

ASYMPTOTIC PRESERVING SCHEME FOR ANISOTROPIC ELLIPTIC EQUATIONS WITH DEEP NEURAL NETWORK

LONG LI, CHANG YANG

School of Mathematics, Harbin Institute of Technology,
No. 92 West Dazhi Street, Nangang District, 150001 Harbin, China

ABSTRACT. In this paper, a new asymptotic preserving (AP) scheme is proposed for the anisotropic elliptic equations. Different from previous AP schemes, the actual one is based on first-order system least-squares for second-order partial differential equations, and it is uniformly well-posed with respect to anisotropic strength. The numerical computation is realized by a deep neural network (DNN), where least-squares functionals are employed as loss functions to determine parameters of DNN. Numerical results show that the current AP scheme is easy for implementation and is robust to approximate solutions or to identify anisotropic strength in various 2D and 3D tests.

CONTENTS

1. Introduction	2
2. An asymptotic preserving scheme for anisotropic elliptic equation based on first order system	3
2.1. Problem statement	3
2.2. An asymptotic preserving scheme based on first order system	4
2.3. APFOS least-squares formulations	8
3. Deep neural network for asymptotic preserving scheme based on first order system	8
3.1. Deep neural network architecture	8
3.2. Deep neural network for APFOS least-squares formulations	9
4. Data-driven discovery in anisotropic elliptic equation	10
4.1. Identification of the anisotropy strength ε	11
4.2. Numerical discretization	11
5. Numerical investigation	12
5.1. Setup for anisotropic elliptic equation	12
5.2. Numerical implementation and numerical results	13
6. Conclusion and perspectives	22
7. Acknowledgements	23
Appendix A. The non-AP least squares schemes for the anisotropic equation (11a)	24
A.1. The non-AP LS scheme for solving the anisotropic equation (11a)	24
A.2. The non-AP LS scheme for identifying the anisotropic strength ε	24
References	25

2010 *Mathematics Subject Classification.* Primary: 68Q25; 68R10; Secondary: 68U05.

Key words and phrases. Asymptotic preserving scheme; Anisotropic elliptic equations; First-order system least-squares formulations; Deep neural network.

1. INTRODUCTION

The study of the anisotropic elliptic equations [8] is very important for two aspects: on the one hand, in mathematical sense, the anisotropic elliptic equations become ill-posed (loss uniqueness of solution) when anisotropic strength, denoted by ε , goes to 0. On the other hand, in numerical sense, the numerical results are polluted once $\varepsilon \ll 1$. Moreover, the anisotropic elliptic equations have important applications in the physics of magnetized plasma, such as confinement plasma [21], plasma of ionosphere [3], plasma propulsion [20], *etc.*

An efficient method for solving the anisotropic elliptic equations is asymptotic preserving (AP) scheme [29]. The basic idea for AP scheme is to identify a unique solution as $\varepsilon \rightarrow 0$. Various types of AP schemes have been proposed in literature, such as the duality-based method [10, 9, 4, 27], the micro-macro method [11], the two field iterated method [12, 28], Tang's method [24, 26], *etc.* In most of these methods, an auxillary variable was introduced to eliminate anisotropy of original anisotropic elliptic equations, so we have a system of second-order elliptic equations.

In this paper, a new AP scheme based on first-order system (FOS) least-squares (LS) is proposed. The first-order system least-squares for second-order elliptic partial differential equations (PDEs) have been studied in [6, 5], where they pointed out two striking features: "on the one hand, it naturally symmetrizes and stabilizes the original problem; on the other hand, value of the corresponding LS functional at the current approximation is an accurate *a posteriori* error estimator". We thus develop APFOS scheme by introducing new auxillary variables, which are equal to directional gradients of solution. Another important difference of our APFOS scheme is that these auxillary variables may not be unique, particularly in the case with closed magnetic field, instead they directional derivatives are unique and used to determine solution of anisotropic elliptic equation. Therefore, our APFOS scheme is uniformly well-posed with respect to $\varepsilon \geq 0$. We refer to Section 2.2 for detailed explanation. In our knowledge, this is the first time to combine AP scheme with FOS of elliptic PDEs, and thus AP scheme may benefit the striking features of FOS. In contrast, the main task of the existing AP schemes in literature is to design a way to uniquely determine auxillary variables. Therefore, our APFOS scheme is free from this design difficult. Finally, we rewrite the APFOS scheme into a LS formulation, that is a minimization problem of LS functional, where boundary conditions are also added with proper scales. Next step is to solve this APFOS-LS formulation efficiently.

Recently, solving PDEs in deep neural network (DNN) framework becomes popular [7, 23, 16]. Compare to classical methods for numerical resolution of PDEs, the DNN is quite different. For instance, in finite element method or spectral method, *etc.*, a discrete functional space is firstly introduced, which has finite dimensions and is an approximation of continuous functional space. Once a basis of the discrete functional space is defined, then unknowns of PDEs are just linear combination of the basis, and the combination coefficients are parameters to be determined. Obviously, more accurate solution means more parameters required. In contrast, in DNN framework, we define a fully nonlinear function, with certain weights and biases to be determined. Thanks to the non-linearity, the DNN solution may describe unknowns of PDEs with few parameters. Particularly, some recent works focus on solving elliptic PDEs in DNN framework [5, 14, 15]. They have shown that, on the one hand, DNN can solve many types of elliptic PDEs, on the other hand, DNN can treat well some particular cases without special effort, which is not the case for classical methods.

