



Stochastic deep collocation method based on neural architecture search and transfer learning for heterogeneous porous media

Hongwei Guo^{2,3,4} · Xiaoying Zhuang^{2,3,4} · Pengwan Chen⁵ · Naif Alajlan¹ · Timon Rabczuk¹

Received: 24 July 2021 / Accepted: 17 December 2021 / Published online: 18 January 2022
© The Author(s) 2022

Abstract

We present a stochastic deep collocation method (DCM) based on neural architecture search (NAS) and transfer learning for heterogeneous porous media. We first carry out a sensitivity analysis to determine the key hyper-parameters of the network to reduce the search space and subsequently employ hyper-parameter optimization to finally obtain the parameter values. The presented NAS based DCM also saves the weights and biases of the most favorable architectures, which is then used in the fine-tuning process. We also employ transfer learning techniques to drastically reduce the computational cost. The presented DCM is then applied to the stochastic analysis of heterogeneous porous material. Therefore, a three dimensional stochastic flow model is built providing a benchmark to the simulation of groundwater flow in highly heterogeneous aquifers. The performance of the presented NAS based DCM is verified in different dimensions using the method of manufactured solutions. We show that it significantly outperforms finite difference methods in both accuracy and computational cost.

Keywords Deep learning · Neural architecture search · Error estimation · Randomized spectral representation · Method of manufactured solutions · Log-normally distributed · Physics-informed · Sensitivity analysis · Hyper-parameter optimization algorithms · Transfer learning

1 Introduction

In recent years, groundwater pollution has become one of the most important environmental problems worldwide. To protect groundwater quality, it is necessary to predict groundwater flow and solute transport. This in turn is based on the theory of porous media. The associated heterogeneity and complexity of the porous media still poses a major challenge for such groundwater flow problems. The hydraulic

conductivity describes the ability of the medium to transmit fluid through pore spaces. It is common to use random fields with a given statistical structure to describe the porous media because of its intrinsic complexity [1]. Freeze [2] showed, that the hydraulic conductivity field can be well characterized by random log-normal distribution. This approach is often used for the flow analysis in saturated zones [3]. Both Gaussian [4] and exponential [5] correlations are commonly chosen for the log-normal probability distribution.

✉ Timon Rabczuk
timon.rabczuk@uni-weimar.de

Hongwei Guo
hongwei.guo@iop.uni-hannover.de

Xiaoying Zhuang
zhuang@iop.uni-hannover.de

Pengwan Chen
pwchen@bit.edu.cn

Naif Alajlan
najlan@ksu.edu.sa

¹ ALISR Laboratory, College of Computer and Information Sciences, King Saud University, P. O. Box 51178, Riyadh 11543, Saudi Arabia

² Computational Science and Simulation Technology, Institute of Photonics, Leibniz Universität Hannover, Appelstr. 11, Hannover 30167, Germany

³ Department of Geotechnical Engineering, Tongji University, 1239 Siping Road, Shanghai 200092, People's Republic of China

⁴ Key Laboratory of Geotechnical and Underground Engineering of Ministry of Education, Tongji University, 1239 Siping Road, Shanghai 200092, People's Republic of China

⁵ State Key Laboratory of Explosion Science and Technology, Beijing Institute of Technology, No. 5, South Street, Zhongguancun, Haidian District, Beijing 100081, People's Republic of China

Different approaches have been used to express the permeability as a function of pore structure parameters [6–8]. Analytical spectral representation methods were first used by Bakr [9] to solve the stochastic flow and solute transport equations perturbed with a random hydraulic conductivity field. If a random field is homogeneous and has zero mean, then it is always possible to represent the process by a Fourier (or Fourier–Stieltjes) decomposition into essentially uncorrelated random components. These random components in turn will give the spectral density function, which is the distribution of variance over different wave numbers k . This theory is widely used to construct hydraulic conductivity fields, and a number of construction methods have been derived such as turning bands method [10], HYDRO_GEN method [11] and Kraichnan algorithm [12]. Ababou et al. [13] used the turning bands method and narrowed down a range of relevant parameters. Wörmann and Kronnäs [13] tested a gradual increase of the heterogeneity of the flow resistance and compared the numerically simulated residence time PDF with the observed based on HYDRO_GEN method. Unlike the other two methods, Kraichnan proposed an approximation algorithm for direct control of random field accuracy, by increasing the modulus through the variance of the log-hydraulic conductivity random field. Inspired by these results, Kolyukhin and Sabelfeld [1] constructed a randomized spectral model (RSM) studying the steady flow in porous media in 3D assuming small fluctuations. We adopted this approach here to generate hydraulic conductivity fields.

Deep learning methods have become very popular since the development of deep neural networks (DNNs) [14, 15]. They have been applied to a variety of problems including image processing [16], object detection [17], speech recognition [18], just to name a few. While the majority of applications employ DNN as regression models, there has been some recent interest in exploiting DNN for the solution of PDEs. Mills et al. for instance solved the Schrödinger equation with convolutional neural networks by directly learning the mapping between the potential and energy [19]. Weinan E et al. presented deep learning-based numerical methods for high-dimensional parabolic PDEs and back-forward stochastic differential equations [20, 21]. Raissi et al. devised a machine learning approach for the solution of linear and nonlinear differential equations using Gaussian Processes [22]. In [23, 24] they presented a so-called physical informed neural network for supervised learning of nonlinear partial differential equations solving for instance Burger's equations and Navier–Stokes equations. Beck et al. [25] solved stochastic differential and Kolmogorov equations with neural networks. For several forward and inverse problems in solid mechanics, we presented a Deep Collocation Method (DCM) [26, 27]. Instead of exploiting the strong form of the boundary value problem, we presented a deep energy

method (DEM) [28–30] which requires the definition of an energy potential instead of a BVP.

In a typical machine learning application, the practitioner must apply appropriate data pre-processing, feature engineering, feature extraction and feature selection methods to make the dataset suitable for machine learning. Following these pre-processing steps, practitioners must perform algorithm selection and hyper-parameter optimization to maximize their final machine learning model's predictive performance. The physics informed neural network (PINN) models discussed in the previous paragraph are no exception though the randomly distributed collocation points are generated for further calculation without the need for data-preprocessing. Most of the time wasted in PINN models are related to the tuning of neural architecture configurations, which strongly influences the accuracy and stability of the approach. Since many of these steps are typically beyond the capabilities of non-experts, automated machine learning (AutoML) has become popular. The oldest AutoML library is AutoWEKA [31], first released in 2013, which automatically selects models and hyper-parameters. Other notable AutoML libraries include auto-sklearn [32], H2O AutoML [33], and TPOT [34]. Neural architecture search (NAS) [35] is a technique for automatic design of neural networks, which allows algorithms to automatically design high-performance network structures based on sample sets. Neural architecture search (NAS) aims to find a configuration comparable to human experts on certain tasks and even discover certain network structures that have not been proposed by humans before, which can effectively reduce the use and implementation cost of neural networks. In [36], the authors add a controller in the efficient NAS, which can learn to discover neural network architectures by searching for an optimal subgraph within a large computational graph. They used parameter sharing between the subgraphs to make the computing process faster. The controller decides which parameter matrices are used by choosing the previous indices. Therefore, in ENAS, all recurrent cells in a search space share the same set of parameters. Liu [37] used a sequence model-based optimization (SMBO) strategy to learn a surrogate model to guide the search of the structure space. To build up a Neural architecture search (NAS) model, it is necessary to conduct dimensionality reduction and identification of valid parameter bounds to reduce the calculation involved in auto-tuning. A global sensitivity analysis can be used to identify valid regions in the search space and subsequently decrease its dimensionality [38], which can serve as a starting point for an efficient calibration process.

The remainder of this paper is organized as follows. In Sect. 2, we describe the physical model of the groundwater flow problem, the randomized spectral method to generate hydraulic conductivity fields as well as the approach of manufactured solutions to verify the accuracy of our model. In

Section 3, we introduce the neural architecture search model. We subsequently present an efficient sensitivity analysis and compare several hyper-parameter optimizers to find an accurate and efficient search method. In Section 4, we briefly describe the employed finite difference method which is used to solve several benchmark problems as comparison. At last, some conclusions are drawn in Section 5.

2 Stochastic analysis of a heterogeneous porous medium

2.1 Darcy equation for groundwater flow problem

Consider the continuity equation for steady-state, aquifer flow in a porous media governed by the Darcy law:

$$q(x) = -K(x)\nabla(h(x)), \tag{1}$$

where q is the Darcy velocity, K is the hydraulic conductivity, h the hydraulic head $h = H + \delta h$ with the mean H and the perturbation δh . To describe the variation of the hydraulic conductivity as a function of the position vector x , it is convenient to introduce the variable

$$Y(x) = \ln K(x), \tag{2}$$

where $Y(x)$ is the hydraulic log-conductivity with the mean $\langle Y \rangle$ and perturbation $Y'(x)$:

$$Y(x) = \langle Y \rangle + Y'(x), \tag{3}$$

with $E[Y'(x)] = 0$, and $Y(x)$ is taken to be a three-dimensional statistically homogeneous random field characterized by its correlation function

$$C_Y(r) = \langle Y'(x+r)Y'(x) \rangle, \tag{4}$$

where r is the separation vector. According to the conservation equation $\nabla \cdot q = 0$, Equation (1) can be rewritten in the following form:

$$E(h) = \sum_{j=1}^N \frac{\partial}{\partial x_j} \left(K(x) \frac{\partial h}{\partial x_j} \right) = 0, \tag{5}$$

which is subjected to the Neumann and Dirichlet boundary conditions

$$\begin{aligned} h(x) &= \bar{h}, x \in \tau_D, \\ q_n(x) &= \bar{q}_n, x \in \tau_N. \end{aligned} \tag{6}$$

with N denoting the dimension. The groundwater flow problem can be boiled down to find a solution h such that Equations (5) and (6) hold; E is an operator that maps elements of vector space H to vector space V :

$$E : H \rightarrow V, \text{ with } h \in H. \tag{7}$$

With Equation (5) and $N = 3$ in domain $D = [0, L_x] \times [0, L_y] \times [0, L_z]$, the Dirichlet boundary and Neumann boundary conditions can be assumed as follows:

$$\begin{cases} h(0, y, z) = -J \cdot L_x, & h(L_x, y, z) = 0, & \forall y \in [0, L_y], z \in [0, L_z], \\ \frac{\partial h}{\partial y}(x, 0, z) = \frac{\partial h}{\partial y}(x, L_y, z) = 0, & & \forall x \in [0, L_x], z \in [0, L_z], \\ \frac{\partial h}{\partial z}(x, y, 0) = \frac{\partial h}{\partial z}(x, y, L_z) = 0, & & \forall x \in [0, L_x], y \in [0, L_y], \end{cases} \tag{8}$$

where J is the mean slope of the hydraulic head in x direction [9]. As suggested by Ababou [13], the scale of the fluctuation should be significantly smaller than the scale of the domain. The lengths L_x, L_y, L_z of the domain are usually set to be ten times larger than λ . A reasonable mesh size Δx could then be

$$\frac{\Delta x}{\lambda} \leq \frac{1}{5}. \tag{9}$$

As Y' is homogeneous and isotropic, we consider two correlation functions: the exponential correlation function [39],

$$C_Y(r) = \sigma_Y^2 \exp\left(-\frac{|r|}{\lambda}\right), \tag{10}$$

and the Gaussian correlation function [40],

$$C_Y(r) = \sigma_Y^2 \exp\left(-\frac{|r|^2}{\lambda^2}\right), \tag{11}$$

where λ is the log conductivity correlation length scale.

2.2 Generate the hydraulic conductivity fields

Due to the intrinsic complexity of heterogeneous porous media, random field theory is implemented for the generation of the heterogeneous field showing a fractal behavior. Applying Wiener–Khinchin theorem to the heterogeneous groundwater flow problem, the Gaussian random field [41] with given spectral density $S(k)$ is just the Fourier transform of the correlation function (in Equation (4)):

$$C_Y(r) = \int_{\mathbb{R}^N} e^{i2\pi k \cdot r} S(k), \tag{12}$$

$$S(k) = \int_{\mathbb{R}^N} e^{-i2\pi k \cdot r} C_Y(r), \tag{13}$$

with $S(k)$ spectral function of the random field $Y'(x)$ and

$$\mathcal{F}\left(\exp\left(-\frac{|r|}{\lambda}\right)\right) = \frac{2\lambda}{1 + 4\pi^2 k^2 r^2}, \tag{14}$$

$$\mathcal{F}(\exp(-\frac{|r|^2}{\lambda^2})) = \lambda\sqrt{\pi}e^{-\pi^2k^2r^2}. \tag{15}$$

Substituting Eqs. (10), (11), (14) and (15) into Eq. (13), respectively, the spectral function under the exponential and the Gaussian correlation coefficient can be derived:

$$S(\mathbf{k}, \lambda) = \sigma_Y^2\lambda^d(1 + (2\pi\mathbf{k}\lambda)^2)^{-\frac{d+1}{2}}, \tag{16}$$

$$S(\mathbf{k}, \lambda) = \sigma_Y^2\pi^{d/2}\lambda^d e^{-(\pi\mathbf{k}\lambda)^2}. \tag{17}$$

A Gaussian homogenous random field in the general case can be retrieved [42]:

$$Y'(\mathbf{x}) = \sqrt{\frac{2\sigma^2}{N}} \sum_{i=1}^N (\xi_1 \cos(2\pi\mathbf{k}_i \cdot \mathbf{x}) + \xi_2 \sin(2\pi\mathbf{k}_i \cdot \mathbf{x})), \tag{18}$$

where ξ_i are mutually independent Gaussian random variables. For the random variable \mathbf{k}_i , we can get its probability density distribution function $p(\mathbf{k})$ and calculate its cumulative distribution function (*cdf*) according to $F(k) = \int_{-\infty}^k p(x)dx$. As long as there exists another uniformly distributed random variable θ , the inverse function

$\mathbf{k} = F^{-1}(\theta)$ can be obtained, and \mathbf{k} must obey the $p(\mathbf{k})$ distribution. More details can be found in Appendix B while the associated python script is summarized in Appendix C. Figures 1 and 2 show the two- and three-dimensional random field space with fixed $\langle k \rangle = 15$, $\sigma^2 = 0.1$ and $N = 1000$.

