling to the interval $[0, 1]$ for all parameters at all nodes, except the output parameter.

The surrogate model is based on a data set of 50 randomly determined input points. Training and validation are performed using the Adam method, with a learning rate of $2e-3$. The validation data fraction was set to 0.2, i.e. 80% of the data is used for training, and 20% for validation. The number of epochs was handled flexibly, but with a maximum of 90000.

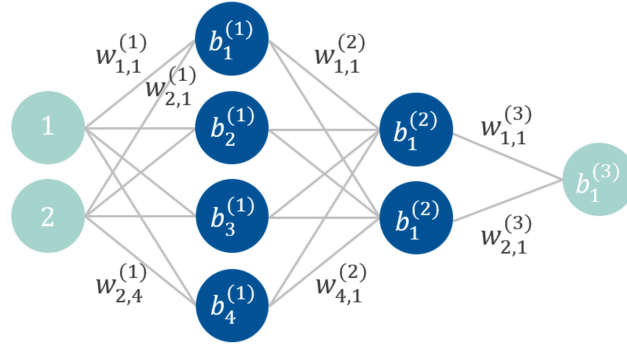


Figure 2. Simple network design, example

Results

Figure 3 shows the flow through a realistic design with a core and drainage as output from a fixed geometry model. The lower conductivity of the core causes the buckles of the water table interface. Note that in this scenario the water table almost touches the downstream surface of the embankment. For increased core conductivity or shortened drainage length, there might be outflow on the downstream side of the dam. High outflow velocities at this boundary lead to the movement of the solid material (mostly sand) and thus are the major causes for dam breaks. Dams should be designed in a way that there is no outflow on the downstream side of the facility.

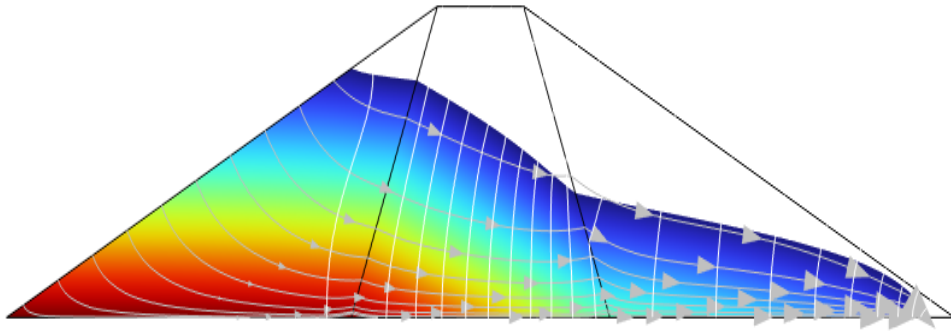


Figure 3. Numerical results within the saturated zone: pressure (red: high, blue: low) and streamlines (grey)

A typical development of the losses with progressing epochs is given in Figure 4. In the beginning both the validation and training losses are decreasing simultaneously. At a certain point of the development (in the figure at around 14000 epochs with losses in the range of $2e-4$) both loss functions start to fluctuate, which indicates some severe re-adjustment of the model. These fluctuations keep on appearing if the DNN continues. At the start of this fluctuation phase losses still decrease, but with the improvement tends to become smaller and smaller. In the figure the minimum value of the loss functions with $5e-6$ is reached after about 40000 epochs. Further continuation of the computations results in increasing loss functions, which surely indicates the state of overfitting.

Modification of Network Design

Various network designs were explored for the data set of 50 points, the Adam method, learning rate $2e-3$, validation rate fraction 0.2 (number of training points: 40, number of validation points: 10), and maximum number of epochs 90.000. The input parameters were allowed to vary as follows: λ between 1 and 10, and d between 0 and 0.5.

The reference model with the above-given design and settings delivered the minimal validation loss of $9.88e-6$ at epoch 50300 and a minimum training loss of $1.76e-5$ at epoch 46331. We experimented with modifications of the network design. The following table shows the complexity of the model, measured by the total number of network parameters, and the performance by the minimum validation loss.

