Please fill in the name of the event you are preparing this manuscript for.	2022 International Petroleum Technology Conference
Please fill in your 5-digit IPTC manuscript number.	22IPTC-22163-MS
Please fill in your manuscript title.	Extended Physics-informed Neural Networks For Solving Fluid Flow Problems In Highly Heterogeneous Media

Please fill in your author name(s) and company affiliation.

Given Name	Surname	Company
Ali	Alhubail	King Abdullah University of Science and Technology
Xupeng	He	King Abdullah University of Science and Technology
Marwa	AlSinan	Saudi Aramco
Hyung	Kwak	Saudi Aramco
Hussein	Hoteit	King Abdullah University of Science and Technology

This template is provided to give authors a basic shell for preparing your manuscript for submittal to an IPTC meeting or event. Styles have been included (Head1, Head2, Para, FigCaption, etc) to give you an idea of how your finalized paper will look before it is published by IPTC. All manuscripts submitted to IPTC will be extracted from this template and tagged into an XML format; IPTC's standardized styles and fonts will be used when laying out the final manuscript. Links will be added to your manuscript for references, tables, and equations. Figures and tables should be placed directly after the first paragraph they are mentioned in. The technical content of your paper WILL NOT be changed. Please start your manuscript below.

Abstract

Utilization of neural networks to solve physical problems has been receiving wide attention recently. These neural networks are commonly named physics-informed neural network (PINN), in which the physics are employed through the governing partial differential equations (PDEs). Traditional PINNs suffer from unstable performance when dealing with flow problems in highly heterogeneous domains. This work presents the applicability of the extended PINN (XPINN) method in solving heterogeneous problems. XPINN can create a full solution model to the solution of the governing PDEs by training the neural network on the PDEs and its constraints such as boundary and initial conditions, and known solution points. The heterogeneous problem is solved by performing domain decomposition, which divides the original heterogeneous domain into various homogeneous sub-domains. Each sub-domain incorporates its own PINN structure. The different PINNs are connected through interface conditions, allowing for information to communicate across the interfaces. These conditions include pressure and flux continuities. Various heterogeneous scenarios are implemented in this study to investigate the robustness of the proposed method. We demonstrate the accuracy of the XPINN model by comparing it with the ground truth solved from high-fidelity simulations. Results show a good match in terms of pressure and velocity with errors of less than 1%. Different interface conditions were tested, and it was observed that without the inclusion of pressure and flux continuities, the solver does not converge to the solution of interest. Sensitivity analysis was performed to explore the effects of the neural network architecture, the weights given to each loss term, and the number of training iterations. Results show that wide and shallow networks performed well due to avoiding the gradient vanishing issue that comes with deeper networks. In addition, balanced weights produced better accuracy in general. Moreover, more training iterations improved the accuracy of the results but at lower rates in later training stages. This paper presents XPINN to solve fluid flow in heterogeneous media. We demonstrate the robustness and accuracy of the proposed XPINN model by comparing it with the ground truth solutions in multiple heterogeneous cases. The model shows good potential and can be readily implemented in reservoir characterization workflow.

Introduction

Understanding and modeling fluid flow in subsurface porous media has been of utmost importance in various applications such as CO₂ sequestration (Hoteit et al., 2019; Omar et al., 2021), geothermal energy extraction (Santoso, Hoteit, et al., 2019; Vahrenkamp et al., 2020), and oil recovery process in hydrocarbon reservoirs (Geiger et al., 2010; Hoteit & Firoozabadi, 2008; Yang et al., 2021). The petroleum engineering field in particular aims to create models that try to encompass all the relevant physics to be able to more accurately predict production rates, make more educated decisions on where to drill new wells, and understand how to properly manage the reservoirs (Kovalchuk & Hadjistassou, 2018).

Modeling fluid flow is complex as there are many physical quantities that interplay with each other. Quantities such as rock porosity and permeability, fluid viscosity, subsurface temperature and pressure, and compressibility of fluids and rocks are all factors that are relevant to fluid flow which can influence each other. In case of multiphase flow, of consideration are also quantities such as capillary pressure, wettability, contact angle, and interfacial tension. More complexity could arise due to the heterogeneity of rock properties across the domain of interest.

The difficulty of solving such a fluid flow problem lies not only in understanding the problem and ensuring the inclusion of all physical quantities relevant to the problem, but also in the construction of the model itself. Properly constructing the model entails appropriately implementing the physics and the interaction of the different quantities across space and time, properly discretizing the partial differential equations (PDE) governing the physics, and employing a suitable grid with a suitable time stepping scheme.