2.3 Defining numerical experimental model

After determining the expression for the stochastic flow analysis in heterogeneous porous materials, we need to set the geometric and physical parameters. It is well known that the number of modes N and the variance σ^2 govern the hydraulic conductivity. For the exponential correlation coefficient, large values of N might lead to a non-differentiable K-field [43]. We set the N values to 500, 1000 and 2000; σ^2 determines the heterogeneity of the hydraulic conductivity, a larger σ^2 indicating a larger heterogeneity. In real geological formations, σ^2 has a wide range of variation. As summarized in Sudicky’s study [44], in low heterogeneous Canadian Forces Base Borden aquifers, it is $\sigma^2 = 0.29$, for Cape Cod it is 0.14, but in highly heterogeneous Columbus aquifers, $\sigma^2 = 4.5$. First-order analysis [9] has been proven

Fig. 1 Two-dimensional hydraulic conductivity field with (a) exponential correlation and (b) Gaussian correlation

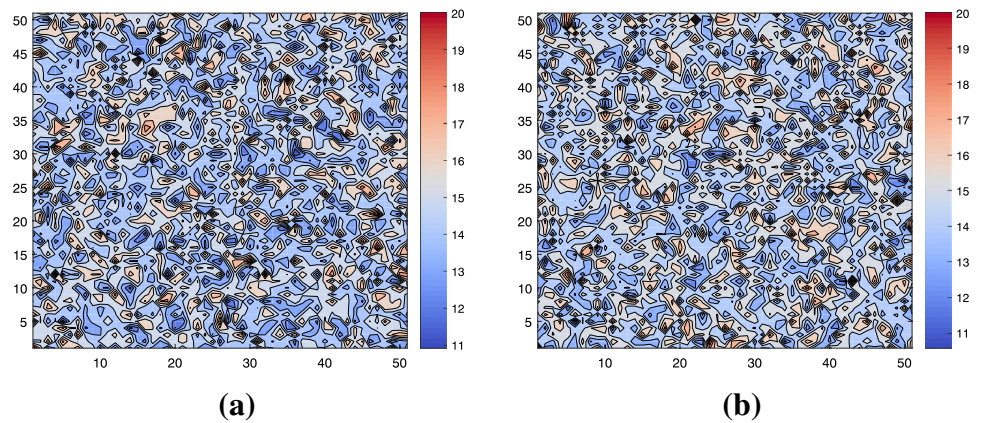
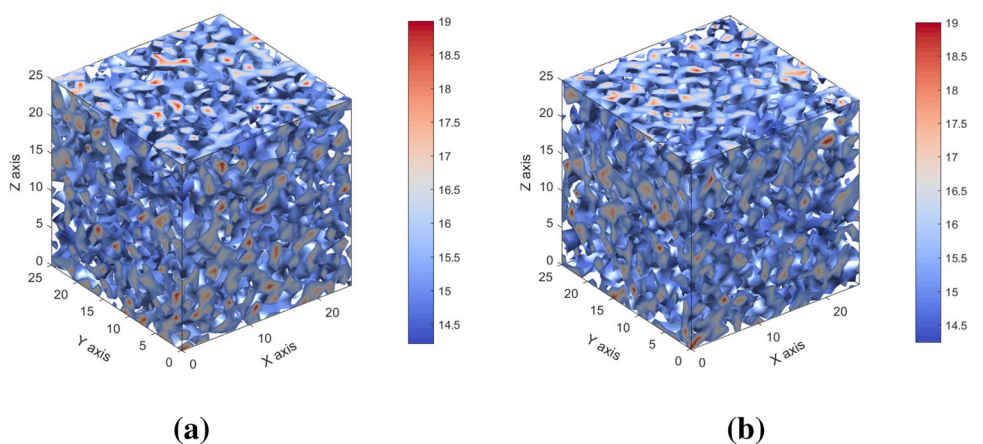


Fig. 2 Three-dimensional hydraulic conductivity field with (a) exponential correlation and (b) Gaussian correlation



as a solid basis for predictions. Numerical simulations [11] indicate that the first-order results are robust and applicable when σ^2 is close to and even above 1. With this approximation, we can get the $e^{(Y)} = \langle K \rangle \exp(-\sigma^2/2)$ for one- and two-dimensional cases [45], and $e^{(Y)} = \langle K \rangle \exp(-\sigma^2/6)$ for three-dimensional cases [46]. In this paper, we set the value of σ^2 to 0.1, 1 and 3, covering the three cases from small to medium and large. The mean hydraulic conductivity is fixed to $\langle K \rangle = 15m/day$, a value representative for gravel or coarse sand aquifers [47]. And we set all the correlation lengths in one- and two-dimensional cases equal 1m, in three dimensional cases, we set them to $\lambda_1 = 0.5m, \lambda_2 = 0.2m$ and $\lambda_3 = 0.1m$. Based on the above settings, we have finalized our test domain:

- One-dimensional groundwater flow $\rightarrow [0, 25]$.
- Two-dimensional groundwater flow $\rightarrow [0, 20] \times [0, 20]$.
- Three-dimensional groundwater flow $\rightarrow [0, 5] \times [0, 2] \times [0, 1]$.

2.4 Manufactured solutions

To verify the accuracy of our model and obtain an error estimation, we use the method of manufactured solution (MMS), which provides a general procedure for generating analytical solutions [48]. Malaya et al. [49] discussed the method of manufactured solutions in constructing an error estimator for solution verification where one simulates the phenomenon of interest with no priori knowledge of the solution. This artificial solution is then substituted into the equations. There will be a residual term since the chosen function is unlikely to be an exact solution to the original partial differential equations. This residual can then be added as a source term. With MMS, the original problem to find the solution of Equation (5) is thus changed to the following form:

$$E(\hat{h}) = \sum_{j=1}^N \left(\frac{\partial}{\partial x_j} \left(K(x) \frac{\partial \hat{h}}{\partial x_j} \right) \right) = \sum_{j=1}^N f_j = f. \tag{19}$$

For operator $E(\hat{h})$, we now get a source term f . By adding the source term to the original governing equation E , a slightly modified governing equation will be obtained:

$$E'(\hat{h}) = E(\hat{h}) - f = 0, \tag{20}$$

which is solved by the manufactured solution \hat{h} . The Neumann and Dirichlet boundary conditions are thus modified as follows:

$$\begin{aligned} \hat{h}(x) &= \hat{h}_{MMS}(x), x \in \tau_D, \\ \hat{q}_n(x) &= -K(x) \hat{h}_{MMS,n}(x), x \in \tau_N. \end{aligned} \tag{21}$$

We adopt the form of the manufactured solution mentioned in Tremblay’s study [48],

$$\hat{h}_{MMS}(x) = a_0 + \sin \left(\sum_{j=1}^N a_j x_j \right), \tag{22}$$

where $\{a_i\}$ are arbitrary non-zero real numbers. When the manufactured solutions (22) are applied on the left side of Eq. (5), we will get a source term f ,

$$f(x_j) = a_j \frac{\partial K(x)}{\partial x_j} \cos \left(\sum_{i=1}^N a_i x_i \right) - a_j^2 K(x) \sin \left(\sum_{i=1}^N a_i x_i \right). \tag{23}$$

To verify the adaptability of our model to different solutions, we also used another form of manufactured solution [49],

$$\hat{h}_{MMS}(x) = a_0 + \sum_{j=1}^N \sin(a_j x_j), \tag{24}$$

where the parameter values are the same as in Equation (22). We can get the source term as follows:

$$f(x_j) = a_j \frac{\partial K(x)}{\partial x_j} \cos(a_j x_j) - a_j^2 K(x) \sin(a_j x_j). \tag{25}$$

This leads to the change of the boundary conditions from Equation (6) to

$$\left\{ \begin{aligned} \hat{h}(0, y, z) &= \hat{h}_{MMS}(0, y, z), \quad \forall y \in [0, L_y], z \in [0, L_z] \\ \hat{h}(L_x, y, z) &= \hat{h}_{MMS}(L_x, y, z), \quad \forall y \in [0, L_y], z \in [0, L_z] \\ \frac{\partial \hat{h}}{\partial y}(x, 0, z) &= \frac{\partial \hat{h}_{MMS}}{\partial y}(x, 0, z), \quad \forall x \in [0, L_x], z \in [0, L_z] \\ \frac{\partial \hat{h}}{\partial y}(x, L_y, z) &= \frac{\partial \hat{h}_{MMS}}{\partial y}(x, L_y, z), \quad \forall x \in [0, L_x], z \in [0, L_z] \\ \frac{\partial \hat{h}}{\partial z}(x, y, 0) &= \frac{\partial \hat{h}_{MMS}}{\partial z}(x, y, 0), \quad \forall x \in [0, L_x], y \in [0, L_y] \\ \frac{\partial \hat{h}}{\partial z}(x, y, L_z) &= \frac{\partial \hat{h}_{MMS}}{\partial z}(x, y, L_z), \quad \forall x \in [0, L_x], y \in [0, L_y] \end{aligned} \right. \tag{26}$$

These source terms can be used as a physical law to describe the system, and also as a basis for evaluating neural networks. The specific form of the constructed solutions and source terms f used in this paper are given in Appendix C.

3 Deep learning-based neural architecture search method

3.1 Modified neural architecture search (NAS) model

The convolutional NAS approach has three main components [35]. The first one is a collection of candidate neural network structures called the search space. The second one is the search strategy and the last one is the performance

evaluation. Inspired by Park [50], we construct the system configuration of the NAS fitted to the PINNs model in Fig. 3. It consists of a sensitivity analyses (SA), search methods, a physics-informed neural networks (NN) generator, which eventually outputs the optimum neural architecture configuration and the corresponding weights and biases. A transfer learning model is eventually built based on the weights, biases and the selected neural network configurations.

3.1.1 Components of convolutional NAS

As already pointed out, the main components of the conventional neural architecture search method are

- Search Space.** The search space defines the architecture that can be represented. Combined with a priori knowledge of the typical properties of architectures well suited to the underlying task, this can reduce the size of the search space and simplify the search. For the model in this study, the priori knowledge of search space is gained from the global sensitive analysis. Figure 4b shows a common global search space with a chain structure. The chain-structured neural network architecture can be

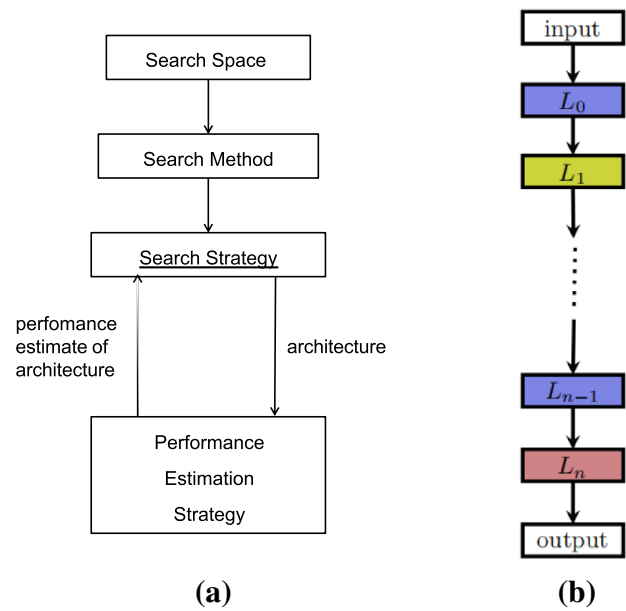
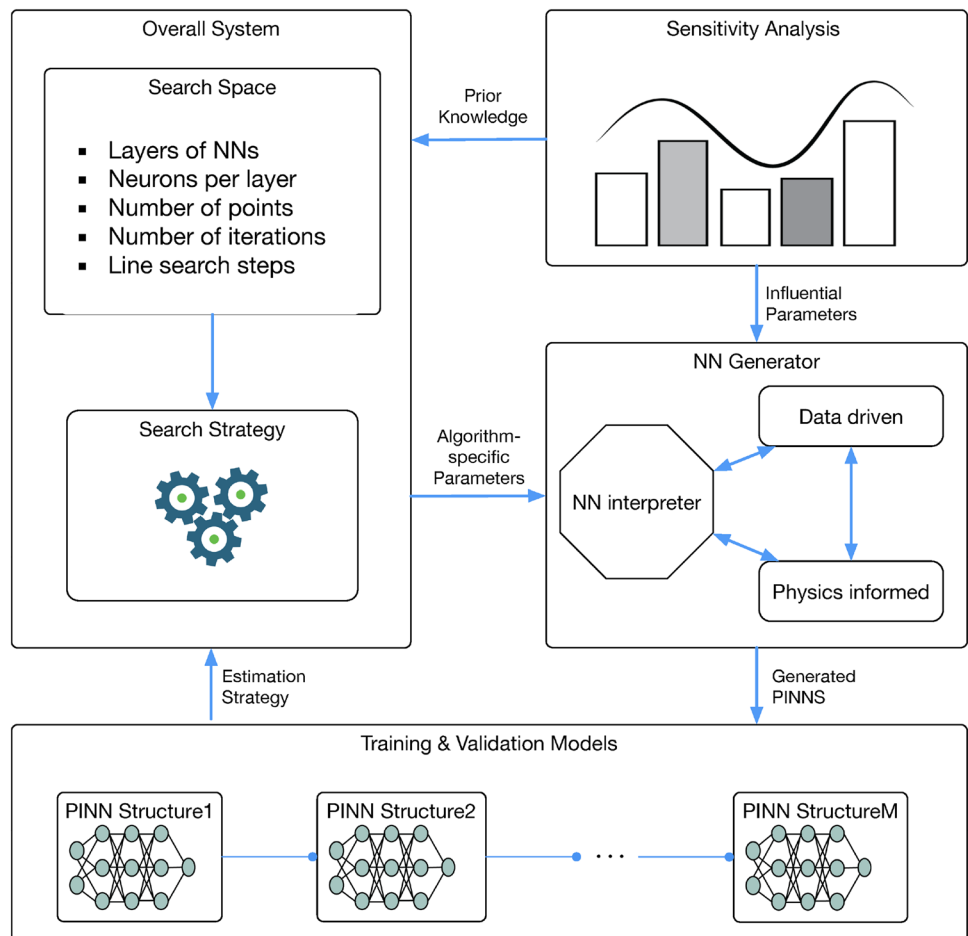


Fig. 4 a Abstract illustration of NAS methods and b search space

Fig. 3 Overall methodology for NAS



written as a sequence of n layers, where the i th layer L_i receives input from layer $i - 1$ and its output is used as input for layer $i + 1$:

$$output = L_n \odot L_{n-1} \odot \dots \odot L_1 \odot L_0, \tag{27}$$

where \odot are operations.

- **Search Method.** The search method is an initial filtering step narrowing down the search space. In this paper, hyperparameter optimizers will be used. The choice of the search space largely determines the difficulty of the optimization problem, which may result in the optimization problem remaining (i) noncontinuous and (ii) high-dimensional. Thus, some prior knowledge of the model features is needed.
- **Performance Estimation Strategy.** The simplest option for a performance estimation strategy is standard training and validation of the data for the architecture. As pointed out in Sect. 2.4, we define the relative error of manufactured solution for the performance estimation strategy:

$$\delta h = \frac{\|\hat{h} - \hat{h}_{MMS}\|_2}{\|\hat{h}_{MMS}\|_2}. \tag{28}$$

3.1.2 Modified NAS

For the modified model shown in Fig. 3, the NAS is divided into four main phases. First, a sensitivity analysis will construct the search space with less human expert knowledge. Secondly, we test several optimization strategies including randomization search method, Bayesian optimization method, Hyperband optimization method, and Jaya optimization method. The third phase is the neural network generation including the generation of physics-informed deep neural networks tailored for a mechanical model based on the information from optimization. The final phase are the training and validation models, with the input neural architectures, which outputs the estimation strategies. A suitable estimation is recommended in Eq. (28).

3.2 Neural networks generator

Mathematicians have developed many tools to approximate functions such as interpolation theory, spectral methods, finite elements, etc. From the perspective of approximation theory, neural networks can be viewed as a nonlinear smooth function approximator. Using the neural network (NN), we can obtain an output value that reflects the quality, validity, etc. of the input data, adjusts the configuration of the neural network based on this result, recalculates the results and repeats these steps until the target is reached. Physics-informed neural networks, on the other hand, add physical conservation law and prior physical knowledge

to the existing neural network, which require substantially less training data and can result in simpler neural network structures, while achieving high accuracy. The diagram of its structure is shown in Fig. 5. In this section, we further formulate the schematic on generation of physics-informed neural networks from two aspects: first, the deep neural network as universal smooth approximation methods is introduced and a simple and generalized way to introduce physics information for flow in heterogeneous media into the deep neural networks.