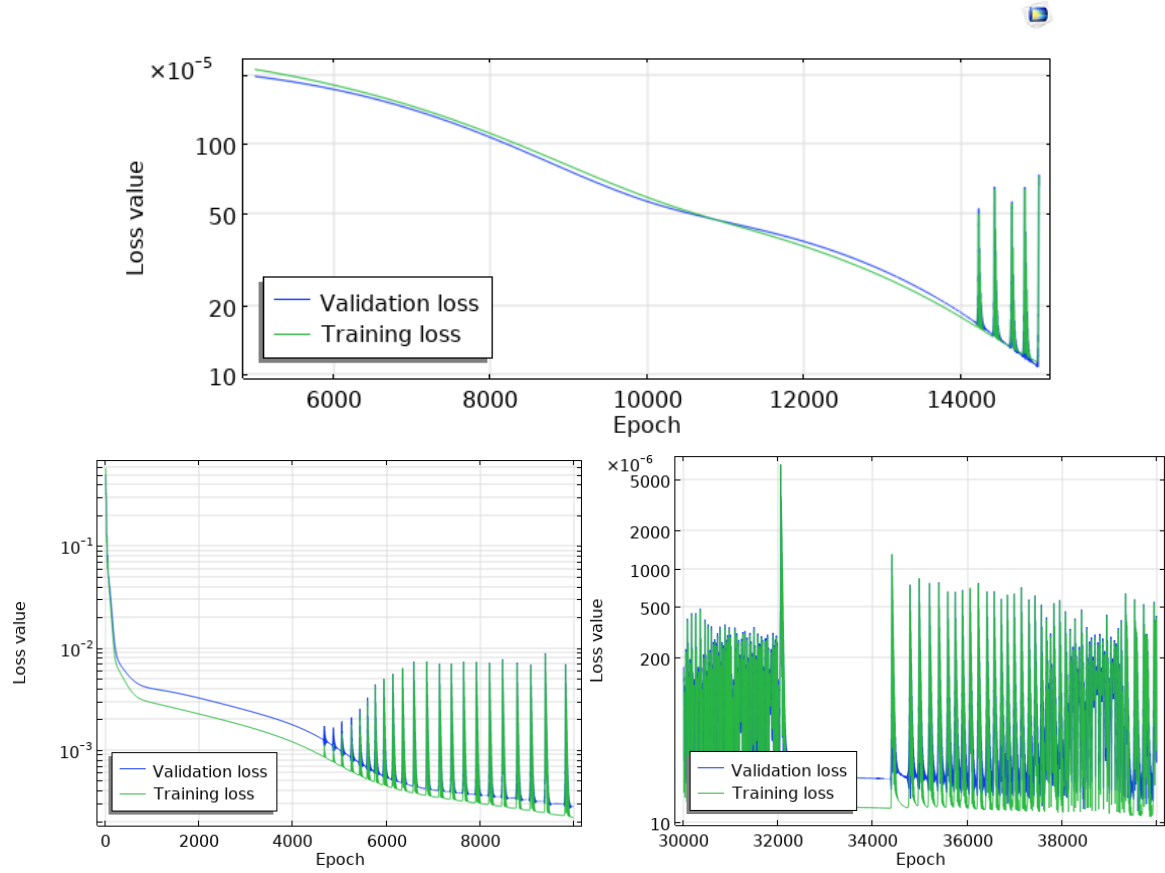


Figure 4: Examples of convergence plots, (first epochs, later epochs (for 2,4,8,16,4,1))

Network architecture	Number of parameters	Validation loss	Epochs
2,4,2,1	25	9.88e-6	50300
2,4,8,2,1	73	5.70e-6	89859
2,4,7,2,1	66	6.71e-6	76279
2,4,16,4,1	165	7.72e-6	38353
2,5,10,2,1	100	8.54e-6	87035
2,5,10,5,1	136	8.04e-6	64187
2,4,8,16,2,1	233	8.21e-6	78713
2,4,8,8,2,1	145	7.79e-6	36404
2,4,8,8,4,1	165	6.66e-6	69816
2,4,8,16,4,1	269	5.45e-6	40380
2,4,8,32,16,1	885	7.34e-6	55407

Table 1. Performance of DNN, depending on the network design

The best performance was reached with the more complex model with 269 network parameters. Compared to the simplest network with only 25 parameters, the performance increased roughly by a factor of 2. However, the time for training taken by the more sophisticated model is much smaller because the optimum before overfitting is reached after 13142 epochs, while the simple network requires more than 8 times more epochs to reach that point.