Generally, four kinds of Neural Networks (NN), including Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), U-Net, and Long-short Term Memory (LSTM), are widely implemented in geoscience and oil & gas areas. Examples include fracture permeability estimation and oil recovery factor forecasting using ANN (He, Zhu, et al., 2021b; He, Qiao, et al., 2021), constructing equivalent permeability model from detailed discrete-fracture characterization using CNN (He et al., 2020; He, Santoso, et al., 2021), fracture recognition from actual outcrop or synthetic resources using U-Net (Santoso, He, et al., 2019), history matching in reservoir simulations and CO₂ leakage forecasting using LSTM (He, Zhu, et al., 2021a; Santoso et al., 2021). Their implementations, however, depend on only the supplied data samples without obeying the physics of the problem and some related constraints. Recently, studies have shown the ability of neural networks to create surrogate models that reflect the solution to physical problems with minimal effort being put into the construction of the model. These types of neural networks are called Physics-Informed Neural Networks (PINN), where the majority of the effort is put into the understanding of the physics to be implemented and optimizing the NN parameters. The mesh-free surrogate model created by PINN approximates the solution to the PDE and the relevant derivatives with no requirements on local time-stepping, gridding, linearization (of nonlinear problems) or committing to assumptions to simplify the PDE (Raissi et al., 2017).

Unlike purely data-drive neural networks which depend on only the supplied datapoints, PINNs utilize the PDE itself as a source of data. The PINN is trained to obey the PDE governing the physics of the problem and to obey all associated constraints. This is done by reflecting the PDE and constraints in the loss function that controls the training of the neural network. In the case of a steady-state fluid flow problem as will be explored in this work, provided to the PINN would be the space domain, the PDE of interest that govern the fluid flow, and the boundary conditions that close the PDE (Raissi et al., 2017; Karniadakis et al., 2021).

Based on the work of Jagtap et al. 2020, Extended PINN (XPINN) uses domain decomposition to divide the original domain into multiple subdomains. This can be used to increase parallelization capacity (i.e. reduce training time) and to increase accuracy as each subdomain can have its PINN structure with its associated hyperparameters (network depth and width, activation function, optimization method, etc.) In this work, however, we expand on the use of XPINN to tackle PDEs with sharp discontinuities in the PDE parameters. We will use XPINN to solve simple fluid flow scenarios in heterogeneous porous media where hydraulic conductivity has sharp discontinuities. Domain decomposition is performed where the domain is divided into parts that have homogeneous properties. We ultimately aim to find the pressure and velocity fields across the entire domain of interest.

Governing Equations

To illustrate the use of XPINN to solve a steady state fluid flow in heterogeneous media problem, we will consider linear flow in 2D space. The equations that govern this physical process are Darcy's law and mass conservation law. To more clearly illustrate how the XPINN method can solve the problem, we will explore a simplified flow problem. We assume isothermal conditions for a steady state laminar flow of a single phase and incompressible fluid with constant viscosity in an isotropic and homogeneous medium while neglecting the effect of gravity. The formulation of Darcy's law after applying these assumptions is

$$\mathbf{u} = -K\nabla p \quad (1)$$

where $\mathbf{u}$ is the Darcy velocity, p is fluid pressure, and K is hydraulic conductivity with $K = k / \mu$ where k is rock permeability and μ is fluid viscosity. Note that permeability is a constant as we will assume it is a diagonal tensor in an isotropic medium. Under the same assumptions, the formulation for the mass conservation law (the continuity equation) is

$$\nabla \cdot \mathbf{u} = q \quad (2)$$

where q is the source term which we set to be zero. Merging Darcy's law and the continuity equation, we obtain what is commonly called the pressure equation,

$$\nabla \cdot (-K\nabla p) = 0 \quad (3)$$

In a homogeneous medium, K is constant, and the formulation becomes

$$K\Delta p = 0 \quad (4)$$

where Δ is the Laplacian operator. The boundary conditions (BC) that will be used to close the above equations are Dirichlet and Neumann boundary conditions. We will use a rectangular 2D domain, thus requiring boundary condition for each of the four sides. In Dirichlet BC, the pressure is known while in Neumann BC, the velocity is known.

Methodology

This work involves the utilization of PINN(s) to infer the solution of a closed PDE system. It is thus required that all governing PDEs and constraints pertinent to the application are supplied and included in the PINNs. In this work, we solve PDEs using PINNs that utilize a feedforward fully connected neural network with tanh activation functions. In the applications of interest to this work, the inputs to these networks are the spatial and temporal coordinates and the outputs are the latent solution(s) to the PDE(s).

