EFFICIENT GROUNDWATER FLOW MODELING USING DEEP NEURAL OPERATORS

Maria Luisa Taccari[*1], **He Wang**[†2], **Somdatta Goswami**[‡3], **Mario De Florio**[§4], **Jonathan Nuttall**[¶5], **Xiaohui Chen**[‖1], **and Peter K. Jimack**[**6]
[1]School of Civil Engineering, University of Leeds, Leeds, UK
[2]Department of Computer Science, University College London, London, UK
[3]Department of Civil and Systems Engineering, Johns Hopkins University, Baltimore, U.S.A
[4]Division of Applied Mathematics, Brown University, Providence, U.S.A
[5]Deltares, Delft, The Netherlands
[6]School of Computing, University of Leeds, Leeds, UK

January 8, 2024

ABSTRACT

This work introduces a cost-effective emulator for groundwater flow modeling using deep neural operators (DeepONet). We address the computational challenges in traditional groundwater modeling by proposing a novel approach that efficiently and accurately forecasts aquifer responses. The model demonstrates excellent performance across various scenarios, including time-dependent and nonlinear systems.

Keywords Deep neural operator · Groundwater flow · Surrogate modeling · Deep learning

1 Introduction

Traditional numerical models, such as MODFLOW [6], have been extensively used for simulating groundwater flows. However, these models often face computational challenges, especially when dealing with large-scale or highly nonlinear systems [2, 3, 9].

Recent progress in deep learning has introduced innovative approaches for functional approximation, making a significant impact in various fields including fluid dynamics and subsurface flows [4, 11, 5]. Deep convolutional neural networks (CNNs) have been successfully employed as surrogate models in dynamic multiphase flow problems [12, 10]. However, these methods have limitations, particularly in their dependency on lattice grid structures and the need for retraining for different mesh resolutions [14].

In response to these challenges, deep operator networks (DeepONets) have emerged as a promising solution. Introduced by Lu et al. [8], DeepONets leverage the universal approximation theorem for operators [1], offering a framework that learns mappings between infinite-dimensional Banach spaces. This approach enables the solution evaluation at any arbitrary spatial and temporal location [7].

[*] cnmlt@leeds.ac.uk
[†] he_wang@ucl.ac.uk
[‡] sgoswam4@jhu.edu
[§] mario_de_florio@brown.edu
[¶] Jonathan.Nuttall@deltares.nl
[‖] X.Chen@leeds.ac.uk
[**] P.K.Jimack@leeds.ac.uk

This study introduces a deep neural operator-based approach as a surrogate model for groundwater flow, offering a computationally efficient alternative to standard numerical methods. We aim to demonstrate its applicability through various experimental setups, addressing both forward and inverse problems in subsurface flows.

2 Problem Statement and Methodology

The movement of groundwater is governed by a PDE that combines Darcy's Law and the principle of mass conservation. Traditional methods like the finite difference apporach based MODFLOW have been extensively used but face limitations in handling large-scale, complex simulations efficiently [6, 3].

Deep neural operators learn complex, non-linear functions in spaces of unbounded dimensions, a concept discussed in [8] and [1], offering an alternative for groundwater level predictions. The traditioanl DeepONet framework comprises two deep neural networks: the branch network and the trunk network. The branch network processes the input function, denoted as u, at fixed sensor points $\{x_1, x_2, \ldots, x_m\}$. The trunk network encodes the spatiotemporal coordinates $\zeta = \{x_i, y_i, t_i\}$, at which the operator's output is evaluated for loss calculation. This learning mechanism operates under a general paradigm where the sensor points $(x_i{}_{i=1}^{m})$ for u evaluations do not require uniform spacing, but they must maintain consistency across evaluations. The branch network inputs $[u(x_1), u(x_2), \ldots, u(x_m)]^T$ and outputs $[b_1, b_2, \ldots, b_q]^T \in \mathbb{R}^q$, whereas the Trunk Network inputs ζ and outputs $[t_1, t_2, \ldots, t_q]^T \in \mathbb{R}^q$. The outputs of these subnetworks are integrated via a dot product, yielding the final output. An additional bias term $b_0 \in \mathbb{R}$ is incorporated to enhance expressiveness, resulting in the approximation $\mathcal{G}(u)(\zeta) \approx \sum_{i=k}^{q} b_k t_k + b_0$. The trainable parameters θ are optimized by minimizing a mean square error loss function.

The study comprises five computational experiments, each addressing a unique aspect of groundwater modeling:

- **E1:** Learning the forward mapping from conductivity fields to hydraulic heads for a stationary well scenario.
- **E2:** Expanding to multiple input functions, including various well locations and varying conductivity fields.
- **E3:** Undertaking inverse modeling to estimate hydraulic conductivity from observed hydraulic heads.
- **E4:** Modeling time-dependent problems with changing pumping rates.
- **E5:** Solving nonlinear problems with head-dependent boundary conditions.