3.2.1 Physics-informed neural network

Physics-informed neural networks generators include neural network interpreters, which represent the configuration of a NN and physical information checkers. The neural network interpreter consists of a deep neural network with multiple layers: the input layer, one or more hidden layers and the output layer. Each layer consists of one or more nodes called neurons, shown in Fig. 5 by small coloured circles, which is the basic unit of computation. For an interconnected structure, every two neurons in neighbouring layers have a connection, which is represented by a weight, see Figure 5. Mathematically, the output of a node is computed by

$$y_i = \sigma_i \left(\sum_j w_j^i z_j^i + b^i \right) \tag{29}$$

with input z^i , weight w^j , bias b^i and activation function σ_i . Now let us define:

Definition 3.1 (Feedforward Neural Network) A generalized neural networks can be written in tuple form $((f_1, \sigma_1), \dots, (f_n, \sigma_n))$, f_i being an affine-line function ($f_i = W_i x + b_i$) that maps $R^{i-1} \rightarrow R^i$ and the activation σ_i mapping $R^i \rightarrow R^i$. The tuple form defines a continuous bounded function mapping R^d to R^n :

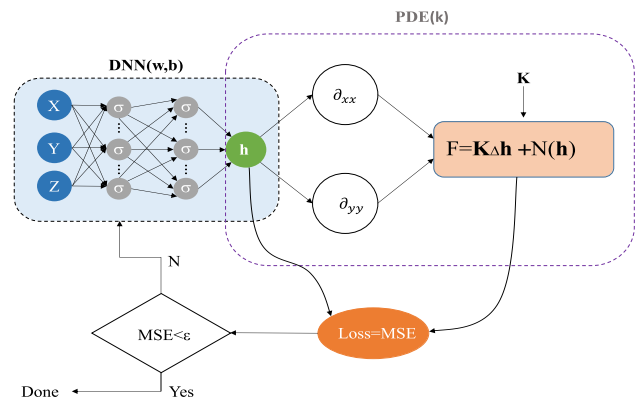


Fig. 5 Physics-informed neural networks

$$FNN : \mathbb{R}^d \rightarrow \mathbb{R}^n, \text{ with } F^n(x;\theta) = \sigma_n \circ f_n \circ \dots \circ \sigma_1 \circ f_1, \quad (30)$$

where d the dimension of the input, n the number of field variables, θ consisting of hyperparameters such as weights and biases and \circ denotes the element-wise composition.

The universal approximation theorem [51, 52] states that this continuous bounded function F with nonlinear activation σ can be adopted to capture the smoothness, nonlinear property of the system. Accordingly, it can be shown that the following theorem holds: [53]:

Theorem 1 *If $\sigma^i \in C^m(R^i)$ is nonconstant and bounded, then F^n is uniformly m -dense in $C^m(R^n)$.*

3.2.2 Deep collocation method

Collocation method is a widely used method seeking numerical solutions for partial differential and integral equations [54]. It is a popular method for trajectory optimization in control theory. A set of randomly distributed points (also known as collocation points) represents a desired trajectory that minimizes the loss function while satisfying a set of constraints. The collocation method is relatively insensitive to instabilities (such as blowing/vanishing gradients with neural networks) and is a viable way to train the deep neural network [55].

The modified Darcy equation (19) can be boiled down to the solution of a second-order differential equations with boundary constraints. Hence, we first discretize the physical domain with collocation points denoted by $\mathbf{x}_\Omega = (x_1, \dots, x_{N_\Omega})^T$. Another set of collocation points are employed to discretize the boundary conditions denoted by $\mathbf{x}_\Gamma(x_1, \dots, x_{N_\Gamma})^T$. Then the hydraulic head \hat{h} is approximated with the aforementioned deep feedforward neural network $\hat{h}^h(x;\theta)$. A loss function can thus be constructed to find the approximate solution $\hat{h}^h(x;\theta)$ by minimizing the governing equation with boundary conditions. Substituting $\hat{h}^h(\mathbf{x}_\Omega;\theta)$ into governing equation, we obtain

$$E'(\mathbf{x}_\Omega;\theta) = K(\mathbf{x})\hat{h}_{,ii}^h(\mathbf{x}_\Omega;\theta) + K_{,i}(\mathbf{x})\hat{h}_{,i}^h(\mathbf{x}_\Omega;\theta) - f(\mathbf{x}_\Omega), \quad (31)$$

which results in a physical informed deep neural network $E'(\mathbf{x}_\Omega;\theta)$. The boundary conditions illustrated in Section 2 can also be expressed by the neural network approximation $\hat{h}^h(\mathbf{x}_\Gamma;\theta)$ as

On Γ_D , we have

$$\hat{h}^h(\mathbf{x}_{\Gamma_D};\theta) = \hat{h}_{MMS}(\mathbf{x}_{\Gamma_D}). \quad (32)$$

On Γ_N ,

$$\hat{q}_n^h(\mathbf{x}_{\Gamma_N};\theta) = -K(\mathbf{x}_{\Gamma_N})\hat{h}_{MMS,n}(\mathbf{x}_{\Gamma_N}). \quad (33)$$

Note the neural network $E'(\mathbf{x};\theta)$, $q(\mathbf{x};\theta)$ shares the same parameters as $\hat{h}^h(x;\theta)$. With the generated collocation points in the domain and on the boundaries as training dataset, the field function can be learned by minimizing the mean square error loss function:

$$L(\theta) = MSE = MSE_{E'} + MSE_{\Gamma_D} + MSE_{\Gamma_N}, \quad (34)$$

with

$$MSE_{E'} = \frac{1}{N_d} \sum_{i=1}^{N_d} \|E'(\mathbf{x}_\Omega;\theta)\|^2,$$

$$MSE_{\Gamma_D} = \frac{1}{N_{\Gamma_D}} \sum_{i=1}^{N_{\Gamma_D}} \|\hat{h}^h(\mathbf{x}_{\Gamma_D};\theta) - \hat{h}_{MMS}(\mathbf{x}_{\Gamma_D})\|^2,$$

$$MSE_{\Gamma_N} = \frac{1}{N_{\Gamma_N}} \sum_{i=1}^{N_{\Gamma_N}} \|\hat{q}_n(\mathbf{x}_{\Gamma_N};\theta) + K(\mathbf{x}_{\Gamma_N})\hat{h}_{MMS,n}(\mathbf{x}_{\Gamma_N})\|^2, \quad (35)$$

where $x_\Omega \in R^N$, $\theta \in R^K$ are the neural network parameters. $L(\theta) = 0$, $\hat{h}^h(x;\theta)$ is a solution to the hydraulic head. Here, the defined loss function measures how well the approximation satisfies the physical law (governing equation) and boundaries conditions. Our goal is to find the a set of parameters θ that the approximated potential $\hat{h}^h(x;\theta)$ minimizes the loss L . If L is a very small value, the approximation $\hat{h}^h(x;\theta)$ is closely satisfying the governing equations and boundary conditions, namely

$$\hat{h}^h = \arg \min_{\theta \in R^K} L(\theta). \quad (36)$$

The solution of groundwater flow problems by the deep collocation method can be reduced to an optimization problem. To train the deep feedforward neural network, the gradient descent based optimization algorithms such as Adam are employed. The idea is to take a descent step at collocation point \mathbf{x}_i with Adam-based learning rates α_i ,

$$\theta_{i+1} = \theta_i + \alpha_i \nabla_{\theta} L(\mathbf{x}_i;\theta_i). \quad (37)$$

The process in Eq. (37) is repeated until a convergence criterion is satisfied. The combined Adam-L-BFGS-B minimization algorithm is used to train the physics-informed neural networks. This strategy consists of training the network first using the Adam algorithm, and after a defined number of iterations, we perform the L-BFGS-B optimization of the loss with a small limit of executions.

The approximation ability of neural networks for solving partial differential equations has been proven by Sirignano et al. [56]. For the stochastic analysis of porous material model, as long as the problem has a unique solution,

s.t. $\hat{h} \in C^2(\Omega)$ with its derivatives uniformly bounded and the heterogeneous hydraulic conductivity function $K(\mathbf{x})$ is assumed to be $C^{1,1}$ (C^1 with Lipschitz continuous derivative), we can conclude that

$$\exists \hat{h}^h \in F^n, \text{ s.t. as } n \rightarrow \infty, L(\theta) \rightarrow 0, \hat{h}^h \rightarrow \hat{h}. \quad (38)$$

More details can be found in [AppendixD](#) and [56].

3.3 Sensitivity analyses (SA)

Sensitivity analysis determines the influence of each parameter of the model on the output. Only the most important ones will be considered in the model calibration process. Parameters that have an impact on the output can be disregarded if they have little or no effect on the model results. This will significantly reduce the workload of model calibration [57–59]. In this work, the parameter sensitivity analysis experiment contributes to the whole NAS model by offering prior knowledge of the DCM, which helps to reduce dimensions of the search space and further improves the computational efficiency for the optimization method.

Global sensitivity analysis methods can be subdivided into qualitative ones such as Morris method [60], Fourier amplitude sensitivity test (FAST) [61] and quantitative analysis methods including Sobol method [62] or extend FAST [63]. Scholars have conducted numerous experiments to compare the advantages and disadvantages between different methods [64–66]. The results shows that Sobol’ method can provide quantitative results for SA, but it requires a large number of runs to obtain stable results. eFAST is more efficient and stable than Sobol’ method and is thus a good

alternative. The method of Morris is able to correctly screen the most and least sensitive parameters for a highly parameterized model with 300 times fewer model evaluations than the Sobol’ method. We will follow an approach proposed by Crosetto [67], i.e., first test all the hyper-parameters using the Morris method, remove the two most and least influential parameters, then filter them again, but with the eFAST method. This yields the highest accuracy in a relatively small amount of time.

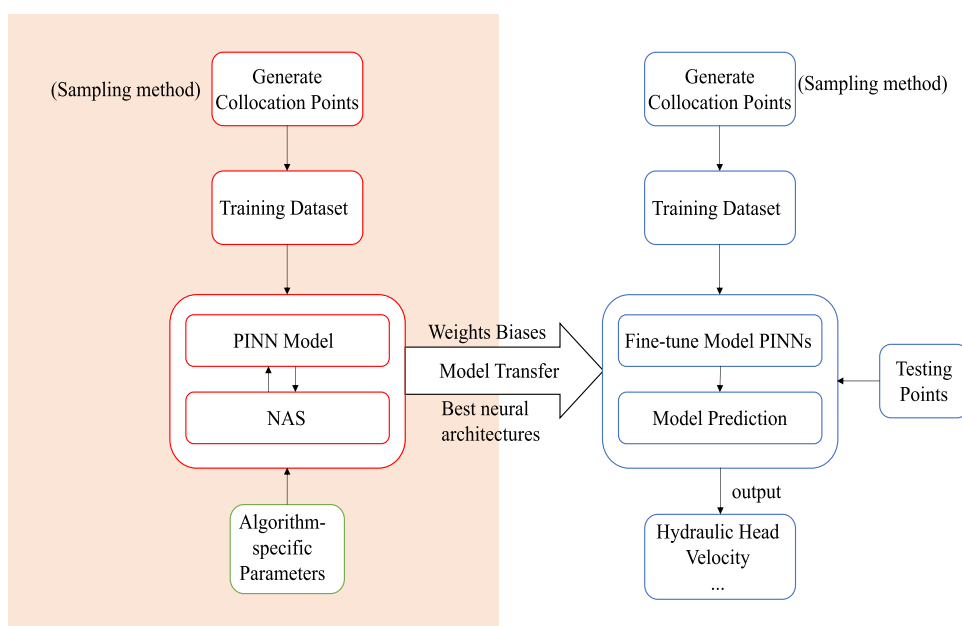
3.4 Search methods for NNs

After the sensitivity analysis, the search space is reduced and a suitable search method is employed to explore the space of the neural architecture. The search method adopts the performance metrics as rewards and learns to generate a high-performance architecture candidate. We employ the classical randomization search method, Bayesian optimization method [68] and some recently proposed optimization methods including the Hyperband algorithm [69] and Jaya algorithm [70].

3.5 Transfer learning (TL)

A combined optimizer is adopted for the model training. To improve the computational efficiency and inherit the learned knowledge from the trained model, transfer learning algorithm is added to train the model. Transfer learning stores knowledge gained while solving one problem and applies it to a different but related problem. The basic architecture of Transfer learning of this Model is shown in Figure 6. It is composed of a Pre-train model and several

Fig. 6 Transfer learning schematic



Fine-tune models. During the neural architecture procedure, the optimum neural architecture configuration is obtained through a hyperparameter optimization algorithm saving the corresponding weights and biases. Then the weights and biases are transferred to the fine-tuning model. It has been proven in the numerical example section that this inheritance method can greatly improve the learning efficiency. For different statistical parameters involved in the random log-hydraulic conductivity field, there is no need to train the whole model from scratch and the solution to the modified Darcy equation is obtained with less iterations, lower learning rates and higher accuracy.

4 Numerical examples

In this section, numerical examples in different dimensions and with various boundary conditions are studied and compared. Firstly, the influence of the exponential and Gaussian correlation functions is discussed. Next, we filter the algorithm-specific parameters by means of a sensitivity analysis and select the parameters that have the greatest impact on the model as our search space. Then, four different hyperparameter optimization algorithms are compared in both accuracy and efficiency identifying a trade-off search method for the NAS model. The relative error in Equation (28) between the predicted results and the manufactured solution is obtained to build the search strategy for the NAS model. These results are then substituted into the PINN. The results of the PINN model are compared to results obtained with FDM. All simulations are done on a 64-bit Windows 10 server with Intel(R) Core(TM) i7-7700HQ CPU, 8GB memory. The

accuracy of the numerical results are compared through the relative error of the hydraulic head defined as

$$\delta\hat{h} = \frac{\|\hat{h}_{predict} - \hat{h}_{MMS}\|}{\|\hat{h}_{MMS}\|} \quad (39)$$

$\|\cdot\|$ referring to the l^2 -norm.

4.1 Comparison of Gaussian and exponential correlations

We first compare the two correlation coefficients, Gaussian and exponential. Those two are the most widely used correlations for random field generation. We calculated the results obtained with these two correlation coefficients for the one-dimensional (1D), two-dimensional (2D), and three-dimensional (3D) stochastic groundwater flow cases, respectively, with the same parameters. The number of hidden layers and the neurons per layer are uniformly set to 6 and 16.

4.1.1 One-dimensional groundwater flow with both correlations

The non-homogeneous 1D flow problem for Darcy equation can be reduced to Eq (19) subjected to Eq (C16). The hydraulic conductivity K is constructed from Eq (18) by the random spectral method, see Eq (C17). The source term f of the manufactured solution, Eq (C15), is obtained from Eq (C18). The detailed derivation can be found in Appendix C.

The relative errors $\delta\hat{h}$ of the predicted hydraulic heads for the exponential and Gaussian correlation of the $\ln(K)$ field are shown in Tables 1 and 2. The Gaussian correlation is more accurate for all N and σ^2 . With transfer learning model,

Table 1 $\delta\hat{h}$ for 1D case computed with exponential correlation in different variance and number of modes

N	σ^2					
	0.1		1		3	
	without TL	with TL	without TL	with TL	without TL	with TL
500	1.184e-3	1.797e-4	1.100e-2	4.884e-4	1.159e-1	5.360e-4
1000	2.437e-2	2.354e-4	9.026e-3	5.282e-4	3.752e-2	1.754e-3
2000	5.789e-4	1.007e-4	3.813e-3	5.939e-4	3.532e-2	4.316e-3

Table 2 $\delta\hat{h}$ for 1D case computed with Gaussian correlation in different variance and number of modes for one-dimensional case

N	σ^2					
	0.1		1		3	
	without TL	with TL	without TL	with TL	without TL	with TL
500	1.211e-4	1.137e-4	9.065e-4	1.204e-4	7.539e-3	1.690e-4
1000	1.317e-4	1.133e-4	8.312e-4	1.200e-4	2.864e-3	3.662e-4
2000	1.158e-4	1.333e-4	2.811e-4	1.538e-4	2.904e-3	5.756e-4

the accuracy even improves. The predicted hydraulic head, the velocity and the manufactured solution for both exponential and Gaussian correlations with $\sigma^2 = 0.1$ and $N = 2000$ are shown in Fig. 7. The predicted results nearly coincide with the manufactured solution Eq (C15) in the 1D domain.

