mathematics

MDPI

Article

Partial Learning Using Partially Explicit Discretization for Multicontinuum/Multiscale Problems with Limited Observation: Dual Continuum Heterogeneous Poroelastic Media Simulation

Aleksei Tyrylgin [1,2], Sergei Stepanov [1], Dmitry Ammosov [1], Aleksandr Grigorev [1] and Maria Vasilyeva [3,*]

[1] Laboratory of Computational Technologies for Modeling Multiphysical and Multiscale Permafrost Processes, North-Eastern Federal University, Yakutsk 677980, Russia; aa.tyrylgin@mail.ru (A.T.); cepe2a@inbox.ru (S.S.); dmitryammosov@gmail.com (D.A.); re5itsme@gmail.com (A.G.)
[2] North-Caucasus Center for Mathematical Research, North-Caucasus Federal University, Stavropol 355017, Russia
[3] Department of Mathematics and Statistics, Texas A&M University, Corpus Christi, College Station, TX 78412, USA
* Correspondence: maria.vasilyeva@tamucc.edu

Abstract: In this paper, we consider the poroelasticity problem in heterogeneous media. The mathematical model is described by a coupled system of equations for displacement and pressure in the coupled dual continuum porous media. We propose a new method based on hybrid explicit–implicit (HEI) learning to solve the poroelasticity problem in dual continuum heterogeneous media. We use a finite element method with standard linear basis functions for spatial approximation. We apply the explicit–implicit time scheme, where the explicit scheme is used for the low-conductive continuum and the implicit scheme for the high-conductive. The fixed-strain splitting scheme is used to accelerate the computation and decouple the flow and mechanics problems. The main idea of the proposed method is partial learning of particular degrees of freedom of the high-conductive continuum's pressure (implicit part of the flow). First, we train a deep neural network (DNN) to obtain values of the implicit part of the flow at some spatial points at some time moments. Then, we apply the Discrete Empirical Interpolation Method (DEIM) combined with Proper Orthogonal Decomposition (POD) to restore the complete implicit parts and perform linear interpolation over time. Consequently, we treat the high-conductive continuum's pressure as a known function and use it to find the other continuum's pressure and displacements. Numerical results for the two-dimensional model problem are presented. The results demonstrate that the proposed method provides fast and accurate predictions.

Keywords: poroelasticity; machine learning; explicit–implicit scheme; discrete empirical interpolation method; proper orthogonal decomposition

MSC: 65M60

Citation: Tyrylgin, A.; Stepanov, S.; Ammosov, D.; Grigorev, A.; Vasilyeva, M. Partial Learning Using Partially Explicit Discretization for Multicontinuum/Multiscale Problems with Limited Observation:Dual Continuum Heterogeneous Poroelastic Media Simulation. *Mathematics* **2022**, *10*, 2629. https://doi.org/10.3390/math10152629

Academic Editor: Fernando Simoes

Received: 13 July 2022
Accepted: 22 July 2022
Published: 27 July 2022

1. Introduction

Accurately describing the stress–strain state of poroelastic bodies is an urgent and essential problem for deformable solid mechanics [1–3]. The mathematical model consists of equations for pressure and displacements. The most important feature of the model is that the equations are coupled. Modern computational technology allows us to solve such problems using various numerical methods. Among them, it is necessary to highlight the finite element and finite difference methods. The finite element method is best suited for spatial approximation of a problem given in a complex domain. At the same time, the finite difference method is commonly used for time approximation.

The most important classification of the finite difference schemes for time approximation is related to their assignment as explicit or implicit ones [4–6]. Implicit schemes are popular in numerical simulation due to their excellent stability, which does not depend on spatial mesh size, time step size, and conductivity. However, these schemes are computationally expensive. In addition, they do not always correctly describe the behavior of the solution. When using implicit schemes, in a sense, we remove high frequencies. On the other hand, explicit schemes are computationally efficient and can capture the solution's dynamics. However, they have conditional stability depending on the grid size, time step size, and conductivity. Therefore, it makes sense in multicontinuum problems to use a combination of explicit and implicit schemes, often called an explicit–implicit or partially explicit scheme. In this combination, we use an implicit scheme for a high-conductivity continuum and an explicit scheme for a low-conductivity continuum.

Another important detail related to time approximation is the use of splitting schemes. When solving systems of partial differential equations numerically, we can use either a coupled scheme or splitting schemes. In the coupled scheme, we solve all the equations at the same time. In splitting schemes, the transition to a new temporal layer is carried out by sequential solution of separate equations. The splitting schemes simplify the construction of difference schemes and reduce required computational resources. For poroelasticity problems, one can note drained, undrained, fixed-strain, fixed-stress, and weighted splitting schemes [7–10].

The above schemes help solve poroelasticity problems in the most computationally efficient way. They all refer in one way or another to temporal approximation. However, there are other ways to speed up computations. For example, one can use numerical homogenization and multiscale methods [11–14]. These methods allow us to solve problems in heterogeneous media using coarse grids, thereby significantly reducing the size of the discrete problem. It is worth noting that one can combine these methods with machine learning to accelerate some steps [14].

Note that one can encounter an issue related to a limited number of possible observation points in applied poroelasticity problems. For example, it can be some measurement devices. The problem of reconstructing the overall picture from the limited observation arises in such situations. Unfortunately, classical spatial interpolation methods cannot give an accurate approximation. For such cases, it is better to use the Discrete Empirical Interpolation Method (DEIM) [15] with Proper Orthogonal Decomposition (POD) [16,17]. POD is a global model reduction method. The method's main idea is to find the most energetic modes and use them as basis functions. These basis functions form the projection basis matrix. The DEIM provides an algorithm for finding interpolation indices (points) and the POD's solution degrees of freedom. One can restore the required function by using the function's values at these interpolation indices and the projection basis matrix.

This paper proposes a new method based on hybrid explicit–implicit (HEI) learning [18] to solve the poroelasticity problem in dual continuum heterogeneous media. In this model, we introduce a pressure field for each continuum, and the effective stress contains each continuum's part in the equation for mechanical deformations [19–22]. We use a finite element method with standard linear basis functions for spatial approximation. We apply the explicit–implicit time scheme, where the explicit scheme is used for the low-conductive continuum and the implicit scheme for the high-conductive one. Furthermore, the fixed-strain splitting scheme is used to accelerate computation. The main idea of the proposed method is partial learning of particular degrees of freedom of the high-conductive continuum's pressure (implicit part of the flow). First, we train a deep neural network (DNN) to obtain values of the implicit part of the flow at some spatial points at some time moments. Then, we apply the DEIM with POD to restore the complete implicit parts and perform linear interpolation over time. Consequently, we treat the high-conductive continuum's pressure as a known function and use it to find the other continuum's pressure and displacements. Our method consists of two stages: offline and online. First, in the offline stage, we generate the POD basis projection matrix, define interpolation indices,

and train DNN. Then, we solve the poroelasticity problem by treating the implicit part of the flow as a known function in the online stage.

Offline stage

1 Construct the POD projection basis matrix and define the interpolation points using DEIM.
2 Generate the training dataset.
- Generate the input data.
- Solve the poroelasticity problem with the partially explicit discretization at each time step.

3 Train the Deep Neural Network to obtain values of the implicit flow part at the interpolation points at some time steps.

Online stage

1 The Deep Neural Network obtains the implicit part of the pressure at the interpolation points at some time moments.
2 The POD projection basis matrix restores the complete implicit parts of the flow at some time moments.
3 Linear interpolation over time.
4 For each time step,
- Compute the explicit flow part using the learned and interpolated implicit one.
- Solve the displacement using the implicit and explicit parts of the flow.

The work has the following structure. First, we present the mathematical model in Section 2 and its approximation in Section 3. Then, Section 3 describes the Discrete Empirical Interpolation Method with Proper Orthogonal Decomposition. Section 4 presents our partial learning partially explicit discretization approach. Next, we present numerical results for two-dimensional model poroelasticity problems in the dual continuum heterogeneous medium in Section 5. Finally, Section 6 summarizes the work.