For a general PDE of the form $f_t + N(\lambda; f) = 0$, we want to find an approximation to the solution $f(\mathbf{x}, t)$. Using PINN we will create a surrogate to the PDE solution which will be given the notation $\hat{f}(\mathbf{x}, t)$. Note that the subscript notation for f denotes finding its derivative with respect to the subscript, e.g. $f_t = \partial f / \partial t$. N is a differential operator that includes the spatial terms in the PDE which are parametrized by λ . We define the residual r which is the value resulting from using the surrogate solution in the evaluation of the PDE, $r = \hat{f}_t + N(\lambda; \hat{f})$. The values of the derivatives are calculated utilizing automatic differentiation, which is based on applying the chain rule on elementary arithmetic operations in reverse mode thereby connecting the output of the neural network to its inputs (Güneş Baydin et al., 2018).

For a PINN to be a proper surrogate model, it must employ all the governing PDEs and imposed constraints in its loss function L . A typical loss function for a PINN can be in the form shown in Eq. (5) where four terms are included.

$$L = W_{IC}L_{IC} + W_{BC}L_{BC} + W_pL_p + W_{PDE}L_{PDE} \quad (5)$$

The first two terms employ the constraints which are imposed on the PDE, namely the initial conditions (L_{IC}) and the boundary conditions (L_{BC}). The third term involves known data points (i.e. known solution points) which are accurate values of the PDE solution (L_p), which in many cases are not available. The fourth term is what employs the PDE in the PINN (L_{PDE}). Each of the four loss terms has an associated weighing term W that controls the significance of the term when training the network. Properly choosing the weighing terms can be imperative in obtaining an accurate surrogate model. In this work, we utilize the mean-squared error function (MSE) for all loss functions. The total loss equation can therefore be written as Eq. (6).

$$L = W_{IC}MSE_{IC} + W_{BC}MSE_{BC} + W_pMSE_p + W_{PDE}MSE_{PDE} \quad (6)$$

The MSE terms can be expressed as

$$MSE_{IC} = \frac{1}{N_{IC}} \sum_{i=1}^{N_{IC}} |\hat{f}(\mathbf{x}_i, t_i) - f_i|^2 \quad (7)$$

$$MSE_{BC} = \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} |\hat{f}(\mathbf{x}_i, t_i) - f_i|^2 \quad (8)$$

$$MSE_p = \frac{1}{N_p} \sum_{i=1}^{N_p} |\hat{f}(\mathbf{x}_i, t_i) - f_i|^2 \quad (9)$$

$$MSE_{PDE} = \frac{1}{N_{PDE}} \sum_{i=1}^{N_{PDE}} |r(\mathbf{x}_i, t_i)|^2 \quad (10)$$

Where N is the number of points used in the calculation of each loss term, and f_i is the known solution value coming from initial conditions, boundary conditions, or real measurements. Note that in Eq. (8) the BC loss function is for Dirichlet type BC, and a different form is to be employed for Neumann BCs which will be explored in later sections. The PDE loss term includes the residual of the PDE which is calculated by evaluating the PINN at the specified N_{PDE} points. The lower the value of the loss functions, the more the surrogate model obeys the constraints and the PDE. By minimizing the four terms (and achieving convergence), the trained PINN is expected to be a good surrogate model to the exact PDE solution.

In the case where there are sharp discontinuities in the parameters of a PDE, the PINN was found to be ineffective in finding an accurate solution to the PDE. Using XPINN, it is possible to solve the closed equation system with heterogeneous parameters after decomposing the domain into multiple non-overlapping subdomains and giving each subdomain its own PINN structure. The decomposition is performed in a way where each subdomain has its own PDE parameter such that it is homogeneous for that entire subdomain. For the XPINN to work, ‘stitching’ the subdomains together at the interfaces is required to make sure that the information propagates between the subdomains. The stitching is performed by the inclusion of additional loss terms in the total loss function. These terms are the interface function continuity term (denoted MSE_{IF}) and the interface residual continuity term (denoted MSE_{IR}). With these additional terms, the total loss function for each subdomain s can be expressed in Eq. (11).

$$\mathbf{L}^{(s)} = \mathbf{W}_{\text{IC}} MSE_{\text{IC}}^{(s)} + \mathbf{W}_{\text{BC}} MSE_{\text{BC}}^{(s)} + \mathbf{W}_{\text{p}} MSE_{\text{p}}^{(s)} + \mathbf{W}_{\text{PDE}} MSE_{\text{PDE}}^{(s)} + \mathbf{W}_{\text{IF}} MSE_{\text{IF}}^{(s)} + \mathbf{W}_{\text{IR}} MSE_{\text{IR}}^{(s)} \quad (11)$$