The $\log(Loss)$ vs. iteration graph for different parameters in constructing the random log-hydraulic conductivity field can be found in Fig. 8 and shows (1) the loss value for the Gaussian correlation is much smaller than the exponential correlation for all σ^2 and N values. (2) With transfer learning, the loss function drops significantly faster and the number of required iterations is greatly reduced. In summary, the Gaussian correlation outperforms the exponential one in generating random log-hydraulic conductivity fields for the PINN.

4.1.2 Two-dimensional groundwater flow with both correlations

To solve the non-homogeneous 2D flow problem for Darcy equation, the manufactured solution in Eq (C19) is adopted. The hydraulic conductivity K is constructed from Eq (18) by the Radom spectral method, see Eq (C21). The source term f is computed according to Eq (C22), see Appendix C for more details. The exponential and Gaussian correlations for the heterogeneous hydraulic conductivity are tested with varying σ^2 and N values. The same conclusion can be drawn from Tables 3 and 4 that with increasing N , the predicted hydraulic head becomes more accurate, however when σ^2 becomes bigger, the accuracy deteriorates in most cases. The PINNs with Gaussian correlation based hydraulic conductivity

Fig. 7 One-dimensional hydraulic head when $\sigma^2 = 0.1$, $N = 2000$ with (a) exponential correlation and b Gaussian correlation

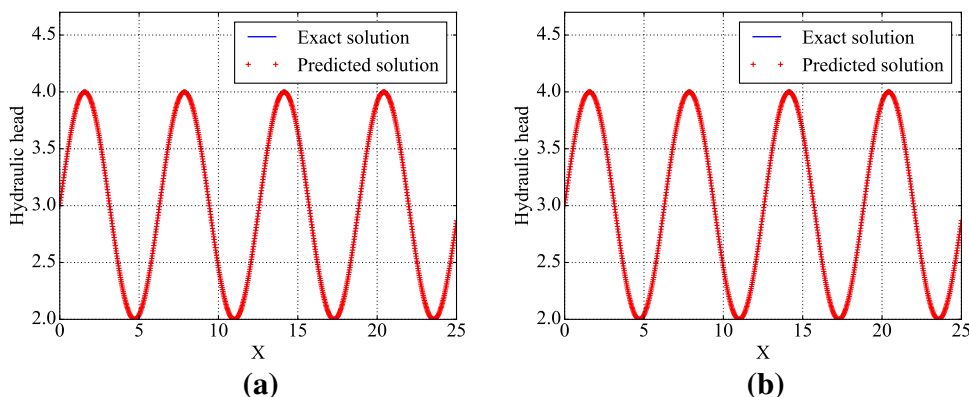


Fig. 8 One-dimensional logarithm loss function with (a) exponential correlation and b Gaussian correlation

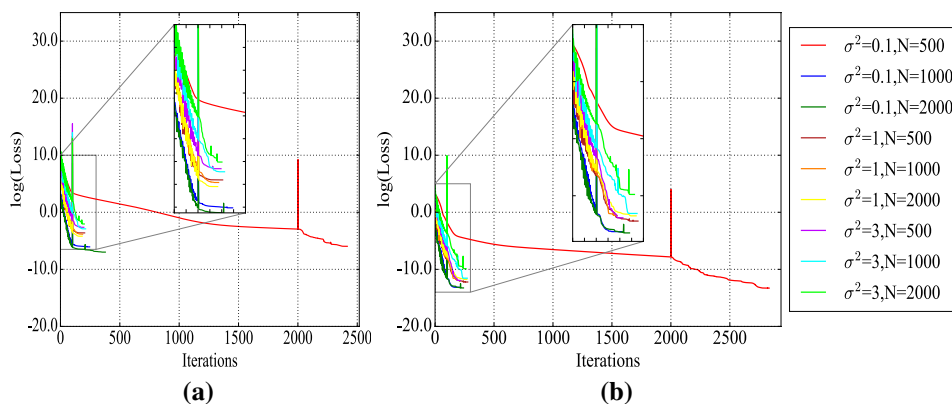


Table 3 δh for 2D case computed with exponential correlation in different variance and number of modes

N	σ^2					
	0.1		1		3	
	Without TL	With TL	Without TL	With TL	Without TL	With TL
500	6.777e-2	9.345e-2	3.817e-2	4.635e-2	2.560e-1	5.5080e-2
1000	1.479e-2	4.832e-2	1.790e-3	8.157e-2	9.739e-2	7.201e-2
2000	7.147e-3	4.829e-2	4.471e-2	4.924e-2	9.357e-2	1.187e-1

outperforms the exponential correlations. The contour plots of the predicted hydraulic head and velocity as well as the manufactured solution for both exponential and Gaussian correlations with $\sigma^2 = 0.1$ and $N = 2000$ are listed in the supplementary material, in Figs. S1, S2, S3 and S4. The predicted physical patterns agree well with the manufactured solution Eq (C15).

The $\log(Loss)$ vs. iteration graph for different parameters in constructing the random log-hydraulic conductivity field is illustrated in Fig. 9. The loss for the PINN with Gaussian correlations is much smaller and decreases faster while the loss is not fully minimized for the exponential correlations. With transfer learning, the loss function converges with less iterations, which largely reduces the training time. Also for the two-dimensional groundwater flow, the Gaussian correlation shows much better performance than the exponential correlation.

4.1.3 Three-dimensional groundwater flow with both correlations

Let us now focus on the 3D non-homogeneous Darcy equation problem [43]. The manufactured solution in Equation (C30) is adopted. The hydraulic conductivity K is constructed according to Eq (C28). The source term f is devised from Eq (C29). The exponential and Gaussian correlations for the heterogeneous hydraulic conductivity and varying σ^2 and N values are tested again. Tables 5 and 6 list the relative error of the hydraulic head for DCM with and without transfer learning. For different σ^2 and N values, the performance of the PINN varies largely for both correlations. The same tendency as in 1D and 2D holds: The Gaussian correlation outperforms the exponential one and transfer learning has a significant impact on the computational cost. The hydraulic head predicted by both correlation functions with $\sigma^2 = 0.1$ and $N = 2000$ are shown in Fig. S5, S6, S7 and S8 (Fig. 10).

Table 4 δh for 2D case computed with Gaussian correlation in different variance and number of modes

N	σ^2					
	0.1		1		3	
	Without TL	With TL	Without TL	With TL	Without TL	With TL
500	9.974e-4	9.842e-4	3.530e-3	7.900e-4	3.053e-2	2.475e-3
1000	2.980e-4	6.954e-4	6.270e-3	1.527e-3	3.855e-2	2.904e-3
2000	7.299e-4	5.717e-4	7.719e-3	1.704e-3	7.486e-2	2.506e-2

Fig. 9 Two-dimensional logarithm loss function with (a) exponential correlation and (b) Gaussian correlation

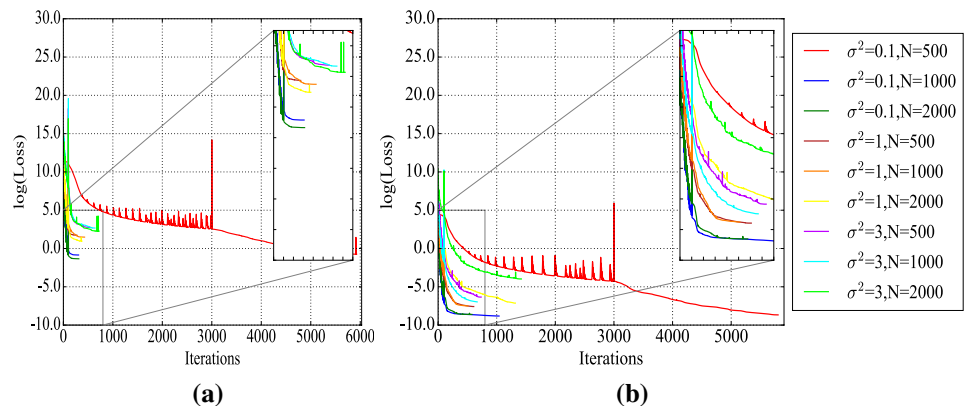


Table 5 δh for 3D case computed with exponential correlation in different variance and number of modes

N	σ^2					
	0.1		1		3	
	Without TL	With TL	Without TL	With TL	Without TL	With TL
500	3.419e-3	1.529e-2	1.131e0	5.885e-2	6.264e-1	7.340e-2
1000	1.219e-1	8.333e-3	3.257e-1	5.668e-2	1.055e0	8.982e-2
2000	5.667e-2	1.230e-2	4.287e-1	6.161e-2	1.204e0	5.313e-2

Table 6 $\delta\hat{h}$ for 3D case computed with Gaussian correlation in different variance and number of modes

N	σ^2					
	0.1		1		3	
	Without TL	With TL	Without TL	With TL	Without TL	With TL
500	1.161e-2	5.439e-3	6.247e-3	1.540e-2	9.294e-2	1.078e-2
1000	1.187e-3	5.620e-3	4.087e-2	1.205e-2	3.004e-1	1.996e-2
2000	8.342e-3	1.661e-2	1.218e-2	1.278e-2	1.562e-1	1.952e-2

Fig. 10 Three dimensional logarithm loss function with (a) exponential correlation and (b) Gaussian correlation

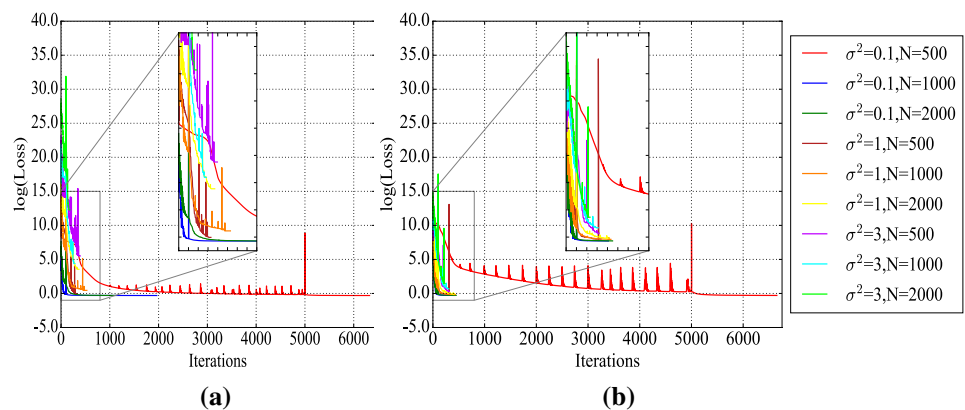


Table 7 Calculation time required in different dimensions

Correlation	Dimension					
	1		2		3	
	Without TL	With TL	Without TL	With TL	Without TL	With TL
Exponential	30s	3.0s	97s	6.5s	58s	7.8s
Gaussian	28s	3.0s	58s	9.8s	52s	5.9s

The computational cost of the DCM with both correlation functions are shown in Table 7. The Gaussian correlation function is not only more accurate but also more efficient.

In summary, the comparison reveals that the loss function in the Gaussian correlation tends to decrease faster than the exponential, and that the error using Gaussian is much smaller and more stable than the exponential one. The Gaussian correlation also requires less computation time. Note also that the loss function of the exponential correlation coefficient leads to gradient explosion when the number of collocation points exceeds a value of 150, while this is not observed from the Gaussian correlation coefficient, even for much larger number of collocation points. Subsequently, we will only use the Gaussian correlation.

4.2 Sensitivity analysis results

The sensitivity analysis should eliminate irrelevant variables to finally reduce the computational cost for the

Table 8 Hyper-parameters and their intervals in groundwater flow problem

Hyper-parameters	Intervals
Layers of NNs	[2, 30]
Neurons per layer	[10, 50]
Number of iterations	[1500, 3000]
Number of collation points	[800, 2000]
Maximum line search of L-BFGS algorithm	[30, 300]

hyperparameter optimizer. The hyper-parameters in this flow problem are listed in Table 8.

The sensitivity analysis results obtained by the hybrid Morris-eFAST method are shown as follows:

From Figs. 11 and 12, we conclude that the number of layers and the neurons have the greatest impact. In contrast, the maximum line search of L-BFGS has almost no effect. So we remove the number of layers and the maxls, and continue to calculate the sensitivity of the remaining three in

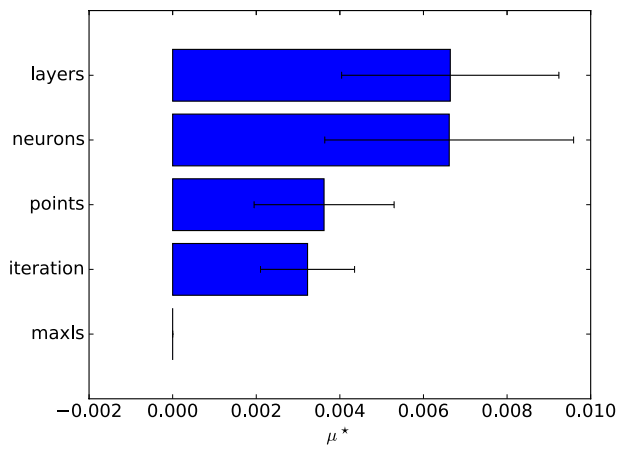


Fig. 11 Sensitivity histogram of Morris

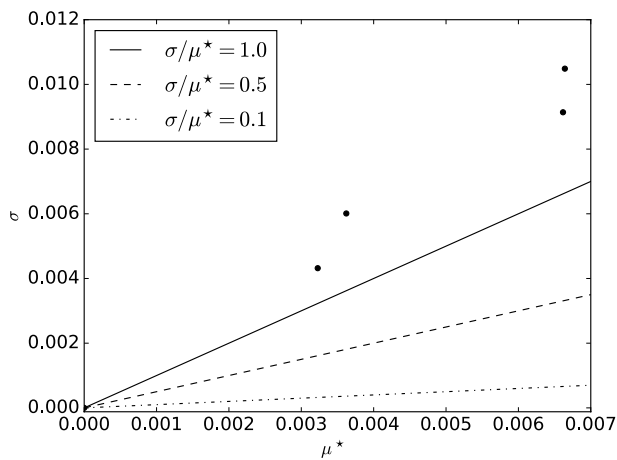


Fig. 12 Morris μ^* and σ computed using the Morris screening algorithm

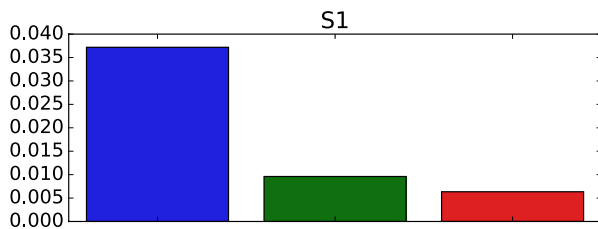


Fig. 13 Sensitivity histogram of eFAST

the eFast model. The results are summarized in Fig. 13. The neurons are the second most important parameter. Hence, the layers and the neurons are chosen as the hyper-parameters in the search space for the automated machine learning approach.

4.3 Hyperparameter optimizations method comparison

To select the most suitable hyperparameter optimization algorithm, we use the two hyperparameters selected from the sensitivity analysis results in Sect. 4.2 as search variables in search space and compute the four algorithms presented in the previous section. All remaining conditions are equal. The horizontal coordinates in Fig. 14 represent the number of neurons per layer and the vertical coordinates refer to the number of hidden layers.

The time required for each method and the search accuracy are shown in Table 9.