Each experiment demonstrates DeepONet's adaptability and effectiveness in addressing various challenges in groundwater modeling. This work presents the results of the first three experiments. For details on the remaining experiments, the readers are referred to our published work [13].

3 Results

In this section, we demonstrate through experiments $E1$, $E2$ and $E3$ that DeepONet can be employed for approximating a range of groundwater flow simulation problems, accurately and efficiently.

In Experiment $E1$, DeepONet is tasked to learn the mapping from hydraulic conductivity to the hydraulic head in a scenario with a fixed well location. The model achieved a mean square error (MSE) of 1.8×10^{-5} during training and 2.5×10^{-4} in testing over $N_{train} = 1000$ and $N_{test} = 200$ samples respectively. Training took 283 seconds for $24,600$ iterations. Figure 1 showcases the model's capability to predict the hydraulic head around the well accurately for a conductivity field unseen during training. DeepONet accurately predicts the pressure buildup and sharp increase of the hydraulic head.

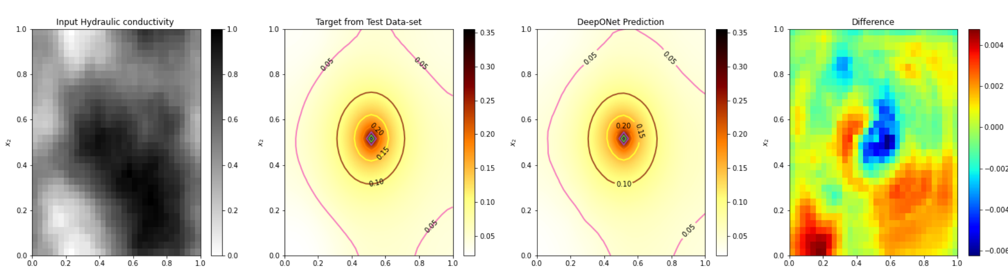

Figure 1: Hydraulic head predictions for two test cases with unseen conductivity fields in Experiment $E1$.

In Experiment $E2$, we addressed the limitations of the vanilla DeepONet in scenarios with varying well locations. The vanilla architecture struggled to accurately capture sharp gradients near the pumping wells when the position of the well was changed for each sample across training and testing dataset. This was evident as the model produced either inaccurate predictions of the location of the pressure front or overly smoothed gradients near the source terms. Multiple approaches to encode the well's location in the branch network were explored, including Cartesian coordinates and Gaussian functions centered at the well, but these methods did not yield satisfactory results. It became clear that informing the trunk network about the well location was crucial for accurate learning. The incorporation of well coordinates into the trunk network input improved the model's ability to localize the basis functions effectively around the wells.

However, in more complex scenarios involving multiple wells or non-point features, this method proves impractical. Consequently, we adopted a novel approach, illustrated in Figure 2, that interlinks the hidden layers of the branch and the trunk networks. In this design, we encode the well location as a binary map, which is then concatenated to the hydraulic conductivity field in the branch network, and outputs from the branch network's pooling layers are integrated with those from each layer of the trunk network. The output from the branch network undergoes flattening and is processed through a dense neural network with a *Sigmoid* activation function. The resultant vector, interpretable as a set of coefficients, matches the dimensionality of the trunk network's corresponding hidden layer, allowing for their combination via an inner product. This combined output then advances through subsequent layers of both the trunk and branch networks, following reshaping and concatenation with the pooling layer's output. Our findings show that this architecture provides precise predictions of the hydraulic head's steep gradients in Experiment $E2$, an improvement over the more generalized predictions of the vanilla DeepONet.

