

GW-PINN: A deep learning algorithm for solving groundwater flow equations



Xiaoping Zhang^a, Yan Zhu^{b,*}, Jing Wang^a, Lili Ju^c, Yingzhi Qian^b, Ming Ye^d, Jinzhong Yang^b

^a School of Mathematics and Statistics, Wuhan University, Wuhan, Hubei 430072, China

^b State Key Laboratory of Water Resources and Hydropower Engineering Science, Wuhan University, Wuhan, Hubei 430072, China

^c Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA

^d Department of Earth, Ocean, and Atmospheric Science, Florida State University, Tallahassee, FL 32306, USA

ARTICLE INFO

Keywords:

Deep learning method

GW-PINN

Locally refined sampling strategy

Hard constraint

Two-stage training strategy

ABSTRACT

Machine learning methods provide new perspective for more convenient and efficient prediction of groundwater flow. In this study, a deep learning method “GW-PINN” without labeled data for solving groundwater flow equations with wells was proposed. GW-PINN takes the physics inform neural network (PINN) as the backbone and uses either the hard or soft constraint in the loss function for training. A locally refined sampling strategy (LRS) is adopted to generate the consistent spatial sampling points for problems with strong hydraulic head change, and then combined with an appropriate temporal sampling scheme to obtain the final spatial-temporal sampling points. A snowball-style two-stage training strategy by dividing the temporal domain into two sub-domains is designed to decrease the sampling points. Five cases were designed to test the training performance of GW-PINN under different sampling strategies and two constraints. The predicted results of GW-PINN were compared with MODFLOW and the analytical solution. The results demonstrate that GW-PINN possesses strong ability in capturing the hydraulic head change for both confined and un-confined aquifers. The hard constraint owns more robust learning ability than the soft constraint. The LRS strategy can generate more accurate results with much fewer sampling points than traditional sampling strategies, and the snowball-style two-stage training strategy is significantly efficient for problems with the drastic change of hydraulic head. Furthermore, the application of GW-PINN as a surrogate model for parameterized groundwater flow equations is illustrated. This study provides an option tool for efficient groundwater flow simulation, especially for those with local refinements are needed.

1. Introduction

With the improvement of computational power and explosive growth of data, deep learning has achieved great success in many fields, such as computation vision, natural language processing, and other fields (Krizhevsky et al., 2012; He et al., 2016). In the past few years, significant effort also has been devoted to use these technologies to solve problems arising in computational and scientific engineering, among which a typical case is numerical solution of partial differential equations (PDEs) of various types (Weinan and Yu, 2018; Jagtap et al., 2020; Raissi et al., 2019; Sirignano and Spiliopoulos, 2018; Zhu and Zabaras, 2018).

What we are concerned with in this paper is the numerical simulation of groundwater flow equations, which is important for groundwater

resources management and protection (Crowe et al., 2004; Knights et al., 2017; Walter et al., 2020). There exists a large amount of mature traditional numerical methods for solving the partial differential equations (PDEs) of groundwater flow, e.g., the finite element method, the finite difference method and the finite volume method (Brunner and Simmons, 2012; Harbaugh et al., 2000; Trefry and Muffels, 2007). Important issues in those spatial discretization methods for solving the PDEs of groundwater flow are the large computational cost and difficult iteration convergence in handling aquifers with strong heterogeneity and complex boundaries or multiple source/sink terms, whereas the local grid refinements are needed (Borsi et al., 2013; Dickinson et al., 2007; Mehl and Hill, 2002; Panday and Langevin, 2012; Vilhelmsen et al., 2011). These models are usually difficult to implement for staff of water resources management agency since very professional numerical

* Corresponding author.

E-mail address: zyan0701@163.com (Y. Zhu).

modeling knowledge and experience are needed. Compared to the traditional numerical methods, the deep learning approach provides a fresh perspective for solving PDEs and possesses many unique advantages: (1) universality: it is a universal method, which can solve different types of PDEs on arbitrary domains; (2) meshless: it only requires the reasonable distribution of sample points, which is much easier than the mesh generation; (3) efficiency: the computational costs of the traditional time-stepping methods, such as the forward Euler scheme, the Crank-Nicolson scheme, etc., grow rapidly as the number of time iterations increases, and the prediction should be carried out sequentially, whereas the deep learning approach can predict the solution at any time very quickly, once the model is well trained, which can be used as a surrogate model for quick prediction for water resources management; (4) convenience: the traditional numerical method can only predict the solution on the grid points, and additional post-processes are needed to obtain the approximate solutions on the non-grid points, while the deep learning approach can predict the solution at arbitrary spatial-temporal points.

To the best of our knowledge, the first attempt of numerical solution of PDEs using neural networks can be traced back at least to the end of the last century (Dissanayake and Phan-Thien, 1994), where the authors proposed an unsupervised method to train the neural network by minimizing

$$\int_{\Omega} (Lu - f)^2 dx + \int_{\partial\Omega} (Bu - g)^2 dx$$

to approximate the solution of the following PDEs,

$$\begin{cases} Lu(x) = f, & x \in \Omega, \\ Bu(x) = g, & x \in \partial\Omega. \end{cases}$$

and two simple examples were considered to verify the effectiveness of the proposed method. However, due to the limitation of computational power at that time, it did not attract enough attention. Until the last decade, this approach has made remarkable progress, and a series of fruitful research works have emerged (Weinan and Yu, 2018; Meng et al., 2020; Raissi et al., 2019; Sirignano and Spiliopoulos, 2018; Zhu and Zabaras, 2018). There are currently two major ways to use deep learning methods to solve PDEs.

The first way is to convert PDEs into their equivalent variational forms, and then solve them using neural networks. The pioneering work is the deep Ritz method proposed by Weinan and Yu, 2018, followed by a series of excellent researches (Liao and Ming, 2019; Müller and Zeinhofer, 2019; Wang and Zhang, 2020; Chen et al., 2020). The deep Nitsche method (Liao and Ming, 2019) extended the deep Ritz method to deal with essential boundary conditions without extra computational costs. Chen et al. (2020) carried out the comparison study for elliptic problems with different boundary conditions, including Dirichlet, Neumann, Robin, and periodic boundary conditions by using the deep Ritz method and the deep Galerkin method (Sirignano and Spiliopoulos, 2018), which observed that the deep Galerkin method works better for problems with smooth solutions while the deep Ritz method works better for problems with low-regularity solutions. Based on the deep Ritz Method, Wang and Zhang (2020) proposed a mesh-free method to solve the interface problems, where two challenging interface problems were investigated, including an elliptic PDE with a discontinuous and heterogeneous diffusion coefficient and a linear elasticity equation with discontinuous stress tensor.

The other way is to use neural networks to directly deal with the original PDEs, and the most representative work is the physical informed neural network (PINN) proposed by Raissi et al. (2019), where the PDE residual is incorporated into the loss function as a regularizer, thereby constraining the space of admissible solutions. The original PINN uses the residual of PDEs as the loss function, which is called as the soft constraint. Different from the soft constraints, Berg and Nystrom (2018)

and Sun et al. (2020) proposed some sort of hard constraint to make the output of the neural networks satisfy the boundary and initial conditions. PINNs have been successfully applied to solve various PDE problems, such as classic time-independent or time-dependent PDEs, fractional PDEs (Pang et al., 2019), and stochastic PDEs (Zhang et al., 2020). However, for high-dimensional problems, especially for time-dependent problems, the amount of data sampling tends to become huge, which would consume unacceptable training time, or even collapse the training of PINN.

In recent years, the deep learning method has been applied for many hydrology problems (Shen, 2018), e.g., groundwater table depth forecasting (Sun et al., 2022), flood forecasting (Lv et al., 2020), runoff prediction (Huang et al., 2020), water quality prediction (Anshuman and Eldho, 2022; Liang et al., 2020), and uncertainty quantification or parameter inversion problems in subsurface flow (Mo et al., 2019; Zhang, 2019; Laloy, 2018). Most machine learning models for groundwater flow prediction were labeled data-driven to learn the relationship between the groundwater table level with the major impact factors, e.g., precipitation, evapotranspiration, groundwater exploration and irrigation. The popular ones included the controlled auto-regressive (CAR) method (Zhao et al., 2020), autoregressive integrated moving average (ARIMA) (Adhikary et al., 2012; Mirzavand and Ghazavi, 2014), artificial neural network (ANN) (Kuo et al. 2007; Sahu et al., 2020), support vector machine (SVM) (Yoon et al., 2011), genetic algorithms (Jha and Sahoo, 2015; Ravansalar et al. 2017) and long short-term memory (LSTM) (Zhang et al., 2018). Recently, Wang et al. (2020) proposed a theory-guided neural network (TgNN) for subsurface flow with heterogeneous hydraulic conductivity field, in which the groundwater flow equation was integrated with the labeled hydraulic head data and additional engineering controls to be constraints in the loss function for subsurface flow modeling. And a weak form TgNN (TgNN-wf) was further proposed for diffusivity equations with local source/sink terms by Xu et al. (2021), and the authors mentioned that TgNN-wf cannot solve the groundwater flow equation containing source/sink terms when constrained only by initial conditions, boundary conditions, and weak form residual losses, and the labeled training data was required to capture the pressure drawdown in cases with wells. The current machine learning methods require a large number of label data and face the over fit problem in training (Srivastava et al., 2014). Very few studies have been proposed for solving groundwater flow equations directly without label data based on the deep learning method, especially for those with local source/sink terms due to the discontinuity at the source/ sink location, which worth paying more attention.

In this paper, a deep learning method to directly solve the PDEs of the groundwater flow without label data called as GW-PINN has been proposed. A locally refined spatial-temporal sampling strategy and a snowball-style two-stage training strategy are proposed and adopted, which make the method suitable for situations with source/sink terms, where local grid refinements are needed. The hard constraint is adopted, which is beneficial to train GW-PINN by only including the initial, boundary losses and residual loss of PDEs, while avoids using the labeled data in the training. Five test cases and one application case are designed to evaluate the accuracy and efficiency of the proposed method. This paper is organized as follows. The description of groundwater flow equations is shown in Section 2, followed by the network architecture of GW-PINN, the sampling strategy, loss functions, and the training strategy introduced in Section 3. The description of the five cases is presented in Section 4. Ablation studies, including the effects of different loss functions, sampling strategies, and training strategies are shown in Section 5, together with comparison results of the test cases with MODFLOW (Harbaugh et al., 2000). Model discussion is shown in Section 6. Finally, some conclusions are drawn in Section 7.

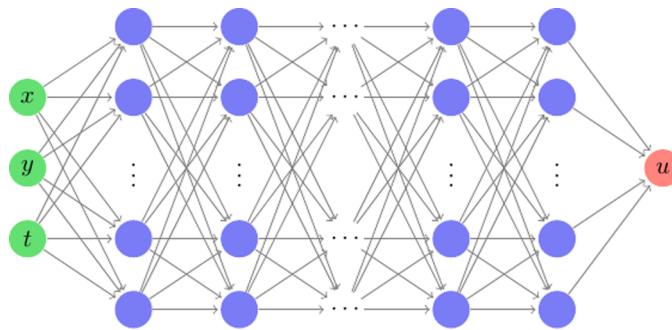


Fig. 1. Network architecture.

2. Groundwater flow equations

Let $\Omega \subset \mathbb{R}^2$ be an open connected domain in the two-dimensional space, $\Omega \times (0, T]$ denotes the spatial-temporal region to be solved, $\Omega \times \{0\}$ denotes the initial region, $\Gamma_1 \times (0, T]$ denotes the boundary of first kind (Dirichlet boundary), and $\Gamma_2 \times (0, T]$ denotes the boundary of second kind (Neumann boundary). Moreover, let $\mathbf{x} = (x, y)$ be a spatial point in Ω and (x, y, t) be a spatial-temporal point, and then the groundwater flow can be described as the following system,

$$\nabla \cdot (\mathbf{K} \nabla h) = \mu \frac{\partial h}{\partial t} + f(x, y, t), \quad (x, t) \in \Omega \times (0, T], \quad (1a)$$

$$h(x, y, t) = g_0(x, y), \quad (x, y, t) \in \Omega \times \{0\}, \quad (1b)$$

$$h(x, y, t) = g_1(x, y, t), \quad (x, y, t) \in \Gamma_1 \times (0, T] \quad (1c)$$

$$\mathbf{K} \nabla h(x, y, t) \cdot \mathbf{n}(x, y) = g_2(x, y, t), \quad (x, y, t) \in \Gamma_2 \times (0, T], \quad (1d)$$

where h is the hydraulic head ([L]); ∇ is the gradient operator defined by $\nabla h = \left(\frac{\partial h}{\partial x}, \frac{\partial h}{\partial y} \right)^T$; g_0, g_1 and g_2 refer to the specified initial value, Dirichlet value and Neumann value of the problem, respectively; and $\mathbf{n}(x, y)$ denotes the unit normal vector outward to Ω on the Neumann boundary. It's remarkable that the specific form of two parameters tensor depends on the type of aquifer, i.e.,

$$\mu = \begin{cases} \mu_e = \mu_s M, & \text{confined aquifer} \\ \mu_d, & \text{unconfined aquifer} \end{cases} \quad (2)$$

and

$$\mathbf{K} = \begin{cases} \mathbf{K}_s M, & \text{confined aquifer} \\ \mathbf{K}_s h, & \text{unconfined aquifer} \end{cases} \quad (3)$$

where $\mu = \mu_e$ is the elastic release coefficient of the confined aquifer ([·]) with the specific storativity μ_s ([L⁻¹]) and the thickness of aquifer M ([L]), or $\mu = \mu_d$ is the specified yield of unconfined aquifer, [·]; $\mathbf{K} = \mathbf{K}_s M$ is the transmissivity tensor if the confined aquifer is considered or $\mathbf{K} = \mathbf{K}_s h$ is the transmissivity tensor if the unconfined aquifer is considered, and,

$$\mathbf{K}_s = \begin{pmatrix} K_{s,x} & 0 \\ 0 & K_{s,y} \end{pmatrix}, \quad (4)$$

denotes the saturated hydraulic conductivity tensor. $K_{s,x}$ and $K_{s,y}$ are the saturated hydraulic conductivity in the x and y directions, respectively, [LT⁻¹]. f is the source/sink term defined by,

$$f = P - E - \sum_{i=1}^{N_w} Q_i \delta(\mathbf{x} - \mathbf{x}_i), \quad (5)$$

where $P = P(x, y, t)$ denotes the amount of precipitation/infiltration to recharge groundwater ([LT⁻¹]); $E = E(x, y, t)$ refers to the

evapotranspiration of groundwater ([LT⁻¹]), and Q_i represents the pumping capacity of pumping well located at (x_i, y_i) ; N_w is the number of pumping wells and δ is the Dirac delta function satisfying the following conditions,

$$\delta(\mathbf{x}) = \begin{cases} 0, & \mathbf{x} \neq 0, \text{ and } \int_{\mathbb{R}^2} \delta(\mathbf{x}) d\mathbf{x} = 1, \\ \infty, & \mathbf{x} = 0, \end{cases} \quad (6)$$

The physical meaning of Eq. (1a) is that it reflects the mass conservation relation of groundwater movement in the aquifer, and represents the water balance relation of the unit volume and the unit time of aquifer under the Darcy's flow condition. That is to say, the difference between mass of water flowing in and out of the unit volume aquifer in the unit time (the left side of Eq. (1a)) is equal to that of water released (or stored) in the unit volume aquifer in the same time (the right side of Eq. (1a)).

3. Proposed method: GW-PINN

Our GW-PINN takes the physics inform neural network (PINN) as the backbone and uses the hard constraint approach in the loss function. For issues where appropriate locally refined spatial meshes are needed, we adopt locally refined sampling strategy to generate the consistent spatial sampling points with the problem, and then combined with appropriate temporal sampling points to obtain the ultimate spatial-temporal sampling points. Consequently, these sampling points will be used as training set to be fed into GW-PINN to train it. Once the network is trained well, it can be used to predict the hydraulic head at any time and at any point in space. In order to train GW-PINN effective and fast, we design a snowball-style two-stage strategy by dividing the temporal domain into two stages and then training respective networks in each stage in sequence. We find that the training cost mainly concentrate in the initial stage, and in this stage the drastic change process of hydraulic head can be captured accurately without large amount of training points. Once the first stage network is trained, the training in the second stage becomes much faster. The details of GW-PINN are shown in the following sections.

3.1. Network architecture

The basic architecture of our GW-PINN for solving Eq. (1) is shown in Fig. 1. The network takes a spatial-temporal point (x, y, t) as an input, and output a scalar u . The input layer is connected to $D - 1$ hidden layers and an output layer, which form a fully-connected neural network of depth D . Letting n_k be the number of neurons in the k -th layer, each hidden layer receives an input $v^{k-1} \in \mathbb{R}^{n_{k-1}}$ from the previous layer output and transforms it to,

$$l_k(v^{k-1}) = W^k v^{k-1} + b^k \quad (7)$$

where $W^k \in \mathbb{R}^{n_k \times n_{k-1}}$ and $b^k \in \mathbb{R}^{n_k}$. The nonlinear activation function $\sigma(\bullet)$ is applied to each component of the transformed vector before sending it to the next layer, other than the last hidden layer. The network thus is a composite of a sequence of nonlinear functions,

$$u(x, y, t; \Theta) = (l_D \circ \sigma \circ l_{D-1} \circ \dots \circ \sigma \circ l_1)(x, y, t) \quad (8)$$

where the operator “ \circ ” denotes the composition and $\Theta = \{W^k, b^k\}_{k=1}^D$ represents the trainable parameters in the network.

3.2. Approximation of the Dirac delta function

In the network setting, we seek for a classic (smooth) solution satisfying the strong form of the PDE Eq. (1). However, if the aquifer contains pumping wells, there exist the Dirac delta functions $\delta(\mathbf{x}, \mathbf{y})$ in the source/sink term. But $\delta(\mathbf{x}, \mathbf{y})$ is not differentiable everywhere, and indeed it can only be well defined in the sense of distribution. Thus, in

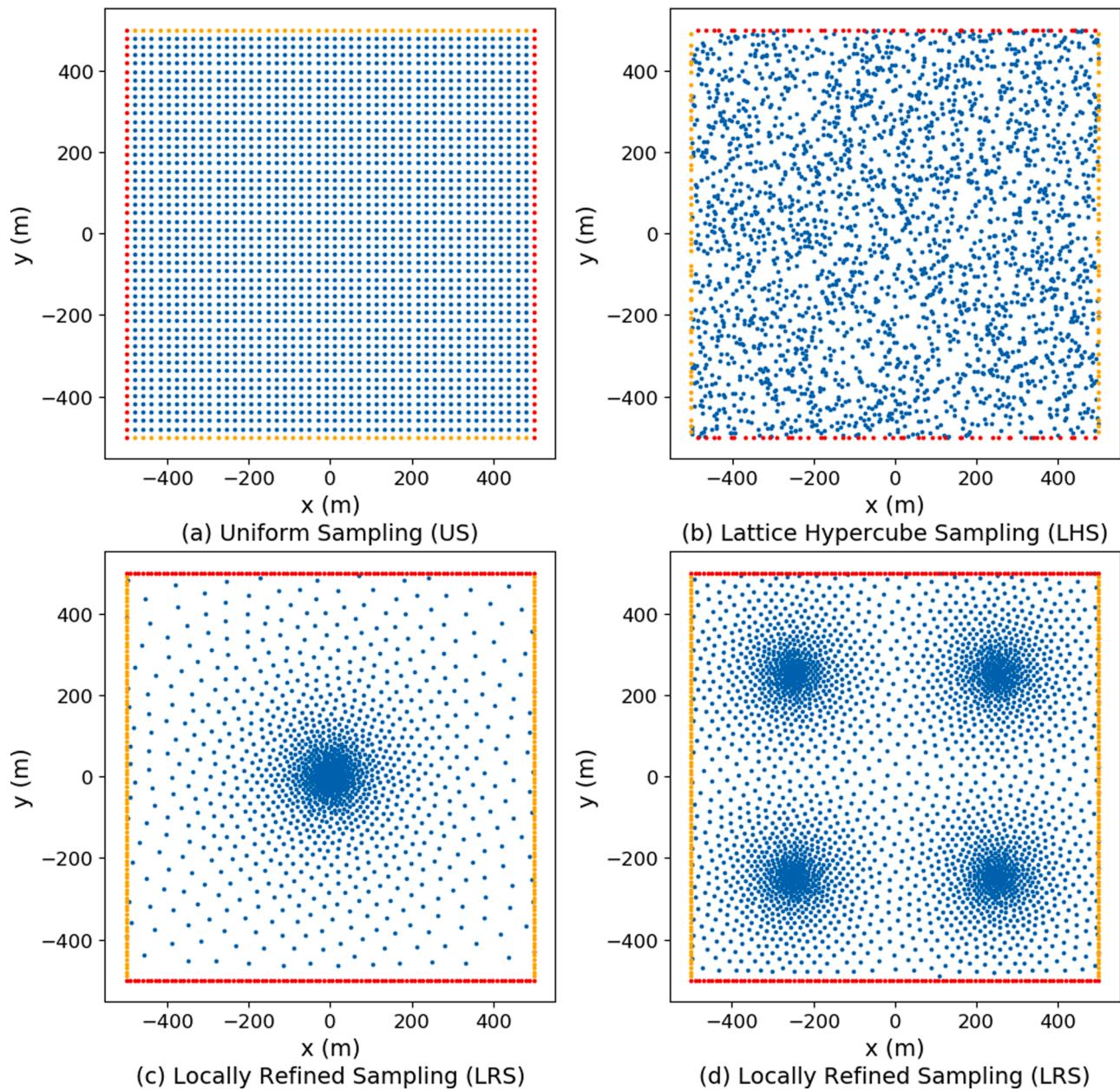


Fig. 2. Different strategies to generate spatial sampling points $\Omega_D, \Gamma_{1,D}$ and $\Gamma_{2,D}$ in different cases.

practice, we approximate the Dirac delta function by a multi-dimensional Gaussian density function,

$$\rho(\mathbf{x}, \mathbf{y}) = \frac{1}{2\pi s^2} e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2s^2}}, \quad (9)$$

where the parameter $s > 0$ denotes the standard deviation of the distribution. As $s \rightarrow 0$, this function approaches to the Dirac delta function.