The Bayesian method gives the best accuracy in the shortest time and will subsequently be adopted. Due to the limited search numbers, the optimal solution searched by the algorithm is not necessarily the best one and will gradually approach the optimal configuration as the number of searches increases. For two and three dimension cases, the optimal configuration are illustrated in Figs. 15 and 16.

The optimal configuration obtained after screening is shown in Table 10. These neural network configurations will be used as input parameters for the next numerical tests.

4.4 Model validation in different dimensions

Now we solve the modified Darcy Eq (19) by the NAS based DCM and the optimized configurations from Sect. 3, i.e. we fix σ_2 to 0.1, and N to 1000. The manufactured solutions can be found in Appendix C. The results are compared to solutions obtained from the finite difference method.

4.4.1 One-dimensional case model validation

The 1D manufactured solution is Eq (C15). We validate the two methods by comparing the hydraulic head in the x -direction in the interval $[0, 25]$, see Fig. 17.

4.4.2 Two-dimensional case model validation

The 2D manufactured solution is Equation (C23) and we focus on the hydraulic head and velocity in the x -direction, along the midline at $y = 10$ within the interval $[0, 20] \times [0, 20]$.

Figure 18 demonstrates that both methods match well with the exact solution for the hydraulic head. However, as seen from Figure 19, the FDM method poorly predicts v_x while the proposed DCM still agrees well.

4.4.3 Three-dimensional case model validation

The manufactured solution for 3D the case is given by Equation (C30). We again compute the hydraulic head and

Fig. 14 Neural network configuration search results with (a) Randomization search method; (b) Bayesian optimization; (c) hyperband optimization; (d) Jaya optimization

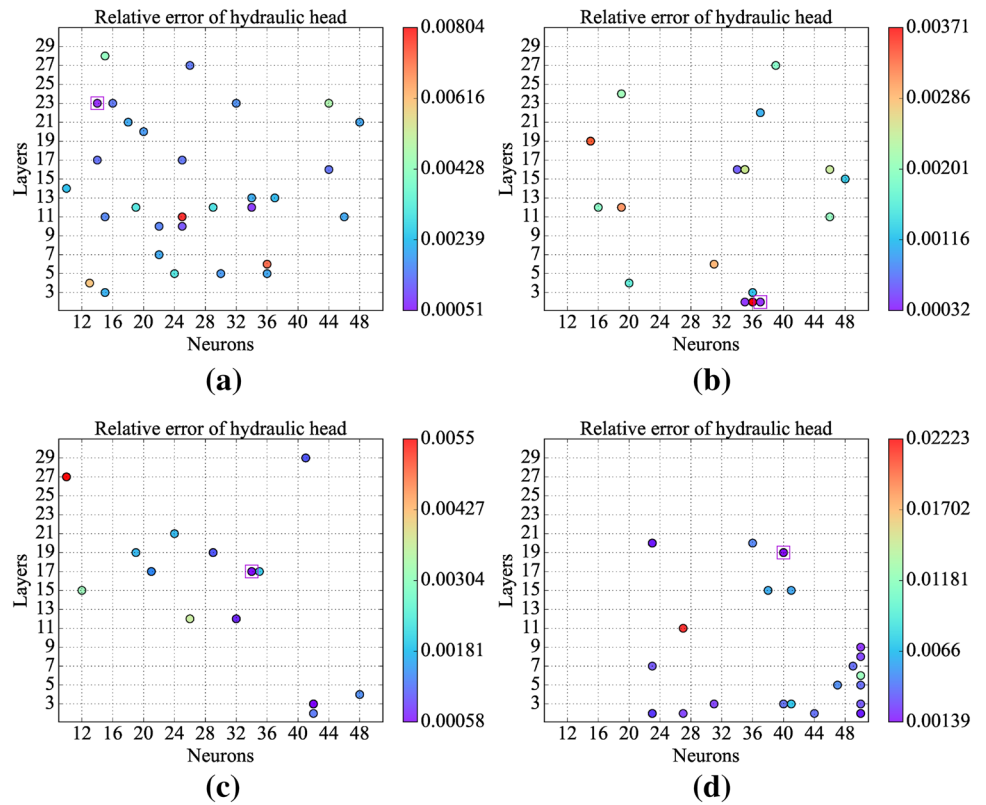


Table 9 Hyper-parameters search results with different algorithms

Algorithms	Time	Relative error
RSM	1830s	0.00051
Bayesian	1395s	0.00032
Hyperband	1449s	0.00058
Jaya	1757s	0.00139

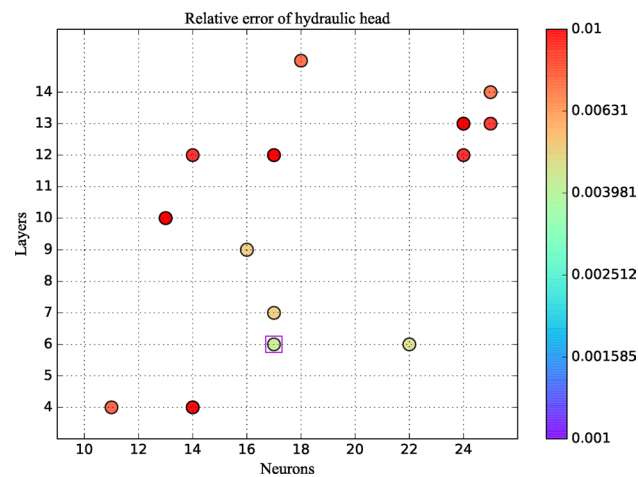


Fig. 15 Neural network configuration search results of Bayesian optimization in two dimension

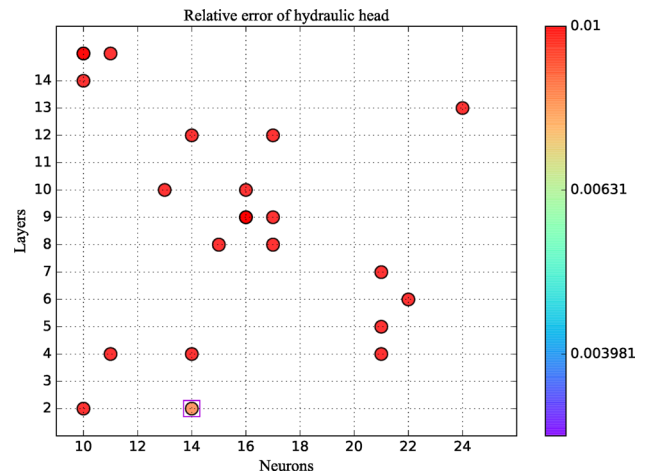


Fig. 16 Neural network configuration search results of Bayesian optimization in three dimension

Table 10 Neural architecture search results with Bayesian optimization

Dimension	Layer	Neurons
1D	2	37
2D	6	17
3D	2	14

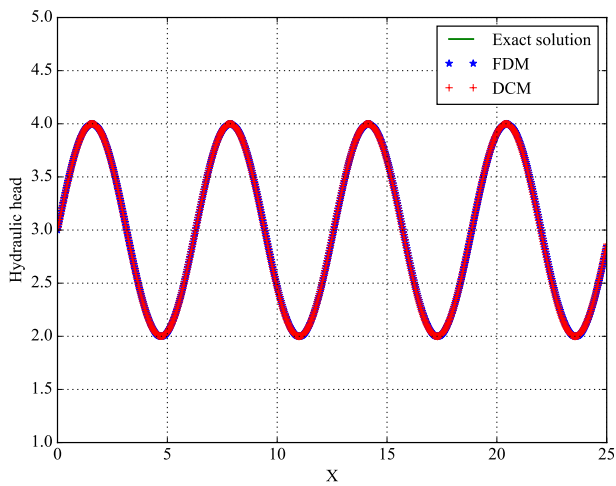


Fig. 17 Hydraulic head calculated by FDM and DCM methods in one dimension

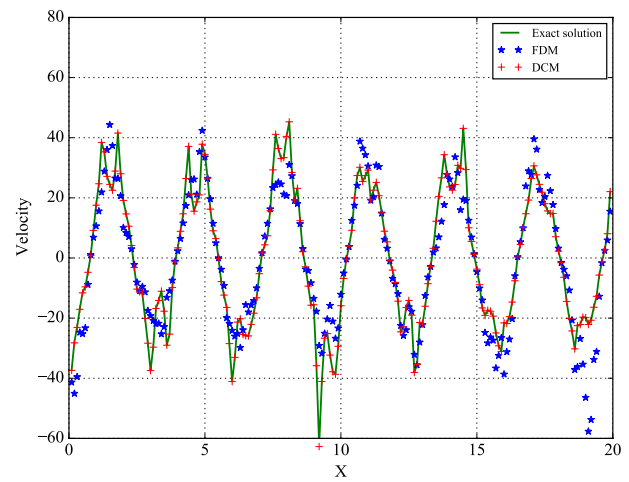


Fig. 19 Velocity in x-direction along $y = 10$ calculated by FDM and DCM methods in two dimension

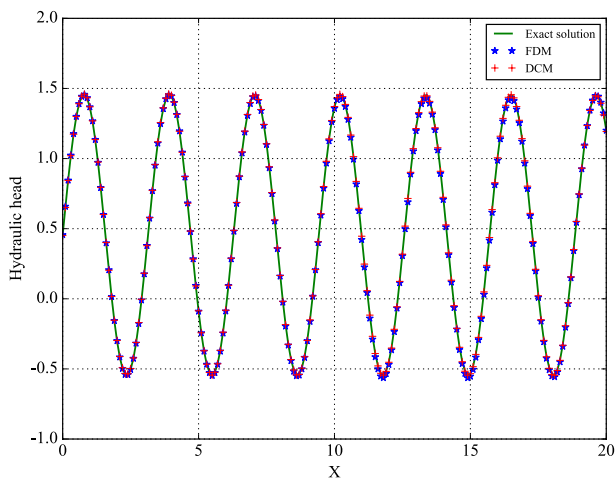


Fig. 18 Hydraulic head along $y = 10$ calculated by FDM and DCM methods in two dimension

velocity in the x -direction at $y = 1, z = 0.5$ over the interval $[0, 5] \times [0, 2] \times [0, 1]$. The results are summarized in Tables 11 and 12. While the results obtained by the FDM is rather poor, the DCM approach still provides solutions close to the exact one (Fig. 20).

The higher the dimensionality of the problem, the more pronounced is the difference between the two methods (FDM and DCM). The FDM method requires an extremely dense discretization, which in turn leads to a high computational cost. DCM yields very accurate results even for very few training points. Transfer learning further reduces the computational cost while simultaneously slightly improving the accuracy. The contour plots for the hydraulic head and velocity are visualized in Figs. 21 and 22.

Table 11 Solving Darcy equation with FDM

Dimension	1	2	3
Results			
Relative error	6.443e-5	0.017	5.711
Time	2.8s	180s	1245s

The isosurface diagrams of the predicted head and velocity are illustrated in Figs. 23 and 24.

5 Conclusion

In this paper, we proposed a NAS-based stochastic DCM employing sensitivity analysis and transfer learning to reduce the computational cost and improve the accuracy. The random spectral method in closed form is adopted for the generation of log-normal hydraulic conductivity fields. It was calibrated to generate the heterogeneous hydraulic conductivity field with Gaussian correlation function. Exploiting sensitivity analysis and comparing hyperparameter selection methods, the Bayesian algorithm was identified as the most suitable optimizer for the search strategy in the NAS model. While the sensitivity analysis and NAS mean additional cost, it still reduces the computational cost for a specified accuracy. Furthermore, for certain type of problems, it is not necessary to repeat this steps. To validate our approach, groundwater flow in highly heterogeneous aquifers are considered.

Since no feature engineering is involved in our PINN, the NAS based DCM can be considered as truly automated

Table 12 Solving Darcy equation with DCM

Results	1		2		3	
	without TL	with TL	without TL	with TL	without TL	with TL
Relative error	1.369e-4	1.195e-4	4.262e-3	4.405e-3	8.915e-3	8.864e-3
Time	14.3s	1.5s	108.5s	9.8s	32.2s	1.9s

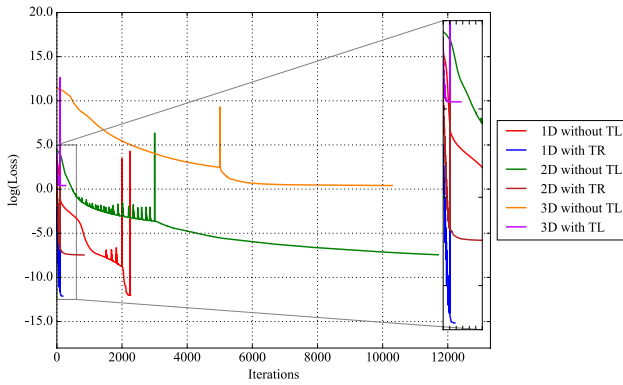
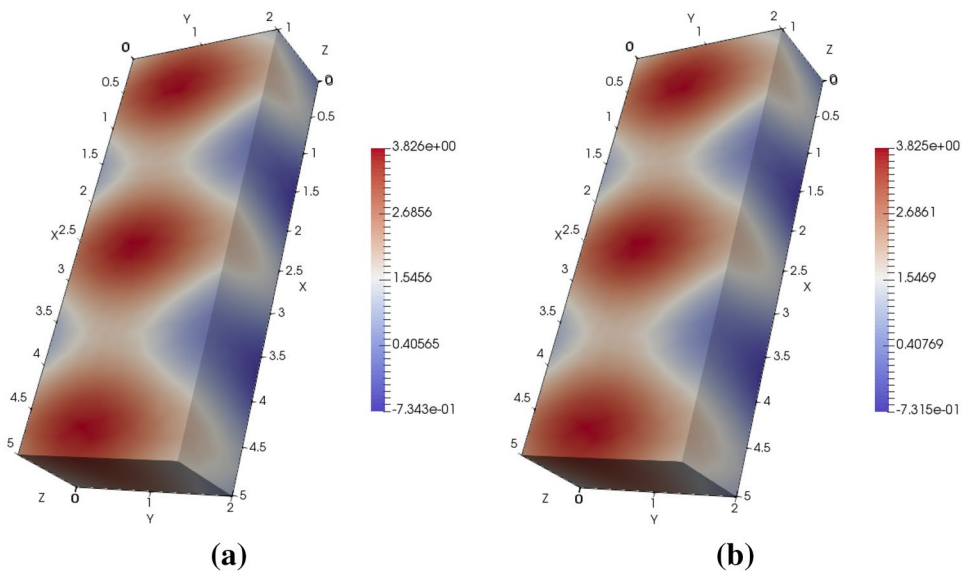


Fig. 20 Logarithm loss function with and without transfer learning

“meshfree” method. It approximates any continuous function. The presented automated DCM is simple to implement as it defines only the definition of the underlying BVP/IBVP and boundary conditions.

Through several numerical examples in 1D, 2D and 3D, we showed that the presented NAS-based DCM significantly outperforms the FDM method in terms of computational efficiency and accuracy. The benefits become more pronounced with increasing dimension and hence ‘complexity’. Note that the presented NAS-based DCM outperforms the FDM

Fig. 21 Three-dimensional hydraulic head when $\sigma^2 = 0.1$, $N = 1000$ with Gaussian correlation (a) exact solution and b predict solution



even if all the steps from sensitivity analysis, optimization and training are accounted for. However, once those deep neural networks are trained, they can be used to evaluate the solution at any desired points with minimal additional computation time. Besides those advantages, the limitations for the proposed stochastic deep collocation method can be encapsulated in the computational cost in neural architecture search model for large multi-scale complex problems and that the gradient-descent based optimizer may get stuck in a local optimal. Those topics will be further investigated in our future research regarding a more generalised and improved NAS-based deep collocation method.