Therefore, we discretize the APFOS-LS functional in DNN framework, that is to substitute solutions of APFOS scheme by a DNN function and to discretize the APFOS-LS functional by a summation

of the equations and boundary condition evaluated on certain collocation points. Minimizing the discrete APFOS-LS functional with respect to weights and biases of DNN yields an optimal DNN, which is approximate solutions of APFOS scheme. The APFOS-LS method in DNN framework is easy to implement since rich tools in deep learning can be directly employed, such as dominant automatic differentiation technique, stochastic gradient descent methods, *etc.* Numerical tests in 2D/3D show our APFOS-LS method is uniformly accurate with respect to ε , even for a difficult test case with closed magnetic field. In contrast, the non-AP LS method does not work for strong anisotropy. This result confirms that DNN framework works only for well-posed PDEs [23].

A reliable prediction also needs a good knowledge of model parameters. In most real applications, such elements are of physical sense but unavailable, e.g., variational data assimilation for weather forecast [18], seismic imaging in geophysics [22], computed tomography in medical imaging [2]. For the anisotropic elliptic equation, the anisotropic strength ε also has an influence on the well-posedness. In this scenario of unknown anisotropic strength ε , it is necessary to determine the coefficient by providing additional *priori* information, which is called inverse problem. In general, a least-squares formulation is established and solved by the gradient descent algorithm. Owing to the ability of tackling strong anisotropy and efficient computation, in this work the APFOS-LS method in the DNN framework is proposed for optimizing the anisotropic strength ε .

This paper is organized as follows. In Section 2, the anisotropic elliptic equation is stated, then we introduce APFOS schemes and their corresponding LS formulations. In Section 3, we first briefly review DNN framework, then the APFOS schemes are numerically resolved based on DNN framework. In Section 4, an APFOS identification scheme, for estimating the anisotropic strength parameter ε , is presented. Finally, in Section 5, numerical implementation is explained and various numerical results are shown.

2. AN ASYMPTOTIC PRESERVING SCHEME FOR ANISOTROPIC ELLIPTIC EQUATION BASED ON FIRST ORDER SYSTEM

2.1. Problem statement. In this paper, the following anisotropic elliptic equation is considered:

$$(1) \quad \begin{cases} -\Delta_{\perp} \phi^{\varepsilon} - \frac{1}{\varepsilon} \Delta_{\parallel} \phi^{\varepsilon} = f^{\varepsilon}, & \text{in } \Omega, \\ \phi^{\varepsilon} = g^{\varepsilon}, & \text{on } \Gamma_D, \\ \nabla_{\perp} \phi^{\varepsilon} \cdot \mathbf{n} + \frac{1}{\varepsilon} \nabla_{\parallel} \phi^{\varepsilon} \cdot \mathbf{n} = 0, & \text{on } \Gamma_N, \end{cases}$$

where Ω is an open set in \mathbb{R}^2 or \mathbb{R}^3 , its boundary is defined by $\partial\Omega = \Gamma_D \cup \Gamma_N$, \mathbf{n} is exterior normal vector of Γ_N . Let \mathbf{b} be a magnetic field, then we can define some parallel or perpendicular differential operators:

$$\begin{aligned} \nabla_{\parallel} \phi^{\varepsilon} &= \mathbf{b} \otimes \mathbf{b} \nabla \phi^{\varepsilon}, & \nabla_{\perp} \phi^{\varepsilon} &= (I - \mathbf{b} \otimes \mathbf{b}) \nabla \phi^{\varepsilon}, \\ \Delta_{\parallel} \phi^{\varepsilon} &= \nabla \cdot \nabla_{\parallel} \phi^{\varepsilon}, & \Delta_{\perp} \phi^{\varepsilon} &= \nabla \cdot \nabla_{\perp} \phi^{\varepsilon}. \end{aligned}$$

The problem (1) is well-posed for any $\varepsilon > 0$. Indeed, in particular, let Ω be a convex polygon, and $f^{\varepsilon} \in L^2(\Omega)$, $g^{\varepsilon} \in H^{3/2}(\Gamma_D)$, then we have a unique solution $\phi^{\varepsilon} \in H_D^2(\Omega)$ of the problem (1), with

$$H_D^2(\Omega) = \{v \in H^2(\Omega) : v|_{\Gamma_D} = g^{\varepsilon}\}.$$

Formally, let ε go to 0, we get a degenerate problem:

$$(2) \quad \begin{cases} \Delta_{\parallel} \phi^0 = 0, & \text{in } \Omega, \\ \phi^0 = g^0, & \text{on } \Gamma_D, \\ \nabla_{\parallel} \phi^0 \cdot \mathbf{n} = 0, & \text{on } \Gamma_N. \end{cases}$$

The problem (2) is ill-posed due to loss of uniqueness of solution, since any function belongs to set

$$H_{D,\parallel}^2 = \{v \in H^2(\Omega) : v|_{\Gamma_D} = g^0, \nabla_{\parallel} v = 0 \text{ in } \Omega\}$$

is solution of (2). In another word, the solution of (2) is in kernel of differential operator ∇_{\parallel} . We then define a limit problem

$$(3) \quad \begin{cases} -\Delta_{\perp} \bar{\phi} = f^0, & \text{in } \Omega, \\ \bar{\phi} = g^0, & \text{on } \Gamma_D, \\ \nabla_{\perp} \bar{\phi} \cdot \mathbf{n} = 0, & \text{on } \Gamma_N, \\ \nabla_{\parallel} \bar{\phi} = 0, & \text{in } \Omega. \end{cases}$$

This limit problem (3) has unique solution $\bar{\phi} \in H_{D,\parallel}^2$ thanks to the constraint $\nabla_{\parallel} \bar{\phi} = 0$ in Ω .