3.3. Sampling strategy

It is crucial to generate appropriate spatial-temporal sampling points for training of the network. By taking into account the features of the source/sink term, we propose adaptive spatial and temporal sampling strategies and training strategies in different cases.

(1) Spatial sampling

We sample the spatial domain in two cases: the inner domain Ω and the boundary Γ , which is composed of the first kind boundary Γ_1 and the

second kind boundary Γ_2 , i.e., $\Gamma = \Gamma_1 \cup \Gamma_2$. Sampling on the boundary Γ can be done through a uniform sampling strategy, such as equidistance nodes in each direction. We denote Γ_D the set of all these sampling points on the boundary Γ , and then divided it into two parts $\Gamma_{1,D}$ and $\Gamma_{2,D}$, which denote the set of sampling points on Γ_1 and Γ_2 , respectively. The generation of the set of sampling points in Ω , denoted as Ω_D , is determined by the property of the source/sink term shown in Eq. (5). Specifically, if the source/sink term only contains the precipitation term P and the evapotranspiration term E , it is enough to simply adopt the uniform sampling (US) strategy or lattice hypercube sampling (LHS) strategy to generate Ω_D , since the changes of these two terms are generally not remarkable, see Fig. 2 (a) and (b). If pumping wells are included in the source/sink term, the variations of hydraulic head depend on the location of the pumping wells. Specifically, the closer to the pumping wells, the more intense the change of the hydraulic head, while the farther away from the pumping wells, the gentler the change of the hydraulic head. Under this circumstance, a locally refined sampling (LRS) strategy is needed to be consistent with the changes of the

(a) Uniform Sampling (US)

(b) Lattice Hypercube Sampling (LHS)

(c) Locally Refined Sampling (LRS)

Fig. 3. Different strategies to generate temporal sampling points $(0, T]_D$.

hydraulic head. To achieve this goal, we use the NodeLab proposed by Mishra (2019) to generate Ω_D , see Fig. 2 (c) for an aquifer with a single well, and Fig. 2 (d) for an aquifer with multiple wells.

(2) Temporal sampling

Since the groundwater flow equation shown in Eq. (1) is time dependent, in addition to the spatial sampling, it is also necessary to conduct the appropriate temporal sampling, and we denote $(0, T]_D$ the set of sampling points in the time domain $(0, T]$. By the prior knowledge of groundwater problems, we know that the change of hydraulic head will eventually become stable as time goes on regardless of the type of aquifers. But whether there are pumping wells or not in the aquifer will affect the change process of hydraulic head. If an aquifer without pumping wells is considered, the overall change of hydraulic head during the time domain is relatively gentle, and therefore, we can adopt the uniform sampling with relatively large time step to obtain $(0, T]_D$ as shown in Fig. 3(a) or the lattice hypercube sampling (LHS) as shown in Fig. 3(b). Otherwise, the hydraulic head will change dramatically in the initial stage due to the influence of the pumping wells, especially near the wells. To capture these changes well, in addition to the locally refined spatial sampling points Ω_D , we also conduct the same strategy in the time direction to obtain $(0, T]_D$ during the initial stage as shown in Fig. 3(c). Such a locally refined sampling strategy is implemented by generating a series of subintervals with equal ratio r in $(0, T]$.

(3) Spatial-temporal sampling

In the previous two subsections, we have already discussed the generation of spatial sampling points and temporal sampling points, respectively. Then we use the spatial sampling $\Omega_D, \Gamma_{1,D}$ and $\Gamma_{2,D}$ and the temporal sampling $(0, T]_D$ to produce the spatial-temporal sampling point sets,

$$\Omega_D^T = \Omega_D \times (0, T]_D, \quad \Omega_D^0 = \Omega_D \times \{0\}, \quad \Gamma_{1,D}^T = \Gamma_{1,D} \times (0, T]_D, \quad \Gamma_{2,D}^T = \Gamma_{2,D} \times (0, T]_D \quad (10)$$

which will be fed into our GW-PINN during the training process.

3.4. Loss Functions

Once the spatial-temporal sampling points are obtained, we need to minimize a loss function to train the neural network. Two types of loss functions, the soft constraint and the hard constraint, are both investigated, where the first is originated from the pioneer work PINNs (Raissi et al., 2019), and the second is inspired from Sun et al. (2020).

(1) Soft constraint

In this case, we directly take the output of the neural network u as the hydraulic head h , i.e., $h = u(x, y, t; \Theta)$, and the soft constraint is expressed as,

$$L(\Theta) = L_{PDE}(\Theta) + \sum_{i=0}^2 \lambda_i L_i(\Theta), \quad (11)$$

where,

$$L_{PDE}(\Theta) = \sum_{(x,y,t) \in \Omega_D \times (0,T)_D} \left[\nabla \cdot (K \nabla h(x, y, t; \Theta)) - \mu \frac{\partial h(x, y, t; \Theta)}{\partial t} - f(x, y, t) \right]^2, \quad (12a)$$

$$L_0(\Theta) = \sum_{(x,y,t) \in (\Omega_D \cup \Gamma_{1,D} \cup \Gamma_{2,D}) \times \{0\}} [h(x, y, t; \Theta) - g_0(x, y, t)]^2, \quad (12b)$$

$$L_1(\Theta) = \sum_{(x,y,t) \in \Gamma_{1,D} \times (0,T)} [h(x, y, t; \Theta) - g_1(x, y, t)]^2, \quad (12c)$$

$$L_2(\Theta) = \sum_{(x,y,t) \in \Gamma_{2,D} \times (0,T)} [K \nabla h(x, y, t; \Theta) \cdot n(x, y) - g_2(x, y, t)]^2. \quad (12d)$$

where $L_{PDE}(\Theta)$ denotes the pointwise PDE residual; $L_0(\Theta), L_1(\Theta), L_2(\Theta)$ measure the initial, first and second boundary errors, respectively, and $\lambda_0, \lambda_1, \lambda_2$ are three hyperparameters for balancing the four terms.

(2) Hard constraint

In this case, we represent the hydraulic head function h as,

$$h(x, y, t; \Theta) = h^*(x, y, t) + d(x, y, t)u(x, y, t; \Theta), \quad (13)$$

where $u(x, y, t; \Theta)$ is the output of the neural network; $h^*(x, y, t)$ is a particular function which exactly satisfies the initial condition and Dirichlet boundary condition, and $d(x, y, t)$ is a function which vanishes on the initial boundary $\Omega \times \{0\}$ and Dirichlet boundary Γ_1 . Due to the introduction of $d(x, y, t)$, it is easy to see that $h(x, y, t)$ automatically satisfies the initial condition and Dirichlet boundary condition, and thus the loss function can be defined as,

$$L(\Theta) = L_{PDE}(\Theta) + \lambda_2 L_2(\Theta), \quad (14)$$

where $L_{PDE}(\Theta)$ and $L_2(\Theta)$ are defined by (12a) and (12d), respectively.

(3) Construction of $h^*(x, y, t)$

$h^*(x, y, t)$ is a particular function which exactly satisfies the initial condition and Dirichlet boundary condition simultaneously, i.e.,

$$h^*(x, y, 0) = g_0(x, y) \quad \forall (x, y) \in \Omega, \quad (15a)$$

$$h^*(x, y, t) = g_1(x, y, t) \quad \forall (x, y, t) \in \Gamma_1 \times (0, T]. \quad (15b)$$

If the initial value and Dirichlet boundary value are chosen as a constant value, $h^*(x, y, t)$ can also be selected as this constant; otherwise, $h^*(x, y, t)$ can be pretrained by a simple fully-connected neural network pretrained by using the initial values and Dirichlet boundary values (Sun et al., 2020).

(4) Construction of $d(x, y, t)$

The simplest choice of $d(x, y, t)$ is the indicator function defined by,

$$d(x, y, t) = \begin{cases} 0, & (x, y, t) \in \Omega \times \{0\} \cup \Gamma_1 \times (0, T] \\ 1, & \text{otherwise} \end{cases}. \quad (16)$$

But it is not smooth and inappropriate to the network-based method. Alternatively, we can choose $d(x, y, t)$ as the distance function from the point (x, y, t) to the initial boundary $\Omega \times \{0\}$ and Dirichlet boundary $\Gamma_1 \times (t_{min}, t_{max}]$, i.e.,

$$d(x, y, t) = \min_{(x', y', t') \in \Omega \times \{0\} \cup \Gamma_1 \times (0, T]} \sqrt{(x - x')^2 + (y - y')^2 + (t - t')^2}. \quad (17)$$

If the domain is regular, the function $d(x, y, t)$ may be constructed explicitly. For example, let $\Omega = [x_{min}, x_{max}] \times [y_{min}, y_{max}]$ with Dirichlet boundary $\Gamma_1 = \{(x, y) : x = x_{min} \text{ or } x_{max}\}$, and initial boundary $\Omega^0 = \{(x, y, t) : t = 0\}$, the function d can be expressed as,

$$d(x, y, t) = \frac{(y - y_{min})(y_{max} - y)t}{(y_{max} - y_{min})^2 T}. \quad (18)$$

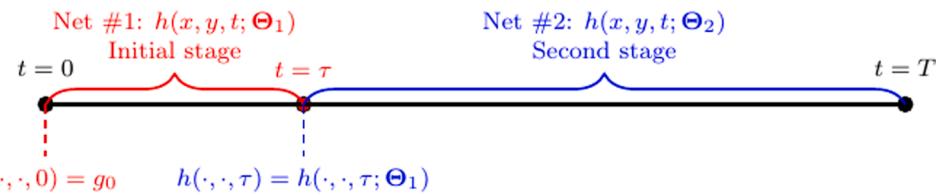


Fig. 4. The snowball-style two-stage training strategy, where the time domain $(0, T]$ is divided into two stages $(0, \tau]$ and $(\tau, T]$. $h(x, y, t; \Theta_1)$ and $h(x, y, t; \Theta_2)$ denote the networks in the initial stage and second stage, respectively, which will be trained in sequence.

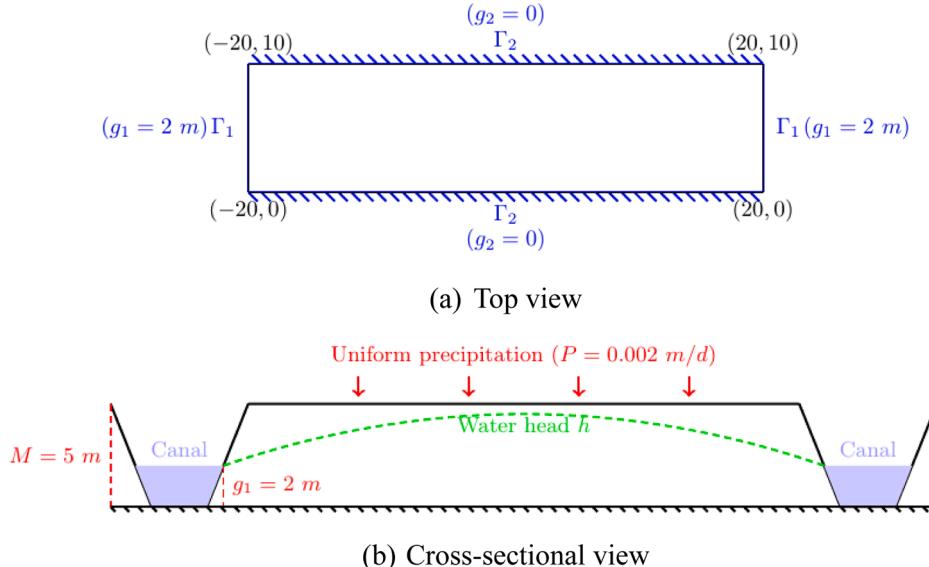


Fig. 5. Problem settings for Test Case 1 that unconfined aquifer with two parallel canals.

For a complex domain Ω , we can pretrain a simple fully-connected neural network $\bar{d}(x, y, t)$ by minimizing the following loss function,

$$L(\bar{d}, d; \Psi) = \sum_{(x, y, t) \in \Omega_D \times [t_{min}, t_{max}]_D} \|\bar{d}(x, y, t; \Psi) - d(x, y, t)\|^2, \quad (19)$$

where Ψ represents the trainable parameters for this network and $d(x, y, t)$ can be precomputed using a certain distance function.

3.5. Training strategy

We can easily get the uniform spatial-temporal sampling points for aquifer without pumping wells. In this case, it is not necessary to use too much sampling points, which makes the training of the neural network relatively easy. Therefore, we here mainly focus on the training strategy for the aquifer with pumping wells. Based on the above discussion, the change process of hydraulic head in this case can be divided into two stages as the initial stage with drastic change of hydraulic head and the second stage with relatively small change of hydraulic head. Precisely, we divide the time domain $(0, T]$ into two subintervals: $(0, \tau]$ and $(\tau, T]$, where τ is the watershed of the two stages which needs to be adjusted empirically or manually. We will train two different networks in two stages in a snowball way, i.e., the first network is trained in the initial stage $(0, \tau]$, and once it is trained well, we use the prediction value of hydraulic head at $t = \tau$ as the initial value of the second stage, and continue to train the second network. To express this idea more clearly, we illustrate the two-stage training strategy in Fig. 4.

4. Testing cases and model evaluation indices

To demonstrate the performance of the proposed GW-PINN for simulating the groundwater problems with different types of aquifer,

five test cases are investigated including confined/unconfined aquifer containing pumping wells, confined aquifer containing multiple pumping wells, and unconfined aquifer with two parallel canals. The details of the test cases and the corresponding evaluation criteria are introduced below.

4.1. Case 1: two parallel canals

In this test, we consider a homogeneous isotropic unconfined aquifer with two canals lying parallel each side with a distance of 40 m. In addition, we set the length of the y direction as 10 m, which forms the simulated domain $\Omega = [-20, 20] \times [0, 10]$ m. The saturated hydraulic conductivity in both x and y directions are $K_{s,x} = K_{s,y} = 0.5$ m/d. The specified yield of unconfined aquifer is $\mu = 0.1$, and the thickness of the aquifer is 5 m. The left and right boundaries are the canals whose water surface height are always 2 m. The upper boundary is recharged by uniform precipitation whose density is 0.002 m/d. That is to say, the source/sink term in Eq. (5) contains only the precipitation term, i.e., $f = P$. The details of problem settings are demonstrated in Fig. 5.

In this case, the steady solution of the hydraulic head can be expressed analytically as,

$$h(x) = \sqrt{4 + \frac{P}{K_s} (400 - x^2)}, \quad (20)$$

where $K_s = K_{s,x} = K_{s,y}$.

4.2. Case 2: single pumping well of the confined aquifer

In this test, we consider the confined aquifer containing a single pumping well ($N_w = 1$ in source/sink term) which is located at the center of the spatial domain $\Omega = [-500, 500] \times [-500, 500]$ m, and has the

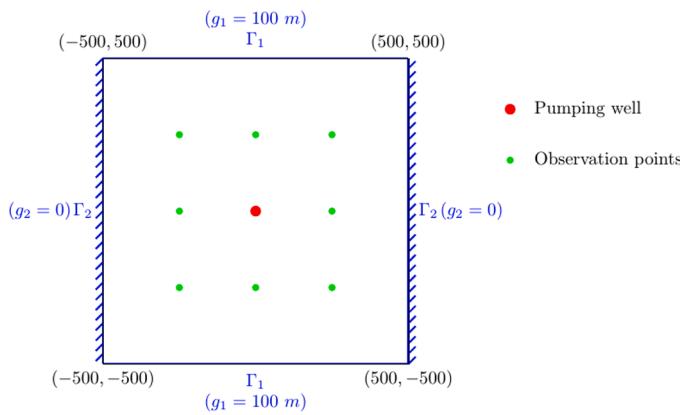


Fig. 6. Problem settings for Test Case 2 that homogenous isotropic confined aquifer containing a single pumping well.

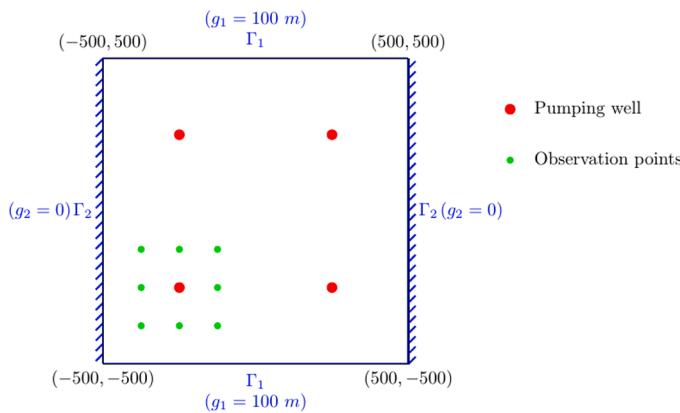


Fig. 7. Problem settings for Test Case 3 that homogenous isotropic confined aquifer containing four pumping wells.

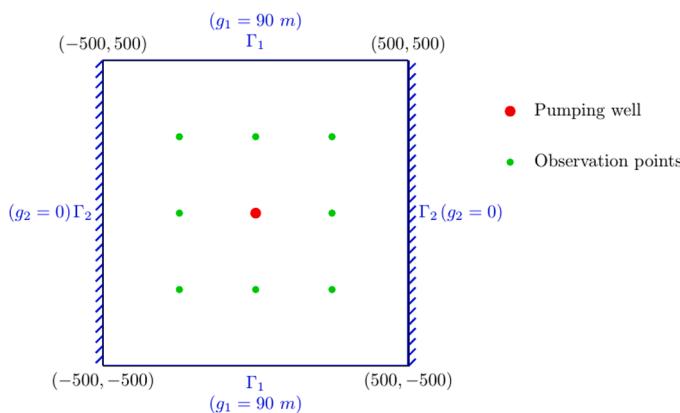


Fig. 8. Problem settings for Test Case 4 that homogenous isotropic unconfined aquifer containing one pumping well.

pumping flow $Q_1 = 10000\text{t/d}$.