2. Problem Formulation

Let $\Omega \subset \mathbb{R}^d$ be a computational domain, where d denotes a geometrical dimension of the problem and equals 2 for a two-dimensional case. We consider a dual continuum model, where the first continuum is high-conductive, and the second continuum is low-conductive. The mathematical model is described by a coupled system of equations for pressures p_1, p_2 and displacements u [23,24]

$$\begin{aligned} &\alpha_1 \frac{\partial \operatorname{div} u}{\partial t} + \frac{1}{M_1}\frac{\partial p_1}{\partial t} - \operatorname{div}(k_1 \operatorname{grad} p_1) + r_{12}(p_1 - p_2) = 0, \quad x \in \Omega, \\ &\alpha_2 \frac{\partial \operatorname{div} u}{\partial t} + \frac{1}{M_2}\frac{\partial p_2}{\partial t} - \operatorname{div}(k_2 \operatorname{grad} p_2) - r_{21}(p_1 - p_2) = g, \quad x \in \Omega, \\ &- \operatorname{div} \sigma(u) + \alpha_1 \operatorname{grad} p_1 + \alpha_2 \operatorname{grad} p_2 = 0, \quad x \in \Omega, \end{aligned} \tag{1}$$

where $k_\alpha = \kappa_\alpha / \mu$, κ_1 and κ_2 are the permeabilities, g is the source term, α_1 and α_2 are the Biot coefficients, M_1 and M_2 are the Biot moduli, r_{mf} is the transfer term between continua, μ is the fluid viscosity, u is the displacement, and σ is the stress tensor.

The relation between the stress and strain tensors is given as

$$\sigma(u) = 2\mu\, \varepsilon(u) + \lambda \operatorname{div} u\, \mathcal{I}, \quad \varepsilon(u) = \frac{1}{2}(\operatorname{grad} u + \operatorname{grad} u^T),$$

where ε is the strain tensor, and λ and μ are the Lamé coefficients.

We consider a system of Equation (1) with the following initial conditions

$$\begin{aligned} &p_1(0) = p_2(0) = p_0, \quad x \in \Omega, \\ &u(0) = u_0, \quad x \in \Omega, \end{aligned} \tag{2}$$

and boundary conditions

$$\begin{aligned} &-k_1\frac{\partial p_1}{\partial n}=0, \quad x\in\partial\Omega, \quad -k_2\frac{\partial p_2}{\partial n}=0, \quad x\in\partial\Omega,\\ &u_1=0,\ (\sigma n)_2=0, \quad x\in\Gamma_L\cup\Gamma_R, \quad u_2=0,\ (\sigma n)_1=0, \quad x\in\Gamma_T\cup\Gamma_B \end{aligned} \tag{3}$$

where boundaries $\Gamma_L\cup\Gamma_R\cup\Gamma_T\cup\Gamma_B=\partial\Omega$; Γ_L, Γ_R, Γ_B, Γ_T denote the left, right, bottom, and top boundaries, respectively.

3. Approximation

Variational formulation. For spatial approximation of the system of Equation (1) with boundary conditions (3), we use a finite element method. We define the following functional spaces

$$W_1=W_2=H^1(\Omega),$$

$$V=\{v\in[H^1(\Omega)]^d : v_1=0 \text{ on } \Gamma_L\cup\Gamma_R \text{ and } v_2=0 \text{ on } \Gamma_T\cup\Gamma_B\}.$$

The variational formulation of the poroelasticity problem can be written as follows. Find $(p_1,p_2,u)\in W_1\times W_2\times V$ such that

$$\begin{aligned} &d_1\left(\frac{\partial u}{\partial t},w_1\right)+c_1\left(\frac{\partial p_1}{\partial t},w_1\right)+b_1(p_1,w_1)+q_{12}(p_1-p_2,w_1)=0, \quad \forall w_1\in W_1,\\ &d_2\left(\frac{\partial u}{\partial t},w_2\right)+c_2\left(\frac{\partial p_2}{\partial t},w_2\right)+b_2(p_2,w_2)-q_{21}(p_1-p_2,w_2)=l(w_2), \quad \forall w_2\in W_2,\\ &a(u,v)+g_1(p_1,v)+g_2(p_2,v)=0, \quad \forall v\in V, \end{aligned} \tag{4}$$

where the bilinear and linear forms are the following

$$b_1(p_1,w_1)=\int_\Omega k_1\,\mathrm{grad}\,p_1\cdot\mathrm{grad}\,w_1\,dx, \quad b_2(p_2,w_2)=\int_\Omega k_2\,\mathrm{grad}\,p_2\cdot\mathrm{grad}\,w_2\,dx$$

$$c_1(p_1,w_1)=\int_\Omega\frac{1}{M_1}p_1w_1\,dx, \quad c_2(p_2,w_2)=\int_\Omega\frac{1}{M_2}p_2w_2\,dx, \quad l(w_2)=\int_\Omega gw_2dx,$$

$$q_{12}(p_1-p_2,w_1)=\int_\Omega r_{12}(p_1-\Pi_{21}p_2)w_1\,dx, \quad q_{21}(p_1-p_2,w_2)=\int_\Omega r_{21}(\Pi_{12}p_1-p_2)w_2\,dx,$$

$$d_1(u,w_1)=\int_\Omega\alpha_1\,\mathrm{div}\,u\,w_1\,dx, \quad d_2(u,w_2)=\int_\Omega\alpha_2\,\mathrm{div}\,u\,w_2\,ds, \quad a(u,v)=\int_\Omega\boldsymbol{\sigma}(u):\boldsymbol{\varepsilon}(v)\,dx,$$

$$g_1(p_1,v)=\int_\Omega\alpha_1\,\mathrm{grad}\,p_1\cdot v\,dx, \quad g_2(p_2,v)=\int_\Omega\alpha_2\,\mathrm{grad}\,p_2\cdot v\,dx,$$

where Π_{12} is the projection operator from W_1 to W_2 defined as

$$\int_\gamma(\Pi_{12}p_1-p_1)w_2\,ds=0, \quad \forall w_2\in W_2.$$

and $\Pi_{21}=\Pi_{12}^*$ (see [25] for details).

Discretization in time. We use an explicit–implicit scheme for the time approximation, where the explicit scheme is used for the low-conductive continuum and the implicit scheme for the high-conductive one. Let $u^n=u(x,t^n)$, $p_1^n=p_1(x,t^n)$, $p_2^n=p_2(x,t^n)$, and $t^n=n\tau$, where $\tau=T/N_t$ is the time step size, T is the final time, and N_t is the count of time steps. Then, the problem can be formulated as follows. For $n=0,\dots,N_t-1$, find $(p_1^{n+1},p_2^{n+1},u^{n+1})\in(W_1\times W_2\times V)$ such that

$$
\begin{aligned}
&d_1\left(\frac{u^{n+1}-u^n}{\tau}, w_1\right) + c_1\left(\frac{p_1^{n+1}-p_1^n}{\tau}, w_1\right) + b_1(p_1^{n+1}, w_1) + q_{12}(p_1^{n+1}-p_2^{n+1}, w_1) = 0,\\
&d_2\left(\frac{u^{n+1}-u^n}{\tau}, w_2\right) + c_2\left(\frac{p_2^{n+1}-p_2^n}{\tau}, w_2\right) + b_2(p_2^n, w_2) - q_{21}(p_1^{n+1}-p_2^{n+1}, w_2) = l(w_2),\\
&a(u^{n+1}, v) + g_1(p_1^{n+1}, v) + g_2(p_2^{n+1}, v) = 0, \quad \forall vs. \in V,
\end{aligned}
\tag{5}
$$

where $\forall(w_1, w_2, v) \in (W_1, W_2, V)$.