Appendix

Data flow in stochastic deep collocation method

The submodules for the stochastic deep collocation method and data flow inside the whole model are illustrated in Figure 25. There are three main submodules involved for stochastic deep collocation method: neural architecture search

Fig. 22 Three-dimensional velocity when $\sigma^2 = 0.1$, $N = 1000$ with Gaussian correlation (a) exact solution and b predict solution

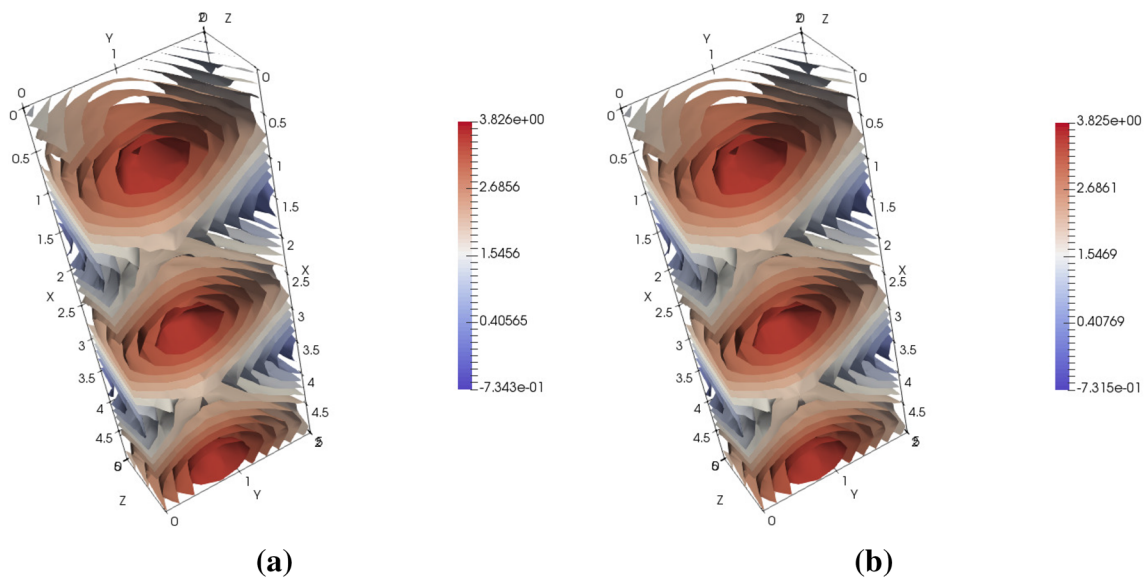
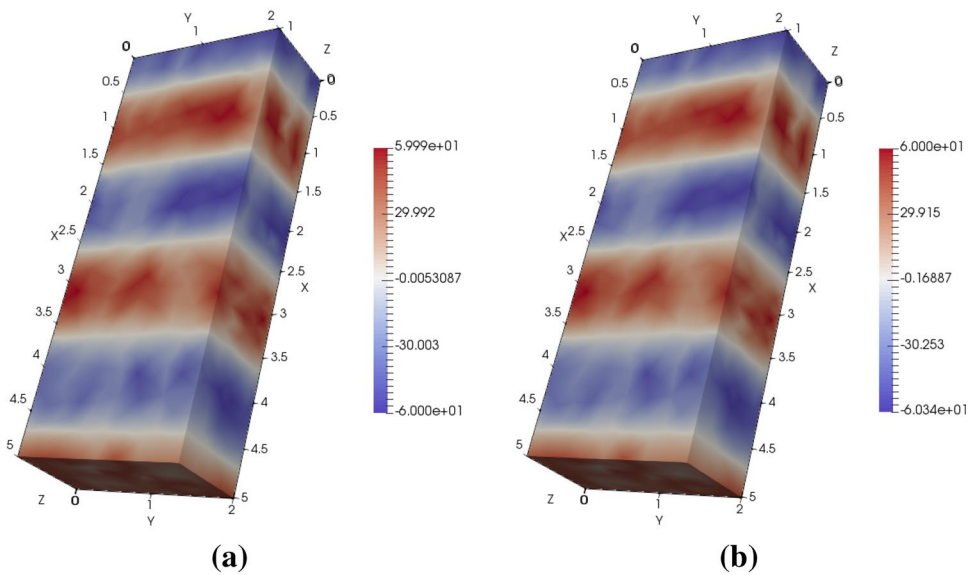


Fig. 23 Three-dimensional isosurface diagram of hydraulic head when $\sigma^2 = 0.1$, $N = 1000$ with Gaussian correlation (a) exact solution and b predict solution

method, physics-informed deep neural networks and a transfer learning technique. First, the neural architecture search model shown on the left module is performed to find a physics-informed deep neural network with optimal performance and then the deep collocation method is constructed based on the deep neural network with searched configurations on the right. To enhance the model generality and efficiency, the neural network settings is inherited for transfer learning. In general, the data first flow inside neural architecture search model to help find an optimal deep neural network architecture and corresponding parameter settings and then to the PINNs model for stochastic analysis of heterogeneous

porous media with initial hydraulic and material parameters. Finally, the data will flow to transfer learning module for stochastic flow analysis in more general cases to reduce the computational costs and enhance the model accuracy and generality.

Choices of random variables \mathbf{k}

From Section 2.2, we can derive the following probability density function (PDF) $p(\mathbf{k})$ for exponential and Gaussian correlations:

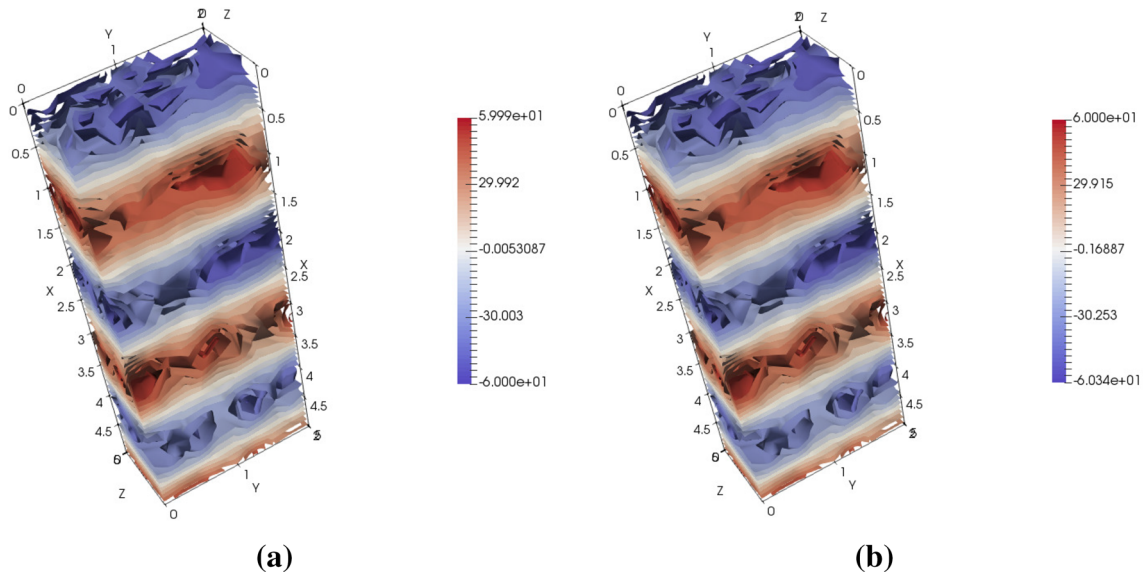


Fig. 24 Three-dimensional isosurface diagram of velocity when $\sigma^2 = 0.1$, $N = 1000$ with Gaussian correlation (a) exact solution and a predict solution

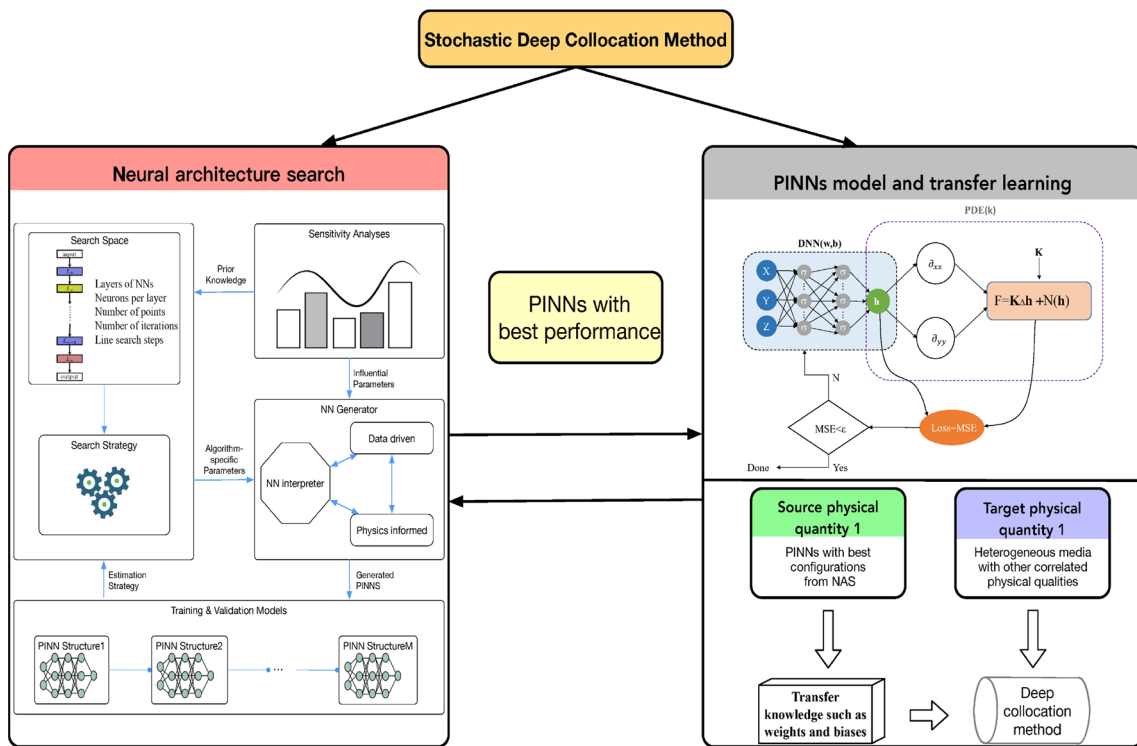


Fig. 25 Data flow in stochastic deep collocation method and its submodules

$$p(\mathbf{k}) = \lambda^d \frac{\Gamma[\frac{d+1}{2}]}{(\pi(1 + (\mathbf{k}\lambda)^2))^{\frac{d+1}{2}}} \tag{B1}$$

$$p(\mathbf{k}) = \pi^{d/2} \lambda^d e^{-(\pi \mathbf{k}\lambda)^2}, \tag{B2}$$

where Γ means the gamma function, with $\Gamma(n) = (n - 1)!$ and $\Gamma(n + 1/2) = \frac{\sqrt{\pi}}{2^n} (2n - 1)!!$ for $n = 1, 2, 3, \dots$

For Gaussian correlations in three dimensional case, the PDF of \mathbf{k} can be transformed into the following form:

$$p(\mathbf{k}) = \left(\sqrt{\pi}\lambda_1 e^{-(\pi k_1 \lambda_1)^2} \right) \left(\sqrt{\pi}\lambda_2 e^{-(\pi k_2 \lambda_2)^2} \right) \left(\sqrt{\pi}\lambda_3 e^{-(\pi k_3 \lambda_3)^2} \right). \tag{B3}$$

Each part of Equation (B3) can be considered as a normal distribution with $\mu = 0$ and $\sigma = \frac{1}{\sqrt{2\pi}\lambda_i}$. Thus, the random vector \mathbf{k} can be simulated by the formula $\mathbf{k} = \frac{1}{\sqrt{2\pi}}(\mu_1/\lambda_1, \mu_2/\lambda_2, \mu_3/\lambda_3)$, where μ_i are independent standard Gaussian random variables.

The \mathbf{k} in the two-dimensional case can be derived by analogy from the above inference as $\mathbf{k} = \frac{1}{\sqrt{2\pi}}(\mu_1/\lambda_1, \mu_2/\lambda_2)$.

For exponential correlations in two-dimensional case, the PDF of \mathbf{k} can be transformed into the following form:

$$p(\mathbf{k}) = \frac{\lambda_1 \lambda_2}{2\pi(1 + (k_1 \lambda_1)^2 + (k_2 \lambda_2)^2)^{\frac{3}{2}}}. \tag{B4}$$

A possible solution to calculate the cumulative distribution function (CDF) is the transformation from Cartesian into polar coordinates, i.e. a representation like:

$$\begin{aligned} k_1 &= r \cos(2\pi \hat{h}) / \lambda_1, \\ k_2 &= r \sin(2\pi \hat{h}) / \lambda_2. \end{aligned} \tag{B5}$$

Here \hat{h} is a uniformly distributed random variable and r is a random variable distribute according to

$$p_r(r) = \frac{2\pi r p(r)}{\lambda_1 \lambda_2}. \tag{B6}$$

Integrating Equation (B6) yields the CDF

$$\begin{aligned} F(r) &= \int_{-\infty}^r p_r(r) dr \\ &= \int_{-\infty}^r \frac{r}{(1+r^2)^{3/2}} dr \\ &= -\frac{1}{(1+r^2)^{1/2}} \Big|_{-\infty}^r \\ &= -\frac{1}{(1+r^2)^{1/2}}. \end{aligned} \tag{B7}$$

Choose a uniformly distributed random variable μ , the inverse function $k = F^{-1}(\mu)$ can be obtained

$$\begin{aligned} \mu &= -\frac{1}{(1+r^2)^{1/2}} \\ \mu^2 &= \frac{1}{1+r^2} \\ r &= \sqrt{1/\mu^2 - 1}. \end{aligned} \tag{B8}$$

Substitute Equation (B5) into Equation (B5), we get the $k_1 = (1/\mu^2 - 1)^{1/2} \cos(2\pi \hat{h}) / \lambda_1$, and $k_2 = (1/\mu^2 - 1)^{1/2} \sin(2\pi \hat{h}) / \lambda_2$.