We specify the south and north boundaries as the Dirichlet boundaries with constant hydraulic head, i.e. $g_1(x,y, t) = 100 \text{ m}$ on $\Gamma_1 \times (0, T]$ with $\Gamma_1 = \{(x, y): |y| = 500\text{m}\}$, and the east and west boundaries as the non-flux boundaries, i.e., $g_2(x,y, t) = 0$ on $\Gamma_2 \times (0, T]$ with $\Gamma_2 = \{(x, y): |x| = 500\text{m}\}$. Moreover, we set the initial condition as $g_0(x,y, t) = 100 \text{ m}$ on $\Omega \times \{0\}$. The remaining relevant parameters are specified as $M = 3\text{m}$, $\mu_s = 10^{-3}\text{m}^{-1}$, $K_{s,x} = K_{s,y} = 33.33\text{m/d}$. The details of problem settings are demonstrated in Fig. 6.

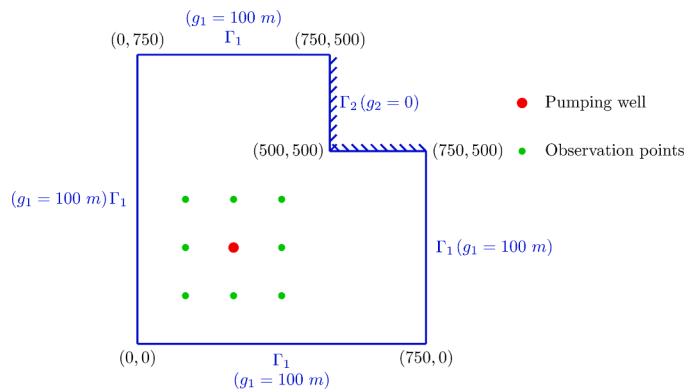


Fig. 9. Problem settings for Test Case 5 that homogenous isotropic confined aquifer with the L-shape domain containing a single pumping well.

4.3. Case 3: confined aquifer containing multiple pumping wells

In this test, we consider the isotropic confined aquifer containing multiple pumping wells. Assume that four pumping wells ($N_w = 4$) are located at $x_1 = (-250, -250)\text{m}$, $x_2 = (250, -250)\text{m}$, $x_3 = (250, 250)\text{m}$, $x_4 = (-250, 250)\text{m}$, and each well has the pumping flow $Q_i = 2500\text{t/d}$, $i = 1, 2, 3, 4$. The other settings, including the geometry and property of aquifer, as well as the initial and boundary conditions, are the same as that of the previous case in 4.2. The details of problem settings are demonstrated in Fig. 7.

4.4. Case 4: Unconfined aquifer containing a single pumping well

This test case is designed to demonstrate the accuracy of GW-PINN in the unconfined aquifer. In this case, we investigate a square homogenous isotropic unconfined aquifer with side length of 1000 m. The specific yield of the unconfined aquifer is set as $\mu_d = 0.1$, and the saturated hydraulic conductivity is set as $K_{s,x} = K_{s,y} = 33.33\text{m/d}$. The left and right sides are specified as non-flux boundaries, i.e., $g_2(x,y, t) = 0$, and the upper and lower sides are assumed to be constant head boundaries with $g_1(x,y, t) = 90\text{m}$. The initial hydraulic head in the simulation region is set to be constant everywhere with $g_0(x,y) = 90\text{m}$. Suppose that a pumping well is located in the center of the region with pumping flow $Q_1 = 40000\text{t/d}$. The well start pumping at the 0-th day, and last to the 20-th day. The problem settings are illustrated in Fig. 8.

4.5. Case 5: confined aquifer with the L-shape domain containing a single pumping well

This test case is designed to demonstrate the accuracy of GW-PINN for irregular domain. In this test, we consider the confined aquifer with the L-shape domain containing a single pumping well ($N_w = 1$ in source/sink term) which is located at (250, 250) m, and has the pumping flow $Q_1 = 2000\text{t/d}$. The shape of the domain and the boundary conditions are shown in Fig. 9. We set the Dirichlet boundaries with a constant hydraulic head 100 m and others as the non-flux boundaries. The initial condition of the whole domain was set as 100 m. The remaining relevant parameters are specified as $M = 3\text{m}$, $\mu_s = 10^{-3}\text{m}^{-1}$, $K_{s,x} = K_{s,y} = 33.33\text{m/d}$.

4.6. Model evaluation indices

The simulation results of MODFLOW or the analytical solution are used as reference results to evaluate the performance of GW-PINN. Two evaluation indices, including the mean absolute error (MAE) and the relative root mean square error (RRMSE) are used to evaluate the errors between GW-PINN and reference results from MODFLOW or the analytic solution as follows,

Table 1
Specific information of observation points for different cases.

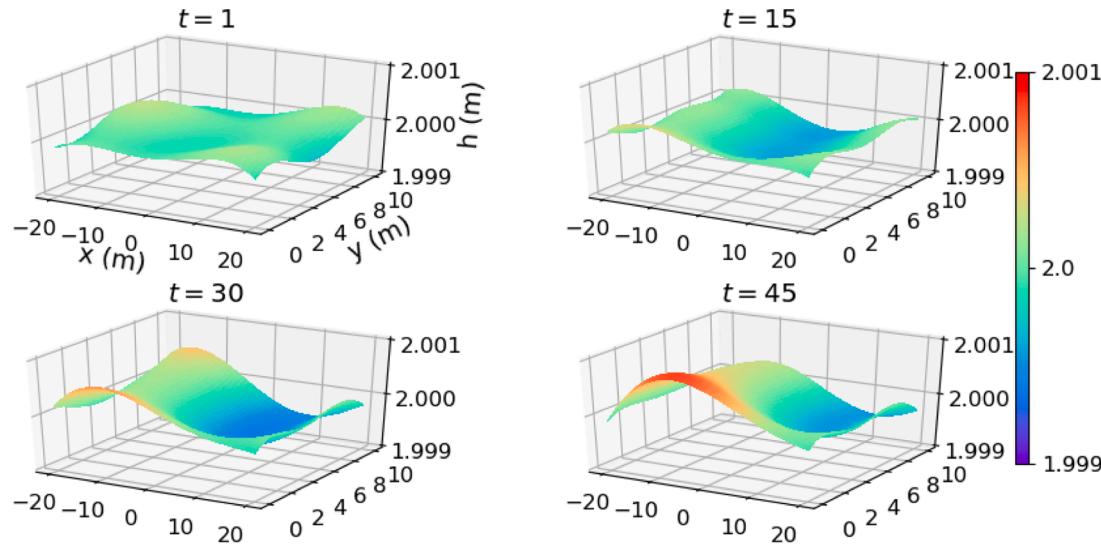
Observation point number	Case 2 and 3		Case 4		Case 5	
	x	y	x	y	x	y
1	-250	-250	-375	-375	150	150
2	0	-250	-250	-375	250	150
3	250	-250	-125	-375	350	150
4	-250	0	-375	-250	150	250
5	250	0	-125	-250	350	250
6	-250	250	-375	-125	150	350
7	0	250	-250	-125	250	350
8	250	250	-125	-125	350	350

$$MAE = \frac{1}{n} \sum_{i=1}^n |h_i - \tilde{h}_i|, \quad (21)$$

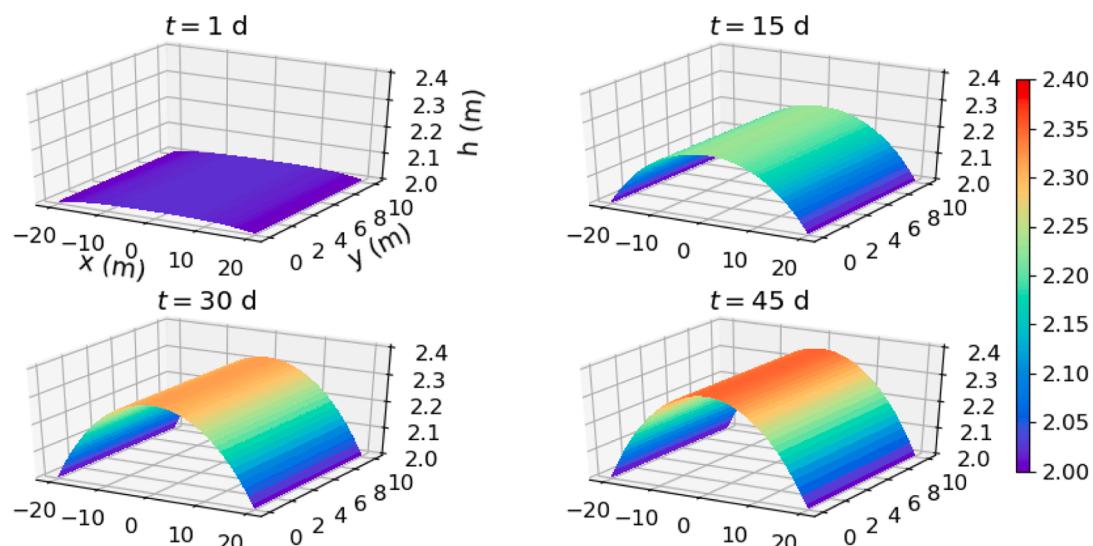
$$RRMSE = \sqrt{\frac{\sum_{i=1}^n (h_i - \tilde{h}_i)^2}{n \times (\frac{1}{n} \sum_{i=1}^n h_i)^2}}, \quad (22)$$

where h_i is the simulated value predicted by GW-PINN; \tilde{h}_i is the simulated value of MODFLOW or the analytic solution; n is the sample number. The MAE value close to 0 and the $RRMSE$ value close to 0% show the accuracy of the model.

For each case, eight observation points were set to calculate the evaluation indices. The coordinates of observation points for different cases are shown in Table 1, and also in Figs. 6–9. The overall errors of all points for Cases 2–5 are also calculated.



(a) Soft constraint



(b) Hard constraint

Fig. 10. 3D surfaces of hydraulic head predicted by our GW-PINN with the soft constraint and the hard constraint at $t = 1, 15, 30, 45$ days for Test Case 1.

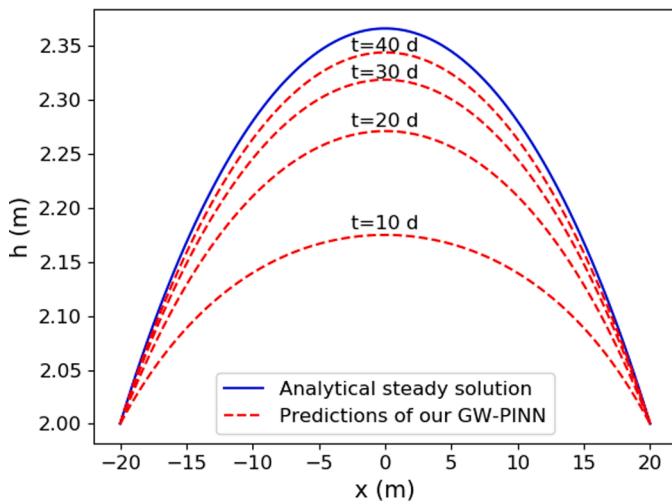


Fig. 11. Predicted hydraulic head at $t = 10, 20, 30, 40$ days, and the analytical steady solution is calculated from Eq. (20).

5. Experimental results

All the experiments are conducted by a single NVIDIA 2080Ti GPU with Pytorch 1.1.0. The simulation results of all test cases will be compared with those produced by the well-known software MODFLOW, except the two parallel canals test case which has the analytical solution. We always use 4 hidden layers (i.e., $D = 4$) and 40 neurons per each hidden layer (i.e., $n_k = 4$) for our GW-PINN except stated otherwise, and the activation function is chosen as the \sin function. The network is trained using Adam optimizer with 2000 epochs followed by LBFGS optimizer with 1000 epochs, and the training process terminates when the absolute difference between the loss values of two adjacent iteration steps is less than a certain threshold value ϵ (we choose $\epsilon = 10^{-8}$). In addition, the parameter s in Eq. (9) is selected as 30. It should be noted that the architecture of GW-PINN, including the number of hidden layers and neurons per each hidden layer, is determined in an empirical way. The training points (x, y, t) are generated by locally-refined spatial sampling and uniform or quasi-uniform temporal sampling; and in the validation or test phase, we generate the input data by uniform sampling for more convenient of visualization.

5.1. Ablation study

In this subsection, we investigate the effect of the loss function, the spatial-temporal sampling strategy and the training strategy on the proposed GW-PINN.

5.1.1. Effect of loss function

As mentioned earlier in Section 3.4, we may use two types of loss function, the soft constraint vs. the hard constraint, to train GW-PINN. Here we study and compare their performance on the training process.

(1) Learning Test Case 1 using the soft constraint and the hard constraint. We first test an unconfined aquifer containing two parallel canals, the detail of problem settings has been described in Test Case 1. In this case, no pumping well is considered in the source term Eq. (5), which makes the change of hydraulic head keeps relatively gentle during the whole evolution process. Therefore, we generate the spatial-temporal sampling points using LHS as the spatial sampling strategy shown in Fig. 2(b) and US as the temporal sampling strategy shown in Fig. 3(a).

The 3D surface of predicted values of hydraulic head by our GW-PINN at $t = 1, 15, 30, 45$ days are presented in Fig. 10, from which we see that numerical performances of the two loss functions are very different. It is observed that the use of soft constraint fails to train GW-

PINN successfully and almost nothing is learned in this case while the use of hard constraint is able to train GW-PINN very well and nice prediction is obtained in this case.

In addition, we find from Fig. 10(b): (1) due to the amount of precipitation, the hydraulic head of the entire surface shows a rising trend; (2) since the hydraulic head are set to be constant on the west and east boundaries, the closer to the center of the area, the faster the hydraulic head rises; (3) the rise of the hydraulic head tends to slow down and remains nearly stable after 45 days due to the volume of water discharged by the canals begins to cover the volume of water recharged by precipitation. The prediction results are well consistent with the hydrological change process under this case.

Further results of quantitative analysis are shown in Fig. 11, where the predictions by our GW-PINN with the hard constraint at different time stamps are shown and compared against the analytical steady-state solution, where the MAE is 0.010 m and the corresponding RRMSE is 0.449%, indicating the good performance of the hard constraint in this case.

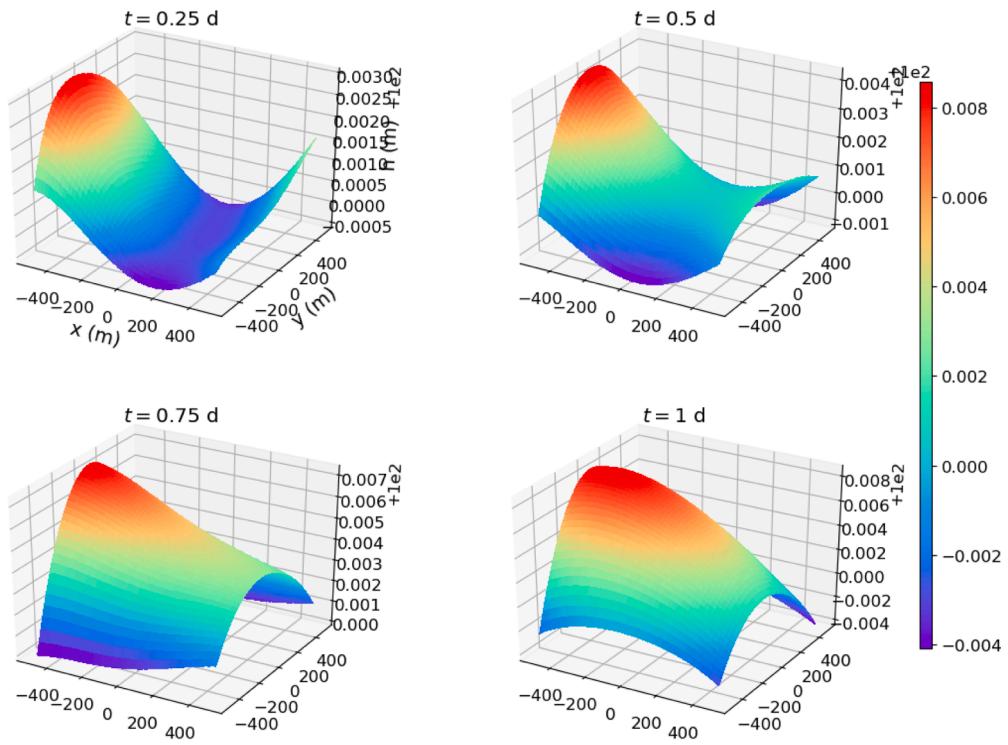
(2) Learning Test Case 2 using the soft constraint and the hard constraint. We next test a confined aquifer containing a single pumping well, and the detailed problem settings are described in Test Case 2. For simplicity, only the initial stage is considered by setting $\tau = 1$ in the snowball-style two-stage training strategy discussed in Section 3.5. In this case, a pumping well is located at the center of the spatial domain, which makes the hydraulic head change dramatically near the pumping well and become slow gradually far away from it during the whole evolution process. Therefore, we recommend spatial-temporal sampling points be generated by using LRS for spatial sampling and LHS for temporal sampling, where the spatial sampling points ($\#\Omega_D \approx 1250$) are shown in Fig. 2(a) and the temporal sampling points ($\#(0, \tau]_D = 50$) shown in Fig. 3(b). The 3D surfaces of predicted values of hydraulic head by our GW-PINN with the soft constraint or hard constraint at 0.25, 0.5, 0.75 and 1 days in the initial stage are presented in Fig. 12. We observe that the similar phenomenon as the previous test. The soft constraint again fails to train GW-PINN while the hard constraint is very suitable to train GW-PINN and produces excellent prediction. It is worth noting that the predictions of GW-PINN are very close to those computed by MODFLOW as demonstrated by the contour maps given in Fig. 13.

5.1.2. Effect of the sampling strategy

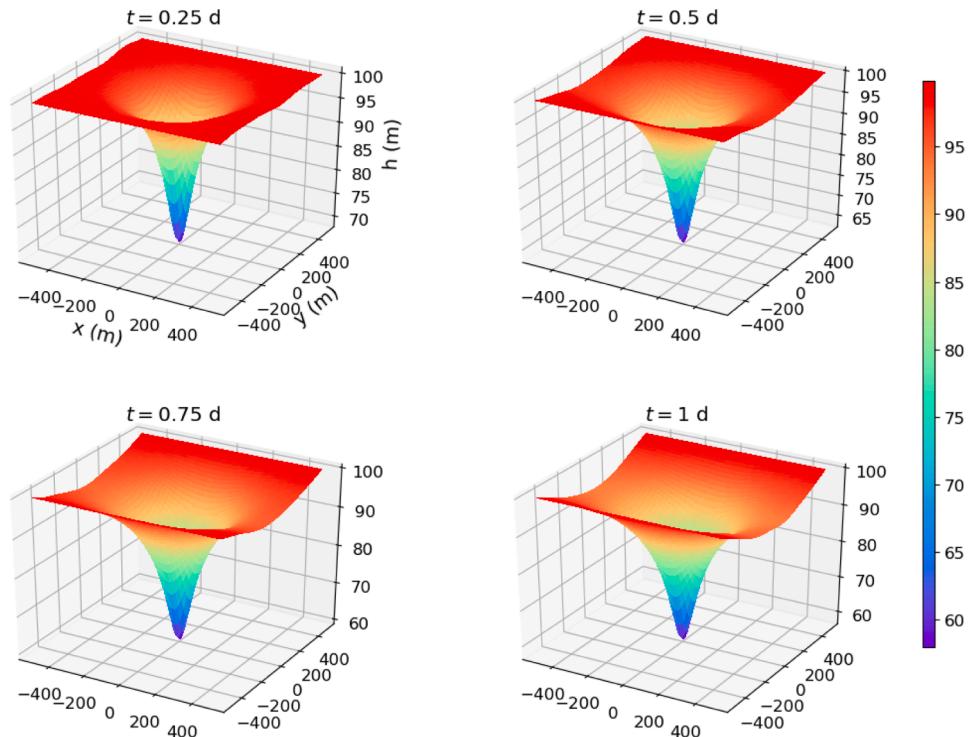
The training quality of GW-PINN depends heavily on the training set, which makes it necessary to choose an effective sampling strategy for spatial-temporal points as discussed in Section 3.3. If there is no pumping well in the aquifer, the sampling method is relatively easy and the uniform or quasi-uniform spatial-temporal strategy can perform well. However, if the pumping wells are included in the aquifer, an appropriate spatial-temporal sampling strategy becomes important. Here we mainly focus on the effect of different sampling strategies on the latter situation.