The interface function continuity loss term works to minimize the difference between the PDE function (i.e. PDE solution) based on the PINN of subdomain s , denoted $\hat{f}^{(s)}(\mathbf{x}, t)$, and the average of the PDE function based on this PINN and the PINNs of the neighboring subdomains s^+ , which are denoted $\hat{f}^{(s^+)}(\mathbf{x}, t)$. This term is expressed in Eq. (12). Note that N_{IF} is the number of common points between each pair of neighboring subdomains s and s^+ .

$$MSE_{\text{IF}} = \sum_{\forall s^+} \left[\frac{1}{N_{\text{IF}}} \sum_{i=1}^{N_{\text{IF}}} \left| \hat{f}^{(s)}(\mathbf{x}_i, t_i) - \frac{\hat{f}^{(s)}(\mathbf{x}_i, t_i) + \hat{f}^{(s^+)}(\mathbf{x}_i, t_i)}{2} \right|^2 \right] \quad (12)$$

The interface residual continuity term, on the other hand, is used to minimize the difference of the PDE residuals between neighboring subdomains s and s^+ , denoted as $r^{(s)}(\mathbf{x}, t)$ and $r^{(s^+)}(\mathbf{x}, t)$, respectively. This term is expressed in Eq. (13). Note that N_{IR} is the number of common points between each pair of neighboring subdomains s and s^+ . The residuals of interest can include quantities other than the governing PDE equation such as the residual of fluxes.

$$MSE_{\text{IR}} = \sum_{\forall s^+} \left[\frac{1}{N_{\text{IR}}} \sum_{i=1}^{N_{\text{IR}}} \left| r^{(s)}(\mathbf{x}_i, t_i) - r^{(s^+)}(\mathbf{x}_i, t_i) \right|^2 \right] \quad (13)$$

The remaining loss terms are calculated as explained in the PINN case but for each subdomain separately. Once the losses of each subdomain are calculated, the total loss can be found using Eq. (14). This total loss is then used in the training of the XPINN.

$$L = \sum_{\forall s} L^{(s)} \quad (14)$$

Application of PINN and XPINN in the Fluid Flow Problem

To showcase the effectiveness of PINN and XPINN in solving the fluid flow in porous media problem, we first investigate the homogeneous case using one PINN. We then investigate XPINN to see if we can solve the same homogeneous problem but after applying domain decomposition and having each subdomain have its own PINN. Finally, we tackle the heterogeneous media problem using XPINN, giving each subdomain its own hydraulic conductivity.

Homogeneous Media Case Using PINN

The first case to be investigated is the homogeneous case ($K = 1$) with only one PINN to solve the fluid flow problem in a 1×1 square domain with simple boundary conditions as shown in **Figure 1**. As can be seen in the figure, the left and right boundaries have a Dirichlet BC while the top and bottom boundaries have a Neumann BC.

Figure 1. Domain and boundary conditions for the homogeneous media case. Green dots inside the domain are randomly selected points used to find the PDE loss while the black dots on the boundaries are randomly selected points used to find the BC loss.

The structure of the PINN has as inputs the coordinates x and y , and as output the pressure p . The fully connected PINN utilizes one hidden layer with 30 nodes connecting the inputs to the outputs. The training is performed using Adam's optimization (Kingma & Ba, 2015) on the total loss. The total loss here includes only the boundary condition and the PDE losses as shown in Eq. (15) as the problem is steady state (no initial conditions required) and with no known solution points. The BC loss is calculated based on random points on the boundaries while the PDE loss is calculated based on random points within the domain.

$$L = W_{BC} MSE_{BC} + W_{PDE} MSE_{PDE} \quad (15)$$

Since there are four boundaries, each will be given its individual loss term,

$$MSE_{BC} = MSE_{BC}^{(T)} + MSE_{BC}^{(B)} + MSE_{BC}^{(L)} + MSE_{BC}^{(R)} \quad (16)$$

where the letters between the brackets denote the boundary of interest for each loss term (T for top, B

for bottom, L for left, and R for right). The first term reflects the top Neumann BC and is expressed as

$$MSE_{BC}^{(T)} = \frac{1}{N_{BC}^{(T)}} \sum_{i=1}^{N_{BC}^{(T)}} |\hat{p}_y(x_i, 1)|^2 \quad (17)$$

where $\hat{p}_y = \partial p / \partial y$ which comes from the y-component of the Darcy velocity, $v = -K p_y$. The value of the derivative is obtained from automatic differentiation. The third term in Eq. (16) reflects the left Dirichlet BC and can be expressed as

$$MSE_{BC}^{(L)} = \frac{1}{N_{BC}^{(L)}} \sum_{i=1}^{N_{BC}^{(L)}} |\hat{p}(0, y_i) - 1|^2 \quad (18)$$