For exponential correlations in three-dimensional case, the PDF of \mathbf{k} can be transformed into the following form:

$$p(\mathbf{k}) = \frac{\lambda_1 \lambda_2 \lambda_3}{\pi^2(1 + (k_1 \lambda_1)^2 + (k_2 \lambda_2)^2 + (k_3 \lambda_3)^2)^2}. \tag{B9}$$

A similar procedure can be used, where spherical instead of polar coordinates are used

$$\begin{aligned} k_1 &= r \sin(\theta) \cos(2\pi \gamma) / \lambda_1, \\ k_2 &= r \sin(\theta) \sin(2\pi \gamma) / \lambda_2, \\ k_3 &= r \cos(\theta) / \lambda_3. \end{aligned} \tag{B10}$$

Here γ is again a uniformly distributed random variable and θ is given as

$$\theta = \arccos(1 - 2\xi), \tag{B11}$$

with ξ being a uniformly distributed random variable. The two random variables were chosen with reference to Weisten’s research on generating random points on the surface of a unit sphere [71]. The radius r is distributed according to

$$p_r(r) = 4\pi r^2 p(r). \tag{B12}$$

The CDF can be calculated as follows:

$$\begin{aligned} F(r) &= \int_{-\infty}^r p_r(r) dr \\ &= \int_{-\infty}^r \frac{4r^2}{\pi(1+r^2)^2} dr \\ &= \frac{2}{\pi} \left(-\frac{r}{1+r^2} \Big|_{-\infty}^r - \int_{-\infty}^r -\frac{1}{1+r^2} dr \right) \\ &= \frac{2}{\pi} \left(\arctan(r) - \frac{r}{1+r^2} \right). \end{aligned} \tag{B13}$$

Choose a uniformly distributed random variable γ_1 , r can be obtained by solving the next Equation (B14):

$$\frac{2}{\pi} \left(\arctan(r) - \frac{r}{1+r^2} \right) = \gamma_1. \tag{B14}$$

Manufactured solutions

For the 1D case, we have selected the following manufactured solution according to benchmark [43]:

$$\hat{h}_{MMS}(x) = 3 + \sin(x), \text{ with } x \in [0, 25]. \tag{C15}$$

This leads to the following Dirichlet boundary conditions :

$$\begin{cases} \hat{h}(0) = 3, \\ \hat{h}(25) = 3 + \sin(25). \end{cases} \quad (C16)$$

The function K is now given by

$$K(x) = C_1 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}))\right), \quad (C17)$$

where we use the shorthand notations $C_1 = \langle K \rangle \exp(-\frac{\sigma^2}{2})$ and $C_2 = \sigma \sqrt{\frac{2}{N}}$. And the source term f has the following form:

$$\begin{aligned} f(x) = & C_1 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}))\right) \\ & \cdot \left((-2\pi)C_2 k_{i,1} \sum_{i=1}^N \sin(\xi_1 + 2\pi(k_{i,1}x + k_{i,2})) \right. \\ & \left. \cos(x) - \sin(x) \right). \end{aligned} \quad (C18)$$

The Python code for the generation of K is shown as follows:

```
import numpy as np
def hydraulic_conductivity(x, N, s, k, k1, k2, xi):
    #Inputs lists
    #x—collocation points
    #N—number of random modes
    #s—the variance
    #k—the mean
    #ki—random variable for every mode
    #xi—random variables.
    c1 = k * np.exp(-s / 2)
    c2 = np.sqrt(s * 2 / N)

    K_inner = np.zeros((x.shape[0], 1), dtype=np.float32)
    K_x_inner = np.zeros((x.shape[0], 1), dtype=np.float32)

    for i in range(int(N)):
        K_inner += c2 * np.cos(xi[i, :] + \
        2 * np.pi * (k1[i, :] * x + k2[i, :]))
        K_x_inner += c2 * (-2 * np.pi * k1[i, :] * \
        np.sin(xi[i, :] + 2 * np.pi * \
        (k1[i, :] * x + k2[i, :])))
    K = (c1 * np.exp(K_inner))
    K_x = (c1 * np.exp(K_inner) * K_x_inner)
    f_source = K_x * np.cos(x) - K * np.sin(x)
    return K, K_x, f_source
```

For the 2D case, we consider the following smooth manufactured solution :

$$\hat{h}_{MMS}(x, y) = 1 + \sin(2x + y), \quad (C19)$$

with $x \in [0, 20]$ and $y \in [0, 20]$,

along with the Dirichlet and Neumann boundary conditions:

$$\begin{cases} \hat{h}(0, y) = 1 + \sin(y), & \forall y \in [0, 20], \\ \hat{h}(20, y) = 1 + \sin(2 \times 20 + y), & \forall y \in [0, 20], \\ \frac{\partial \hat{h}}{\partial y}(x, 0) = \cos(2x), & \forall x \in [0, 20], \\ \frac{\partial \hat{h}}{\partial y}(x, 20) = \cos(2x + 20), & \forall x \in [0, 20]. \end{cases} \quad (C20)$$

The function K is now given by

$$K(x, y) = C_1 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y))\right), \quad (C21)$$

where we use the shorthand notations C_1 and C_2 same as in 1-dimensional case. And the source term f has the following form:

$$\begin{aligned}
 f(x, y) = & 2C_1 C_2 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y))\right) \\
 & \cdot \sum_{i=1}^N \left(-2\pi k_{i,1} \sin(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y))\right) \\
 & \cos(2x + y) \\
 & - 5C_1 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y))\right) \\
 & \sin(2x + y) \\
 & + C_1 C_2 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y))\right) \\
 & \cdot \sum_{i=1}^N \left(-2\pi k_{i,2} \sin(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y))\right) \\
 & \cos(2x + y).
 \end{aligned} \tag{C22}$$

An alternative manufactured solution is

$$\begin{aligned}
 \hat{h}_{MMS}(x, y) = & 1 + \sin(2x) + \sin(y), \\
 & \text{with } x \in [0, 20] \text{ and } y \in [0, 20],
 \end{aligned} \tag{C23}$$

along with the Dirichlet and Neumann boundary conditions:

$$\begin{cases}
 \hat{h}(0, y) = 1 + \sin(y), & \forall y \in [0, 20], \\
 \hat{h}(20, y) = 1 + \sin(2 \times 20) + \sin(y), & \forall y \in [0, 20], \\
 \frac{\partial \hat{h}}{\partial y}(x, 0) = \cos(0), & \forall x \in [0, 20], \\
 \frac{\partial \hat{h}}{\partial y}(x, 20) = \cos(20), & \forall x \in [0, 20].
 \end{cases} \tag{C24}$$

The source term f has the following form:

$$\begin{aligned}
 f(x, y) = & 2C_1 C_2 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y))\right) \\
 & \cdot \sum_{i=1}^N \left(-2\pi k_{i,1} \sin(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y))\right) \cos(2x) \\
 & - C_1 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y))\right) \\
 & (4\sin(2x) + \sin(y)) \\
 & + C_1 C_2 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y))\right) \\
 & \cdot \sum_{i=1}^N \left(-2\pi k_{i,2} \sin(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y))\right) \cos(y).
 \end{aligned} \tag{C25}$$

For the 3D case, we consider the following smooth manufactured solution:

$$\begin{aligned}
 \hat{h}_{MMS}(x, y, z) = & 1 + \sin(3x + 2y + z), \\
 & \text{with } x \in [0, 5], y \in [0, 2] \text{ and } z \in [0, 1],
 \end{aligned} \tag{C26}$$

along with the Dirichlet and Neumann boundary conditions:

$$\begin{cases}
 \hat{h}(0, y, z) = 1 + \sin(2y + z), & \forall y \in [0, 2], \forall z \in [0, 1], \\
 \hat{h}(5, y, z) = 1 + \sin(3 \times 5 + 2y + z), & \forall y \in [0, 2], \forall z \in [0, 1], \\
 \frac{\partial \hat{h}}{\partial y}(x, 0, z) = 2\cos(3x + z), & \forall x \in [0, 5], \forall z \in [0, 1], \\
 \frac{\partial \hat{h}}{\partial y}(x, 2, z) = 2\cos(3x + 2 \times 2 + z), & \forall x \in [0, 5], \forall z \in [0, 1], \\
 \frac{\partial \hat{h}}{\partial z}(x, y, 0) = \cos(3x + 2y), & \forall x \in [0, 5], \forall y \in [0, 2], \\
 \frac{\partial \hat{h}}{\partial z}(x, y, 1) = \cos(3x + 2y + 1), & \forall x \in [0, 5], \forall y \in [0, 2].
 \end{cases} \tag{C27}$$

The function K is now given by

$$K(x, y, z) = C_1 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z))\right), \tag{C28}$$

where we use the shorthand notations C_2 same as in 1-dimensional case, but $C_1 = \langle K \rangle \exp(-\frac{\sigma^2}{6})$. And the source term f has the following form:

$$\begin{aligned}
 f(x, y, z) = & 3C_1 C_2 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z))\right) \\
 & \cdot \sum_{i=1}^N \left(-2\pi k_{i,1} \sin(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z))\right) \\
 & \cos(3x + 2y + z) \\
 & + 2C_1 C_2 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z))\right) \\
 & \cdot \sum_{i=1}^N \left(-2\pi k_{i,2} \sin(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z))\right) \\
 & \cos(3x + 2y + z) \\
 & + C_1 C_2 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z))\right) \\
 & \cdot \sum_{i=1}^N \left(-2\pi k_{i,3} \sin(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z))\right) \\
 & \cos(3x + 2y + z) \\
 & - 14C_1 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z))\right) \\
 & \cdot \sin(3x + 2y + z).
 \end{aligned} \tag{C29}$$

An alternative manufactured solution is

$$\begin{aligned}
 \hat{h}_{MMS}(x, y, z) = & 5 + \sin(3x) + \sin(2y) + \sin(z), \\
 & \text{with } x \in [0, 5], y \in [0, 2] \text{ and } z \in [0, 1],
 \end{aligned} \tag{C30}$$

along with the Dirichlet and Neumann boundary conditions:

$$\begin{cases} \hat{h}(0, y, z) = 5 + \sin(2y) + \sin(z), & \forall y \in [0, 2], \forall z \in [0, 1], \\ \hat{h}(5, y, z) = 5 + \sin(3 \times 5) + \sin(2y) + \sin(z), & \forall y \in [0, 2], \forall z \in [0, 1], \\ \frac{\partial \hat{h}}{\partial x}(x, 0, z) = 2\cos(0), & \forall x \in [0, 5], \forall z \in [0, 1], \\ \frac{\partial \hat{h}}{\partial y}(x, 2, z) = 2\cos(2 \times 2), & \forall x \in [0, 5], \forall z \in [0, 1], \\ \frac{\partial \hat{h}}{\partial z}(x, y, 0) = \cos(0), & \forall x \in [0, 5], \forall y \in [0, 2], \\ \frac{\partial \hat{h}}{\partial z}(x, y, 1) = \cos(1), & \forall x \in [0, 5], \forall y \in [0, 2]. \end{cases} \tag{C31}$$

And the source term f has the following form:

$$\begin{aligned} f(x, y, z) = & 3C_1 C_2 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 \right. \\ & \left. + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z))\right) \\ & \cdot \sum_{i=1}^N \left(-2\pi k_{i,1} \sin(\xi_1 \right. \\ & \left. + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z)) \right) \cos(3x) \\ & + 2C_1 C_2 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 \right. \\ & \left. + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z))\right) \\ & \cdot \sum_{i=1}^N \left(-2\pi k_{i,2} \sin(\xi_1 \right. \\ & \left. + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z)) \right) \cos(2y) \\ & + C_1 C_2 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 \right. \\ & \left. + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z))\right) \\ & \cdot \sum_{i=1}^N \left(-2\pi k_{i,3} \sin(\xi_1 \right. \\ & \left. + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z)) \right) \cos(z) \\ & - C_1 \exp\left(C_2 \sum_{i=1}^N \cos(\xi_1 \right. \\ & \left. + 2\pi(k_{i,1}x + k_{i,2}y + k_{i,3}z))\right) \\ & \cdot (9\sin(3x) + 4\sin(2y) + \sin(z)). \end{aligned} \tag{C32}$$

Approximation proof

Here we will give a proof of the convergence of physics-informed neural network approximating hydraulic head in solving the proposed model. First, we need to assume that this partial differential equation has a unique solution, s.t. $\hat{h} \in C^2(\Omega)$ with its derivatives uniformly bounded and the heterogeneous hydraulic conductivity function $K(\mathbf{x})$ to be $C^{1,1}$ (C^1 with Lipschitz continuous derivative). The smoothness of the K field is essentially determined by the correlation of the random field Y' . According to [72], the smoothness conditions are fulfilled if the correlation of Y' has a Gaussian shape and is infinitely differentiable. For the source term, the smoothness of the source term is determined by the constructed manufactured solution \hat{h}_{MMS} , in Equations 22 and 24, which is obvious continuous and infinitely differentiable $f \in C^\infty(\Omega)$.

Theorem 2 *With assumption that Ω is compact and considering measures ℓ_1, ℓ_2 , and ℓ_3 whose supports are constrained in Ω, Γ_D , and Γ_N . Also, the governing Equation (19) subject to boundary conditions (21) is assumed to have a unique classical solution and conductivity function $K(\mathbf{x})$ is assumed to be $C^{1,1}$ (C^1 with Lipschitz continuous derivative). Then $\forall \epsilon > 0, \exists \lambda > 0$, which may dependent on $\sup_{\Omega} \|\hat{h}_{ii}\|$ and $\sup_{\Omega} \|\hat{h}_i\|$, s.t. $\exists \hat{h}^h \in F^n$, that satisfies $L(\theta) \leq \lambda \epsilon$.*

Proof For governing Equation (19) subject to boundary conditions (21), according to Theorem 1, $\forall \epsilon > 0, \exists \hat{h}^h \in F^n$, s.t.

$$\sup_{x \in \Omega} \|\hat{h}_{,i}(\mathbf{x}_\Omega) - \hat{h}_{,i}^h(\mathbf{x}_\Omega)\|^2 + \sup_{x \in \Omega} \|\hat{h}_{,ii}(\mathbf{x}_\Omega) - \hat{h}_{,ii}^h(\mathbf{x}_\Omega)\|^2 < \epsilon. \tag{D33}$$

Recalling that the Loss is constructed in the form shown in Equation (34), for MSE_G , applying triangle inequality, and obtains:

$$\begin{aligned} \|G(\mathbf{x}_\Omega; \theta)\|^2 & \leq \|K(\mathbf{x}_\Omega) \hat{h}_{,ii}^h(\mathbf{x}_\Omega; \theta)\|^2 \\ & + \|K_{,i}(\mathbf{x}_\Omega) \hat{h}_{,i}^h(\mathbf{x}_\Omega; \theta)\|^2 + \|f(\mathbf{x}_\Omega)\|^2. \end{aligned} \tag{D34}$$

Also, considering the $C^{1,1}$ conductivity function $K(\mathbf{x})$, $\exists M_1 > 0, M_2 > 0, \exists x \in \Omega, \|K(\mathbf{x})\| \leq M_1, \|K_{,i}(\mathbf{x})\| \leq M_2$. From Equation (D33), it can be obtained that

$$\begin{aligned} \int_{\Omega} K_{,i}^2(\mathbf{x}_\Omega) (\hat{h}_{,i}^h - \hat{h}_{,i})^2 d\ell_1 & \leq M_2^2 \epsilon^2 \ell_1(\Omega) \\ \int_{\Omega} K^2(\mathbf{x}_\Omega) (\hat{h}_{,ii}^h - \hat{h}_{,ii})^2 d\ell_1 & \leq M_1^2 \epsilon^2 \ell_1(\Omega). \end{aligned} \tag{D35}$$

On boundaries Γ_D and Γ_N ,

$$\int_{\Gamma_D} (\hat{h}^h(\mathbf{x}_{\Gamma_D}; \theta) - \hat{h}(\mathbf{x}_{\Gamma_D}; \theta))^2 d\ell_2 \leq \varepsilon^2 \ell_2(\Gamma_D)$$

$$\int_{\Gamma_N} K^2(\mathbf{x}_{\Gamma_N}) \left(\hat{h}_{,n}^h(\mathbf{x}_{\Gamma_N}; \theta) - \hat{h}_{,n}(\mathbf{x}_{\Gamma_N}; \theta) \right)^2 d\ell_3 \leq M_1^2 \varepsilon^2 \ell_3(\Gamma_N). \quad (\text{D36})$$