Splitting scheme. We can solve the poroelasticity problem using the fixed-strain splitting scheme. The fixed-strain splitting scheme decouples the flow and mechanics system solving them sequentially at each time step. Then, we have the following problem formulation. For $n = 1, \ldots N_t - 1$, find

- Pressures $(p_1^{n+1}, p_2^{n+1}) \in (W_1 \times W_2)$ such that

$$
\begin{aligned}
&d_1\left(\frac{u^{n}-u^{n-1}}{\tau}, w_1\right) + c_1\left(\frac{p_1^{n+1}-p_1^n}{\tau}, w_1\right) + b_1(p_1^{n+1}, w_1) + q_{12}(p_1^{n+1}-p_2^{n+1}, w_1) = 0,\\
&d_2\left(\frac{u^{n}-u^{n-1}}{\tau}, w_2\right) + c_2\left(\frac{p_2^{n+1}-p_2^n}{\tau}, w_2\right) + b_2(p_2^n, w_2) - q_{21}(p_1^{n+1}-p_2^{n+1}, w_2) = l(w_2),
\end{aligned}
\tag{6}
$$

  where $\forall(w_1, w_2) \in (W_1, W_2)$.
- Displacements u^{n+1} such that

$$
a(u^{n+1}, v) + g_1(p_1^{n+1}, v) + g_2(p_2^{n+1}, v) = 0, \tag{7}
$$

  where $\forall vs. \in V$.

Discrete formulation. Let $\mathcal{T}^h$ be a grid partition of the computational domain Ω into finite elements and $W_{1,h} \subset W_1$, $W_{2,h} \subset W_2$ and $V_h \subset V$ be the finite element subspaces on $\mathcal{T}^h$.

$$
u = \sum u_i \Phi_i, \quad p_1 = \sum p_{1,i}\phi_{1,i} \quad p_2 = \sum p_{2,i}\phi_{2,i} \tag{8}
$$

where Φ, ϕ_1, and ϕ_2 are the linear basis functions defined on $\mathcal{T}^h$.

The discrete formulation of the poroelasticity problem using the explicit–implicit time scheme will have the following form for coupled and splitting schemes.

Coupled scheme. For $n = 0, \ldots, N_t - 1$, find $(p_1^{n+1}, p_2^{n+1}, u^{n+1})$ such that

$$
\begin{aligned}
&B_1 p_1^{n+1} + C_1\frac{p_1^{n+1}-p_1^n}{\tau} + D_1\frac{u^{n+1}-u^n}{\tau} + Q_{12}(p_1^{n+1}-p_2^{n+1}) = 0,\\
&B_2 p_2^{n} + C_2\frac{p_2^{n+1}-p_2^n}{\tau} + D_2\frac{u^{n+1}-u^n}{\tau} - Q_{21}(p_1^{n+1}-p_2^{n+1}) = L,\\
&Au^{n+1} + G_1 p_1^{n+1} + G_2 p_2^{n+1} = 0.
\end{aligned}
\tag{9}
$$

where

$$
\begin{aligned}
B_1 = [b_1, ij], \quad b_{1,ij} = b_1(\phi_{1,j}, \phi_{1,i}), \quad B_2 = [b_2, ij], \quad b_{2,ij} = b_1(\phi_{2,j}, \phi_{2,i}),\\
C_1 = [c_1, ij], \quad c_{1,ij} = c_1(\phi_{1,j}, \phi_{1,i}), \quad C_2 = [c_2, ij], \quad c_{2,ij} = c_2(\phi_{2,j}, \phi_{2,i}),\\
D_1 = [d_1, ij], \quad d_{1,ij} = d_1(\Phi_j, \phi_{1,i}), \quad D_2 = [d_2, ij], \quad d_{2,ij} = d_2(\Phi_j, \phi_{2,i}),\\
Q_{12} = [q_{12}, ij], \quad q_{12,ij} = q_{12}(\phi_{1,j}, \phi_{1,i}), \quad Q_{21} = [q_{21}, ij], \quad q_{21,ij} = q_{21}(\phi_{2,j}, \phi_{2,i}),\\
G_1 = [g_1, ij], \quad g_{1,ij} = g_1(\phi_j, \Phi_{1,i}), \quad G_2 = [g_2, ij], \quad g_{2,ij} = g_2(\phi_j, \Phi_{2,i}),
\end{aligned}
$$

$$L = [l, i], \quad l_i = l_(\phi_{2,i}), \quad A_u = [a_u, ij], \quad a_{u,ij} = a_u(\Phi_j, \Phi_i).$$

For $n = 1, \ldots, N_t - 1$, find

- Pressures (p_1^{n+1}, p_2^{n+1}) such that

$$\begin{aligned} B_1 p_1^{n+1} + C_1 \frac{p_1^{n+1} - p_1^n}{\tau} + D_1 \frac{u^n - u^{n-1}}{\tau} + Q_{12}(p_1^{n+1} - p_2^{n+1}) = 0, \\ B_2 p_2^n + C_2 \frac{p_2^{n+1} - p_2^n}{\tau} + D_2 \frac{u^n - u^{n-1}}{\tau} - Q_{21}(p_1^{n+1} - p_2^{n+1}) = L. \end{aligned} \tag{10}$$

- Displacements u^{n+1} such that

$$A u^{n+1} + G_1 p_1^{n+1} + G_2 p_2^{n+1} = 0. \tag{11}$$

In the next section, we will consider the Discrete Empirical Interpolation Method we will use for our partial learning approach.

4. Discrete Empirical Interpolation Method with Proper Orthogonal Decomposition

When we need to reconstruct the solution in a heterogeneous medium as accurately as possible using only several points, we cannot apply the classical spatial interpolation methods. The Discrete Empirical Interpolation Method (DEIM) is better suited for such situations. The procedure of such spatial interpolation consists of the following steps:

1. Computing the Proper Orthogonal Decomposition (POD) basis functions;
2. Determining the interpolation nodes using the DEIM algorithm.

In our work, we want to use the DEIM to spatially interpolate the pressure of the high-conductive continuum. In this case, after we train the neural network to acquire values at several points, we can reconstruct the pressure in the whole area.

Computing the Proper Orthogonal Decomposition (POD) basis functions. To compute snapshot functions, we save the first continuum's pressure data into matrix B with every second time step in each row saved, solving the problem five times with different parametric functions. Thus, we obtain a matrix B with N_s by M elements:

$$B = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N_s} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N_s} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M,1} & a_{M,2} & \cdots & a_{M,N_s} \end{pmatrix},$$

where N_s is the number of snapshots, and M is the number of degrees of freedom (mesh nodes in our case).

Next, we assemble a symmetric positive covariance matrix of dimension $M \times M$.

$$R = BB^T$$

Then, we determine eigenvalues λ_k and eigenvectors ψ_k in matrix R. We sort the eigenvalues in descending order $\lambda_1 > \lambda_2 > \cdots > \lambda_m > \ldots$.

Finally, we choose m eigenvectors ψ_k corresponding to the first m eigenvalues λ_k ($m \ll M$). These eigenvectors will be our basis functions.

Determining the interpolation nodes using the DEIM algorithm. After calculating the basis functions, we can represent the desired function (in our case, p_1) in the following form.

$$p_1 \approx \Psi c, \tag{12}$$

where $p_1 = (p_{1,1}, \ldots, p_{1,M})^T$, $\Psi = (\psi_1, \ldots, \psi_m)$, $c = (c_1, \ldots, c_m)^T$.

To determine p_1, we need to know c. Note that we can find c using m rows. Suppose we can know the values of p_1 in any m nodes (rows). In this case, our task is to determine m optimal nodes (rows). We achieve this using the DEIM algorithm [15].

The DEIM finds m distinct interpolation nodes $(v_1, \ldots, v_m)$ and assembles the DEIM interpolation nodes matrix $P = (e_{n_1}, \ldots, e_{n_m})$, where e_{n_i} is the n_i-th column of the identity matrix I_M. After that, we can restore p_1 as follows.

$$p_1 \approx \Psi(P^T\Psi)^{-1}P^T p_1, \tag{13}$$

where $P^T p_1$ samples p_1 at m components only.