The question now is how to find a way such that a unique solution is uniformly defined for any $\varepsilon \geq 0$, which is the main subject of next subsection.

2.2. An asymptotic preserving scheme based on first order system. In the sequel, we will omit the superscript ε if there is no confusion.

Aligned 2D case. To give the basic idea, let us first consider a simple case in 2D, that is the magnetic field lines \mathbf{b} aligned with z -axis:

$$(4) \quad \begin{cases} -\partial_x^2 \phi - \frac{1}{\varepsilon} \partial_z^2 \phi = f, & \text{in } \Omega, \\ \phi = 0, & \text{on } \Gamma_D, \\ \partial_z \phi = 0, & \text{on } \Gamma_N, \end{cases}$$

where $\Omega = (0, 1) \times (0, 1)$, $\Gamma_D = \{0, 1\} \times (0, 1)$, $\Gamma_N = (0, 1) \times \{0, 1\}$. The equation (4) is well-posed for any $\varepsilon > 0$. Formally let $\varepsilon \rightarrow 0$, the equation (4) becomes ill-posed, since any function ϕ depending only on x and satisfying $\phi = 0$ on Γ_D is the solution of degenerate problem.

A new asymptotic preserving (AP) scheme is introduced by introducing the ansatz:

$$(5) \quad \partial_x \phi = \tau \text{ and } \partial_z \phi = \varepsilon \sigma, \quad \text{in } \Omega.$$

Thus the new AP system reads:

$$(6a) \quad \begin{cases} -\partial_x \tau - \partial_z \sigma = f, & \text{in } \Omega, \\ \partial_x \phi = \tau, & \text{in } \Omega, \\ \partial_z \phi = \varepsilon \sigma, & \text{in } \Omega, \end{cases}$$

with boundary conditions:

$$(6b) \quad \begin{cases} \phi = 0, & \text{on } \Gamma_D, \\ \partial_z \phi = 0, & \text{on } \Gamma_N, \\ \sigma = 0, & \text{on } \Gamma_N. \end{cases}$$

Next theorem presents the well-posedness of the AP system (6a)-(6b) (APFOS scheme):

Theorem 2.1. *Let*

$$H_D^1 = \{v \in H^1(\Omega) : v|_{\Gamma_D} = 0\}, \quad H_N^1 = \{v \in H^1(\Omega) : v|_{\Gamma_N} = 0\}.$$

For any $\varepsilon \geq 0$, $f \in L^2(\Omega)$, we have a unique triple $(\phi, \sigma, \tau) \in H_D^1 \times H_N^1 \times H^1(\Omega)$ being the solution of the AP system (6).

Proof. It is evident that the AP system (6) is well-posed for any $\varepsilon > 0$. Indeed, the AP system (6) is equivalent to the equation (4) for $\varepsilon > 0$.

Now we will focus on the case $\varepsilon = 0$, where the system (6a) becomes

$$(7) \quad \begin{cases} -\partial_x \tau - \partial_z \sigma = f, & \text{in } \Omega, \\ \partial_x \phi = \tau, & \text{in } \Omega, \\ \partial_z \phi = 0, & \text{in } \Omega. \end{cases}$$

The constraint $\partial_z \phi = 0$ in Ω of (7) implies ϕ only depending on x , *i.e.* $\phi(x, z) \equiv \phi(x)$, which also implies $\tau(x, z) \equiv \tau(x)$ thanks to $\partial_x \phi = \tau$.

Let $\bar{\Omega} = (0, 1)$. Integrating the system (7) on z -axis, we have

$$(8) \quad \begin{cases} -\partial_x \bar{\tau} = \bar{f}, & \text{in } \bar{\Omega}, \\ \partial_x \bar{\phi} = \bar{\tau}, & \text{in } \bar{\Omega}, \\ \bar{\phi} = 0, & \text{on } \{0, 1\}, \end{cases}$$

where the symbol bar means average value along z -axis of a function, *e.g.*

$$\bar{f}(x) = \int_0^1 f(x, z) dz.$$

The system (8) admits a unique couple of solutions $(\bar{\phi}, \bar{\tau}) \in \{v \in H^1(\bar{\Omega}) : v(0) = v(1) = 0\} \times H^1(\bar{\Omega})$ [13].

Finally, we reformulate the first equation of (7) as

$$\partial_z \sigma = -\partial_x \tau - f,$$

which gives σ , thanks to boundary condition of σ , by integration, *i.e.*

$$\sigma(x, z) = -z \partial_x \tau(x) - \int_0^z f(x, s) ds.$$

□

Remark 2.2. *If the boundary conditions at the end of z -axis of (6b) is replaced by periodic boundary conditions, *i.e.**

$$(9) \quad \begin{cases} \phi(0, z) = \phi(1, z) = 0, & z \in \bar{\Omega}, \\ \partial_z \phi(x, 0) = \partial_z \phi(x, 1), & x \in \bar{\Omega}, \\ \sigma(x, 0) = \sigma(x, 1), & x \in \bar{\Omega}, \end{cases}$$

then the couple (ϕ, τ) in the APFOS scheme remains well-defined, but σ is not unique if $\varepsilon = 0$. Indeed, we have

$$\sigma(x, z) = -z \partial_x \tau(x) - \int_0^z f(x, s) ds + c(x),$$

for any $c(x)$ depending only on x . However, the partial derivative $\partial_z \sigma$ is unique since $\partial_z \sigma = -\partial_x \tau - f$.