Influence of Scaling

The effect of scaling was studied for the simplest 2-4-2-1 network and the best 2,4,8,16,4,1 network. No scaling for all parameters resulted in a validation loss of 1.06e-5 at 32862 epochs for the former and 1.00e-5 at 38617 epochs for the latter. If scaling is performed for all, input and output, parameters the validation loss amounts to 0.0068 at the 61557 epochs for the simplest network and to 0.011 at 23603 epochs for the best network. Comparison with the results obtained with the reference approach (scaling of input, no scaling for output parameters!) shows for all cases that the global scaling or non-scaling approaches yield less accurate predictions.

Influence of Learning Rate

The influence of the learning rate was examined for the simplest network 2,4,2,1-layer architecture. Table 2 shows the validation loss and the corresponding number of epochs for selected values of the learning rate. Best performance was reached for the learning rate of 2.5e-3, worst performance for 1e-3.

Learning rate	0.5e-3	1e-3	1.5e-3	2e-3	2.5e-3	3e-3	3.5e-3
Validation loss	1.38e-5	1.52e-5	8.81e-6	9.88e-6	7.86e-6	8.62e-6	8.86e-6
Epochs	89544	89555	80167	50300	87236	89645	88744

Table 2. Performance of DNN, depending on the learning rate

Influence of Activation Function in Output Layer

The activation function of a layer determines how the output is computed from the input values and the bias. In the above noted terminology (Figure 2) the output of node (neuron) j using the activation function f can be written as:

$$y_j = f(\sum_i x_{ij} w_{ij} + b_j)$$

The indices i denote the neurons of the preceding layer, x_{ij} are the inputs and b_j the bias. In the notation of the parameters the superscripts for the layer are omitted for simplicity. Using the linear term in the brackets as it is does the simplest activation. One may say, there is no activation function, or that f is the identity function. Other common activation functions are the sigmoid $f(x)=1/(1+\exp(-x))$, ReLu $f(x)=\max(0,x)$, and $f(x)=\tanh(x)$. In COMSOL the latter is the default. The exponential linear unit ELU is defined by:

$$f(x) = \begin{cases} x & \text{for } x > 0 \\ \alpha(e^x - 1) & \text{for } x \leq 0 \end{cases}$$

with tuning factor α , which by default is set to 1. In the first evaluation of the activation functions we keep the default $\tanh(x)$ for all layers and change the activation function only for the output layer. The following table shows the results. Here the linear choice, i.e., no activation function, delivers the most accurate result.

The effect of the activation function was examined for the 2,4,2,1 layer network with learning rate 2e-3. After the first runs with 25000 as maximum number of epochs, a second series of runs was started in which 90000 epochs were performed. In the cells of Table 3 the results of the first series is noted above, for the second below.

Activation function in output	tanh	linear	sigmoid	ELU	ReLU
Validation loss	3.26e-5 9.88e-6	2.80e-5 7.92e-6	2.40e-5 1.51e-5	1.75e-5 9.78e-6	2.02e-5 2.02e-5
Corresponding number of epochs	24843 50300	24818 33168	25000 90000	24915 33511	1 1

Table 3. Performance of DNN, depending on the activation function

For fewer epochs the best results were obtained using the ELU function for activation. If more epochs were performed, the tanh and linear functions reached a higher improvement. Finally the linear function turned out most accurate. The sigmoid function showed the least improvement by doing more epochs. Moreover, the sigmoid always reached the best only at the maximum of allowed epochs. The ReLU gave good estimates in the first epoch, that strangely were not improved in further epochs.

Change of Method

COMSOL Multiphysics® offers two methods for trying and validation. Both methods were applied for the proposed network with the learning rate of 2e-3 and all other parameters being identical. Figure 5 visualizes the surrogate functions obtained by both runs in dependence of the input parameters. On one axis the drainage length varies from 0 to 0.5 m, on the other axis the permeability ratio varies from 1 to 10. The left sub-figure results, if the SDG method is used. The right figure is obtained by using Adam..

The error, measured by the validation loss, amounts to $1.8\text{e-}6$ for Adam and 0.0043 for SGD, which demonstrates the superiority of the Adam method. This may also be confirmed by visual inspection of the figures. The surface on the right has a more regular shape and doesn't show unrealistic rise and decline at the three blue corners.