Figure 1 presents an example of selected interpolation nodes. We will use this interpolation method for our machine learning approach presented in the next section.

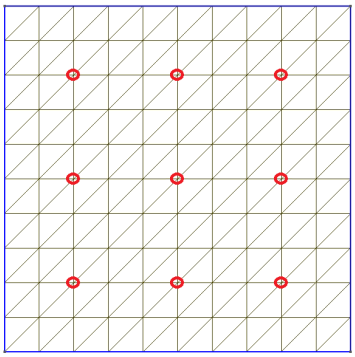

Figure 1. Selected vertices from the computational mesh.

5. Machine Learning Approach

Training neural networks to solve the poroelasticity problem is complex and time-consuming. As a result, it makes more sense to make training the computationally expensive part of the solution—the implicit flow part. Due to the inefficiency of training all degrees of freedom, we propose training only a portion of the solution. We train the neural network to obtain the high-conductive continuum's pressure at multiple mesh nodes at some time steps. Then, we apply the DEIM spatial interpolation and linear time interpolation. As a result, we treat the high-conductive continuum's pressure as a known function that we may use to find the other pressure.

A basic DNN is made up of layers, each of which uses a differentiable function to translate one volume of activations to another. The layers of a neuron network are nonlinear transformations. The previous layer's output result is transmitted to the following layer. The neurons of the previous layer are linked to the neurons of the current layer, and the connection data are a weighting parameter. As a result, the neural network training aims to perfect the weights. Deep neural networks use a nonlinear operation to minimize the amount of input data and simplify the task. We use TensorFlow's most straightforward DNN variation in the sequential model [26]. The generation of the dataset used to train is essential in machine learning. The final need is to create an appropriate neural network architecture.

A neural network function $\mathcal{N}$ of L—layers with samples x—input data and y—output data may be represented in the following form in deep learning.

$$\mathcal{N}^L(x, \theta) = \sigma^L(W^L\sigma^{L-1}(\cdots\sigma^2(W^2\sigma^1(W^1x + b^1) + b^2)\cdots) + b^L),$$

where Ws are weight matrices, bs are bias vector, and σs are the activation functions, where $\sigma - >$ RELU $= \max(0, x)$.

Let us write:

- The first output layer: $\mathcal{N}^1(x^0) = Y^1$, i.e., $Y^1 = \sigma^1(W^1x + b^1)$;
- For i's layer: $Y^i = \sigma^i(W^i x^{i-1} + b^i)$, where i = 1,2,..., L.

For estimating the output y, a neural network called $\mathcal{N}^L$ is employed. The goal of the neural network is to solve an optimization problem to identify parameters θ^*.

$$\theta^* = \text{argmin}_\theta \frac{1}{N} \sum_{j=1}^{N} ||y_j - \mathcal{N}(x_j;\theta)||_2^2,$$

where N is the number of the samples. Here, $\frac{1}{N}\sum_{j=1}^{N} ||y_j - \mathcal{N}(x_j;\theta)||_2^2$ is the loss function—$L(\theta)$.

We generate a network $\mathcal{N}^L(x,\theta)$ using:

- Activation function: ReLU (Rectified Linear Unit) activation function for all layers (first input and hidden layers), no activation function at the last output layer;
- DNN structure: 2 hidden layers, each layer comprises 12 neurons;
- Kernel initializer: normal for input and output layers and he_normal for all hidden layers;
- Training optimizer: Adam.

We chose the activation function ReLU because it has shown effectiveness in deep neural network training without experiencing a vanishing gradient problem.

Therefore, our method consists of two stages: offline and online. In the offline stage, we compute the POD basis functions and interpolation nodes, generate (or receive) data, and train the neural network to obtain high-conductive continuum pressure in the interpolation nodes. In the online stage, we can already solve the problem for an arbitrary set of input parameters. We pass the input data to the neural network, which gives the pressure values at the interpolation nodes. We perform spatial and temporal interpolation and solve the problem by treating the pressure of the high-permeability continuum as a known function. Figure 2 depicts a block diagram of our method's offline and online stages.

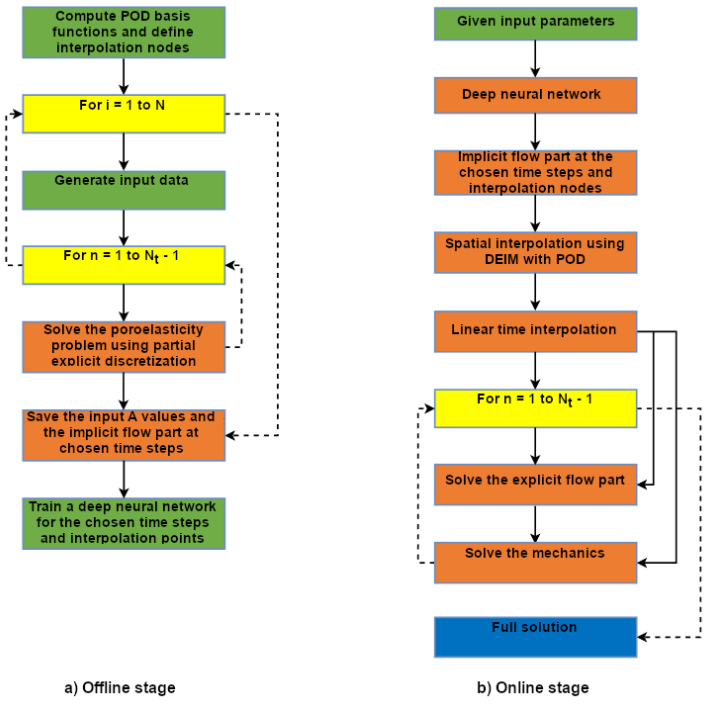

Figure 2. Flow charts of the offline and online stages of partial learning using partially explicit discretization with limited observation.

6. Numerical Results

This section considers the numerical solution of the poroelasticity problems in dual continuum heterogeneous media. As a computational domain, we consider $\Omega = [0,1] \times [0,1]$. We use a structured computational mesh with 121 vertices and 200 cells (see Figure 3). For model problem parameters, we set $M_1 = M_2 = 1$, $\alpha_1 = \alpha_2 = 1.0$, $\nu = 0.3$. The calculation is performed by $T_{max} = 1$ with time step $\tau = 0.01$. For the initial conditions, we set $p_0 = 0$ and $u_0 = 0$. The eterogeneous coefficients for elasticity modulus E and heterogeneous permeability k for the first and second continua are presented in Figure 4. We set the parametric function $g = \sum_{i=1}^{N_a} A_i \sin(i\pi x) \sin(i\pi y) \sin(it)$, where $N_a = 5$, and the value range $A_i \in [-1,1]$.

Note that this paper focuses mainly on developing a solution method, so we consider only the two-dimensional case. However, our approach can be beneficial for more complex problems, such as modeling three-phase and three-dimensional materials. Solving such problems will not require any significant changes to the algorithm.

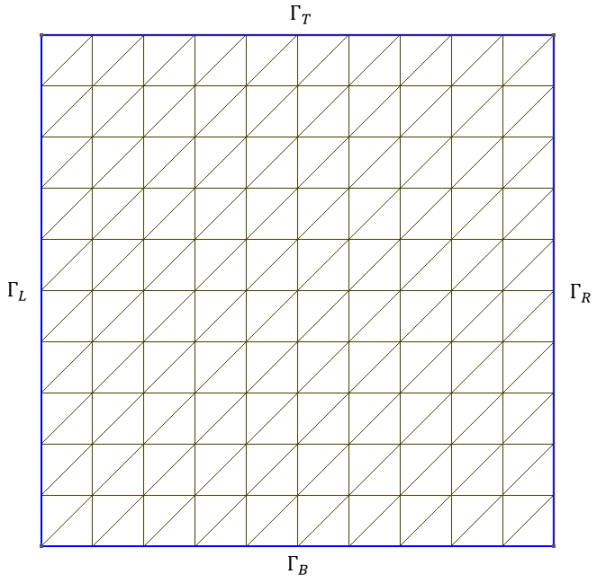

Figure 3. Computational domain.