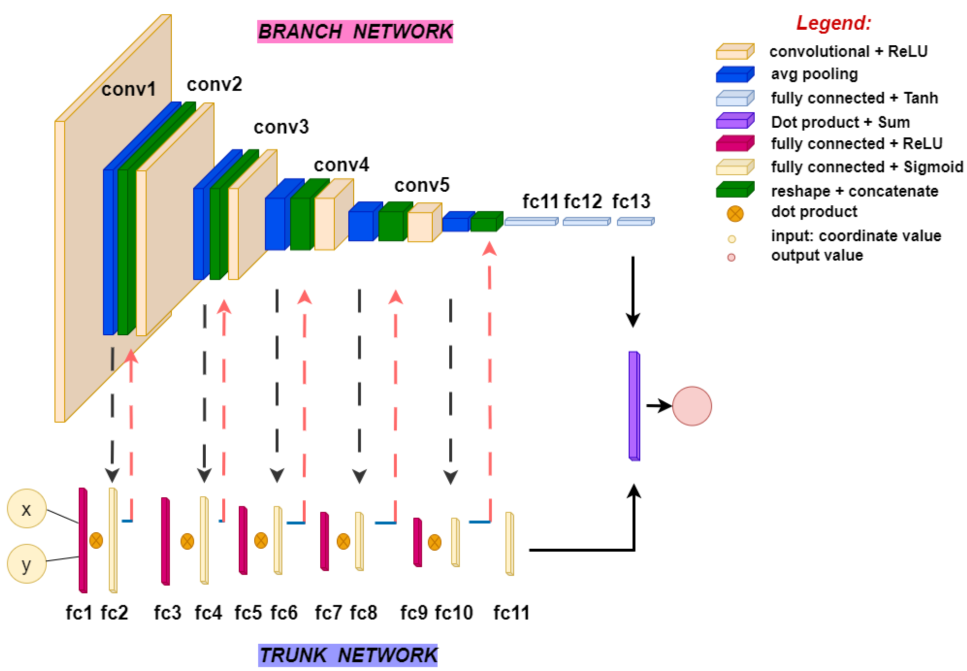

Figure 2: Schematic of the novel DeepONet architecture proposed for experiment $E2$.

Figure 3 demonstrates the results obtained using the proposed DeepONet architecture, where the model accurately captures sharp gradients the well locations, that are typically situated at distinct locations for each training and testing sample. The training process for this experiment took approximately 324 seconds, and the error metrics on the training and testing datasets were 6.6×10^{-5} and 2.6×10^{-4}, respectively. Extended analyses included scenarios with two randomly located pumping wells, as described in [13], where the proposed architecture continued to demonstrate superior performance.

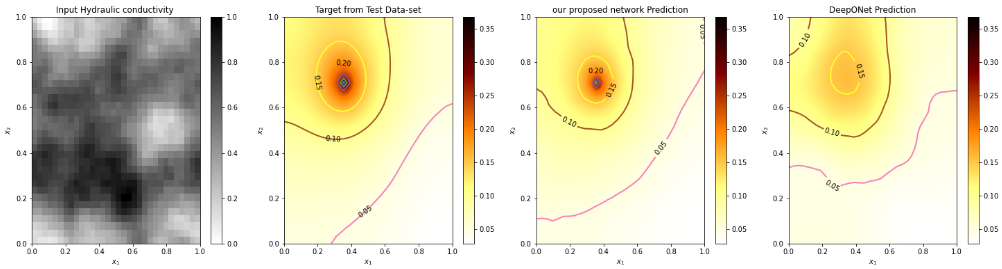

Figure 3: Hydraulic head predictions with varying well locations using vanilla and modified DeepONet architectures in Experiment $E2$.

Finally, Experiment $E3$ delves into solving inverse problems, where DeepONet estimates the conductivity field based on known hydraulic head information. The operator network demonstrates trustworthy predictions, with a MSE test error 1.46×10^{-2} and a standard deviation 5.3×10^{-3}, obtained by conducting five experiments. Incorporating 20% of the hydraulic conductivity data as supplementary input led to a 29.5% reduction in the MSE error for test samples. This improvement was even more pronounced, reaching 37.6%, with 40% of hydraulic conductivity data as additional input. We also observed that providing the network with observations of hydraulic head at different time intervals further refined the predictions.

Figure 4 displays a test case that highlights the model's proficiency in predicting the conductivity field from hydraulic head data. Specifically, the first column shows the unknown hydraulic conductivity field and the location of sparse observations used as additional input. The second column presents the input hydraulic head. The third column depicts the outcomes of the simulated inverse procedure, while the final column illustrates the hydraulic head corresponding to the predicted conductivity, as calculated using a conventional solver.

It is important to consider the ill-posed nature of this problem, particularly when based on sparse hydraulic head observations. Multiple conductivity maps could lead to an identical set of sparse Hydraulic head observations, a likelihood that increases with greater sparsity. Therefore, the results should be interpreted within this context.

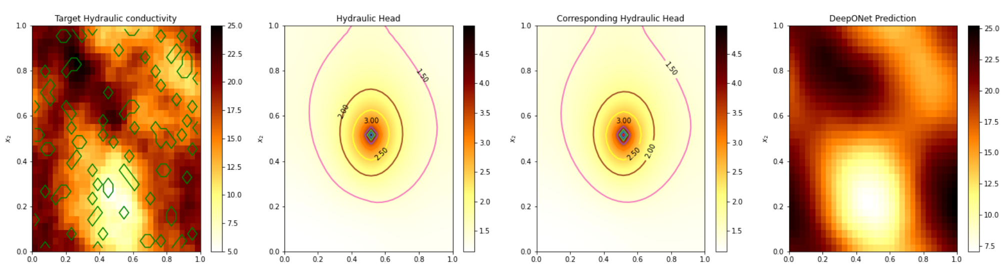

Figure 4: Inverse analysis in Experiment $E3$: Estimating conductivity fields from hydraulic head observations.