General 2D case. Let us consider now general 2D anisotropic elliptic equation:

$$(10) \quad \begin{cases} -\Delta_{\perp} \phi - \frac{1}{\varepsilon} \Delta_{\parallel} \phi = f, & \text{in } \Omega, \\ \phi = g, & \text{on } \Gamma_D, \\ \nabla_{\perp} \phi \cdot \mathbf{n} + \frac{1}{\varepsilon} \nabla_{\parallel} \phi \cdot \mathbf{n} = 0, & \text{on } \Gamma_N. \end{cases}$$

The magnetic field and its orthogonal vector are defined by

$$\mathbf{b} = (b_x, b_z)^T, \quad \mathbf{b}^{\perp} = (-b_z, b_x)^T.$$

Similar to the aligned 2D case, the APFOS scheme for general 2D case is given by

$$(11a) \quad \begin{cases} -\nabla \cdot (\tau \mathbf{b}^\perp) - \nabla \cdot (\sigma \mathbf{b}) = f, & \text{in } \Omega, \\ \nabla_\perp \phi := \mathbf{b}^\perp \otimes \mathbf{b}^\perp \nabla \phi = \tau \mathbf{b}^\perp, & \text{in } \Omega, \\ \nabla_\parallel \phi := \mathbf{b} \otimes \mathbf{b} \nabla \phi = \varepsilon \sigma \mathbf{b}, & \text{in } \Omega, \end{cases}$$

with boundary conditions:

$$(11b) \quad \begin{cases} \phi = g, & \text{on } \Gamma_D, \\ \nabla_\perp \phi \cdot \mathbf{n} + \frac{1}{\varepsilon} \nabla_\parallel \phi \cdot \mathbf{n} = 0, & \text{on } \Gamma_N, \\ \tau \mathbf{b}^\perp \cdot \mathbf{n} + \sigma \mathbf{b} \cdot \mathbf{n} = 0, & \text{on } \Gamma_N. \end{cases}$$

It is clear that, by taking $\mathbf{b} = (0, 1)^T$, $\Omega = (0, 1) \times (0, 1)$, $\Gamma_D = \{0, 1\} \times (0, 1)$, $\Gamma_N = (0, 1) \times \{0, 1\}$, the APFOS scheme (11) in general 2D case is nothing but the APFOS scheme (6).

The well-posedness of the APFOS scheme (11) in general 2D case is summarized in next theorem:

Theorem 2.3. *Suppose that the ends of magnetic field lines only cross on Γ_N and the boundary Γ_D coincides with the magnetic field lines. Let*

$$H_{D,g}^1 = \{v \in H^1(\Omega) : v|_{\Gamma_D} = g\}.$$

Then for any $\varepsilon \geq 0$, $f \in L^2(\Omega)$, we have a unique couple $(\phi, \sigma, \tau) \in H_{D,g}^1 \times H^1(\Omega) \times H^1(\Omega)$ being the solution of the APFOS scheme (11).

Proof. We will only consider the case $\varepsilon = 0$, whereas the well-posedness for the case $\varepsilon > 0$ is evident. Let us make a change of variable as

$$x = x(X, Z), \quad z = z(X, Z),$$

with scale factors $h_X(X, Z)$ and $h_Z(X, Z)$ respectively. The Z -axis is aligned with the magnetic field, while X -axis is perpendicular with the magnetic field, *i.e.*

$$\hat{X} = -b_z \hat{x} + b_x \hat{z}, \quad \hat{Z} = b_x \hat{x} + b_z \hat{z},$$

where \hat{X} and \hat{Z} are unit basis in new curvilinear coordinates. For simplicity, in the sequel we suppose that $(X, Z) \in [X_0, X_{\max}] \times [Z_0, Z_{\max}]$. However, the general case can be proved similarly. Thanks to the change of variable, the system (11a) become

$$(12) \quad \begin{cases} -\partial_X(h_Z \tau) - \partial_Z(h_X \sigma) = h_X h_Z f, & \text{in } \Omega, \\ \partial_X \phi = h_X \tau = \frac{h_X}{h_Z}(h_Z \tau), & \text{in } \Omega, \\ \partial_Z \phi = \varepsilon h_Z \sigma = 0, & \text{in } \Omega. \end{cases}$$

Let us denote

$$T = h_Z \tau, \quad \Sigma = h_X \sigma, \quad F = h_X h_Z f.$$

so the system (12) becomes

$$(13) \quad \begin{cases} -\partial_X T - \partial_Z \Sigma = F, & \text{in } \Omega, \\ \partial_X \phi = \frac{h_X}{h_Z} T, & \text{in } \Omega, \\ \partial_Z \phi = 0, & \text{in } \Omega. \end{cases}$$