The second and fourth terms are expressed similarly to the above terms depending on the type of the BC. The PDE loss term utilizes the pressure equation and is expressed as

$$MSE_{PDE} = \frac{1}{N_{PDE}} \sum_{i=1}^{N_{PDE}} |\hat{p}_{xx}(x_i, y_i) + \hat{p}_{yy}(x_i, y_i)|^2 \quad (19)$$

where $\hat{p}_{xx} = \partial^2 p / \partial x^2$ and $\hat{p}_{yy} = \partial^2 p / \partial y^2$. The values of the derivatives are obtained from automatic differentiation. We specify the number of points N for each boundary loss and for the PDE loss to be 100 points. The location of the points was set to change randomly for each training iteration. Note that the PDE points can fall inside the domain and on the boundaries. **Figure 2** shows the schematic of the process. Note that the blue nodes in the figure represent the hidden layer, and that there are 30 nodes but only four are shown in the diagram for simplicity.

Figure 2. Diagram of the PINN used to solve the homogeneous fluid flow problem.

The initial weights for the PINN are initialized using Xavier initialization method (Glorot & Bengio, 2010) and the training is performed using Adam's optimization scheme which is iterated for 50,000 iterations. After performing the training, we obtain the results shown in **Figure 3** for the pressure, the x-direction Darcy velocity and the y-direction Darcy velocity. The first row of plots in the figure shows the quantities for the entire 2D domain while the second row shows the quantities for a cross-section across the domain. The pressure decreases linearly from left to right as anticipated and is obeying the boundary conditions appropriately. The largest difference from the exact solution was found to be 3.0×10^{-5} . For the x-direction Darcy velocity, the value of one is obtained with errors in the order of 10^{-5} . Similarly for the y-direction Darcy velocity, the value is zero with errors in the order of 10^{-5} . It can be seen that the PINN provides a solution to the PDE with good accuracy.

Figure 3. The solution to the homogeneous media fluid flow problem governed by the pressure equation utilizing unit hydraulic conductivity using PINN. The first row of plots are the pressure, x-direction Darcy velocity, and y-direction Darcy velocity in the 2D square domain. The values of these three quantities for the red line crossing the domain are shown in the second row of plots in the figure.

Homogeneous Media Case Using XPINN

To solve the same problem above using XPINN, domain decomposition is performed where we divided the domain into five subdomains. All subdomains have a hydraulic conductivity of one, keeping the problem homogeneous. Each subdomain has its own PINN that utilizes different weights and biases from the other sub-domains. **Figure 4** shows the 2D domain after it was decomposed into five subdomains, which were set to be of the same size.

Figure 4. The different subdomains after performed domain decomposition into different subdomains. The subdomains are connected by interfaces that are accounted for in the PINN of each subdomain. Random points are selected within each subdomain and for each subdomain BC and interface to calculate the associated loss for that subdomain.

The PINN of each subdomain has as inputs the coordinates x and y , and as output the pressure p . The hidden layer for the PINN of each subdomain has 30 nodes. The training is performed using Adam's optimization on the total loss. The total loss here includes the BC and the PDE losses as well as interface

losses as shown in Eq. (20). Each subdomain is given a number (s) which is written in the superscript of each term in the loss functions for clarity.

$$L^{(s)} = W_{BC} MSE_{BC}^{(s)} + W_{PDE} MSE_{PDE}^{(s)} + W_{IF} MSE_{IF}^{(s)} + W_{IR} MSE_{IR}^{(s)} \quad (20)$$

By trial and error, we found suitable weights (Wang, et al., 2020) that allow for good convergence of the loss function. The values are seen in Eq. (21).

$$L^{(s)} = 100 MSE_{BC}^{(s)} + MSE_{PDE}^{(s)} + 100 MSE_{IF}^{(s)} + MSE_{IR}^{(s)} \quad (21)$$

The BC loss is calculated based on random points on the boundaries, the PDE loss is calculated based on random points within the domain, and the interface losses are calculated based on random points on the interface. Taking into consideration the leftmost subdomain ($s=1$), we see that this subdomain has three BCs, top, bottom and left. The loss function for the top BC is expressed as

$$MSE_{BC}^{(1,T)} = \frac{1}{N_{BC}^{(1,T)}} \sum_{i=1}^{N_{BC}^{(1,T)}} |\hat{p}_y^{(1)}(x_i, 1)|^2 \quad (22)$$

whereas for the left BC, the loss function is expressed as

$$MSE_{BC}^{(1,L)} = \frac{1}{N_{BC}^{(1,L)}} \sum_{i=1}^{N_{BC}^{(1,L)}} |\hat{p}^{(1)}(0, y_i) - 1|^2 \quad (23)$$