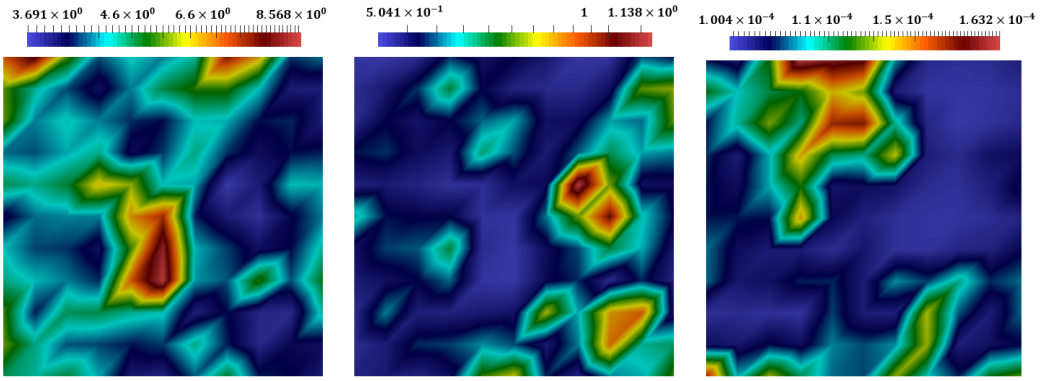

Figure 4. Elasticity parameter E (**left**) and heterogeneous permeabilities k_1 (**center**) and k_2 (**right**).

GMSH software was used to construct the computational domains and meshes [27]. The numerical realization of the problem was based on finite element approximation using the FEniCS computing platform [28]. For partial learning implementation, we used Keras [29]—a high-level API for the TensorFlow machine learning platform [26]. Finally, visualization of the numerical results was based on the ParaView software [30].

Our neural network aimed to find the high-conductive continuum pressure at nine interpolation points. As mentioned before, we used the Discrete Empirical Interpolation Method (DEIM) to perform the spatial interpolation. The first step of the DEIM was generating snapshot functions. To generate them, we solved the problem five times with different values of A_i. We saved the results every second time step. Therefore, we obtained $N = 250$ snapshots. Then, we solved a spectral problem and obtained nine basis functions. Using the DEIM algorithm, we selected nine optimal interpolation nodes. Next, we trained the neural network to obtain the high-conductive continuum pressure at these interpolation points.

The neural network considered random A_i values as the input and the high-conductive continuum pressure at nine interpolation points as output. Thus, the input data size was five, and the output data size was nine. We solved the problem 300 times with randomly generated A_i values to prepare a training dataset. For a test dataset, we solved the problem 20 times with A_i that were absent from the training dataset. Then, we performed spatial and time interpolations.

We used relative errors L_2 between the reference solution and the proposed approach's solution to compare the results. We performed multiple tests to determine the errors caused by different steps of our approach.

$$e^{p_1}_{L^2} = \left(\frac{\int_\Omega (p_1^{ref} - p_1^{appr})^2 dx}{\int_\Omega (p_1^{ref})^2 dx} \right)^{1/2} \cdot 100\%, \quad e^{p_2}_{L^2} = \left(\frac{\int_\Omega (p_2^{ref} - p_2^{appr})^2 dx}{\int_\Omega (p_2^{ref})^2 dx} \right)^{1/2} \cdot 100\%,$$

$$e^{u}_{L^2} = \left(\frac{\int_\Omega (u^{ref} - u^{appr})^2 dx}{\int_\Omega (u^{ref})^2 dx} \right)^{1/2} \cdot 100\%,$$

where the superscript ref denotes the reference solution, and the superscript $appr$ denotes the approximate solution. We considered different methods depending on the part of the error we wanted to evaluate.

Figures 5–8 present the distributions of pressures for the first and second continua and the displacements in x_1 and x_2 directions at different time steps (t_m with $m = 0.4, 0.8, 1$). In each figure, we depicted the solutions using the coupled explicit–implicit scheme, the split explicit–implicit scheme, and the proposed approach (from top to bottom). Because all results were very similar, we can conclude that our proposed method can provide good accuracy.

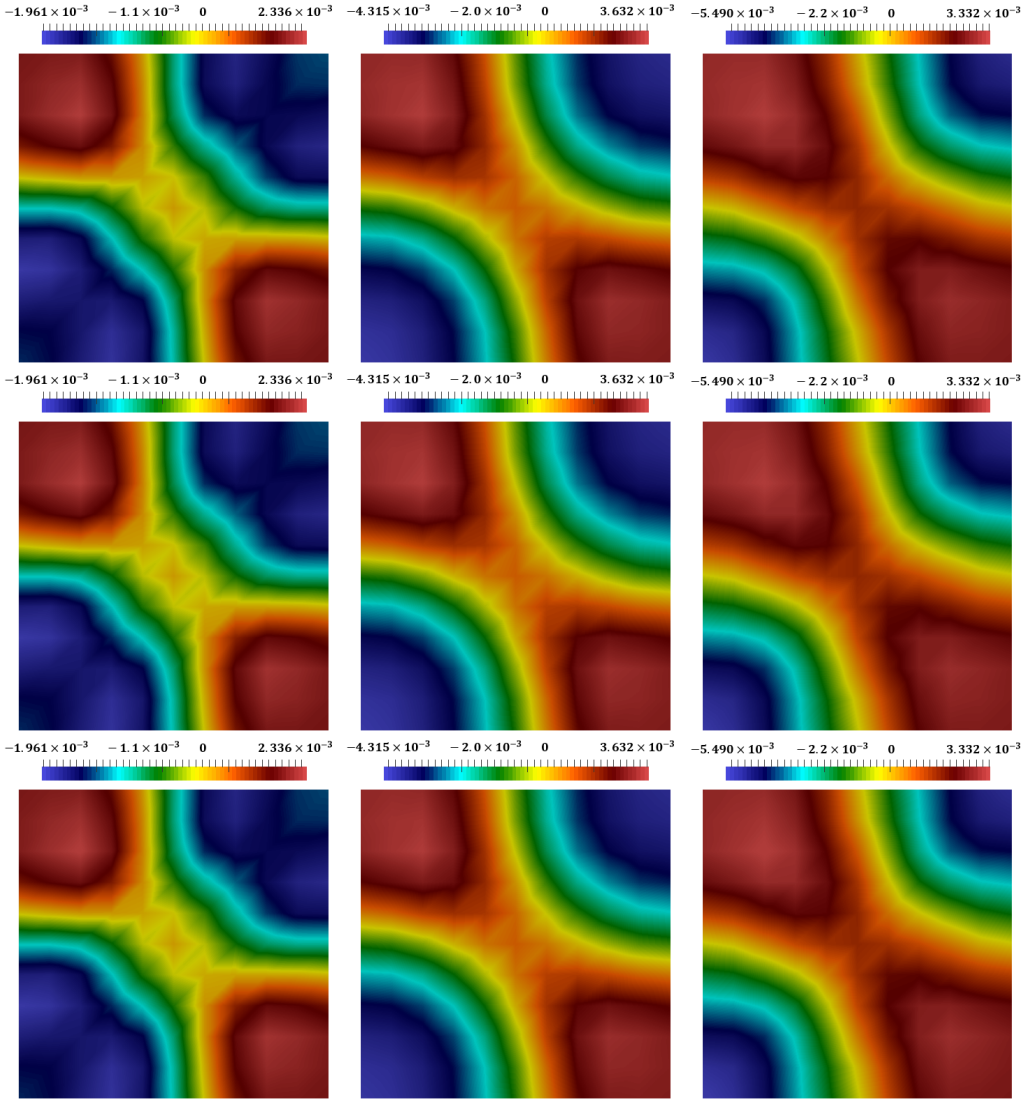

Figure 5. Distribution of pressure for the first continuum at different times, t_m with $m = 0.4, 0.8, 1$ (from **left** to **right**). (**The first row**): the coupled explicit–implicit solution. (**The second row**): the split explicit–implicit solution. (**The third row**): the proposed approach's solution.