Thanks to the third equation of (13), we have

$$\phi(X, Z) \equiv \phi(X).$$

Integrating the first equation of (13) along magnetic field lines \mathbf{b} (Z -axis) yields

$$-\partial_X \bar{T}(X) - (\Sigma(X, Z_{\max}) - \Sigma(X, Z_0)) = \bar{F}(X),$$

where the symbol bar means average value along Z -axis of a function, *e.g.*

$$\bar{F}(X) = \frac{1}{Z_{\max} - Z_0} \int_{Z_0}^{Z_{\max}} F(X, Z) dZ.$$

Thanks to the third equation of (11b), we have

$$\sigma(X, Z_0) = -\bar{T}(X) \left(\frac{\mathbf{b}^\perp \cdot \mathbf{n}}{\mathbf{b} \cdot \mathbf{n}} \frac{h_X}{h_Z} \right) (X, Z_0), \quad \sigma(X, Z_{\max}) = -\bar{T}(X) \left(\frac{\mathbf{b}^\perp \cdot \mathbf{n}}{\mathbf{b} \cdot \mathbf{n}} \frac{h_X}{h_Z} \right) (X, Z_{\max}).$$

Moreover, thanks to the assumption on the magnetic field, we have $g = \bar{g}$ on $\{X_0, X_{\max}\}$. Thus

$$(14) \quad \begin{cases} -\partial_X \bar{T} + C(X) \bar{T} = \bar{f}, & \text{in } \bar{\Omega}, \\ \bar{T} = \frac{h_Z}{h_X} \partial_X \bar{\phi}, & \text{in } \bar{\Omega}, \\ \bar{\phi} = \bar{g}, & \text{on } \{X_0, X_{\max}\}, \end{cases}$$

where $\bar{\Omega} = [X_0, X_{\max}]$, $C(X) = \left(\frac{\mathbf{b}^\perp \cdot \mathbf{n}}{\mathbf{b} \cdot \mathbf{n}} \frac{h_X}{h_Z} \right) (X, Z_{\max}) - \left(\frac{\mathbf{b}^\perp \cdot \mathbf{n}}{\mathbf{b} \cdot \mathbf{n}} \frac{h_X}{h_Z} \right) (X, Z_0)$. The system (14) permits unique solution provided $C(X)$ is bounded in $[X_0, X_{\max}]$ [13], which is true thanks to the assumption of theorem. Therefore, we have a unique couple $(\phi, \tau) = (\bar{\phi}, \frac{1}{h_X} \partial_X \bar{\phi}) \in H_{D,g}^1 \times H^1(\Omega)$.

Finally, following the last lines of the proof of Theorem 2.1, we deduce the uniqueness of σ , thus we achieve the proof. \square

We remark that, on the one hand, for the case with closed magnetic field, that is periodic boundary in Z -axis, we have

$$\Sigma(X, Z_0) = \Sigma(X, Z_{\max}).$$

Then, following the same arguments in the proof of Theorem 2.3, we have again a unique couple $(\phi, \tau) \in H_{D,g}^1 \times H^1(\Omega)$. On the other hand, for the same reason as that in the aligned case, we lose the uniqueness of the variable σ for the case $\varepsilon = 0$ with closed magnetic field.

General 3D case. The general 3D anisotropic elliptic equation takes the same form as in (10). However, the magnetic field and its orthogonal vectors are defined differently

$$\mathbf{b} = (b_x, b_y, b_z)^T, \quad \mathbf{b}^{\perp,1} = \frac{\mathbf{B}^{\perp,1}}{|\mathbf{B}^{\perp,1}|}, \quad \mathbf{b}^{\perp,2} = \frac{\mathbf{B}^{\perp,2}}{|\mathbf{B}^{\perp,2}|},$$

with

$$\mathbf{B}^{\perp,1} = \mathbf{b} \times (0, 0, 1)^T, \quad \mathbf{B}^{\perp,2} = \mathbf{b} \times \mathbf{B}^{\perp,1}.$$

Therefore, we need introduce correspondingly three auxiliary variables. The APFOS scheme for general 3D case is given by

$$(15a) \quad \begin{cases} -\nabla \cdot (\tau \mathbf{b}^{\perp,1} + \chi \mathbf{b}^{\perp,2}) - \nabla \cdot (\sigma \mathbf{b}) = f, & \text{in } \Omega, \\ \nabla \phi \cdot \mathbf{b}^{\perp,1} = \tau, & \text{in } \Omega, \\ \nabla \phi \cdot \mathbf{b}^{\perp,2} = \chi, & \text{in } \Omega, \\ \nabla \phi \cdot \mathbf{b} = \varepsilon \sigma, & \text{in } \Omega, \end{cases}$$

with boundary conditions:

$$(15b) \quad \begin{cases} \phi = g, & \text{on } \Gamma_D, \\ \nabla_{\perp} \phi \cdot \mathbf{n} + \frac{1}{\varepsilon} \nabla_{\parallel} \phi \cdot \mathbf{n} = 0, & \text{on } \Gamma_N, \\ (\tau \mathbf{b}^{\perp,1} + \chi \mathbf{b}^{\perp,2}) \cdot \mathbf{n} + \sigma \mathbf{b} \cdot \mathbf{n} = 0, & \text{on } \Gamma_N. \end{cases}$$

By following the same arguments shown in general 2D case, we could deduce that the APFOS scheme (15) in general 3D case is also well-posed.

2.3. APFOS least-squares formulations. This part is devoted to introducing APFOS least-squares formulations for APFOS scheme (11) and (15).