For the PDE loss function, the loss can be calculated using

$$MSE_{PDE}^{(1)} = \frac{1}{N_{PDE}^{(1)}} \sum_{i=1}^{N_{PDE}^{(1)}} |\hat{p}_{xx}^{(1)}(x_i, y_i) + \hat{p}_{yy}^{(1)}(x_i, y_i)|^2 \quad (24)$$

This subdomain has only one associated interface which separates subdomain 1 from subdomain 2. We need to satisfy both the function continuity and residual continuity for this interface. The interface function continuity loss function is expressed as

$$MSE_{IF}^{(1)} = \frac{1}{N_{IF}^{(1)}} \sum_{i=1}^{N_{IF}^{(1)}} \left| \hat{p}^{(1)}(0.2, y_i) - \frac{\hat{p}^{(1)}(0.2, y_i) + \hat{p}^{(2)}(0.2, y_i)}{2} \right|^2 \quad (25)$$

The interface residual continuity loss function will be comprised of three terms: the pressure equation residual term, the x-flux residual term, and the y-flux residual term,

$$MSE_{IR}^{(1)} = MSE_{IR}^{(1, PDE)} + MSE_{IR}^{(1, xFlux)} + MSE_{IR}^{(1, yFlux)} \quad (26)$$

It was found that without the inclusion of the flux terms, the XPINN does not converge to the solution of interest, hence why they were added to the IR term. The PDE (pressure equation) residual continuity term is expressed as

$$MSE_{IR}^{(1, PDE)} = \frac{1}{N_{IR}^{(1)}} \sum_{i=1}^{N_{IR}^{(1)}} \left| K^{(1)} (\hat{p}_{xx}^{(1)}(0.2, y_i) + \hat{p}_{yy}^{(1)}(0.2, y_i)) - K^{(2)} (\hat{p}_{xx}^{(2)}(0.2, y_i) + \hat{p}_{yy}^{(2)}(0.2, y_i)) \right|^2 \quad (27)$$

The x-flux and y-flux continuity terms are expressed as

$$MSE_{IR}^{(1, xFlux)} = \frac{1}{N_{IR}^{(1)}} \sum_{i=1}^{N_{IR}^{(1)}} \left| K^{(1)} \hat{p}_x^{(1)}(0.2, y_i) - K^{(2)} \hat{p}_x^{(2)}(0.2, y_i) \right|^2 \quad (28)$$

$$MSE_{IR}^{(1, yFlux)} = \frac{1}{N_{IR}^{(1)}} \sum_{i=1}^{N_{IR}^{(1)}} \left| K^{(1)} \hat{p}_y^{(1)}(0.2, y_i) - K^{(2)} \hat{p}_y^{(2)}(0.2, y_i) \right|^2 \quad (29)$$

Figure 5 shows the schematic of the process for the first subdomain. The remaining subdomains undergo the same process, albeit with different boundaries and different interfaces. A total loss can be calculated from combining the losses of the five subdomains using

$$L = \sum_{s=1}^5 L^{(s)} \quad (30)$$

Figure 5. Diagram of the PINN for subdomain 1 used to find the total loss for this subdomain. Each subdomain undergoes the same process to obtain its own loss, and then the losses of each subdomain are added into a total loss which is used for training the XPINN.

The training is performed to minimize the total loss, and a similar training process is performed as the PINN case for 50,000 iterations. The results after the training can be seen in **Figure 6**. Similarly to the PINN case, the pressure decreases linearly from left to right as anticipated and is obeying the boundary conditions appropriately. The largest difference from the exact solution was found to be 6.3×10^{-5} . For the x-direction Darcy velocity, the value of one is obtained with errors in the order of 10^{-4} . Similarly for the y-direction Darcy velocity, the value is zero with errors in the order of 10^{-4} . It can be seen that the XPINN provides a solution to the PDE with good accuracy but with lesser accuracy when compared to the PINN case.

Figure 6. The solution to the homogeneous media fluid flow problem governed by the pressure equation utilizing unit hydraulic conductivity using XPINN. The first row of plots are the pressure, x-direction Darcy velocity, and y-direction Darcy velocity in the 2D square domain. The values of these three quantities for the red line crossing the domain are shown in the second row of plots in the figure.

Heterogeneous Media Case Using XPINN

After it was seen that XPINN can handle the homogeneous case, we now tackle the heterogeneous case. Each subdomain has its own hydraulic conductivity value K and the same equations are used from the homogeneous XPINN case. The values for K for subdomains one through five are 100, 1, 100, 1, and 100, respectively. Performing the training for 50,000 iterations, we obtain the results seen in **Figure 7**. The plots in the first row are the results of p , u and v from XPINN, the second row is the result of the three quantities obtained from a numerical simulator (considered to be real values), and the third row is a comparison between the results of XPINN and the numerical simulator involving the points falling on the red dashed line.