Therefore, using Equations (D35) and (D36), as $n \rightarrow \infty$

$$L(\theta) = \frac{1}{N_\Omega} \sum_{i=1}^{N_\Omega} \left\| K(\mathbf{x}_\Omega) \hat{h}_{,ii}^h(\mathbf{x}_\Omega; \theta) + K_{,i}(\mathbf{x}_\Omega) \hat{h}_{,i}^h(\mathbf{x}_\Omega; \theta) - f(\mathbf{x}_\Omega) \right\|^2 +$$

$$\frac{1}{N_{\Gamma_D}} \sum_{i=1}^{N_{\Gamma_D}} \left\| \hat{h}^h(\mathbf{x}_{\Gamma_D}; \theta) - \hat{h}_{MMS}(\mathbf{x}_{\Gamma_D}) \right\|^2 +$$

$$\frac{1}{N_{\Gamma_N}} \sum_{i=1}^{N_{\Gamma_N}} \left\| -K(\mathbf{x}_{\Gamma_N}) \frac{\partial \hat{h}(\mathbf{x}_{\Gamma_N}; \theta)}{\partial n} + K(\mathbf{x}_{\Gamma_N}) \frac{\partial \hat{h}_{MMS}}{\partial n} \right\|^2$$

$$\leq \frac{1}{N_\Omega} \sum_{i=1}^{N_\Omega} \left\| K(\mathbf{x}_\Omega) \hat{h}_{,ii}^h(\mathbf{x}_\Omega; \theta) \right\|^2$$

$$+ \frac{1}{N_\Omega} \sum_{i=1}^{N_\Omega} \left\| K_{,i}(\mathbf{x}_\Omega) \hat{h}_{,i}^h(\mathbf{x}_\Omega; \theta) \right\|^2$$

$$+ \frac{1}{N_\Omega} \sum_{i=1}^{N_\Omega} \left\| f(\mathbf{x}_\Omega) \right\|^2$$

$$+ \frac{1}{N_{\Gamma_D}} \sum_{i=1}^{N_{\Gamma_D}} \left\| \hat{h}^h(\mathbf{x}_{\Gamma_D}; \theta) - \hat{h}_{MMS}(\mathbf{x}_{\Gamma_D}) \right\|^2 +$$

$$\frac{1}{N_{\Gamma_N}} \sum_{i=1}^{N_{\Gamma_N}} \left\| -K(\mathbf{x}_{\Gamma_N}) \frac{\partial \hat{h}(\mathbf{x}_{\Gamma_N}; \theta)}{\partial n} + K(\mathbf{x}_{\Gamma_N}) \frac{\partial \hat{h}_{MMS}}{\partial n} \right\|^2$$

$$\leq (M_2^2 + M_1^2 + 1) \varepsilon^2 \ell_1(\Omega) + \varepsilon^2 \ell_2(\Gamma_D) + M_1^2 \varepsilon^2 \ell_3(\Gamma_N) = K\varepsilon. \quad (\text{D37})$$

□

With the hold of Theorem 2 and conditions that Ω is a bounded open subset of \mathbb{R} , $\forall n \in \mathbb{N}_+$, $\hat{h}^h \in F^n \in L^2(\Omega)$, it can be concluded from Sirignano et al. [56] that

Theorem 3 $\forall p < 2$, $\hat{h}^h \in F^n$ converges to \hat{h} strongly in $L^p(\Omega)$ as $n \rightarrow \infty$ with \hat{h} being the unique solution to the potential problems.

In summary, for feedforward neural networks $F^n \in L^p$ space ($p < 2$), the approximated solution $\hat{h}^h \in F^n$ will converge to the solution to this PDE.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s00366-021-01586-2>.

Acknowledgements The authors extend their appreciation to the Distinguished Scientist Fellowship Program (DSFP) at King Saud University for funding this work.

Funding Open Access funding enabled and organized by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Kolyukhin Dmitry, Sabelfeld Karl (2005) Stochastic flow simulation in 3d porous media. Monte Carlo Methods Appl 11(1):15–37
2. Freeze RA (1975) A stochastic-conceptual analysis of one-dimensional groundwater flow in nonuniform homogeneous media. Water Resour Res 11(5):725–741
3. Matheron G, De Marsily G (1980) Is transport in porous media always diffusive? A counter example. Water Resour Res 16(5):901–917
4. Kolyukhin D, Sabelfeld K (2010) Stochastic flow simulation and particle transport in a 2d layer of random porous medium. Transp Porous Media 85:347–373
5. Gelhar LW (1986) Stochastic subsurface hydrology from theory to applications. Water Resour Res 22(9S):S135S–S145S
6. Carman Phillip C (1997) Fluid flow through granular beds. Chem Eng Res Design 75:S32–S48
7. Rumpf H, Gupte AR (1975) The influence of porosity and grain size distribution on the permeability equation of porous flow. Chem Ing Technol 43(6):367–375
8. Pape H, Clauser C, Iffland J (2000) Variation of permeability with porosity in sandstone diagenesis interpreted with a fractal pore space model. Pure Appl Geophys 157:603–619
9. Bakr et al (1983) Stochastic analysis of spatial variability in subsurface flows 1. Comparison of one- and three-dimensional flows. Water Resour Res 19(1):161–180
10. Mantoglou A, Wilson JL (1982) The turning bands method for simulation of random fields using line generation by a spectral method. Water Resour Res 18(5):1379–1394
11. Bellin A, Rubin Y (1996) Hydrogen: a spatially distributed random field generator for correlated properties. Stoch Hydrol Hydraul 10:253–278
12. Kraichnan RH (1970) Diffusion by a random velocity field. Phys Fluids 13(1):22–31
13. Ababou R, McLaughlin D, Gelhar LW (1989) Numerical simulation of three-dimensional saturated flow in randomly heterogeneous porous media. Transp Porous Media 4(6):549–565
14. Hinton Geoffrey E, Osindero Simon, Teh Yee-Whye (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554
15. LeCun Yann, Bengio Yoshua, Hinton Geoffrey (2015) Deep learning. Nature 521(7553):436
16. Yang Liping, MacEachren Alan, Mitra Prasenjit, Onorati Teresa (2018) Visually-enabled active deep learning for (geo) text and image classification: a review. ISPRS Int J Geo-Inf 7(2):65

17. Zhao Z-Q, Zheng P, Xu S-t, Wu X (2019) Object detection with deep learning: a review. *IEEE Trans Neural Netw Learn Syst* 30(11):3212–3232
18. Nassif AB, Shahin I, Attili I, Azzeh M, Shaalan K (2019) Speech recognition using deep neural networks: a systematic review. *IEEE Access* 7:19143–19165
19. Mills K, Spanner M, Tamblyn I (2017) Deep learning and the Schrödinger equation. *Phys Rev A* 96(4):042113
20. Weinan E, Han Jiequn, Jentzen Arnulf (2017) Deep learning-based numerical methods for high-dimensional parabolic partial differential equations and backward stochastic differential equations. *Commun Math Stat* 5(4):349–380
21. Han Jiequn, Jentzen Arnulf, Weinan E (2018) Solving high-dimensional partial differential equations using deep learning. *PNAS; Proc Natl Acad Sci* 115(34):8505–8510
22. Raissi Maziar, Perdikaris Paris, Karniadakis George E (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
23. Raissi Maziar, Em Karniadakis George (2018) Hidden physics models: machine learning of nonlinear partial differential equations. *J Comput Phys* 357:125–141
24. Raissi Maziar (2018) Deep hidden physics models: deep learning of nonlinear partial differential equations. *J Mach Learn Res* 19(1):932–955
25. Beck Christian, Weinan E, Jentzen Arnulf (2019) Machine learning approximation algorithms for high-dimensional fully nonlinear partial differential equations and second-order backward stochastic differential equations. *J Nonlinear Sci* 29(4):1563–1619
26. Anitescu Cosmin, Atroschenko Elena, Alajlan Naif, Rabczuk Timon (2019) Artificial neural network methods for the solution of second order boundary value problems. *Comput Mater Contin* 59(1):345–359
27. Guo Hongwei, Zhuang Xiaoying, Rabczuk Timon (2019) A deep collocation method for the bending analysis of kirchhoff plate. *Comput Mater Continua* 59(2):433–456
28. Samaniego Esteban, Anitescu Cosmin, Goswami Somdatta, Nguyen-Thanh Vien Minh, Guo Hongwei, Hamdia Khader, Zhuang X, Rabczuk T (2020) An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Comput Methods Appl Mech Eng* 362:112790
29. Nguyen-Thanh Vien Minh, Zhuang Xiaoying, Rabczuk Timon (2020) A deep energy method for finite deformation hyperelasticity. *Eur J Mech-A/Solids* 80:103874
30. Goswami Somdatta, Anitescu Cosmin, Chakraborty Souvik, Rabczuk Timon (2020) Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theor Appl Fract Mech* 106:102447
31. Thornton C, Hutter F, Hoos HH, Leyton-Brown K (2013) Auto-weka: combined selection and hyperparameter optimization of classification algorithms. In: *Proceedings of the 19th ACM SIG-KDD international conference on Knowledge discovery and data mining*, pp 847–855
32. Feurer M, Klein A, Eggensperger K, Springenberg J, Blum M, Hutter F (2015) Efficient and robust automated machine learning. In: *Advances in neural information processing systems*, pp 2962–2970
33. H2O.ai. *H2O AutoML*, June 2017. H2O version 3.30.0.1
34. Le Trang T, Fu W, Moore JH (2020) Scaling tree-based automated machine learning to biomedical big data with a feature set selector. *Bioinformatics* 36(1):250–256
35. Elsken Thomas, Metzen Jan Hendrik, Hutter Frank et al (2019) Neural architecture search: a survey. *J Mach Learn Res* 20(55):1–21
36. Pham H, Guan M, Zoph B, Le Q, Dean J (2018) Efficient neural architecture search via parameters sharing. In: *International Conference on Machine Learning*, PMLR, pp 4095–4104
37. Liu , Zoph B, Neumann M, Shlens J, Hua W, Li L-J, Fei-Fei L, Yuille A, JHuang, Murphy K (2018) Progressive neural architecture search. In: *Proceedings of the European Conference on Computer Vision (ECCV)*, pp 19–34
38. Fraikin Nicolas, Funk Kilian, Frey Michael, Gauterin Frank (2019) Dimensionality reduction and identification of valid parameter bounds for the efficient calibration of automated driving functions. *Automot Engine Technol* 4(1–2):75–91
39. Salandin P, Fiorotto V (1998) Solute transport in highly heterogeneous aquifers. *Water Resour Res* 34:949–961
40. Dentz M, Kinzelbach H, Attinger S, Kinzelbach W (2002) Temporal behaviour of a solute cloud in a heterogeneous porous medium. 3. Numerical simulations. *Water Resour Res* 38(7):1118–1130
41. Heße Falk, Prykhodko Vladyslav, Schlüter Steffen, Attinger Sabine (2014) Generating random fields with a truncated power-law variogram: a comparison of several numerical methods. *Environ Model Softw* 55:32–48
42. Kramer PR, Kurbanmuradov O, Sabelfeld K (2007) Comparative analysis of multiscale gaussian random field simulation algorithms. *J Comput Phys* 226(1):897–924
43. Alecsa Cristian D, Boros Imre, Frank Florian, Knabner Peter, Nechita Mihai, Prechtel Alexander, Rupp Andreas, Suci Nicolae (2020) Numerical benchmark study for flow in highly heterogeneous aquifers. *Adv Water Res* 138:103558
44. Sudicky EA, Illman WA, Goltz IK, Adams JJ, McLaren RG (2010) Heterogeneity in hydraulic conductivity and its role on the macroscale transport of a solute plume: From measurements to a practical application of stochastic flow and transport theory. *Water Resour Res* 46(1):W01508
45. Attinger Sabine (2003) Generalized coarse graining procedures for flow in porous media. *Comput Geosci* 7(4):253–273
46. Gelhar Lynn W, Axness Carl L (1983) Three-dimensional stochastic analysis of macrodispersion in aquifers. *Water Resour Res* 19(1):161–180
47. Dagan G (1989) *Flow and transport in porous formations*. Springer, Berlin
48. Tremblay D, Etienne S, Pelletier D (2006) Code verification and the method of manufactured solutions for fluid-structure interaction problems. In: *36th AIAA Fluid Dynamics Conference and Exhibit*, pp 3218
49. Malaya Nicholas, Estacio-Hiroms Kemelli C, Stogner Roy H, Schulz Karl W, Bauman Paul T, Carey Graham F (2013) Masa: a library for verification using manufactured and analytical solutions. *Eng Comput* 29(4):487–496
50. Kang-moon P, Shin D, Yoo Y (2020) Evolutionary neural architecture search (NAS) using chromosome non-disjunction for korean grammaticality tasks. *Appl Sci* 10(10):3457
51. Funahashi Ken-Ichi (1989) On the approximate realization of continuous mappings by neural networks. *Neural Netw* 2(3):183–192
52. Hornik Kurt, Stinchcombe Maxwell, White Halbert (1989) Multilayer feedforward networks are universal approximators. *Neural Netw* 2(5):359–366
53. Hornik Kurt (1991) Approximation capabilities of multilayer feedforward networks. *Neural netw* 4(2):251–257
54. Atluri SN (2005) *Methods of computer modeling in engineering & the sciences*, vol 1. Tech Science Press, Palmdale
55. Liaqat A, Fukuhara M, Takeda T (2003) Optimal estimation of parameters of dynamical systems by neural network collocation method. *Comput Phys Commun* 150(3):215–234
56. Sirignano Justin, Spiliopoulos Konstantinos (2018) Dgm: a deep learning algorithm for solving partial differential equations. *J Comput Phys* 375:1339–1364

57. Gardner RH, O’neill RV, Mankin JB, Carney JH (1981) A comparison of sensitivity analysis and error analysis based on a stream ecosystem model. *Ecol Model* 12(3):173–190
58. Henderson-Sellers B, Henderson-Sellers A (1996) Sensitivity evaluation of environmental models using fractional factorial experimentation. *Ecol Model* 86(2–3):291–295
59. Majkowski Jacek, Ridgeway Joanne M, Miller Donald R (1981) Multiplicative sensitivity analysis and its role in development of simulation models. *Ecol Model* 12(3):191–208
60. Morris Max D (1991) Factorial sampling plans for preliminary computational experiments. *Technometrics* 33(2):161–174
61. McRae Gregory J, Tilden James W, Seinfeld John H (1982) Global sensitivity analysis—a computational implementation of the fourier amplitude sensitivity test (fast). *Comput Chem Eng* 6(1):15–25
62. Nossent Jiri, Elsen Pieter, Bauwens Willy (2011) Sobol’ sensitivity analysis of a complex environmental model. *Environ Model Softw* 26(12):1515–1525
63. Zhang JingXiao, Wei Su et al (2012) Sensitivity analysis of cere-wheat model parameters based on efast method. *J China Agric Univ* 17(5):149–154
64. Wang Anqi, Solomatine Dimitri P (2019) Practical experience of sensitivity analysis: comparing six methods, on three hydrological models, with three performance criteria. *Water* 11(5):1062
65. Herman JD, Kollat JB, Reed PM, Wagener T (2013) Method of Morris effectively reduces the computational demands of global sensitivity analysis for distributed watershed models. *Hydrol Earth Syst Sci* 17(7):2893–2903
66. Loic Brevault, Mathieu Balesdent, Nicolas Bérend, and Rodolphe Le Riche. Comparison of different global sensitivity analysis methods for aerospace vehicle optimal design. In *10th World Congress on Structural and Multidisciplinary Optimization, WCSMO-10*, 2013
67. Crosetto Michele, Tarantola Stefano (2001) Uncertainty and sensitivity analysis: tools for gis-based model implementation. *Int J Geogr Inf Sci* 15(5):415–437
68. Snoek J, Larochelle H, Adams RP (2012) Practical bayesian optimization of machine learning algorithms. In: *Advances in neural information processing systems*, pp 2951–2959
69. Li Lisha, Jamieson Kevin, DeSalvo Giulia, Rostamizadeh Afshin, Talwalkar Ameet (2017) Hyperband: a novel bandit-based approach to hyperparameter optimization. *J Mach Learn Res* 18(1):6765–6816
70. Rao RV (2019) *Jaya: an advanced optimization algorithm and its engineering applications*. Springer International Publishing AG, Switzerland
71. Weisstein EW (2002) Sphere point picking. <https://mathworld.wolfram.com/>
72. Cramér H, Leadbetter MR (2013) *Stationary and related stochastic processes: sample function properties and their applications*. Dover Publications, New York

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.