(1) Spatial sampling. After 50 temporal sampling points are generated by using LHS, we take three different type of spatial sampling strategies, including US, LHS and LRS, to show their effects on the training of our GW-PINN in the initial stage of Test Case 2. Note that the results for Test Case 2 predicted by our GW-PINN using LRS as the spatial sampling strategy, including 3D surface and contour map of the hydraulic head, have been very satisfactory as shown in Figs. 12(b) and 13, where the number of spatial sampling points is 1250 approximately.

Here we first use US and LHS to generate approximately 2500 spatial sampling points respectively (as shown in Fig. 2(a) and (b)), and combine them with the aforementioned temporal sampling points to form the spatial-temporal sampling points, and then feed them into GW-PINN to train it. The corresponding 3D surfaces of predicted hydraulic head are shown in Fig. 14, from which we can see that both US and LHS



(a) Soft constraint



(b) Hard constraint

Fig. 12. 3D surfaces of hydraulic head predicted by GW-PINN using the soft constraint and the hard constraint at $t = 0.25, 0.5, 0.75, 1$ days in the initial stage for Test Case 2.

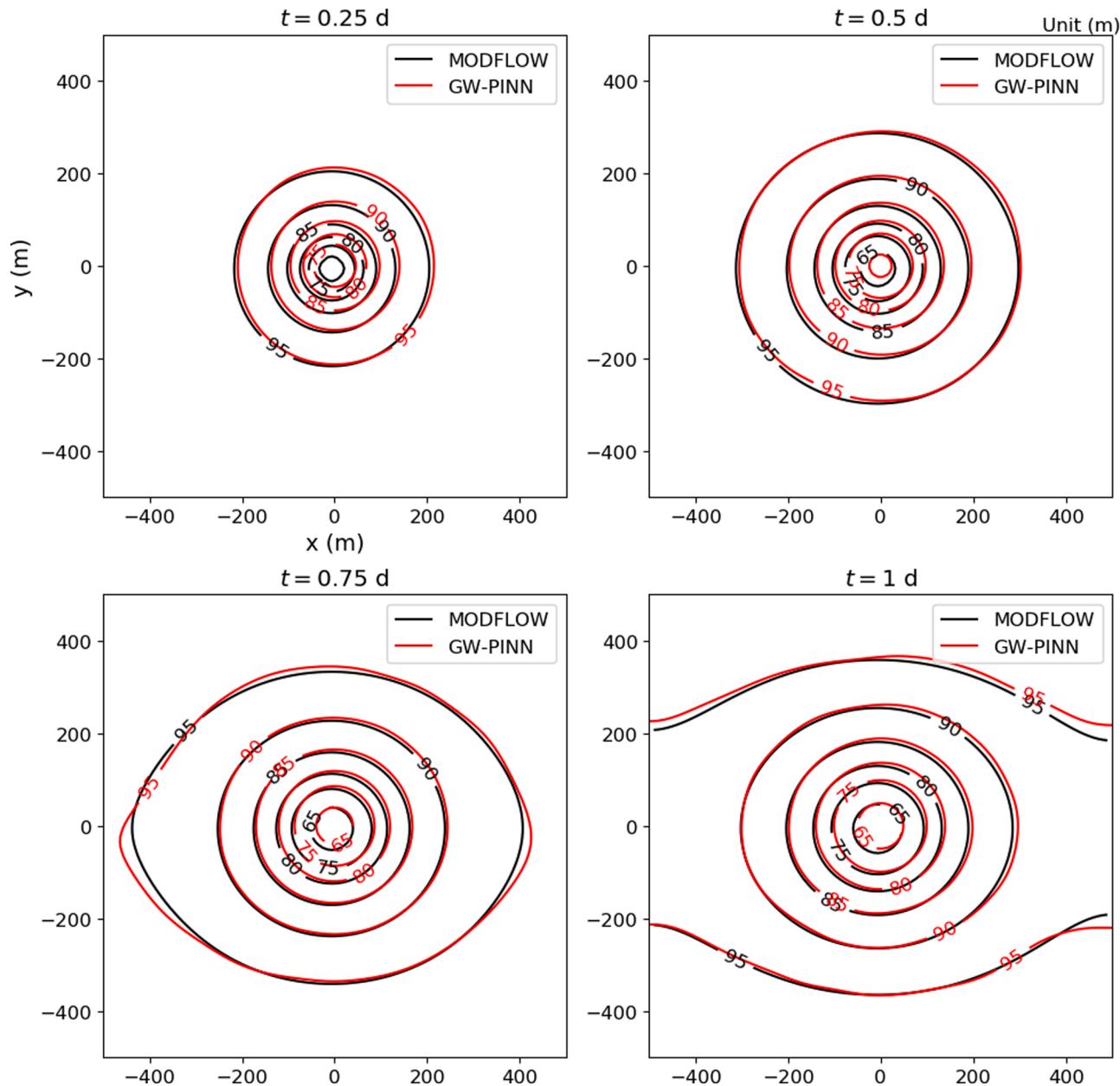


Fig. 13. Contour maps of hydraulic head predicted by GW-PINN with the hard constraint at $t = 0.25, 0.5, 0.75, 1$ days in the initial stage for Test Case 2.

failed to capture the change of hydraulic head near the pumping well accurately, compared to the result of Fig. 12(b). The main reason for this phenomenon is that the main change of hydraulic head is concentrated in the vicinity of the pumping well, and when US or LHS is used as spatial sampling strategy, the sampling points falling in the adjacent area of the pumping well are actually not enough to make GW-PINN to capture the details of the change of hydraulic head, especially when the total number of spatial sampling points is not large.

Next, we use US and LHS to generate more spatial sampling points (approximately 6000) to train GW-PINN again. The predicted results are shown in Figs. 15 and 16, from which we can observe that, along with the increase of spatial sampling points, the 3D surfaces visually seem to be greatly improved (especially for LHS), and close to those shown in Fig. 12(b). However, they are still inaccurate and the corresponding contour maps are far from those calculated by MODFLOW. Although we have reasons to believe that with the continuous increase of spatial sampling points, the prediction ability of GW-PINN will be further improved, there are also other troubling problems, for example, the

sharp increase in the amount of training data will make the training of GW-PINN slowly, even intractable.

A natural solution for this issue is probably to use LRS to generate locally refined spatial sampling points (as shown in Fig. 2(c)), where most of them are concentrated near the pumping well, and gradually become sparse when further away from the well. As can be seen from the predicted results shown in Fig. 12(b), although the number of spatial sampling points ($\#\Omega_D \approx 1250$) is only about a fifth of those by US and LRS ($\#\Omega_D \approx 6000$), the predictions by our GW-PINN using LRS are much more accurate.

In summary, we recommend using the locally-refined strategy for spatial sampling if the aquifer contains pumping wells. Otherwise, uniform or quasi-uniform spatial sampling strategy is apt for training the GW-PINN.

(2) *Temporal sampling.* We continue to discuss the effect of temporal sampling strategy on the training of GW-PINN. By fixing the LRS as the spatial sampling strategy, we use two different temporal sampling

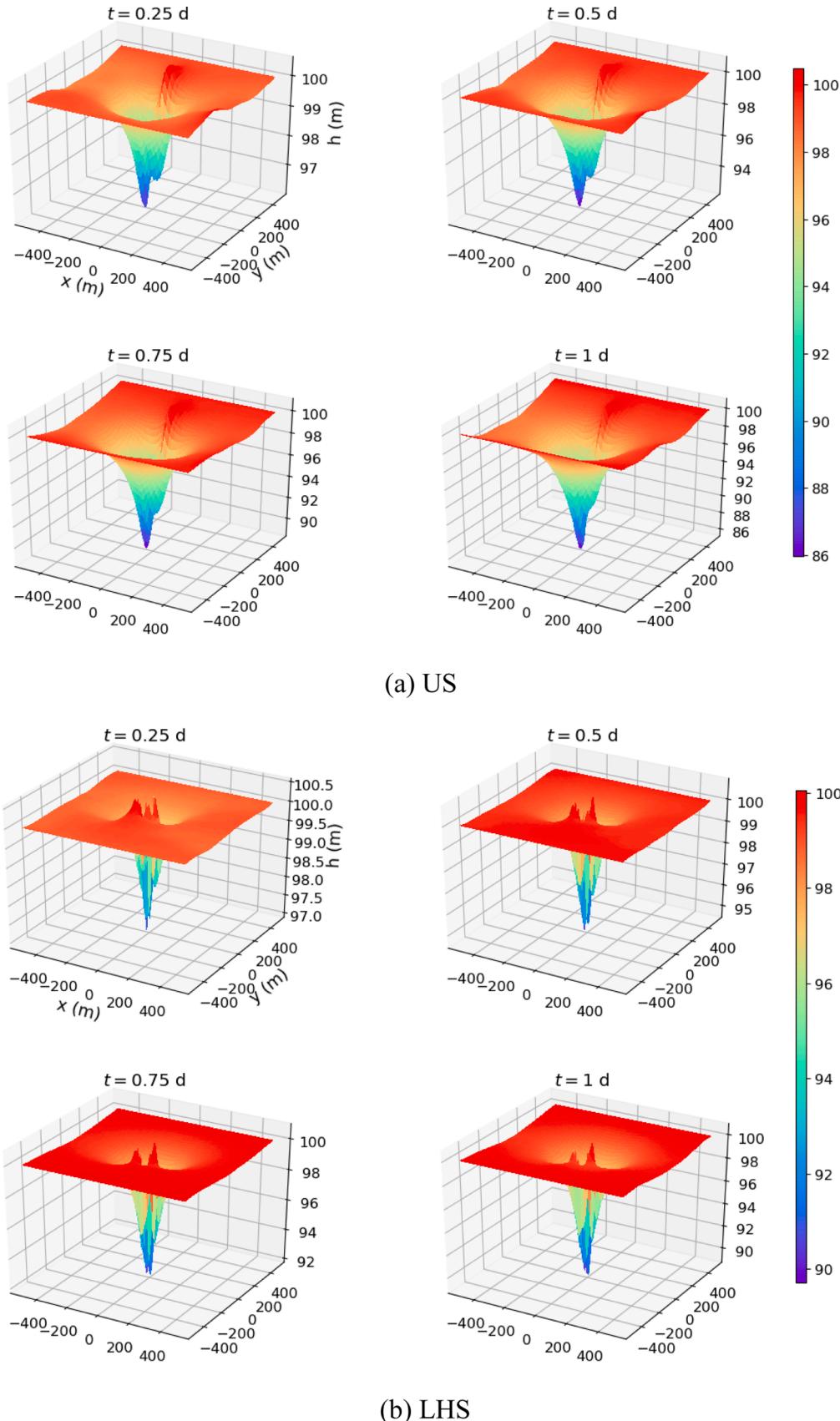
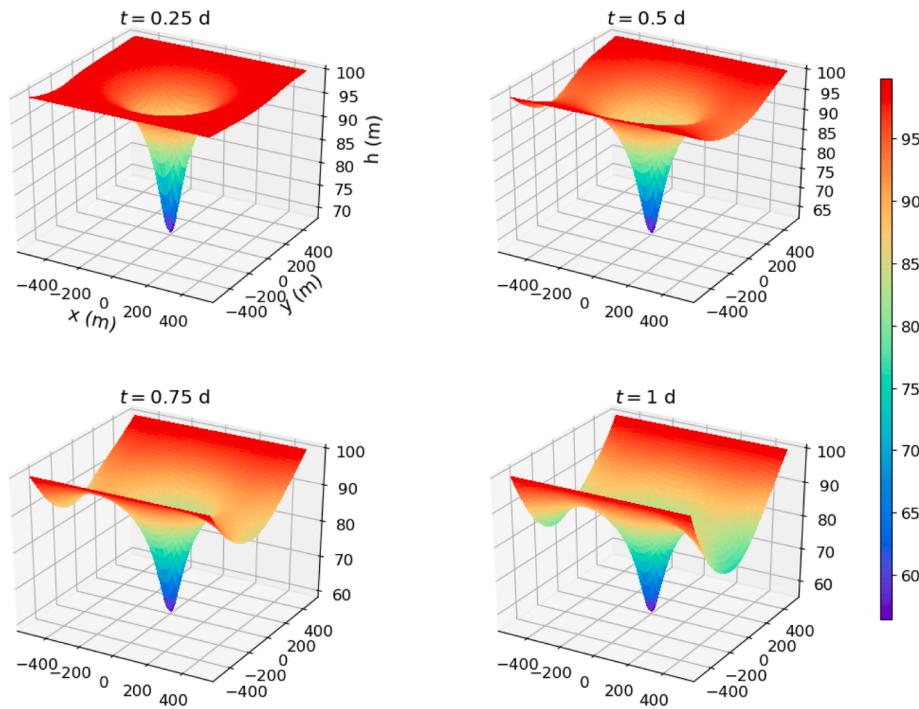
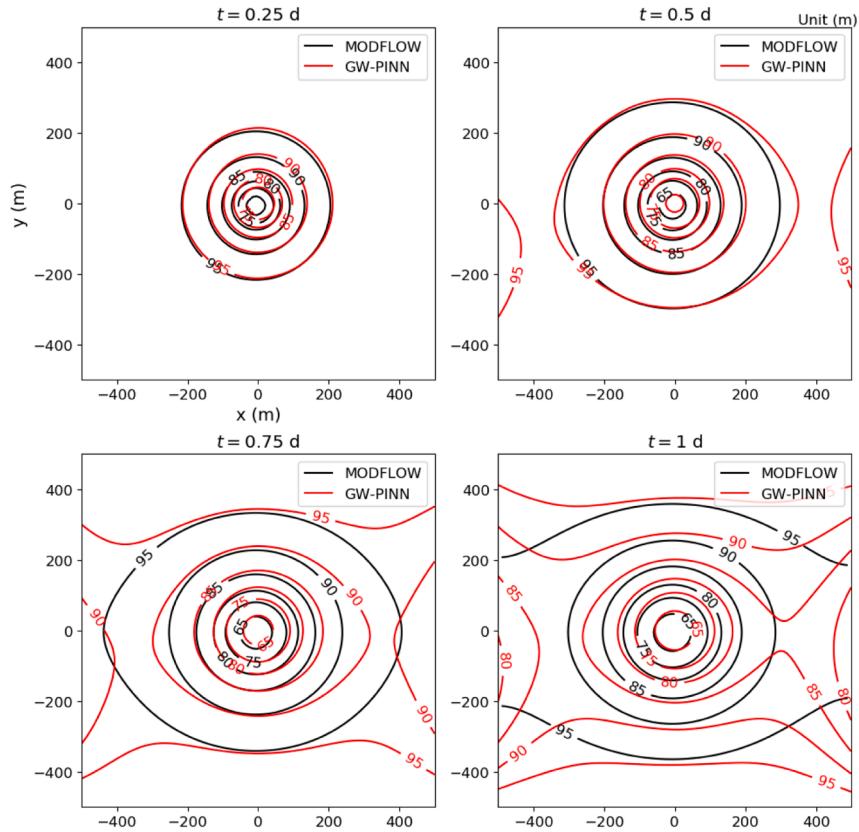


Fig. 14. 3D surface of hydraulic head predicted by our GW-PINN at different time-stamps in the first stage for Test Case 2, where US or LHS is used to generate the spatial sampling points ($\#\Omega_D \approx 2500$) and LHS is used to generate the temporal sampling points ($\#(0, \tau_D) \approx 50$).

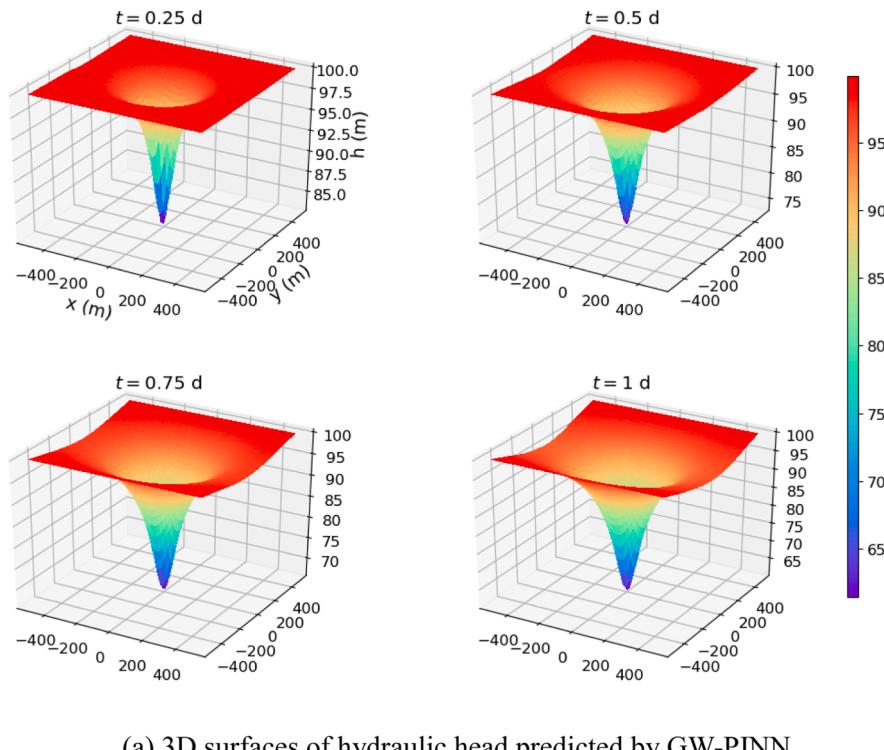


(a) 3D surfaces of hydraulic head predicted by GW-PINN

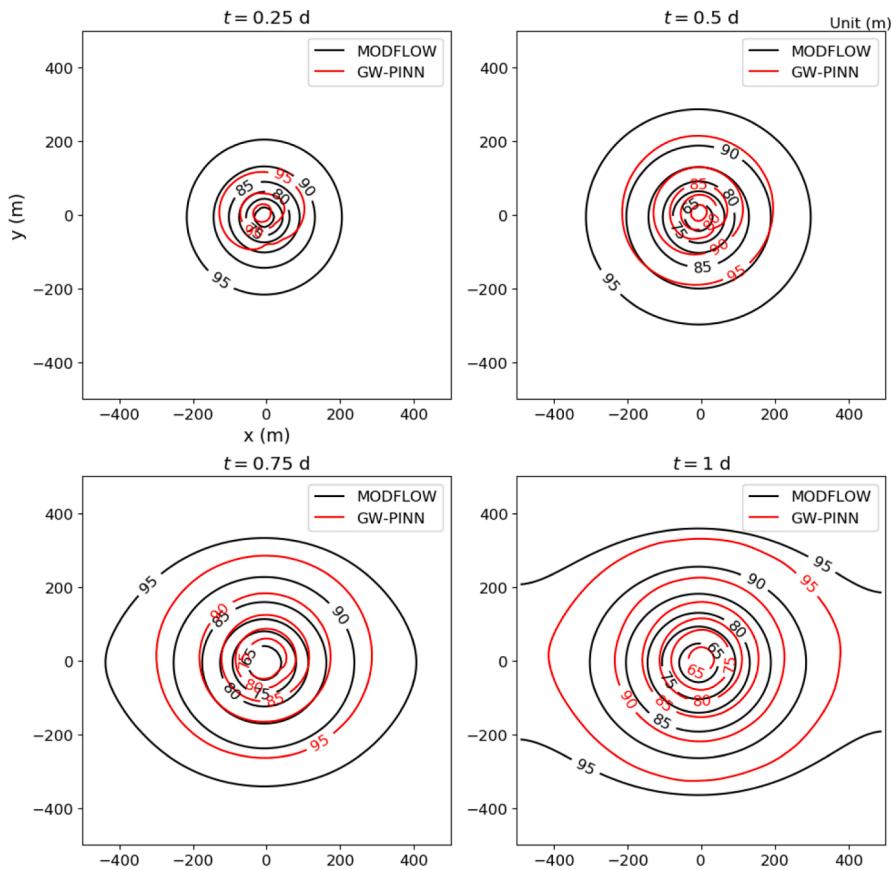


(b) Contour maps of hydraulic head predicted by GW-PINN and MODFLOW

Fig. 15. 3D surfaces and contour maps of hydraulic head predicted by our GW-PINN at different times in the first stage for Test Case 2, where US is used to generate the spatial sampling points ($\#\Omega_D \approx 6000$) and LHS is used to generate the temporal sampling points ($\#(0, \tau]_D \approx 50$).