For 2D case, the APFOS-LS formulation is to find $(\phi, \tau, \sigma) \in (H^1(\Omega))^3$ such that

$$(16a) \quad \mathcal{G}(\phi, \tau, \sigma; \mathbf{f}) = \min_{(\psi, \xi, \zeta) \in (H^1(\Omega))^3} \mathcal{G}(\psi, \xi, \zeta; \mathbf{f}),$$

where $\mathbf{f} = (f, g)$, and the APFOS-LS functional with proper scales is

$$(16b) \quad \begin{aligned} \mathcal{G}(\psi, \xi, \zeta; \mathbf{f}) = & \|\nabla \cdot (\xi \mathbf{b}^{\perp}) + \nabla \cdot (\zeta \mathbf{b}) + f\|_{0,\Omega}^2 + \|\nabla \psi \cdot \mathbf{b}^{\perp} - \xi\|_{0,\Omega}^2 + \|\nabla \psi \cdot \mathbf{b} - \varepsilon \zeta\|_{0,\Omega}^2 \\ & + \|\psi - g\|_{1/2,\Gamma_D}^2 + \|\varepsilon \nabla_{\perp} \psi \cdot \mathbf{n} + \nabla_{\parallel} \psi \cdot \mathbf{n}\|_{-1/2,\Gamma_N}^2 + \|\xi \mathbf{b}^{\perp} \cdot \mathbf{n} + \zeta \mathbf{b} \cdot \mathbf{n}\|_{-1/2,\Gamma_N}^2, \end{aligned}$$

for all $(\psi, \xi, \zeta) \in (H^1(\Omega))^3$. There exists $(\phi, \tau, \sigma) \in (H^1(\Omega))^3$, which can minimize APFOS-LS functional (16b). Moreover, (ϕ, τ) is also the unique solution to the APFOS scheme (11) such that $(\phi, \tau) \in H_{D,g}^1 \times H^1(\Omega)$.

Similarly, for 3D APFOS scheme (15), the APFOS-LS formulation is to find $(\phi, \tau, \chi, \sigma) \in (H^1(\Omega))^4$ such that

$$(17a) \quad \mathcal{G}(\phi, \tau, \chi, \sigma; \mathbf{f}) = \min_{(\psi, \xi, \kappa, \zeta) \in (H^1(\Omega))^4} \mathcal{G}(\psi, \xi, \kappa, \zeta; \mathbf{f}).$$

where the APFOS-LS functional is as follows

$$(17b) \quad \begin{aligned} \mathcal{G}(\psi, \xi, \kappa, \zeta; \mathbf{f}) = & \|\nabla \cdot (\xi \mathbf{b}^{\perp,1} + \kappa \mathbf{b}^{\perp,2}) + \nabla \cdot (\zeta \mathbf{b}) + f\|_{0,\Omega}^2 + \|\nabla \psi \cdot \mathbf{b}^{\perp,1} - \xi\|_{0,\Omega}^2 \\ & + \|\nabla \psi \cdot \mathbf{b}^{\perp,2} - \kappa\|_{0,\Omega}^2 + \|\nabla \psi \cdot \mathbf{b} - \varepsilon \zeta\|_{0,\Omega}^2 + \|\psi - g\|_{1/2,\Gamma_D}^2 \\ & + \|\varepsilon \nabla_{\perp} \psi \cdot \mathbf{n} + \nabla_{\parallel} \psi \cdot \mathbf{n}\|_{-1/2,\Gamma_N}^2 + \|\xi \mathbf{b}^{\perp,1} \cdot \mathbf{n} + \kappa \mathbf{b}^{\perp,2} \cdot \mathbf{n} + \zeta \mathbf{b} \cdot \mathbf{n}\|_{-1/2,\Gamma_N}^2, \end{aligned}$$

for all $(\psi, \xi, \kappa, \zeta) \in (H^1(\Omega))^4$.

3. DEEP NEURAL NETWORK FOR ASYMPTOTIC PRESERVING SCHEME BASED ON FIRST ORDER SYSTEM

3.1. Deep neural network architecture. We will briefly describe the deep neural network (DNN) structure used in this paper. For a more detailed review of DNN structure, we refer to [5] and references therein.

Simply to say, a deep neural network defines a nonlinear function

$$\mathcal{N} : \mathbf{x} \in \mathbb{R}^d \rightarrow \mathbf{y} = \mathcal{N}(\mathbf{x}) \in \mathbb{R}^c,$$

where d and c are dimensions of input \mathbf{x} and output \mathbf{y} , respectively. $\mathcal{N}(\mathbf{x})$ is a composite function of many different layers of functions as

$$\mathcal{N}(\mathbf{x}) = \mathcal{N}^{(L)} \circ \dots \circ \mathcal{N}^{(2)} \circ \mathcal{N}^{(1)}(\mathbf{x}),$$

where L is the number of layers. The layer function $\mathcal{N}^{(l)} : \mathbb{R}^{n_{l-1}} \rightarrow \mathbb{R}^{n_l}$ is defined by

$$\mathcal{N}^{(l)}(\mathbf{x}^{(l-1)}) = \varphi^{(l)}(\mathcal{W}^{(l)}\mathbf{x}^{(l-1)} + \mathcal{B}^{(l)}), \quad l = 1, \dots, L-1,$$

where $\mathbf{x}^{(l-1)} \in \mathbb{R}^{n_{l-1}}$ is the output of layer $l-1$, it is also the input of layer l . $\mathcal{W}^{(l)} \in \mathbb{R}^{n_l \times n_{l-1}}$, $\mathcal{B}^{(l)} \in \mathbb{R}^{n_l}$ are called network weights and biases respectively. The application $\varphi^{(l)} : \mathbb{R} \rightarrow \mathbb{R}$ is an activation function, which is the soul of DNN, it enables the multi-layer network structure to learn complex things, complex data, and to represent the nonlinear and complex arbitrary function mapping between input and output. At the same time, the activation function can also perform data normalization to map the input data to a certain range, on the one hand, it prevents the risk of data overflow; on the other hand, since each input data of the network often has different physical meanings and different dimensions, data normalization makes all components change within a common range, so that the input components are given equal importance at the beginning of network training. In this paper, the hyperbolic tangent function is taken as the activation function