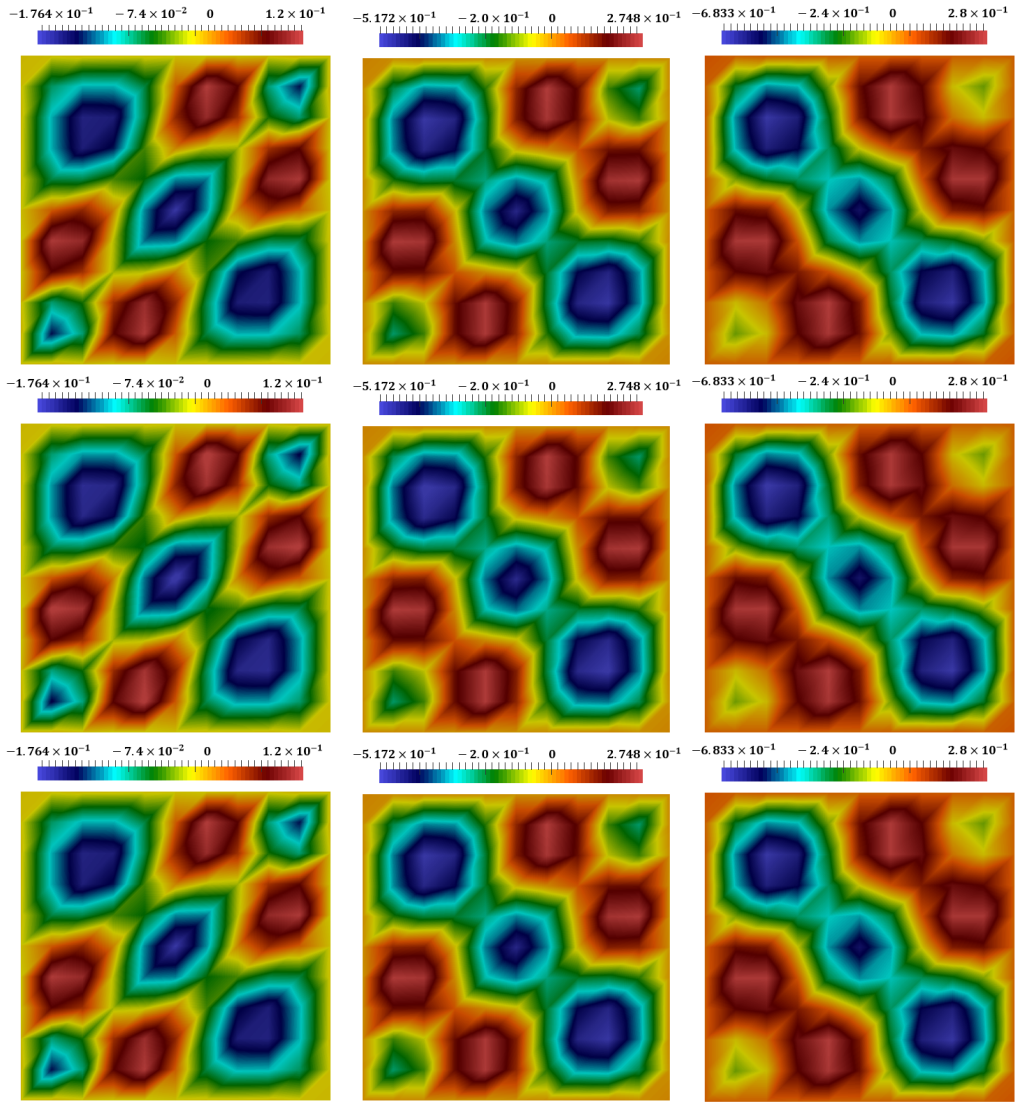

Figure 6. Distribution of pressure for the second continuum at different times, t_m with $m = 0.4, 0.8, 1$ (from **left** to **right**). (**The first row**): the coupled explicit–implicit solution. (**The second row**): the split explicit–implicit solution. (**The third row**): the proposed approach's solution.

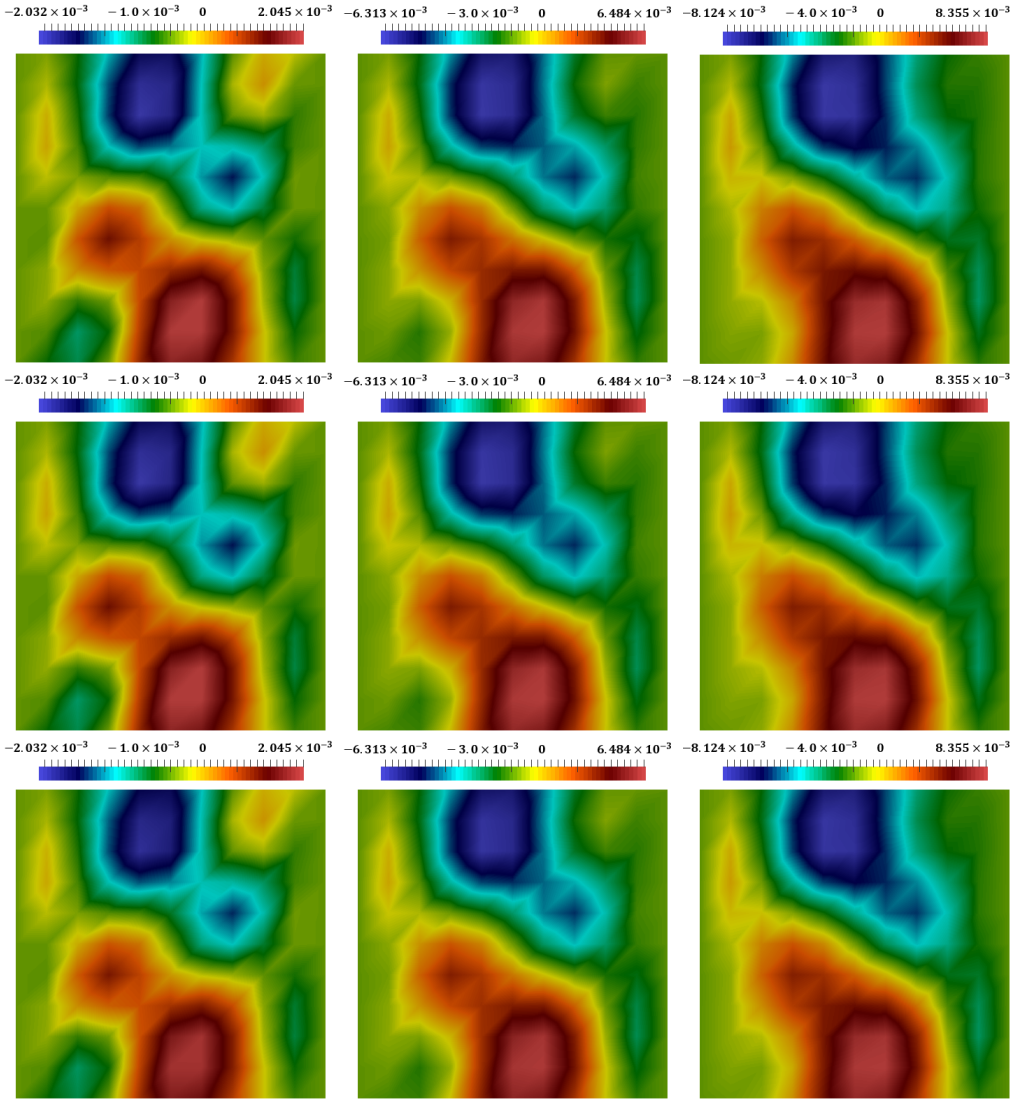

Figure 7. Distribution of displacement in x_1 direction at different times, t_m with $m = 0.4, 0.8, 1$ (from **left** to **right**). (**The first row**): the coupled explicit–implicit solution. (**The second row**): the split explicit–implicit solution. (**The third row**): the proposed approach's solution.