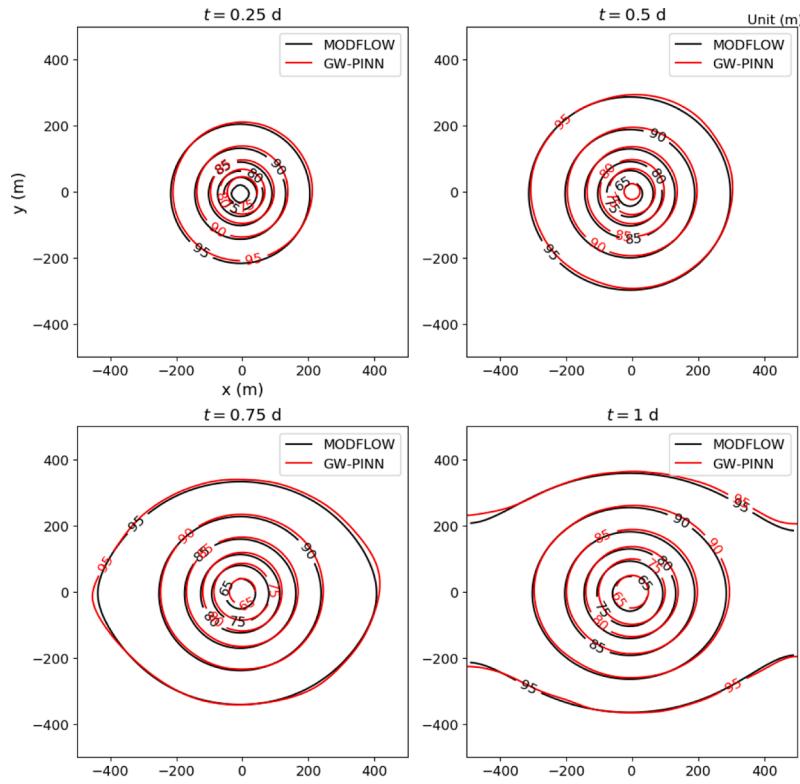


(a) 3D surfaces of hydraulic head predicted by GW-PINN

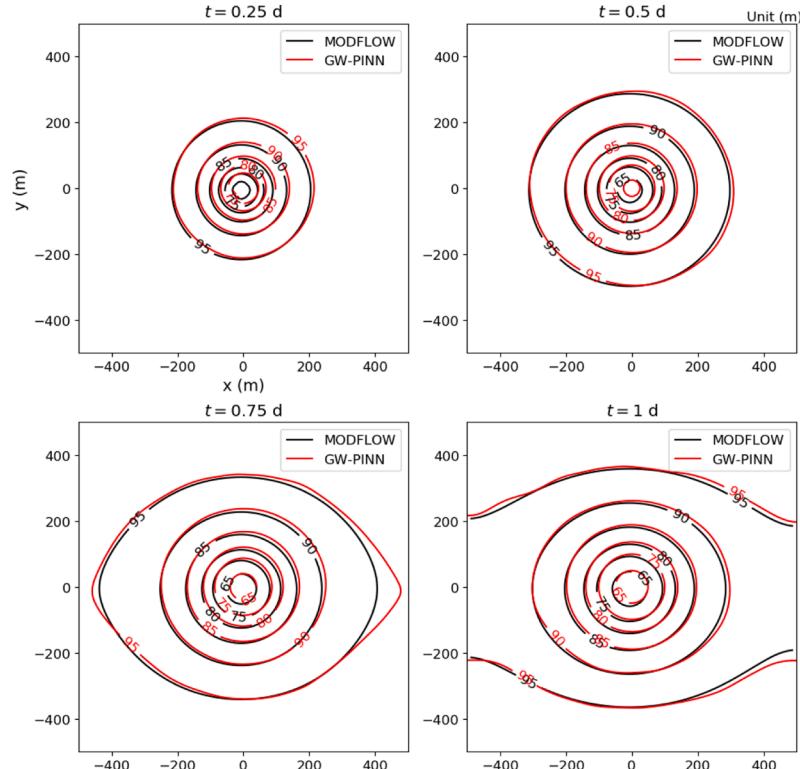


(b) Contour maps of hydraulic head predicted by GW-PINN and MODFLOW

Fig. 16. 3D surfaces and contour maps of hydraulic head predicted by our GW-PINN at different times in the first stage for Test Case 2, where LHS is used to generate the spatial sampling points ($\#\Omega_D \approx 6000$) and LHS is used to generate the temporal sampling points ($\#(0, \tau]_D \approx 50$).



(a) US for temporal sampling



(b) LRS for temporal sampling

Fig. 17. Contour maps of the hydraulic head predicted by our GW-PINN at different times in the first stage, where US or LRS is used to generate the temporal sampling points ($\#(0, \tau]_D \approx 50$), and LRS to generate the spatial sampling points ($\#\Omega_D \approx 1250$).

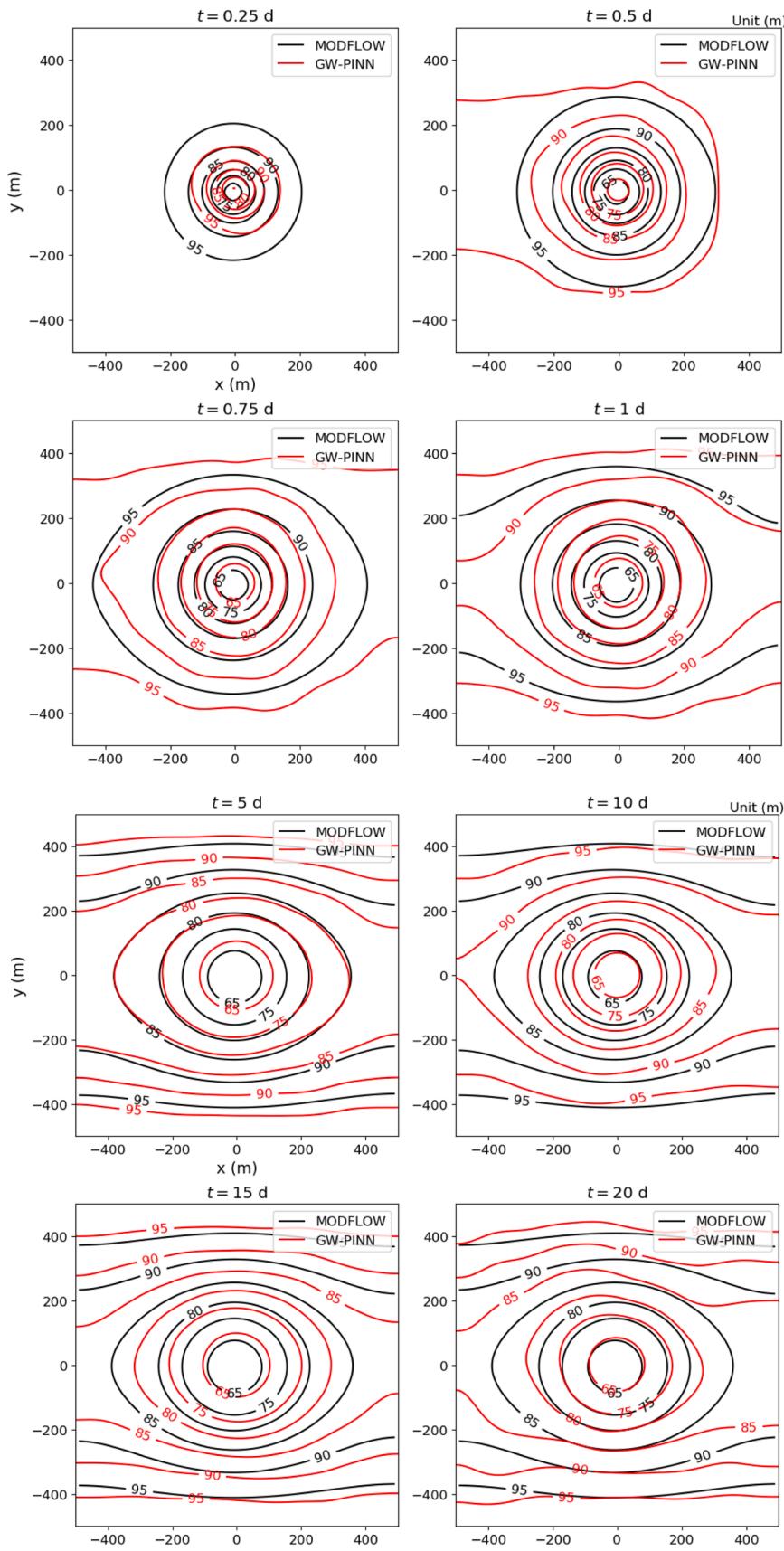


Fig. 18. Contour maps of the hydraulic head predicted by our GW-PINN at different times for Test Case 2, where the global training strategy is adopted.

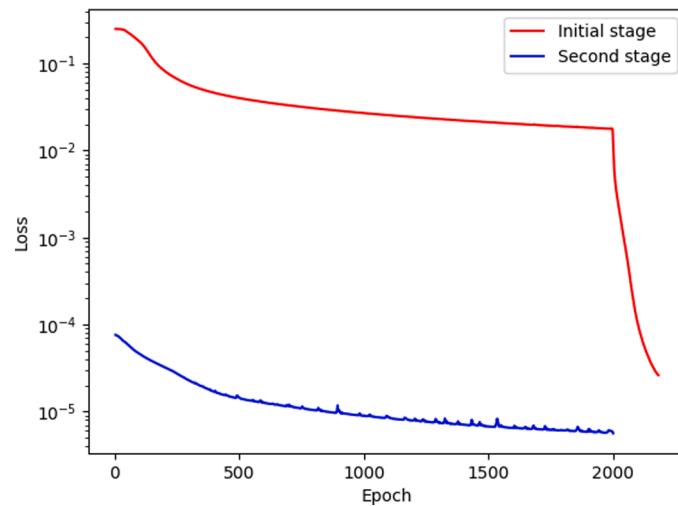


Fig. 19. Loss curves at two stages. **Fig. 20.** 3D surfaces and contour maps of hydraulic head predicted by our GW-PINN at different times in the second stage for Test Case 2, where the snowball-style two-stage training strategy is used.

strategies, including US and LRS, to generate temporal sampling points (as shown in Fig. 3(a) and (c)), where the ratio of LRS is chosen as $r = 1.04$ here. Contour maps of hydraulic head predicted by GW-PINN using US or LRS as temporal sampling strategy are shown in Fig. 17, from which we see that both temporal sampling strategies work similarly on the predictive performances of our GW-PINN. It can be concluded that the training of our GW-PINN is insensitive to the temporal sampling strategies.

5.1.3. Effect of training strategy

Groundwater equation is a time evolution equation, and it is usually necessary to carry out long-term simulation for groundwater simulation. A naive “global” way is to train GW-PINN in the whole spatial-temporal domain. To capture the change of the hydraulic head at different levels, enough temporal sampling points are needed to train GW-PINN when the spatial sampling strategy is fixed. The simulation results predicted by our GW-PINN using the global training strategy are shown in Fig. 18, from which we see that although the overall trend of hydraulic head has been learned, the details are not sufficiently accurate. Along with the increase of temporal sampling points, the predictive ability of the GW-PINN could be further improved, on the other hand, this will lead to a sharp increase of training data, which make the training more difficult.

A better way to train the GW-PINN is a snowball-style shown in Section 3.5, which is implemented by first dividing the temporal domain of Eq. (1) into two stages, and then train the network in each stage in sequence, see Fig. 4 for the details of the overall training process. The advantage of such a strategy is to capture the drastic change of the hydraulic head quickly and accurately in the initial stage by using as few temporal sampling points as possible, and in the following stage, the change of hydraulic head becomes gradually stable, which makes the training process relatively easier. This can be verified by the loss curve shown in Fig. 19, from which we observe that the overall training cost is concentrated in the initial stage, and once the network in the initial stage is trained well, the training of the network in the second stage becomes very smooth, where the whole loss value is maintained at a very low level. It only needs about 2000 Adam iterations to complete the training. The simulation results predicted by our GW-PINN, including the 3D surfaces and contour maps at different times in the second stage, are shown in Fig. 19, where the contour maps predicted by GW-PINN are very close to those calculated by MODFLOW. Furthermore, we also conduct a quantitative analysis at eight observation points listed in Table 1 at $t = 20$ days, and the results are reported in Table 2, where the MAE in Test Case 2 at 20 d is 0.43 m and the RRMSE is only 0.59%,

which are clearly within the acceptable range.

5.2. Performance results for Test Cases 3 and 4

To further demonstrate the performance of our GW-PINN, we will test a confined aquifer containing multiple pumping wells (see the problem settings for Test Case 3) and an unconfined aquifer containing a single pumping well (see the problem settings for Test Case 4). Based on the ablation studies, we use the hard constraint as the loss function, LRS strategy to generate spatial sampling points, LHS strategy to generate temporal sampling points, and the snowball-style training two-stage strategy with setting $\tau = 1$ for training in all the following numerical tests.

Let us first consider Test Case 3, simulation results predicted by our GW-PINN, including the 3D surfaces and contour maps at different times are demonstrated in Fig. 21 for the initial stage and Fig. 22 for the second stage. The corresponding quantitative evaluation results are reported in Table 2. From Figs. 21 and 22, it is easy to find that the change process of hydraulic head in this case looks like that of Case 2, but four depression funnels were formed near the four wells. Furthermore, contour maps of hydraulic head demonstrate that the simulation results predicted by our GW-PINN are very close to those calculated by MODFLOW at different times. We can see from Table 2 that the MAE in Test Case 3 at 20 days is 0.1 m and the corresponding RRMSE is 0.14 %, where the error is within the acceptable range.

Next, we consider Test Case 4, which is a relatively complicated case, to show the capacity of our GW-PINN. Different from the networks setting in other cases, we add one more layer in the fully-connected network, i.e., 5 hidden layers (i.e., $D = 6$) and 40 neurons per each hidden layer (i.e., $n_k = 40$) for the architecture of our GW-PINN. It can be seen from the 3D surfaces in Figs. 23 and 24 that the change process of the hydraulic head in this case is similar to that of Case 2. Furthermore, the corresponding contour maps show that the simulation results predicted by our GW-PINN are very close to those calculated by MODFLOW. As reported in Table 2, the MAE in Test Case 4 at 20 days is 0.17 m and the corresponding RRMSE is 0.20%. The error is within the acceptable range.

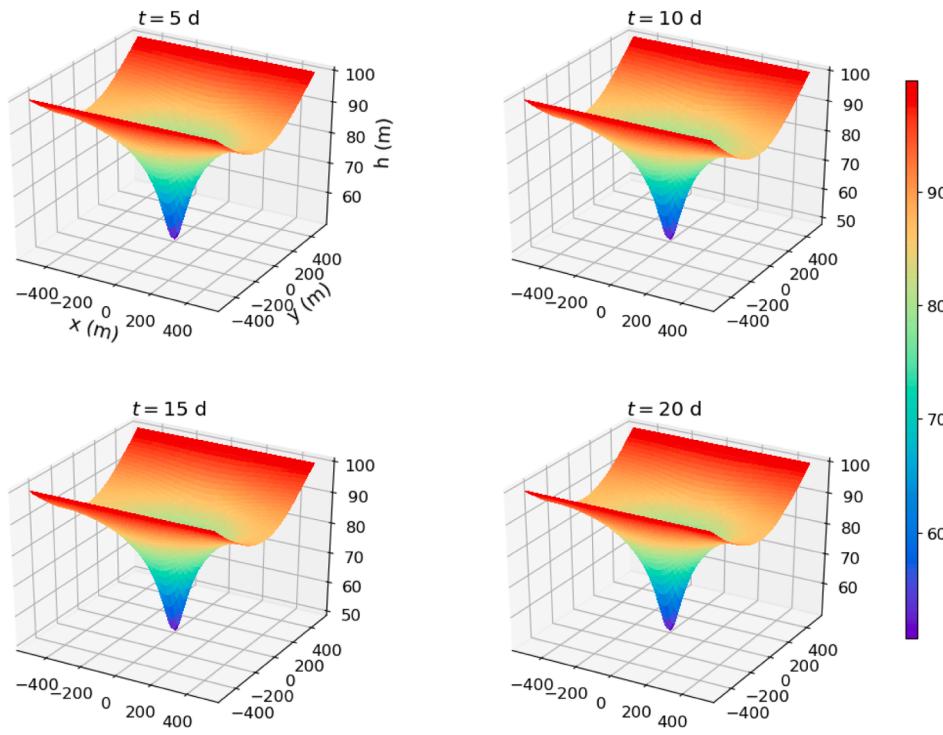
5.3. Performance results for Test Case 5 with the L-shape domain

For Test Case 5 with the L-shape domain, simulation results predicted by our GW-PINN, including the 3D surfaces and contour maps at different times are demonstrated in Fig. 25. The corresponding quantitative evaluation results are reported in Table 2. The contour maps of hydraulic head demonstrate that the simulation results predicted by our GW-PINN are very close to those calculated by MODFLOW at different times. We can see from Table 2 that the MAE in Test Case 5 at 1 day is 0.052 m and the corresponding RRMSE is 0.058%, where the error is within the acceptable range. This case demonstrates the accuracy of GW-PINN for irregular domain, which has possible application for real-world cases with complex geometry.