$$\varphi(x) = \tanh(x), \quad x \in \mathbb{R}.$$

The function \tanh is very smooth, thus it will not infect the regularity of solution. Notice that the image of \tanh is $[-1, 1]$, it is better to rescale the input of the first layer to $[-1, 1]^d$. Finally, for the last layer L , there is no need to use activation function, hence

$$\mathcal{N}^{(L)}(\mathbf{x}^{(L-1)}) = \mathcal{W}^{(L)}\mathbf{x}^{(L-1)} + \mathcal{B}^{(L)}.$$

It is clear that the DNN function \mathcal{N} is uniquely determined by weights and biases

$$\mathcal{W}^{(l)}, \mathcal{B}^{(l)}, \quad l = 1, \dots, L,$$

which have totally $N = \sum_{l=1}^L n_l \times (n_{l-1} + 1)$ parameters. Particularly, for anisotropic problem, we have $n_L = 1$; for APFOS scheme in 2D case, $n_L = 3$; for APFOS scheme in 3D case, $n_L = 4$. The DNN function \mathcal{N} will be regarded as an approximation for resolution of the least-squares formulations in next part.

3.2. Deep neural network for APFOS least-squares formulations. The idea of deep neural network for least-squares (LS) formulations is to substitute the continuous LS formulations by their discrete approximations, where the unknowns are DNN function \mathcal{N} . Then minimizing the discrete LS formulations gives the optimal weights and biases.

Let us consider 2D APFOS-LS formulation, in which the approximations $\hat{\phi}(\mathbf{x}, \Theta)$, $\hat{\tau}(\mathbf{x}, \Theta)$, $\hat{\sigma}(\mathbf{x}, \Theta)$ are also determined by the location $\mathbf{x} \in \mathbb{R}^2$ and the weights and biases denoted as $\Theta \in \mathbb{R}^N$. The discrete APFOS-LS formulation reads

$$(18a) \quad \hat{\mathcal{G}}[\hat{\phi}, \hat{\tau}, \hat{\sigma}; \mathbf{f}](\Theta) = \min_{\Theta \in \mathbb{R}^N} \hat{\mathcal{G}}[\hat{\psi}, \hat{\xi}, \hat{\zeta}; \mathbf{f}](\tilde{\Theta}),$$

and the following discrete APFOS-LS functional is employed

$$\begin{aligned}
& \hat{\mathcal{G}}[\hat{\psi}, \hat{\xi}, \hat{\zeta}; \mathbf{f}](\tilde{\Theta}) \\
&= \frac{1}{N_f} \sum_{i=1}^{N_f} \left(\nabla \cdot (\hat{\xi}(\mathbf{x}_i, \tilde{\Theta}) \mathbf{b}^\perp(\mathbf{x}_i)) + \nabla \cdot (\hat{\zeta}(\mathbf{x}_i, \tilde{\Theta}) \mathbf{b}(\mathbf{x}_i)) + f(\mathbf{x}_i) \right)^2 \\
(18b) \quad &+ \frac{1}{N_f} \sum_{i=1}^{N_f} \left((\nabla \hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) \cdot \mathbf{b}^\perp(\mathbf{x}_i) - \hat{\xi}(\mathbf{x}_i, \tilde{\Theta}))^2 + (\nabla \hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) \cdot \mathbf{b}(\mathbf{x}_i) - \varepsilon \hat{\zeta}(\mathbf{x}_i, \tilde{\Theta}))^2 \right) \\
&+ \frac{\beta_D}{N_d} \sum_{j=1}^{N_d} (\hat{\psi}(\mathbf{x}_j, \tilde{\Theta}) - g(\mathbf{x}_j))^2 + \frac{\beta_N}{N_n} \sum_{k=1}^{N_n} (\varepsilon \nabla_\perp \hat{\psi}(\mathbf{x}_k, \tilde{\Theta}) \cdot \mathbf{n}(\mathbf{x}_k) + \nabla_\parallel \hat{\psi}(\mathbf{x}_k, \tilde{\Theta}) \cdot \mathbf{n}(\mathbf{x}_k))^2 \\
&+ \frac{\beta_N}{N_n} \sum_{k=1}^{N_n} (\hat{\xi}(\mathbf{x}_i, \tilde{\Theta}) \mathbf{b}^\perp(\mathbf{x}_k) \cdot \mathbf{n}(\mathbf{x}_k) + \hat{\zeta}(\mathbf{x}_k, \tilde{\Theta}) \mathbf{b}(\mathbf{x}_k) \cdot \mathbf{n}(\mathbf{x}_k))^2.
\end{aligned}$$

where N_f , N_d , N_n are numbers of collocation points in Ω , Γ_D , Γ_N respectively. The collocation points can be uniformly sampled in the domain or in some random way, which will be specified in Section 5. The norms on the boundaries are approximated by weighted L^2 norms, and β_D , β_N represent weights in the discrete APFOS-LS functionals. Now minimizing (18b) with respect to Θ yields an optimal DNN structure solution to the APFOS scheme (11).