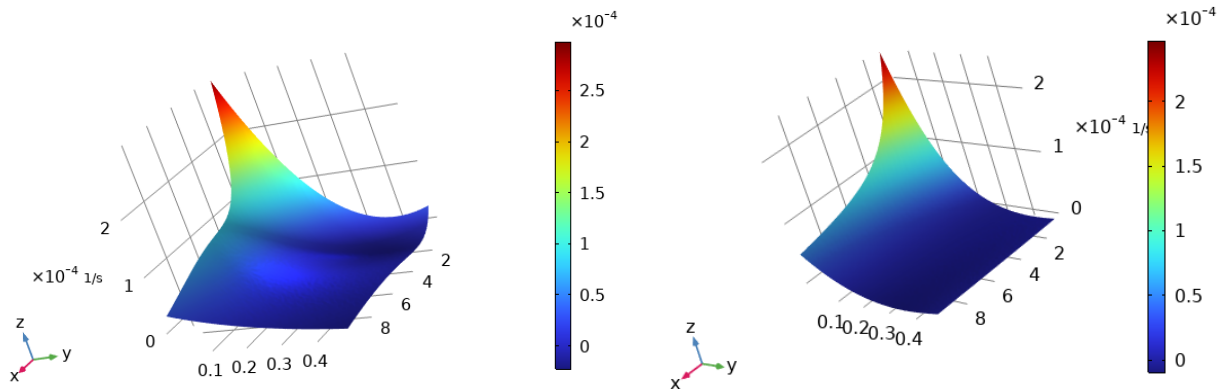


Figure 5: Surrogate Function for SGD (left) and Adam (right)

Conclusions

I examined the influence of various options concerning the neural network design and solution strategies: number of hidden layers and neurons, scaling, learning rate, and choice of activation function. The following conclusions could be drawn from the experiments: (1) the 'tanh' activation function should be used as the default for all nodes; (2) all parameters at all nodes should be scaled to the interval $[0,1]$, except the output parameter, which should not be scaled; (3) $2\text{e-}3$ is the best starting value for the learning rate; (4) 'Adam' should be used as the method. For good results, i.e. low validation loss, the DNN network should not be too simple, but it should also not be too complex in order to keep the number of parameters limited and, depending on this, the computation time. Our best results were obtained with a network of 4 hidden layers, with 4, 8, 16, and 4 nodes, which leads to 269 network parameters

References

- [1] L. Casagrande, "Näherungsmethoden zur Bestimmung von Art und Menge der Sickerung durch geschüttete Dämme", Thesis, Technische Hochschule, Vienna, 1932.
- [2] P.Y. Polubarinova-Kochina, "Theory of groundwater movement", Princeton Univ. Press, Princeton, 1962.
- [3] M. Muskat, "The Flow of Homogeneous Fluids through Porous Media", McGraw-Hill, New York, 1937.
- [4] J. Kozeny, "Grundwasserbewegung bei freiem Spiegel, Fluss- und Kanal-versickerung". *Wasserkraft und Wasserwirtschaft*, Vol. 26, 28, 1931.
- [5] E. Holzbecher, "Modelling Flow through Earthen Embankments", 9th International Symposium on Flash Floods in Wadi Systems (ISFF9), 2025 (Proceedings to appear).
- [6] M. Nouri and F. Salmasi, "Predicting Seepage of Earth Dams using Artificial Intelligence Techniques", *Irrigation Sciences and Engineering*, Vol. 42(1), pp. 83-97, 2019
- [7] A.A.J. Jamal, "Analysis and estimation of seepage through homogeneous dam without filter", *Diyala Journal of Engineering Sciences*, Vol. 9(2), pp. 83-94, 2016.
- [8] R.S. Abd, R.H. Irzooki, and A. Al-Obaidi, "Easy equation for estimating seepage discharge through earth dams based on impermeable foundation, Water Resources in Iraq: Perspectives and Prognosis", ICWRPP 2022, IOP Publishing 1120, 012023, 2022.
- [9] I. Rehamnia, A.H.S. Al-Janabi, S.Sh. Sammen, B.T. Pham, and I. Prakash, "Prediction of seepage flow through earthfill dams using machine learning models" *Hydro Research*, Vol. 7, pp. 131-139, 2024.
- [10] E. Holzbecher, "Flow through earth embankments and dams - a comparison of numerical approaches", COMSOL Conference, Amsterdam, 2025.