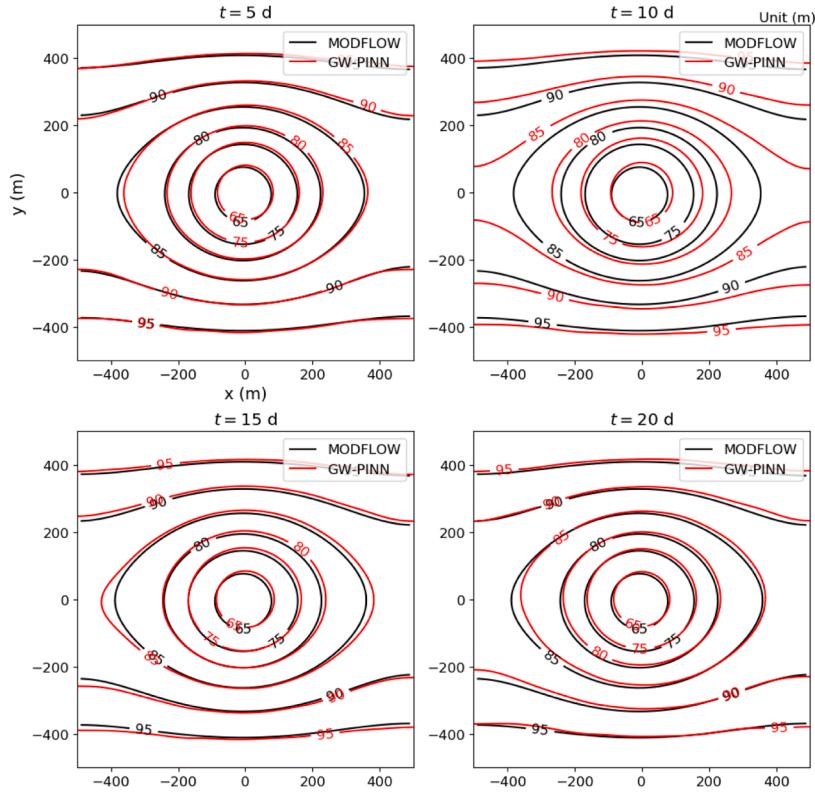
6. Model discussion

6.1. Model application as a surrogate model for parameterized groundwater flow

In the previous sections, we investigate GW-PINN for solving different types of groundwater flow, and satisfactory predicted results are obtained compared to the numerical results provided by MODFLOW. However, the potential advantages of the network-based method exist that it can serve as a surrogate model for solving parameterized PDEs. For example, we reconsider case 2 by extending the fixed pumping capacity into variable ones. In other words, the pumping capacity in Eq. (5) is now a parameter in a certain range $[Q_{min}, Q_{max}]$, here we assume $Q_{min} = 5000 \text{ t/d}$ and $Q_{max} = 10000 \text{ t/d}$. This scenario is very useful in the



(a) 3D surfaces of hydraulic head predicted by GW-PINN



(b) Contour maps of hydraulic head predicted by GW-PINN and MODFLOW

Fig. 20. 3D surfaces and contour maps of hydraulic head predicted by our GW-PINN at different times in the second stage for Test Case 2, where the snowball-style two-stage training strategy is used.

Table 2

Computed hydraulic head of each observation point and values of evaluation indices of Test Cases 2-5.

Case number	Mesh scheme	Observation point number								MAE(m)	RRMSE(%)	Overall error	
		1	2	3	4	5	6	7	8			MAE(m)	RRMSE(%)
Case 2	MODFLOW	87.82	83.93	88.20	80.42	81.22	88.09	84.56	88.44	/	/	0.47	1.07
	GW-PINN	88.60	84.62	88.21	80.88	80.80	88.02	84.10	87.87	0.43	0.59		
Case 3	MODFLOW	93.92	92.92	93.97	88.97	89.07	87.76	86.78	87.82	/	/	0.21	0.67
	GW-PINN	93.86	92.92	93.86	88.99	88.92	87.68	86.60	87.61	0.10	0.14		
Case 4	MODFLOW	88.50	87.98	88.55	87.50	87.61	88.54	88.06	88.59	/	/	0.21	0.43
	GW-PINN	88.37	87.85	88.37	87.38	87.39	88.37	87.85	88.37	0.17	0.20		
Case 5	MODFLOW	97.58	96.33	97.28	96.33	95.98	97.28	95.97	96.82	/	/	0.056	0.155
	GW-PINN	97.62	96.38	97.26	96.38	95.89	97.25	95.90	96.76	0.052	0.058		

agricultural water management, because the relevant managers sometimes need to monitor the influence of the pumping capacity on the groundwater level. To tackle this problem, we can use GW-PINN to solve it once and for all. To this aim, we need to make a slight modification on the original GW-PINN, that is, to use (x, y, t, Q) instead of (x, y, t) as the input of the network. The spatial-temporal-capacity sampling point set is generated as follows:

$$\Omega_D^{T,Q} = \Omega_D \times (0, T]_D \times [Q_{min}, Q_{max}]_D, \quad (23)$$

where $[Q_{min}, Q_{max}]_D$ denotes the uniform sampling points in $[Q_{min}, Q_{max}]$. The loss functions are the same as those in Eq. (12). To avoid the limitation of insufficient memory, the mini-patch training strategy is adopted and only Adam optimizer is used in the whole training process. Once the training is finished (about 200 epochs), we can predict the groundwater level at arbitrary spatial and temporal points in the solution domain for any pumping capacity. The corresponding predicted results are demonstrated in Figs. 26–29, where the pumping capacity Q is chosen as 5550, 6575, 7810 and 8920 t/d, respectively. It should be noted that such values are not appeared in the training set $[Q_{min}, Q_{max}]_D$, and the results of both 3D surfaces and contour maps are very satisfactory, compared with the numerical solutions provided by MODFLOW using very fined meshes. In this case, there are 1250 spatial sampling points, 50 temporal sampling points and 50 points for pumping capacity with a uniform interval of 100 t/d. The training time is about 60 min.

6.2. Model advantages and disadvantages

Compared with current proposed machine learning methods to solve the groundwater flow equation (Wang et al., 2020; Xu et al., 2021), GW-PINN takes the advantage in that it can solve the groundwater flow equation containing source/sink terms directly when constrained only by residual losses of PDEs, while no label data are needed. The overall errors are fairly well (Table 2), showing high accuracy. The input data are (x, y, t) , which are unsupervised and can be obtained in a random and flexible way. This is meaningful for accurately obtaining the large hydraulic gradient adjacent to the source/sink terms, e.g., wells, by using the locally refined sampling strategy. This sampling method is similar to those numerical models by using refined grids adjacent to wells. The hard constrain is proposed in this study to keep the output being satisfied by the initial and boundary conditions, which owns more appropriated physical meaning. This method avoids the over fit problem encountered by those data-drive machine learning methods. Another advantage of GW-PINN is that it can be used as a surrogate model, making the prediction task be easy and efficient. The prediction takes ignorable computational cost with the well-trained GW-PINN and no simulation experience is required, which is beneficial for those staff of water resources management agency. No meshes are used in GW-PINN while only sampling points without topological structure are used, which is easy to carry out.

It should be noted that GW-PINN is not proposed to compete with the traditional numerical methods. One major limitation of these network-based models including GW-PINN to solve PDEs is the computational

cost during the training period when applying it to a real-world case, especially with large area domain and more complex boundary and initial conditions. In view of the particularity of the point source problem, the spatial locally-refined sampling strategy we adopted greatly reduces the training points without affecting the final effect, so as to effectively improve the training process. Nevertheless, further studies, including more effective constraints and more advanced training strategies, should be investigated.

7. Conclusions

In this paper, we have proposed a deep learning method “GW-PINN” for solving groundwater flow equations with source/sink terms from a new perspective. The hard constraint and spatial-temporal sampling strategy as well as the training strategy are proposed and investigated through a series of ablation studies. The major conclusions are as follows

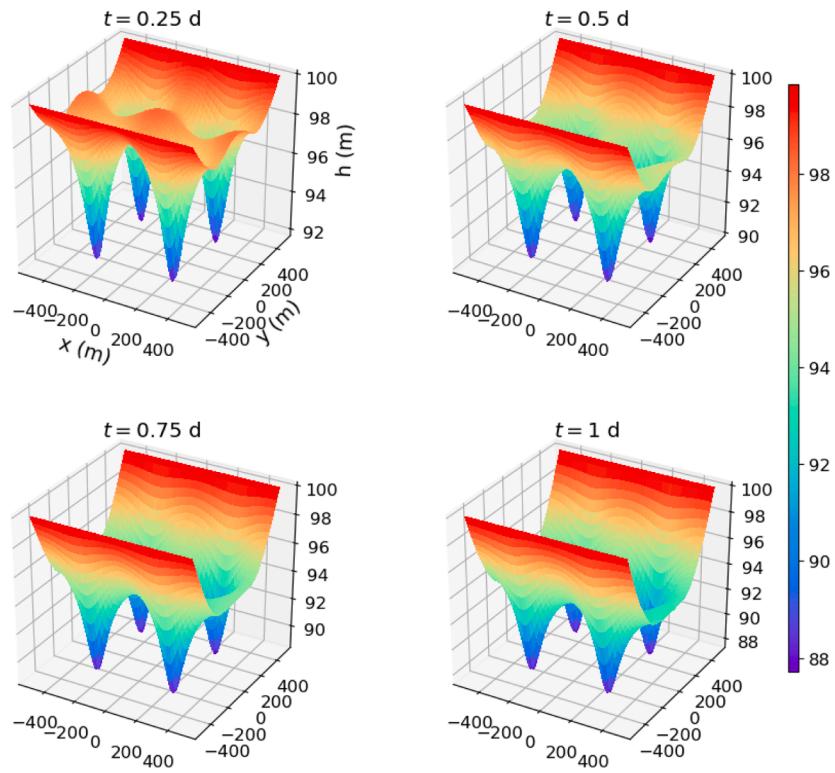
- (1) The proposed GW-PINN is valid for capturing the hydraulic head of groundwater flow, especially for those with local source/sink terms.
- (2) The loss function using the hard constraint is more suitable to train GW-PINN and can produce more accurate prediction than the soft constraint.
- (3) The proposed locally refined spatial sampling (LRS) strategy can generate more accurate results with much fewer sampling points than the uniform sampling (US) strategy or lattice hypercube sampling (LHS) strategy.
- (4) The temporal sampling strategy owns slight impact on the predictions of GW-PINN.
- (5) The proposed snowball-style two-stage training strategy is more efficient to capture the drastic change of the hydraulic head than using the global training strategy.
- (6) GW-PINN can be used as a surrogate model for parameterized groundwater flow prediction.
- (7) Future studies can focus on increasing the training efficiency to extend the application of GW-PINN for more complex and larger area groundwater flow prediction.

CRediT authorship contribution statement

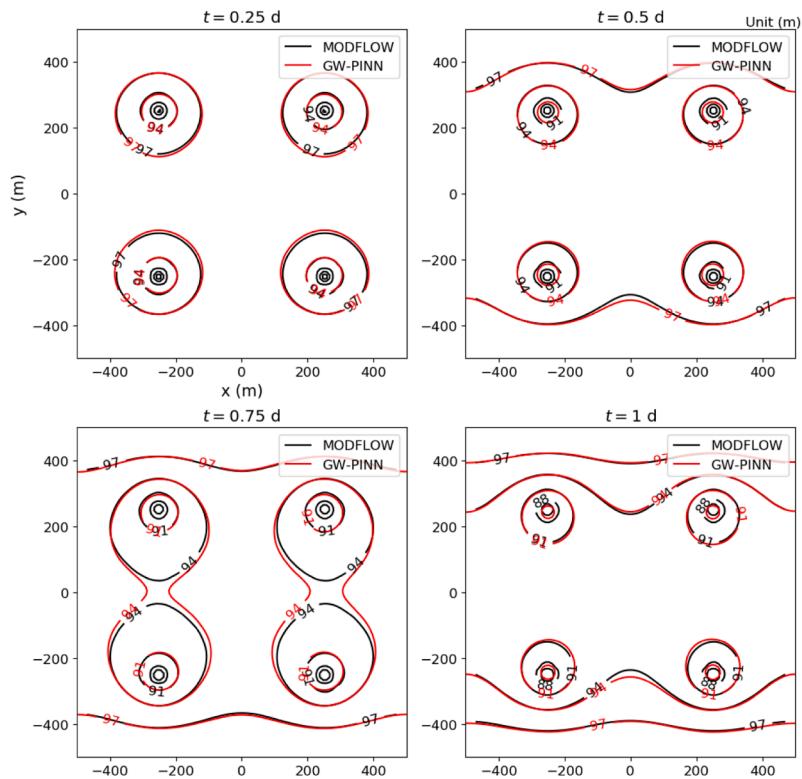
Xiaoping Zhang: Methodology, Validation, Writing – original draft.
Yan Zhu: Conceptualization, Methodology, Writing – review & editing.
Jing Wang: Methodology, Writing – review & editing. **Lili Ju:** Methodology. **Yingzhi Qian:** Writing – review & editing. **Ming Ye:** Writing – review & editing. **Jinzhong Yang:** Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

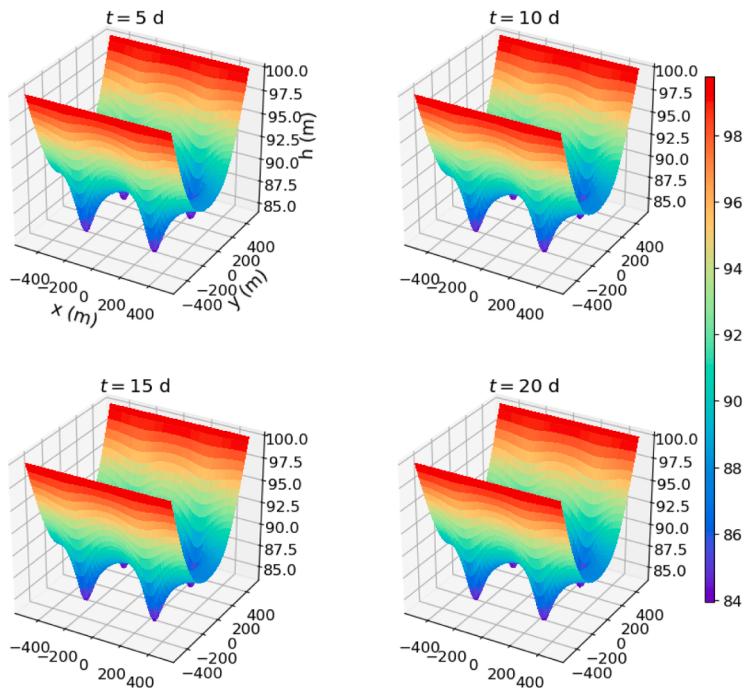


(a) 3D surfaces of hydraulic head predicted by GW-PINN

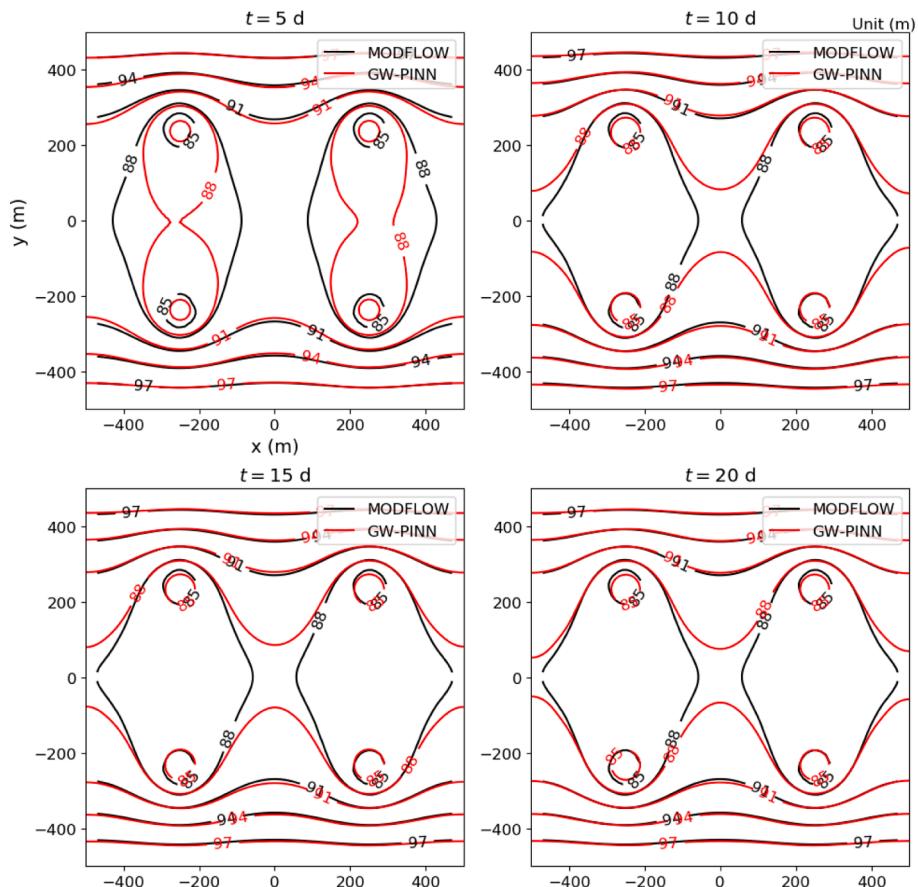


(b) Contour maps of hydraulic head predicted by GW-PINN and MODFLOW

Fig. 21. 3D surfaces and contour maps of the hydraulic head predicted by our GW-PINN at different times in the initial stage for Test Case 3.

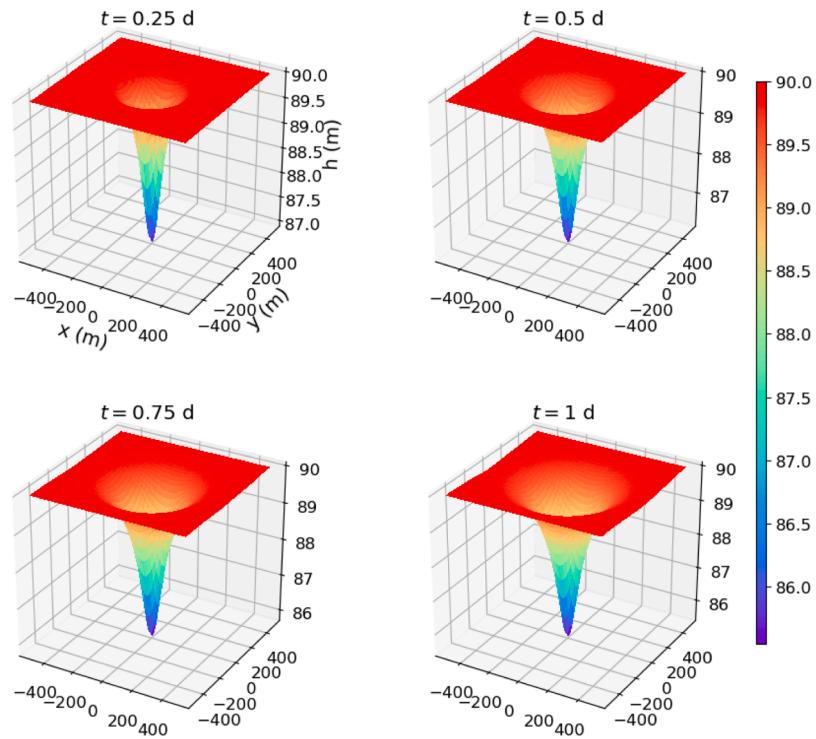


(a) 3D surfaces of hydraulic head predicted by GW-PINN

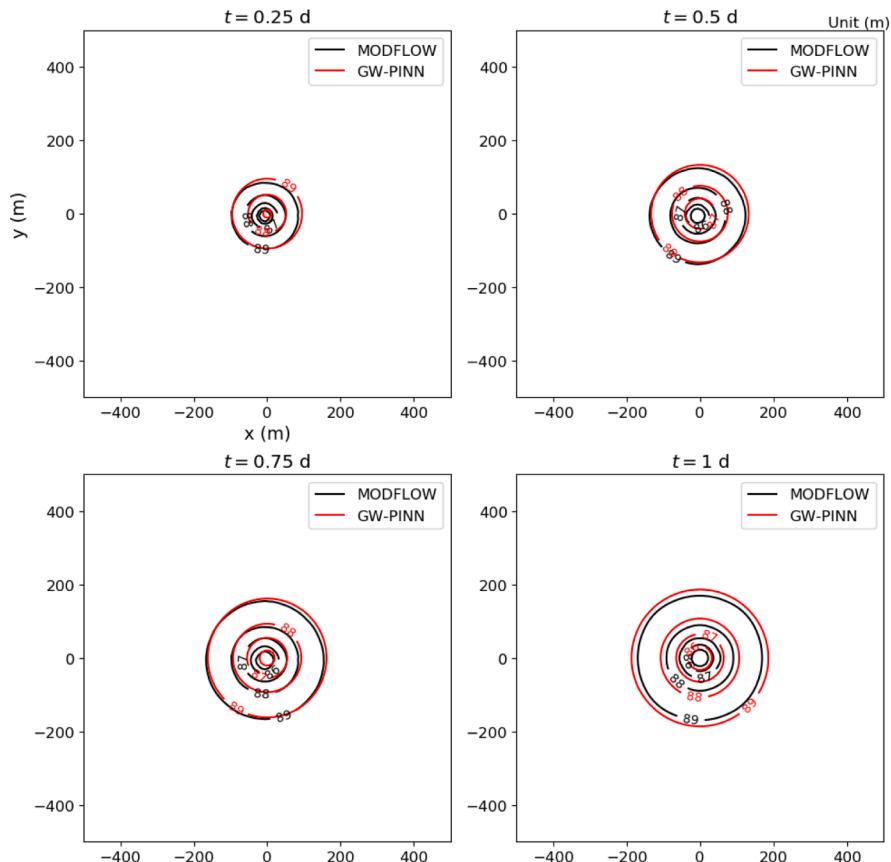


(b) Contour maps of hydraulic head predicted by GW-PINN and MODFLOW

Fig. 22. 3D surfaces and contour maps of the hydraulic head predicted by our GW-PINN at different times in the second stage for Test Case 3.