4 Summary and Discussion

The DeepONet framework has shown remarkable capabilities in simulating groundwater flow. Our experiments covered scenarios with fixed and variable pumping well locations. The integration of the branch network with the trunk network was crucial in predicting solutions for variable well locations. This novel approach allowed for greater flexibility and accuracy, overcoming limitations in the model's original design.

While the current study primarily focused on single and two-well configurations, future work aims to expand this to multiple wells and to scenarios with varying abstraction rates, aquifer geometries, and interactions with surface water systems. The integration of multi-fidelity and multi-modality data sources also presents an exciting frontier, considering the varied and sparse nature of real-world data.

Acknowledgments

This work was supported by the Leeds-York-Hull Natural Environment Research Council (NERC) Doctoral Training Partnership (DTP) Panorama under grant NE/S007458/1. We would like to acknowledge the support provided by Deltares and we express our sincere gratitude to Bennie Minnema for his invaluable contribution to designing the non-linear test case. Furthermore, we express our gratitude to Professor George Karniadakis for generously hosting the first author at Brown University within the CRUNCH group. His mentorship and insightful inputs have significantly contributed to the research development and understanding in this study.

References

[1] Tianping Chen and Hong Chen. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE transactions on neural networks*, 6 4:911–7, 1995.

[2] Rolf Farrell, Marcel Ververs, Paul Davison, Paul Howlett, and Mark Whiteman. Splicing recharge and groundwater flow models in the environment agency national groundwater modelling system. *Geological Society, London, Special Publications*, 408(1):55–69, 2017.

[3] Joseph D Hughes, Martijn J Russcher, Christian D Langevin, Eric D Morway, and Richard R McDonald. The modflow application programming interface for simulation control and software interoperability. *Environmental Modelling & Software*, 148:105257, 2022.

[4] George Em Karniadakis, Ioannis G Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.

[5] Dmitrii Kochkov, Jamie A Smith, Ayya Alieva, Qing Wang, Michael P Brenner, and Stephan Hoyer. Machine learning–accelerated computational fluid dynamics. *Proceedings of the National Academy of Sciences*, 118(21):e2101784118, 2021.

[6] C. P. Kumar. Numerical modelling of ground water flow using modflow. *Indian Journal of Science*, 2:86–92, 02 2013.

[7] Samuel Lanthaler, Siddhartha Mishra, and George E Karniadakis. Error estimates for deeponets: A deep learning framework in infinite dimensions. *Transactions of Mathematics and Its Applications*, 6(1):tnac001, 2022.

[8] L. Lu, P. Jin, G. Pang, Z. Zhang, and G. E. Karniadakis. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nature Machine Intelligence*, 3(3), March 2021. doi: https://doi.org/10.1038/s42256-021-00302-5.

[9] Marjolein Mens, Bennie Minnema, Koen Overmars, and Bart Van den Hurk. Dilemmas in developing models for long-term drought risk management: The case of the national water model of the netherlands. *Environmental Modelling & Software*, 143:105100, 2021.

[10] Shaoxing Mo, Yinhao Zhu, Nicholas Zabaras, Xiaoqing Shi, and Jichun Wu. Deep convolutional encoder-decoder networks for uncertainty quantification of dynamic multiphase flow in heterogeneous media. *Water Resources Research*, 55(1):703–728, 2019.

[11] Esteban Samaniego, Cosmin Anitescu, Somdatta Goswami, Vien Minh Nguyen-Thanh, Hongwei Guo, Khader Hamdia, X Zhuang, and T Rabczuk. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, 362:112790, 2020.

[12] Maria Luisa Taccari, Jonathan Nuttall, Xiaohui Chen, He Wang, Bennie Minnema, and Peter K Jimack. Attention u-net as a surrogate model for groundwater prediction. *Advances in Water Resources*, 163:104169, 2022.

[13] Maria Luisa Taccari, He Wang, Somdatta Goswami, Mario De Florio, Jonathan Nuttall, Xiaohui Chen, and Peter K. Jimack. Developing a cost-effective emulator for groundwater flow modeling using deep neural operators. *Journal of Hydrology*, page 130551, 2023.

[14] Yinhao Zhu and Nicholas Zabaras. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. *Journal of Computational Physics*, 366:415–447, 2018.