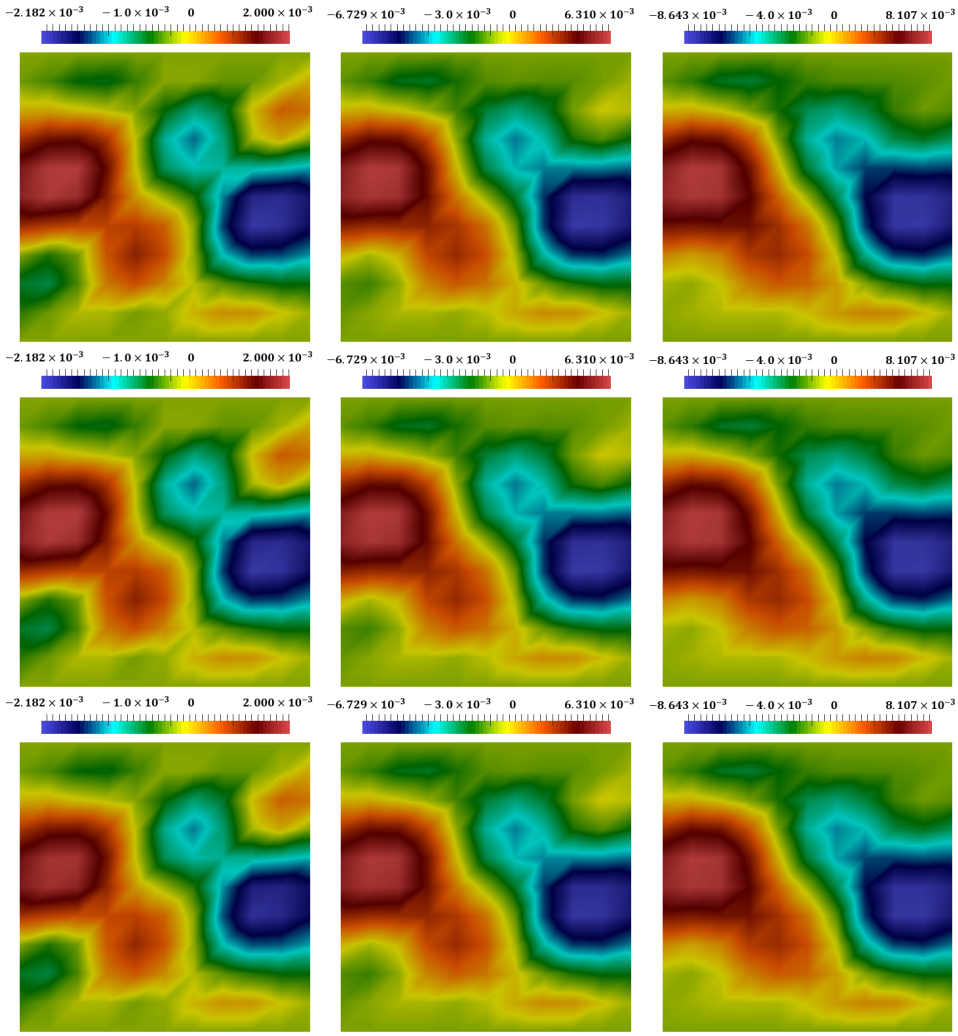

Figure 8. Distribution of displacement in x_2 direction at different times, t_m with $m = 0.4, 0.8, 1$ (from **left** to **right**). (**The first row**): the coupled explicit–implicit solution. (**The second row**): the split explicit–implicit solution. (**The third row**): the proposed approach's solution.

In Figure 9, we studied the dependence of accuracy on the computational mesh. As a reference mesh, we took the finest mesh (6561 vertices) and compared its results with the results of coarser meshes (121, 441, and 1681 vertices) for the first and second continuum and displacement. Figures 10 and 11 describe the distributions of strain and stress at the final time.

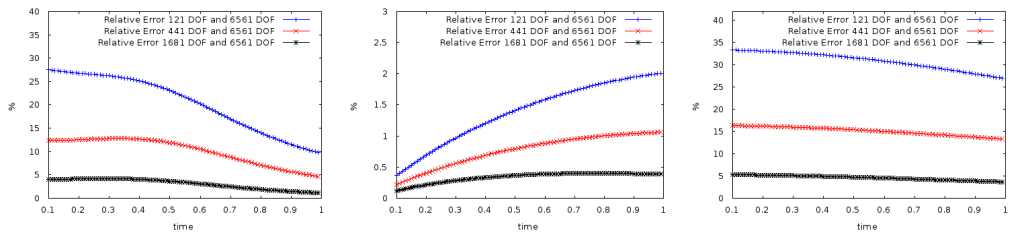

Figure 9. Relative L_2 errors in % for the reference mesh (6561 vertices) with other meshes (121, 441, and 1681 vertices) for the first and second continuum and displacement (from **left** to **right**).

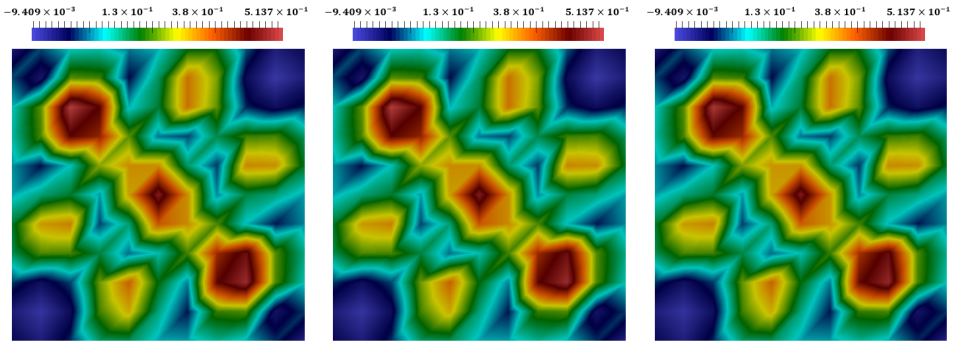

Figure 10. Distributions of the stress at the final time. The coupled explicit–implicit solution, the split explicit–implicit solution, and the proposed approach's solution (from **left** to **right**).

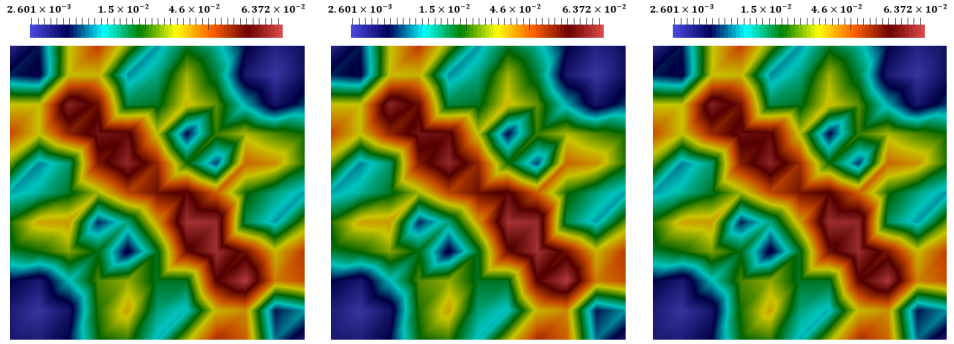

Figure 11. Distributions of the strain at the final time. The coupled explicit–implicit solution, the split explicit–implicit solution, and the proposed approach's solution(from **left** to **right**).

7. Conclusions

This work considered the poroelasticity problem in dual continuum heterogeneous media. For the mathematical model, we used dual continuum flow and the effective stress that contained terms of both continua. We applied a finite element method with standard linear basis functions for the spatial approximation. Furthermore, we used the explicit–implicit scheme for the time approximation, where the explicit scheme was used for the low-conductive continuum, and the implicit scheme was applied for the high-conductive one.