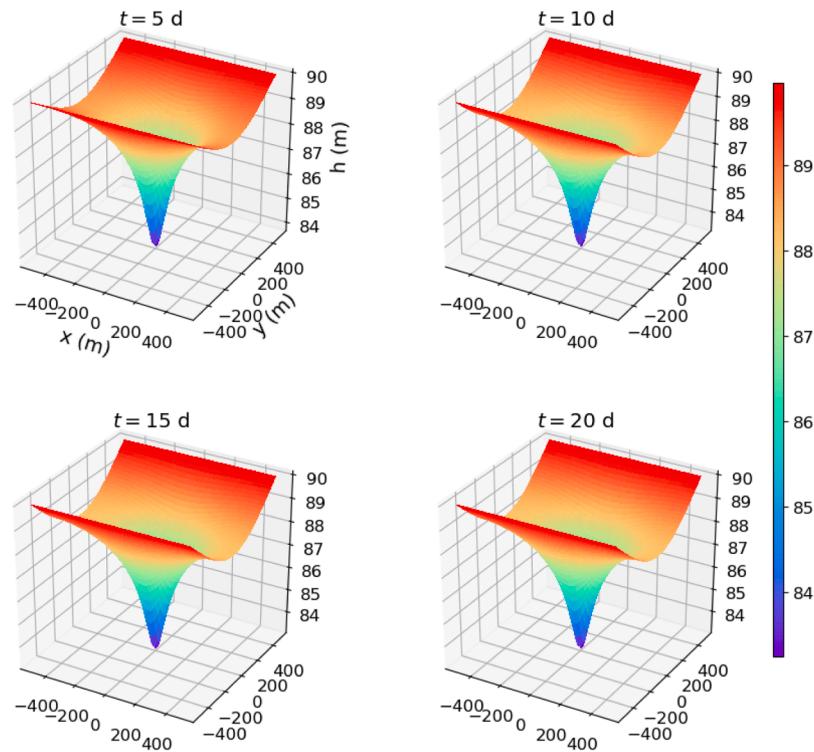


(a) 3D surfaces of hydraulic head predicted by GW-PINN

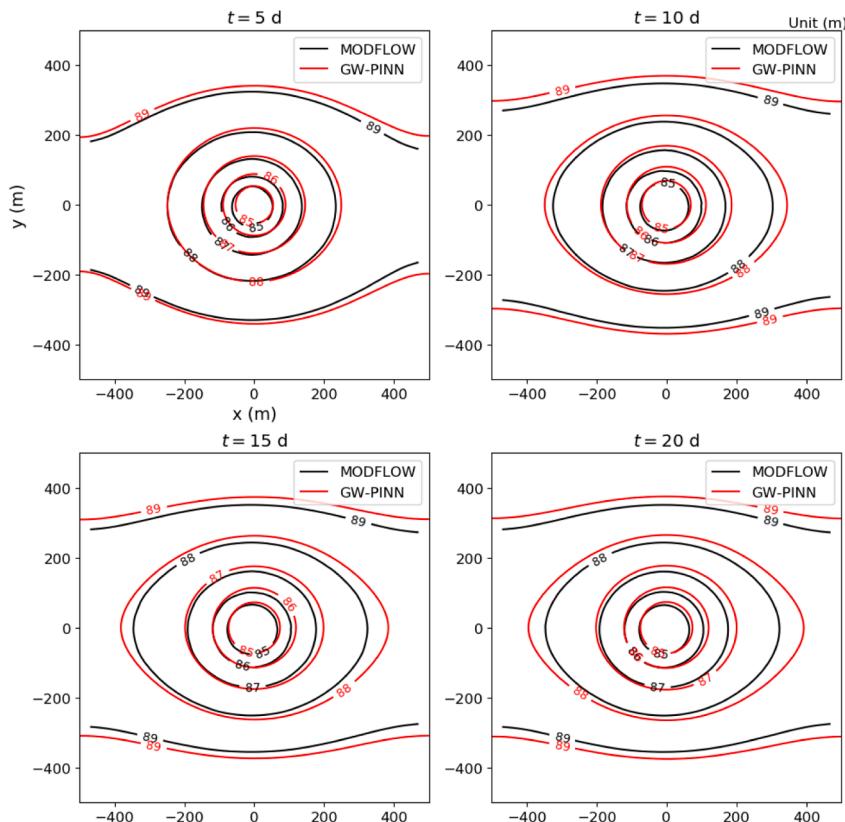


(b) Contour maps of hydraulic head predicted by GW-PINN and MODFLOW

Fig. 23. 3D surfaces and contour maps of the hydraulic head predicted by our GW-PINN at different times in the initial stage for Test Case 4.

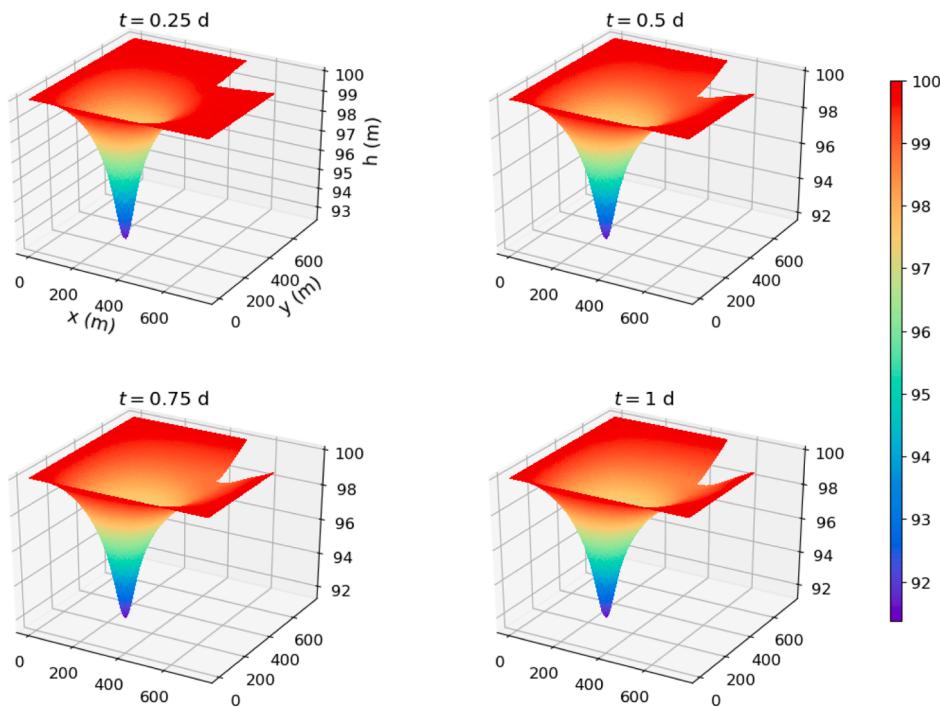


(a) 3D surfaces of hydraulic head predicted by GW-PINN

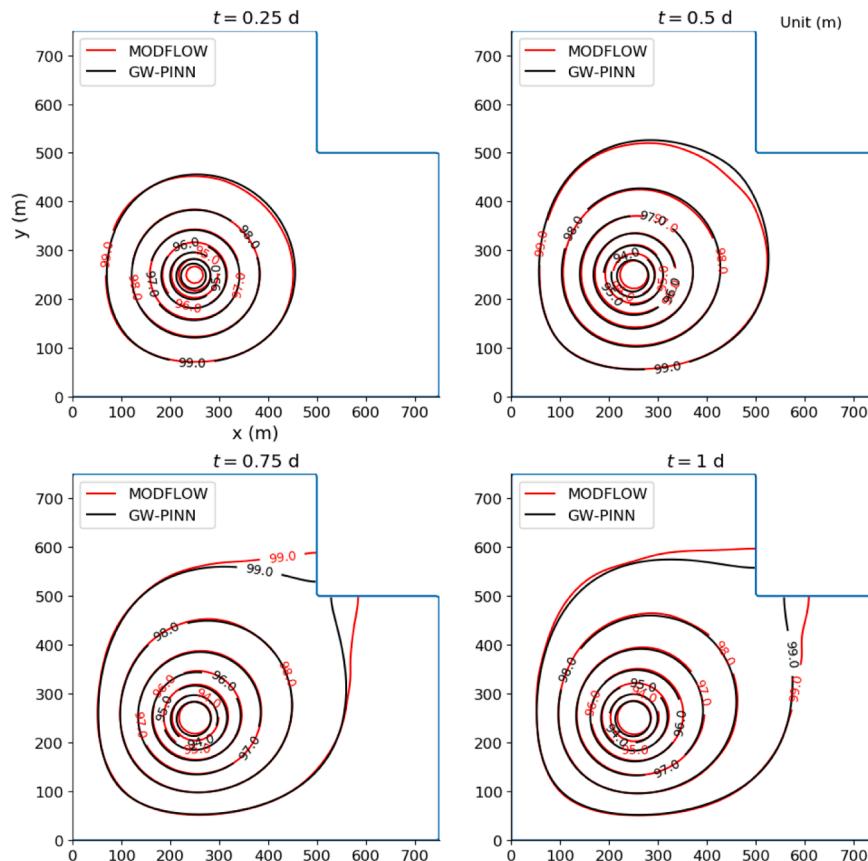


(b) Contour maps of hydraulic head predicted by GW-PINN and MODFLOW

Fig. 24. 3D surfaces and contour maps of the hydraulic head predicted by our GW-PINN at different times in the second stage for Test Case 4.

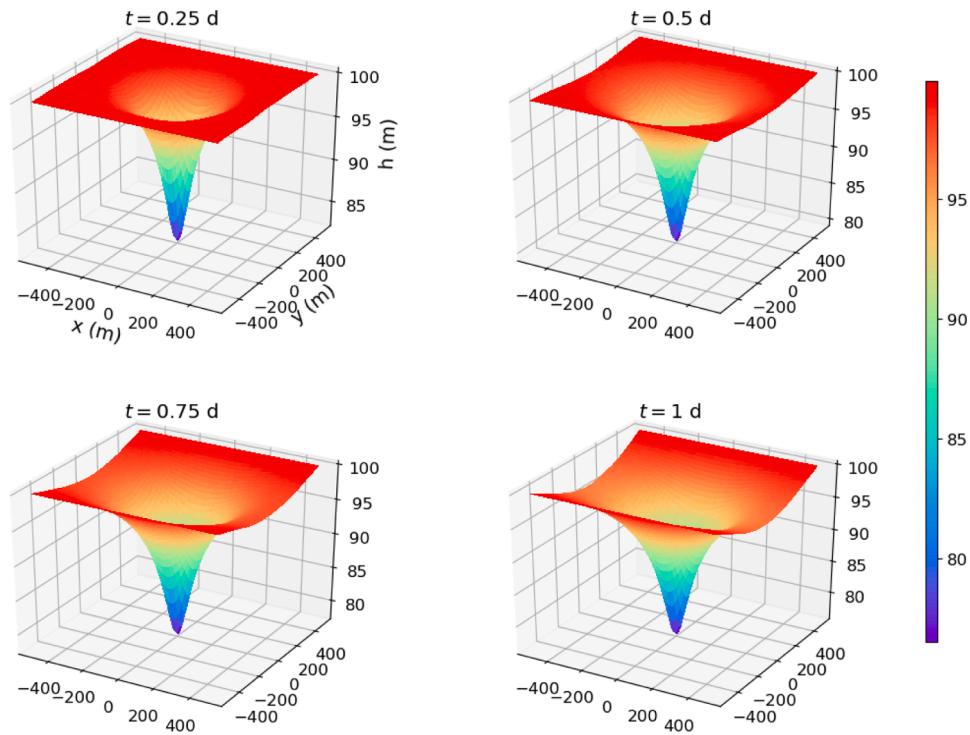


(a) 3D surfaces of hydraulic head predicted by GW-PINN

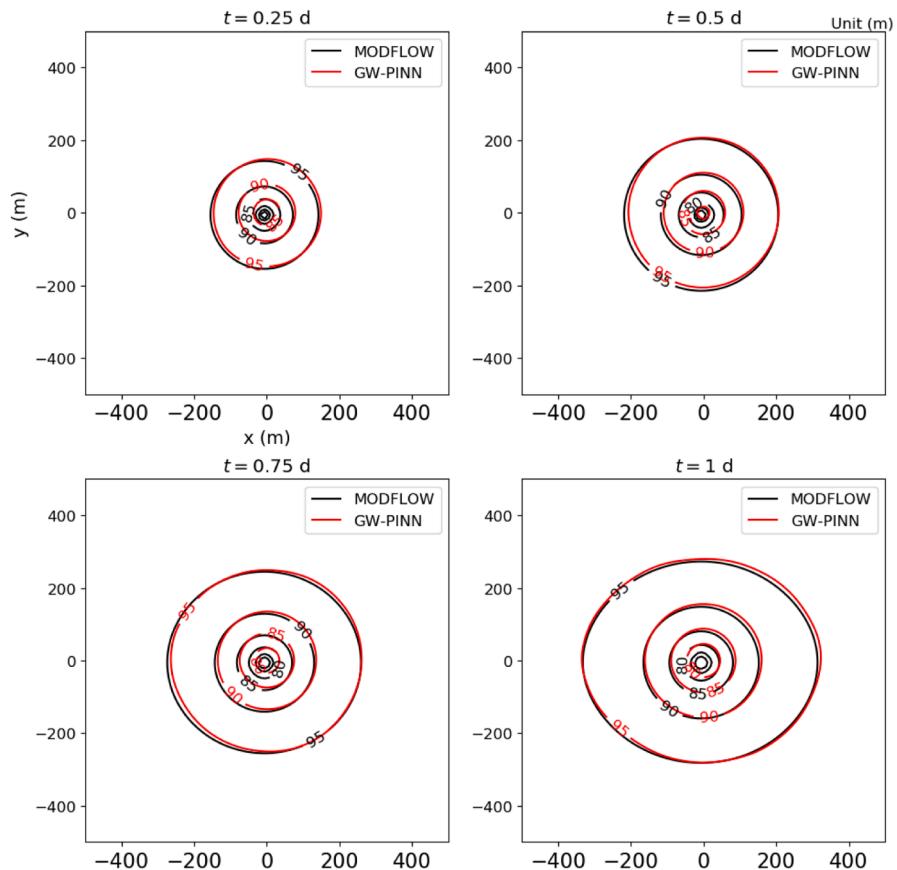


(b) Contour maps of hydraulic head predicted by GW-PINN and MODFLOW

Fig. 25. 3D surfaces and contour maps of hydraulic head predicted by our GW-PINN at different times for Test Case 5.

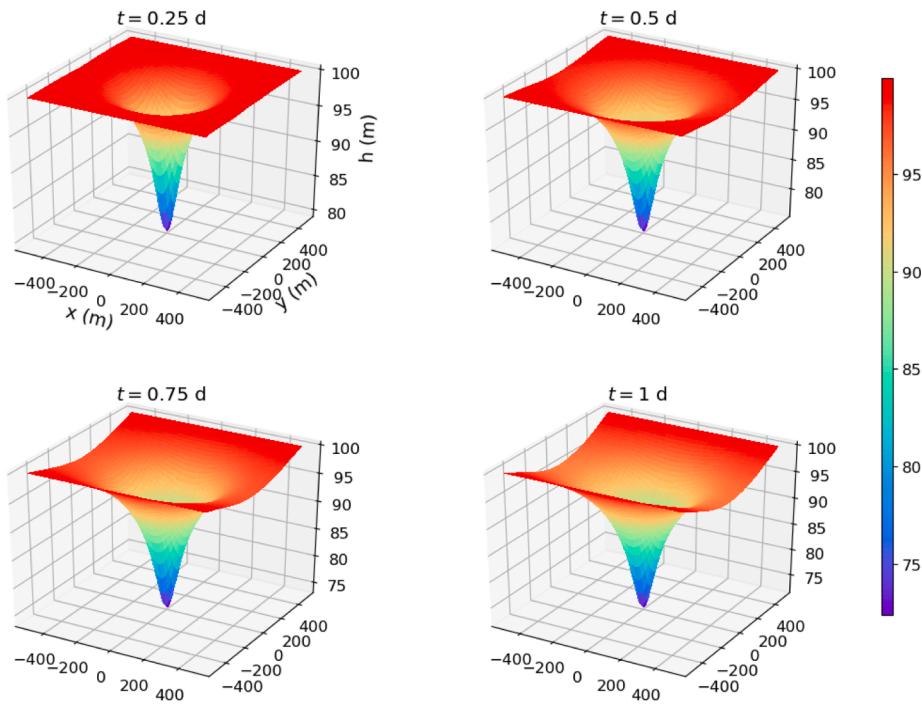


(a) 3D surfaces of hydraulic head predicted by GW-PINN

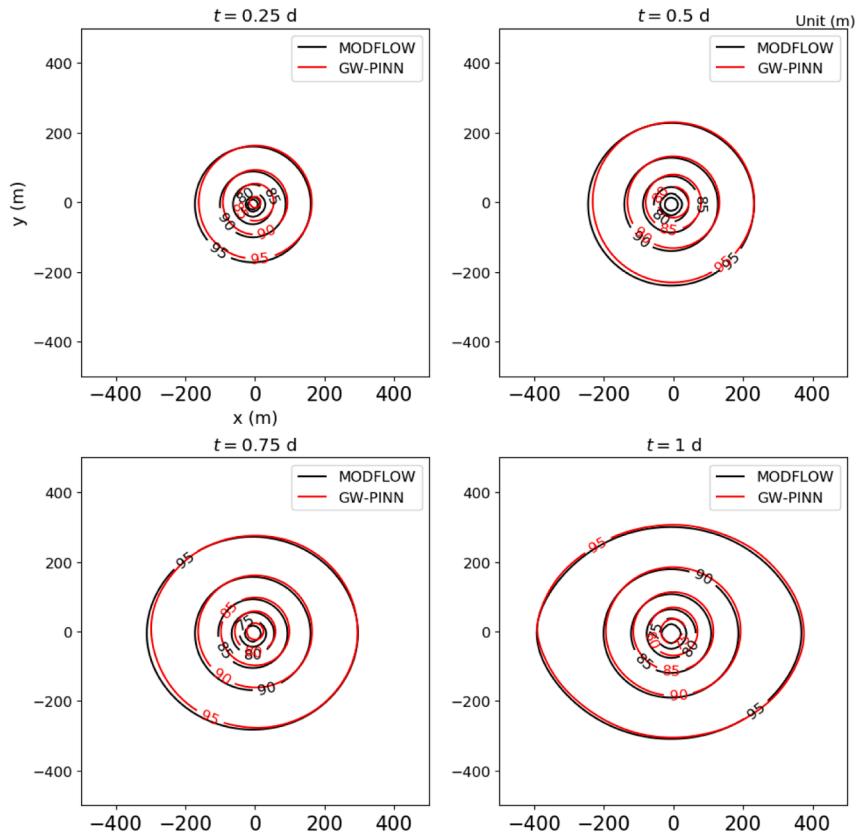


(b) Contour maps of hydraulic head predicted by GW-PINN and MODFLOW

Fig. 26. 3D surfaces and contour maps between predicted solutions by GW-PINN and numerical solutions by MODFLOW, where $Q = 5550 \text{ t/d}$.