It can be seen that the pressure profile obeys the boundary conditions and follows the result of the numerical simulator well. The largest difference in pressure was found to be 6.2×10^{-3} which occurred at an interface between two subdomains. For the x-direction Darcy velocity, the value of 2.46 is obtained with errors in the order of 10^{-4} (with the true value being $500/203 \approx 2.46305$). For the y-direction Darcy velocity, the value is zero with errors in the order of 10^{-4} . It can be seen that the XPINN provides a solution to the heterogeneous problem with a good accuracy.

Figure 7. The solution to the heterogeneous media fluid flow problem using XPINN and a numerical simulator. The hydraulic conductivity of the subdomains from left to right are 100, 1, 100, 1, and 100. The first row of plots are the pressure, x-direction Darcy velocity, and y-direction Darcy velocity in the 2D square domain from XPINN. The second row are the same quantities but from the numerical simulator. The third row compares the XPINN with the numerical simulator across the red dashed line.

An error map was created for the pressure and for the x-direction Darcy velocity which can be seen in **Figure 8**. For the pressure error map, it can be seen that the largest errors occur at the interfaces. This could be due to the choice of the interface function continuity term utilizing the average of pressure for two neighboring subdomains. For the velocity error map, the errors appear to be almost uniform across the domain, unlike the pressure error map.

Figure 8. Error map for pressure (left) and x-direction Darcy velocity (right). The maps show the absolute difference between the XPINN results and the results from the numerical simulator. For the pressure, the difference is largest at the interfaces separating the subdomains. For the velocity, it can be seen that the error is more or less uniform across the domain.

The progression of all loss values that were used in the heterogeneous flow problem can be seen in **Figure 9**. The plots in the left column are the BC and PDE losses for each subdomain separately, while on the right column, the first four plots are the IF and IR losses for each interface, and the fifth plot is the total loss based on the source of the loss. For all these losses, a general trend is observed where a sharp decrease occurs at the initial portion of the training which later changes to a more gradual decrease. It is worth noting that the weights were not balanced which can in turn affect the values of the losses.

Figure 9. The progression of all loss functions as the XPINN is trained. The left column includes the BC and PDE losses for the subdomains, while the right column includes the IF and IR losses for the interfaces, as well as the total loss categorized by source.

Conclusions

The rapid development in neural networks has allowed for their use in solving physical problems in a simplified way. It is possible to give neural networks the ability to be physics-informed by including in their loss function all the relevant PDE and their associated constraints. Even with a simple neural network structures having a single hidden layer with thirty nodes, it was possible to solve fluid flow in porous media problems and with good accuracy. The main conclusions are summarized as follows:

- For a homogeneous fluid flow in porous media problem, it was possible to use a single PINN to obtain accurate solutions to the PDE.
- Solving a PDE that has sharp discontinuities in its parameters utilizing a single PINN leads to inaccurate results. However, proper domain decomposition using XPINN to make each subdomain homogeneous allowed for these scenarios to be tackled with good accuracy.
- XPINN required stitching the subdomains by applying all appropriate interface terms. Missing some terms can make the subdomains not interact properly and thus leading to inaccurate solutions.
- In the heterogeneous media case, relatively large errors in pressure were at the interfaces as compared to the errors within each subdomain. Finding ways to reduce this error could lead to a much more improved accuracy.
- The training process can be long as it requires a large number of iterations to produce accurate results. In this work, 50,000 iterations were performed to reach a point where the loss terms are not reducing considerably with each new iteration.
- Training the PINN or XPINN, even for a large number of iterations, does not necessarily mean that they will converge to the correct solution. In this case, some of the parameters that could be modified to improve accuracy include the weights associated with the loss terms, the PINN structure and hyperparameters, interface terms in case of usage of XPINN, the number and the location of points used to calculate losses, and the training algorithm.
- PINN and XPINN schemes are in their early stages, and further development can allow them to compete with the well-established numerical methods.

Acknowledgments

We would like to thank King Abdullah University of Science and Technology (KAUST) for providing a license of MATLAB.