We proposed a new method based on hybrid explicit–implicit (HEI) learning to solve the poroelasticity problem in dual continuum heterogeneous media. The method's main idea was to train a Deep Neural Network to predict the implicit flow part at some node at some time steps. Then, we performed the DEIM interpolation and linear time interpolation. After that, we treated the high-conductive continuum's pressure as a known function and solved the whole problem. We considered the two-dimensional model problem to test the proposed method. We solved the problem using various methods to evaluate errors in the different steps of our method. The results demonstrated that the proposed approach can successfully solve the poroelasticity problems in dual continuum heterogeneous media.

Author Contributions: Conceptualization, implementation, investigation, writing, A.T., S.S., D.A., A.G. and M.V. All authors have read and agreed to the published version of the manuscript.

Funding: The work of S.S. and D.A. is supported by the Russian government's project Science and Universities 121110900017-5 aimed at supporting junior laboratories and the Russian Science Foundation grant 22-11-20027. A.T. is supported by the North-Caucasus Center for Mathematical Research under agreement N. 075-02-2021-1749 with the Ministry of Science and Higher Education of the Russian Federation.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Meirmanov, A. Mathematical Models for Poroelastic Flows. 2014. Available online: https://link.springer.com/book/10.2991/978-94-6239-015-7 (accessed on 16 December 2013).
2. Castelletto, N.; Klevtsov, S.; Hajibeygi, H.; Tchelepi, H.A. Multiscale two-stage solver for Biot's poroelasticity equations in subsurface media. *Comput. Geosci.* **2019**, *23*, 207–224. [CrossRef]
3. Iliev, O.; Kolesov, A.; Vabishchevich, P. Numerical solution of plate poroelasticity problems. *Transp. Porous Media* **2016**, *115*, 563–580. [CrossRef]
4. Vabishchevich, P.N.; Vasil'eva, M.V. Explicit-implicit schemes for convection-diffusion-reaction problems. *Numer. Anal. Appl.* **2012**, *5*, 297–306. [CrossRef]
5. Quevedo, R.; Roehl, D. A novel and efficient sequential-explicit technique for poroelasticity problems. *Comput. Geotech.* **2021**, *138*, 104334. [CrossRef]
6. Almani, T.; Kumar, K.; Singh, G.; Wheeler, M.F. Stability of multirate explicit coupling of geomechanics with flow in a poroelastic medium. *Comput. Math. Appl.* **2019**, *78*, 2682–2699. [CrossRef]
7. Kim, J. Sequential Methods for Coupled Geomechanics and Multiphase Flow. Doctoral Dissertation, Stanford University, Stanford, CA, USA, 2010.
8. Kim, J.; Tchelepi, H.A.; Juanes, R. Stability and convergence of sequential methods for coupled flow and geomechanics: Drained and undrained splits. *Comput. Methods Appl. Mech. Eng.* **2011**, *200*, 2094–2116. [CrossRef]
9. Kim, J.; Tchelepi, H.A.; Juanes, R. Stability and convergence of sequential methods for coupled flow and geomechanics: Fixed-stress and fixed-strain splits. *Comput. Methods Appl. Mech. Eng.* **2011**, *200*, 1591–1606. [CrossRef]
10. Vabishchevich, P.N.; Vasil'eva, M.V.; Kolesov, A.E. Splitting scheme for poroelasticity and thermoelasticity problems. *Comput. Math. Math. Phys.* **2014**, *54*, 1305–1315. [CrossRef]
11. Akkutlu, I.Y.; Efendiev, Y.; Vasilyeva, M.; Wang, Y. Multiscale model reduction for shale gas transport in poroelastic fractured media. *J. Comput. Phys.* **2018**, *353*, 356–376. [CrossRef]
12. Vasilyeva, M.; Chung, E.T.; Efendiev, Y.; Kim, J. Constrained energy minimization based upscaling for coupled flow and mechanics. *J. Comput. Phys.* **2019**, *376*, 660–674. [CrossRef]
13. Ammosov, D.; Vasilyeva, M.; Chung, E.T. Generalized Multiscale Finite Element Method for thermoporoelasticity problems in heterogeneous and fractured media. *J. Comput. Appl. Math.* **2022**, *407*, 113995. [CrossRef]
14. Vasilyeva, M.; Tyrylgin, A. Machine learning for accelerating macroscopic parameters prediction for poroelasticity problem in stochastic media. *Comput. Math. Appl.* **2021**, *84*, 185–202. [CrossRef]
15. Chaturantabut, S.; Sorensen, D.C. Nonlinear model reduction via discrete empirical interpolation. *SIAM J. Sci. Comput.* **2010**, *32*, 2737–2764. [CrossRef]
16. Wang, Z.; Akhtar, I.; Borggaard, J.; Iliescu, T. Proper orthogonal decomposition closure models for turbulent flows: A numerical comparison. *Comput. Methods Appl. Mech. Eng.* **2012**, *237*, 10–26. [CrossRef]
17. Wang, Z.; Akhtar, I.; Borggaard, J.; Iliescu, T. Two-level discretizations of nonlinear closure models for proper orthogonal decomposition. *J. Comput. Phys.* **2011**, *230*, 126–146. [CrossRef]
18. Efendiev, Y.; Leung, W.T.; Lin, G.; Zhang, Z. HEI: Hybrid explicit-implicit learning for multiscale problems. *arXiv* **2021**, arXiv:2109.02147.
19. Boutin, C.; Royer, P. On models of double porosity poroelastic media. *Geophys. Suppl. Mon. Not. R. Astron. Soc.* **2015**, *203*, 1694–1725. [CrossRef]
20. Zhang, J.; Bai, M.; Roegiers, J.C. Dual-porosity poroelastic analyses of wellbore stability. *Int. J. Rock Mech. Min. Sci.* **2003**, *40*, 473–483. [CrossRef]
21. Barenblatt, G.I.; Zheltov, I.P.; Kochina, I. Basic concepts in the theory of seepage of homogeneous liquids in fissured rocks [strata]. *J. Appl. Math. Mech.* **1960**, *24*, 1286–1303. [CrossRef]
22. Jänicke, R.; Larsson, F.; Runesson, K. A poro-viscoelastic substitute model of fine-scale poroelasticity obtained from homogenization and numerical model reduction. *Comput. Mech.* **2020**, *65*, 1063–1083. [CrossRef]
23. Tyrylgin, A.; Vasilyeva, M.; Spiridonov, D.; Chung, E.T. Generalized Multiscale Finite Element Method for the poroelasticity problem in multicontinuum media. *J. Comput. Appl. Math.* **2020**, *374*, 112783. [CrossRef]
24. Tyrylgin, A.; Vasilyeva, M.; Ammosov, D.; Chung, E.T.; Efendiev, Y. Online Coupled Generalized Multiscale Finite Element Method for the Poroelasticity Problem in Fractured and Heterogeneous Media. *Fluids* **2021**, *6*, 298. [CrossRef]
25. D'angelo, C.; Quarteroni, A. On the coupling of 1d and 3d diffusion-reaction equations: application to tissue perfusion problems. *Math. Model. Methods Appl. Sci.* **2008**, *18*, 1481–1504. [CrossRef]

26. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2015. Available online: tensorflow.org (accessed on 2 November 2016).
27. Geuzaine, C.; Remacle, J.F. A three-dimensional finite element mesh generator with built-in pre-and post-processing facilities. *Int. J. Numer. Methods Eng.* **2020**, *11*, 79.
28. Logg, A.; Mardal, K.A.; Wells, G. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2012; Volume 84.
29. Chollet, F.; Watson, M.; Bursztein, E.; Zhu, Q.S.; Jin, H. keras. 2015. Available online: https://github.com/fchollet/keras (accessed on 27 March 2015).
30. Ahrens, J.; Geveci, B.; Law, C. Paraview: An end-user tool for large data visualization. *Vis. Handb.* **2005**, 717–735 .