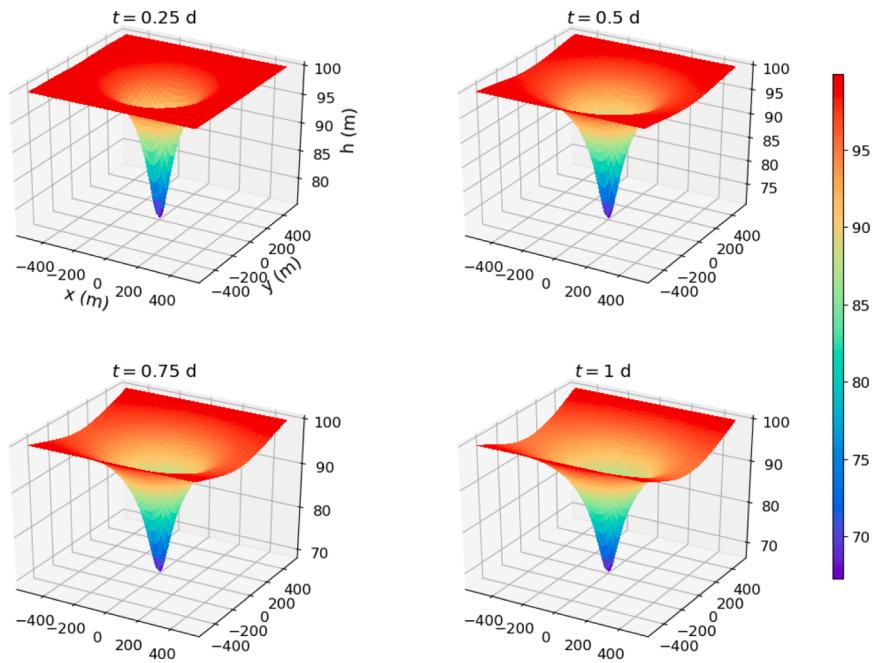


(a) 3D surfaces of hydraulic head predicted by GW-PINN

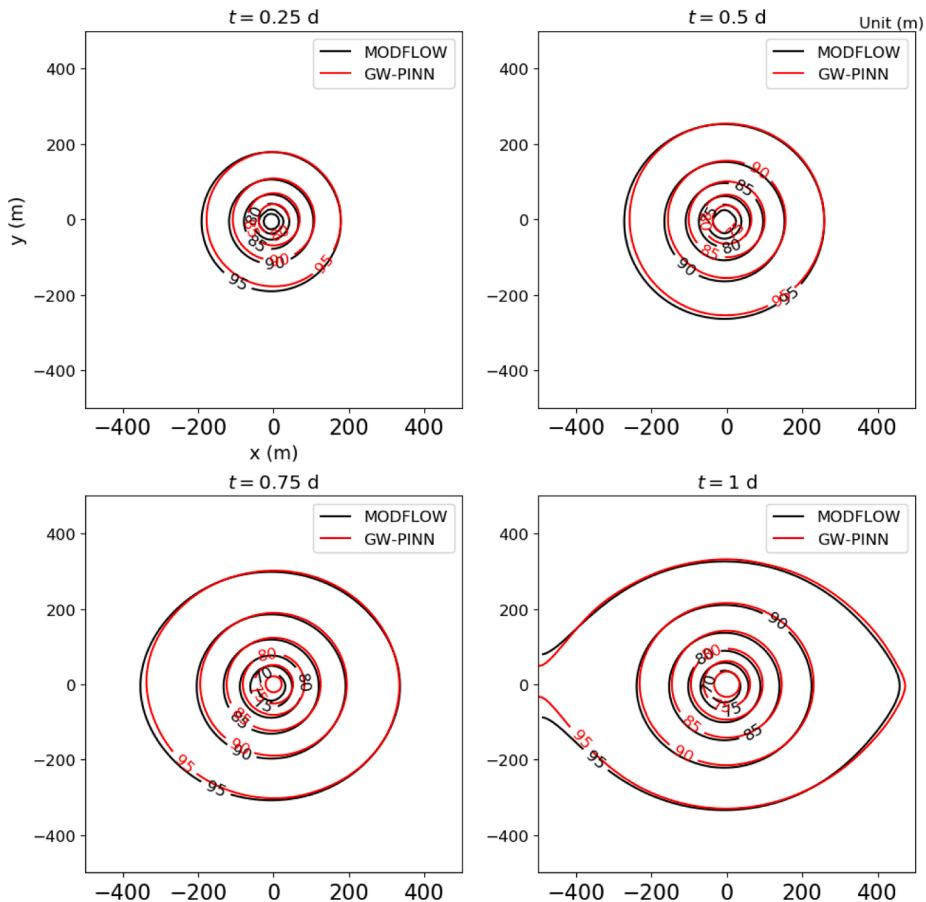


(b) Contour maps of hydraulic head predicted by GW-PINN and MODFLOW

Fig. 27. 3D surfaces and contour maps between predicted solutions by GW-PINN and numerical solutions by MODFLOW, where $Q = 6575$ t/d.

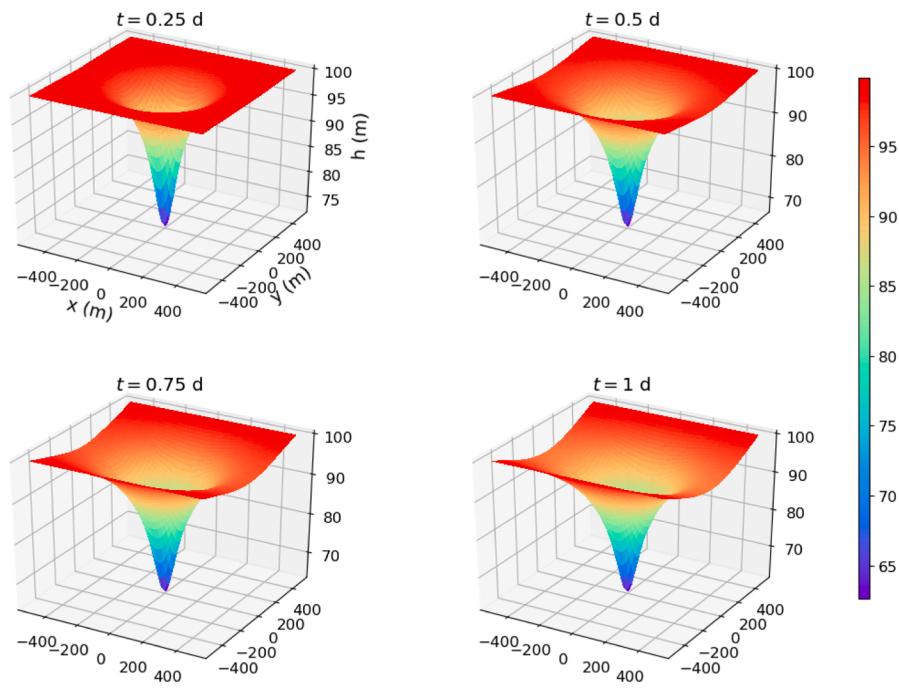


(a) 3D surfaces of hydraulic head predicted by GW-PINN

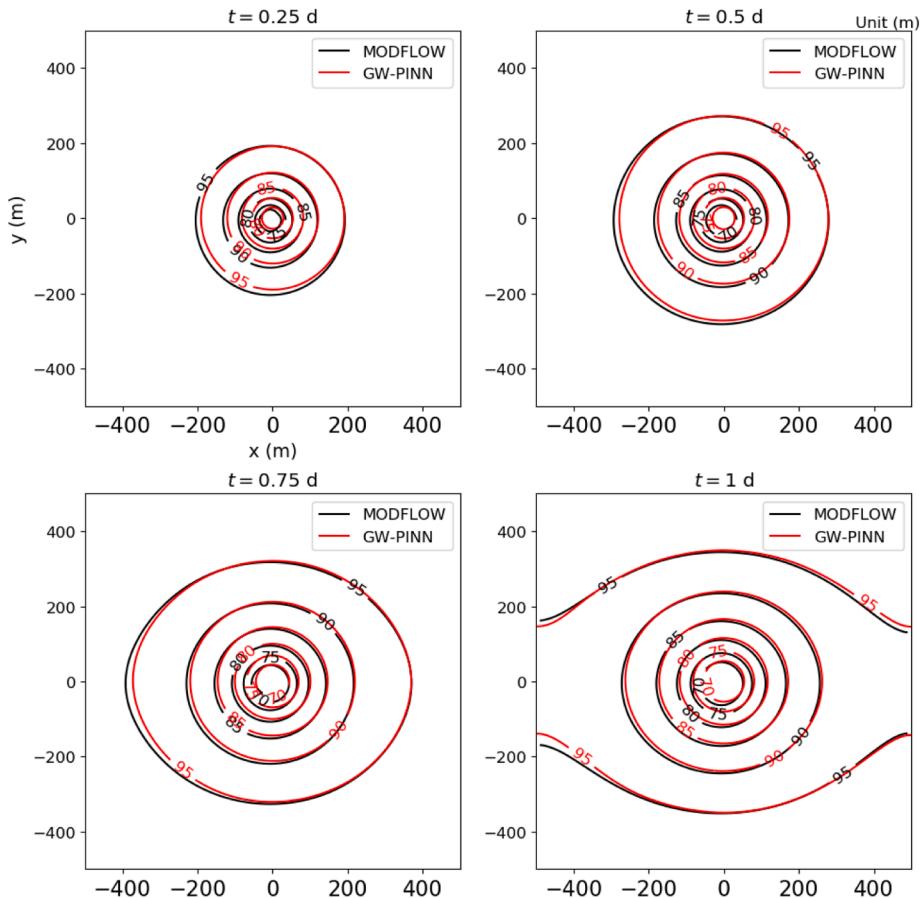


(b) Contour maps of hydraulic head predicted by GW-PINN and MODFLOW

Fig. 28. 3D surfaces and contour maps between predicted solutions by modified GW-PINN and numerical solutions by MODFLOW, where $Q = 7810$ t/d



(a) 3D surfaces of hydraulic head predicted by GW-PINN



(b) Contour maps of hydraulic head predicted by GW-PINN and MODFLOW

Fig. 29. 3D surfaces and contour maps between predicted solutions by modified GW-PINN and numerical solutions by MODFLOW, where $Q = 8920$ t/d.

Acknowledgments

X. Zhang's research was partially supported by National Key Research and Development Program of China (2021YFD1900805-02). Y. Zhu's research was partially supported by National Natural Science Foundation of China through Grants 52179041. L. Ju's research is partially supported by US Department of Energy grant DE-SC0022254.

References

- Adhikary, S.K., Rahman, M.M., Gupta, A.D., 2012. A stochastic modelling technique for predicting groundwater table fluctuations with time series analysis. *Int. J. Appl. Sci. Eng.* 1 (2), 238–249.
- Anshuman, A., Eldho, T.I., 2022. Entity aware sequence to sequence learning using LSTMs for estimation of groundwater contamination release history and transport parameters. *J. Hydrol.* 608, 127662.
- Berg, J., Nyström, K., 2018. A unified deep artificial neural network approach to partial differential equations in complex geometries. *Neurocomputing* 317, 28–41.
- Borsi, I., Rossetto, R., Schifani, C., Hill, M., 2013. Modeling unsaturated zone flow and runoff processes by integrating MODFLOW-LGR and VSF, and creating the new CFL package. *J. Hydrol.* 488, 33–47.
- Brunner, P., Simmons, C., 2012. HydroGeoSphere: a fully integrated, physically based hydrological model. *Groundwater* 50 (2), 170–176.
- Chen, J., Du, R., Wu, K., 2020. A comparison study of deep Galerkin method and deep Ritz method for elliptic problems with different boundary conditions. *Commun. Math. Res.* 36 (3), 354–376.
- Crowe, A.S., Shikaze, S.G., Ptacek, C.J., 2004. Numerical modelling of groundwater flow and contaminant transport to Point Pelee marsh, Ontario, Canada. *Hydrol. Process.* 18 (2), 293–314.
- Dickinson, J., James, S., Mehl, S., Hill, M., Leake, S., Zyvoloski, G., Faunt, C., Eddebarh, A., 2007. A new ghost-node method for linking different models and initial investigations of heterogeneity and nonmatching grids. *Adv. Water Res.* 30, 1722–1736.
- Dissanayake, M., Phan-Thien, N., 1994. Neural-network-based approximations for solving partial differential equations. *Commun. Numer. Methods Eng.* 10 (3), 195–201.
- Harbaugh, A., Banta, E., Hill, M., McDonald, M., 2000. MODFLOW-2000, The US Geological Survey modular ground-water model-User guide to modularization concepts and the ground-water flow process. US Geol. Surv. OpenFile Rep.
- He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, pp. 770–778.
- Gao, S., Huang, Y., Zhang, S., Han, J., Wang, G., Zhang, M., Lin, Q., 2020. Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation. *J. Hydrol.* 589, 125188.
- Jagtap, A., Kharazmi, E., Karniadakis, G., 2020. Conservative physics-informed neural networks on discrete domains for conservation laws: applications to forward and inverse problems. *Comput. Methods Appl. Mech. Eng.* 365, 113028.
- Jha, M.K., Sahoo, S., 2015. Efficacy of neural network and genetic algorithm techniques in simulating spatio-temporal fluctuations of groundwater. *Hydrol. Process.* 29 (5), 671–691.
- Knights, D., Parks, K.C., Sawyer, A.H., David, C.H., Browning, T.N., Danner, K.M., Wallace, C.D., 2017. Direct groundwater discharge and vulnerability to hidden nutrient loads along the Great Lakes coast of the United States. *J. Hydrol.* 554, 331–341.
- Krizhevsky, A., Sutskever, I., Hinton, G., 2012. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* 25 (2), 1097–1105.
- Kuo, J.T., Hsieh, M.H., Lung, W.S., She, N., 2007. Using artificial neural network for reservoir eutrophication prediction. *Ecol. Model.* 200 (1), 171–177.
- Laloy, E., Héroult, R., Jacques, D., Linde, N., 2018. Training-image based geostatistical inversion using a spatial generative adversarial neural network. *Water Resour. Res.* 54 (1), 381–406.
- Liang, Z., Zou, R., Chen, X., Ren, T., Su, H., Liu, Y., 2020. Simulate the forecast capacity of a complicated water quality model using the long short-term memory approach. *J. Hydrol.* 581, 124432.
- Liao, Y., Ming, P., 2019. Deep Nitsche method: Deep Ritz method with essential boundary conditions. <https://arxiv.org/pdf/1912.01309v1.pdf>.
- Lv, N., Liang, X., Chen, C., Zhou, Y., Li, J.i., Wei, H., Wang, H., 2020. A long short-term memory cyclic model with mutual information for hydrology forecasting: a case study in the Xixian Basin. *Adv. Water Resour.* 141, 103622.
- Mehl, S., Hill, M., 2002. Development and evaluation of a local grid refinement method for block-centered finite-difference groundwater models using shared nodes. *Adv. Water Resour.* 25, 497–511.
- Meng, X., Li, Z., Zhang, D., Karniadakis, G., 2020. PPINN: Parareal physics-informed neural network for time-dependent PDEs. *Comput. Methods Appl. Mech. Eng.* 370, 113250.
- Mirzavand, M., Ghazavi, R., 2014. A stochastic modelling technique for groundwater level forecasting in an arid environment using time series methods. *Water Resour. Manag.* 29 (4), 1315–1328.
- Mishra, P., 2019. NodeLab. <https://github.com/pankajkmishra/NodeLab>.
- Mo, S., Zabaras, N., Shi, X., Wu, J., 2019. Deep autoregressive neural networks for high dimensional inverse problems in groundwater contaminant source identification. *Water Resour. Res.* 55 (5), 3856–3881.
- Müller, J., Zeinhofer, M., 2019. Deep Ritz revisited. <https://arxiv.org/abs/1912.03937>.
- Panday, S., Langevin, C., 2012. Improving sub-grid scale accuracy of boundary features in regional finite-difference models. *Adv. Water Resour.* 41, 65–75.
- Pang, G., Lu, L., Karniadakis, G., 2019. FPINNs: fractional physics-informed neural networks. *SIAM J. Sci. Comput.* 41 (4), A2603–A2626.
- Raissi, M., Perdikaris, P., Karniadakis, G., 2019. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* 378, 686–707.
- Ravansalar, M., Rajaei, T., Kisi, O., 2017. Wavelet linear genetic programming: a new approach for modeling monthly streamflow. *J. Hydrol.* 549, 461–475.
- Sahu, R.K., Müller, J., Park, J., Varadharajan, C., Arora, B., Faybishenko, B., Agarwal, D.A., 2020. Impact of input feature selection on groundwater level prediction from a multi-layer perceptron neural network. *Front. Water* 2, 573034.
- Shen, C., 2018. A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resour. Res.* 54 (11), 8558–8593.
- Sirignano, J., Spiliopoulos, K., 2018. DGM: a deep learning algorithm for solving partial differential equations. *J. Comput. Phys.* 375, 1339–1354.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15, 1929–1958.
- Sun, J.C., Hu, L.T., Li, D.D., Sun, K.N., Yang, Z.Q., 2022. Data-driven models for accurate groundwater level prediction and their practical significance in groundwater management. *J. Hydrol.* 608, 127630.
- Sun, L., Gao, H., Pan, S., Wang, J., 2020. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Comput. Methods Appl. Mech. Eng.* 361, 112732.
- Trefry, M., Muffels, C., 2007. FEFLOW: a finite-element groundwater flow and transport modeling tool. *Groundwater* 45 (5), 525–528.
- Vilhelmsen, T., Christensen, S., Mehl, S., 2011. Evaluation of MODFLOW-LGR in connection with a synthetic regional-scale model. *Groundwater* 50, 118–132.
- Walter, D.A., Masterson, J.P., Finkelstein, J.S., Monti, J., Misut, P.E., Fienen, M.N., 2020. Simulation of groundwater flow in the regional aquifer system on Long Island, New York, for pumping and recharge conditions in 2005–15: U.S. Geol. Surv. Sci. Invest. Rep., 75 p., 10.3133/sir202005091.
- Wang, N.Z., Zhang, D.X., Chang, H.B., Li, H., 2020. Deep learning of subsurface flow via theory-guided neural network. *J. Hydrol.* 584, 124700.
- Wang, Z., Zhang, Z., 2020. A mesh-free method for interface problems using the deep learning approach. *J. Comput. Phys.* 400, 108963.
- Weinan, E., Yu, B., 2018. The deep Ritz method: a deep learning-based numerical algorithm for solving variational problems. *Commun. Math. Stat.* 6.
- Xu, R., Wang, N.Z., Zhang, D.X., 2021. Solution of diffusivity equations with local sources/sinks and surrogate modeling using weak form Theory-guided Neural Network. *Adv. Water Resour.* 153, 103941.
- Zhao, T.X., Zhu, Y., Ye, M., Zhang, X.P., Yang, J.Z., Wu, J.W., 2020. Machine-learning methods for water table depth prediction in seasonal freezing-thawing areas. *Groundwater* 58 (3), 419–431.
- Zhang, D., Guo, L., Karniadakis, G., 2020. Learning in modal space: solving time-dependent stochastic PDEs using physics-informed neural networks. *SIAM J. Sci. Comput.* 42 (2), A639–A665.
- Zhang, D., Lu, L., Guo, L., Karniadakis, G.E., 2019. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *J. Comput. Phys.* 397, 108850.
- Zhang, J., Zhu, Y., Zhang, X., Ye, M., Yang, J., 2018. Developing a long short-term memory (LSTM) based model for predicting water table depth in agricultural areas. *J. Hydrol.* 561, 918–929.
- Zhu, Y., Zabaras, N., 2018. Bayesian deep convolutional encoder-decoder networks for surrogate modeling and uncertainty quantification. *J. Comput. Phys.* 366, 415–447.