References

- Das, M. K., Mukherjee, P. P., & Muralidhar, K. (2018). *Mechanical Engineering Series Modeling Transport Phenomena in Porous Media with Applications*. <http://www.springer.com/series/1161>
- Glorot, X. & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research*, 9. 249-256.
<http://proceedings.mlr.press/v9/glorot10a.html>
- Güneş Baydin, A., Pearlmutter, B. A., Andreyevich Radul, A., & Mark Siskind, J. (2018). Automatic differentiation in machine learning: A survey. *Journal of Machine Learning Research*, 18, 1–43.
- Geiger, S., Cortis, A., & Birkholzer, J. T. (2010). Upscaling solute transport in naturally fractured porous media with the continuous time random walk method. *Water Resources Research*, 46(12).
- He, X., Santoso, R., & Hoteit, H. (2020). Application of machine-learning to construct equivalent continuum models from high-resolution discrete-fracture models. In *International Petroleum Technology Conference*. OnePetro.
- He, X., Zhu, W., Santoso, R., Alsinan, M., Kwak, H., & Hoteit, H. (2021a). CO2 Leakage Rate Forecasting Using Optimized Deep Learning. In *SPE Annual Technical Conference and Exhibition*. OnePetro.
- He, X., Santoso, R., Alsinan, M., Kwak, H., & Hoteit, H. (2021). Constructing Dual-Porosity Models from High-Resolution Discrete-Fracture Models Using Deep Neural Networks. In *SPE Reservoir Simulation Conference*. OnePetro.
- He, X., Zhu, W., Santoso, R., Alsinan, M., Kwak, H., & Hoteit, H. (2021b). Fracture Permeability Estimation Under Complex Physics: A Data-Driven Model Using Machine Learning. In *SPE Annual Technical Conference and Exhibition*. OnePetro.
- He, X., Qiao, T., Santoso, R., Hoteit, H., AlSinan, M. M., & Kwak, H. T. (2021, November 1). Gas Injection Optimization Under Uncertainty in Subsurface Reservoirs: An Integrated Machine Learning-Assisted Workflow. *ARMA/DGS/SEG 2nd International Geomechanics Symposium*.
- Hoteit, H., & Firoozabadi, A. (2008). Numerical modeling of two-phase flow in heterogeneous permeable media with different capillarity pressures. *Advances in Water Resources*, 31(1), 56–73.
- Hoteit, H., Fahs, M., & Soltanian, M. R. (2019). Assessment of CO2 injectivity during sequestration in depleted gas reservoirs. *Geosciences*, 9(5), 199.
- Omar, A., Addassi, M., Vahrenkamp, V., & Hoteit, H. (2021). Co-Optimization of CO2 Storage and Enhanced Gas Recovery Using Carbonated Water and Supercritical CO2. *Energies*, 14(22), 7495.
- Santoso, R., He, X., & Hoteit, H. (2019). Application of machine-learning to construct simulation models from high-resolution fractured formation. In *Abu Dhabi International Petroleum Exhibition & Conference*. OnePetro.
- Santoso, R., Hoteit, H., & Vahrenkamp, V. (2019). Optimization of energy recovery from geothermal reservoirs undergoing re-injection: conceptual application in Saudi Arabia. In *SPE middle east oil and gas show and conference*. OnePetro.
- Santoso, R., He, X., Alsinan, M., Kwak, H., & Hoteit, H. (2021). Bayesian Long-Short Term Memory for History Matching in Reservoir Simulations. In *SPE Reservoir Simulation Conference*. OnePetro.
- Vahrenkamp, V., Finkbeiner, T., Patzek, T., & Hoteit, H. (2020). Maturing Geothermal Energy for Saudi Arabia. In *82nd EAGE Annual Conference & Exhibition* (Vol. 2020, pp. 1–4). European Association of Geoscientists & Engineers.
- Yang, X., Shao, Q., Hoteit, H., Carrera, J., Younes, A., & Fahs, M. (2021). Three-dimensional natural convection, entropy generation and mixing in heterogeneous porous medium. *Advances in Water Resources*, 155, 103992.
- Jagtap, A. D., & Karniadakis, G. E. (2020). Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial

- differential equations. *Communications in Computational Physics*, 28(5), 2002–2041.
<https://doi.org/10.4208/CICP.OA-2020-0164>
- Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., & Yang, L. (2021). Physics-informed machine learning. *Nature Reviews Physics*, May. <https://doi.org/10.1038/s42254-021-00314-5>
- Kingma, D. P., & Ba, J. L. (2015). Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–15.
- Kovalchuk, N., & Hadjistassou, C. (2019). Laws and principles governing fluid flow in porous media. *European Physical Journal E*, 42(5). <https://doi.org/10.1140/epje/i2019-11819-6>
- Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2017). *Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. Part I*, 1–22.
<http://arxiv.org/abs/1711.10561>
- Wang, S., Teng, Y., & Perdikaris, P. (2020). *Understanding and mitigating gradient pathologies in physics-informed neural networks*. 1–28. <http://arxiv.org/abs/2001.04536>