Similarly, for 3D case, the discrete APFOS-LS formulation reads

$$(19a) \quad \hat{\mathcal{G}}[\hat{\phi}, \hat{\tau}, \hat{\chi}, \hat{\sigma}; \mathbf{f}](\Theta) = \min_{\Theta \in \mathbb{R}^N} \hat{\mathcal{G}}[\hat{\psi}, \hat{\xi}, \hat{\kappa}, \hat{\zeta}; \mathbf{f}](\tilde{\Theta}),$$

where the discrete APFOS-LS functional is

$$\begin{aligned}
(19b) \quad & \hat{\mathcal{G}}[\hat{\psi}, \hat{\xi}, \hat{\kappa}, \hat{\zeta}; \mathbf{f}](\tilde{\Theta}) \\
&= \frac{1}{N_f} \sum_{i=1}^{N_f} \left(\nabla \cdot (\hat{\xi}(\mathbf{x}_i, \tilde{\Theta}) \mathbf{b}^{\perp,1}(\mathbf{x}_i)) + \nabla \cdot (\hat{\kappa}(\mathbf{x}_i, \tilde{\Theta}) \mathbf{b}^{\perp,2}(\mathbf{x}_i)) + \nabla \cdot (\hat{\zeta}(\mathbf{x}_i, \tilde{\Theta}) \mathbf{b}(\mathbf{x}_i)) + f(\mathbf{x}_i) \right)^2 \\
&+ \frac{1}{N_f} \sum_{i=1}^{N_f} \left((\nabla \hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) \cdot \mathbf{b}^{\perp,1}(\mathbf{x}_i) - \hat{\xi}(\mathbf{x}_i, \tilde{\Theta}))^2 + (\nabla \hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) \cdot \mathbf{b}^{\perp,2}(\mathbf{x}_i) - \hat{\kappa}(\mathbf{x}_i, \tilde{\Theta}))^2 \right) \\
&+ \frac{1}{N_f} \sum_{i=1}^{N_f} (\nabla \hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) \cdot \mathbf{b}(\mathbf{x}_i) - \varepsilon \hat{\zeta}(\mathbf{x}_i, \tilde{\Theta}))^2 \\
&+ \frac{\beta_D}{N_d} \sum_{j=1}^{N_d} (\hat{\psi}(\mathbf{x}_j, \tilde{\Theta}) - g(\mathbf{x}_j))^2 + \frac{\beta_N}{N_n} \sum_{k=1}^{N_n} (\varepsilon \nabla_\perp \hat{\psi}(\mathbf{x}_k, \tilde{\Theta}) \cdot \mathbf{n}(\mathbf{x}_k) + \nabla_\parallel \hat{\psi}(\mathbf{x}_k, \tilde{\Theta}) \cdot \mathbf{n}(\mathbf{x}_k))^2 \\
&+ \frac{\beta_N}{N_n} \sum_{k=1}^{N_n} (\hat{\xi}(\mathbf{x}_i, \tilde{\Theta}) \mathbf{b}^{\perp,1}(\mathbf{x}_k) \cdot \mathbf{n}(\mathbf{x}_k) + \hat{\kappa}(\mathbf{x}_i, \tilde{\Theta}) \mathbf{b}^{\perp,2}(\mathbf{x}_k) \cdot \mathbf{n}(\mathbf{x}_k) + \hat{\zeta}(\mathbf{x}_k, \tilde{\Theta}) \mathbf{b}(\mathbf{x}_k) \cdot \mathbf{n}(\mathbf{x}_k))^2.
\end{aligned}$$

4. DATA-DRIVEN DISCOVERY IN ANISOTROPIC ELLIPTIC EQUATION

As a direct application of APFOS scheme, in this part we focus on data-driven discovery in anisotropic elliptic equation. More specifically, we are interested in identification of the anisotropy strength ε . Only 2D cases are considered here, the ones for 3D can be extended similarly.

Let us point out that data-driven discovery can be regarded as inverse problem, that means to use some *priori* information of solutions, called observation, to estimate parameters of equations. The way of estimating parameters can be made by adding those *priori* information of solutions to LS functionals introduced in Section 2.3 and Section 3.2, then we can find optimal parameter estimations by solving the new LS formulations.

4.1. Identification of the anisotropy strength ε . Suppose that $\phi_e \in H_D^2(\Omega)$ is the solution of the anisotropic equation (10). For 2D APFOS scheme (11), the identification APFOS-LS formulation is to find $\varepsilon \in \mathbb{R}^+$ and $(\phi, \tau, \sigma) \in (H^1(\Omega))^3$ such that

$$(20a) \quad \mathcal{G}(\varepsilon, \phi, \tau, \sigma; \mathbf{f}) = \min_{\tilde{\varepsilon} \in \mathbb{R}^+, (\psi, \xi, \zeta) \in (H^1(\Omega))^3} \mathcal{G}(\tilde{\varepsilon}, \psi, \xi, \zeta; \mathbf{f}),$$

where $\mathbf{f} = (f, g)$, the APFOS-LS functional is as follows

$$(20b) \quad \begin{aligned} \mathcal{G}(\tilde{\varepsilon}, \psi, \xi, \zeta; \mathbf{f}) = & \|\psi - \phi_e\|_{0,\Omega}^2 \\ & + \|\nabla \cdot (\xi \mathbf{b}^\perp) + \nabla \cdot (\zeta \mathbf{b}) + f\|_{0,\Omega}^2 + \|\nabla \psi \cdot \mathbf{b}^\perp - \xi\|_{0,\Omega}^2 + \|\nabla \psi \cdot \mathbf{b} - \tilde{\varepsilon} \zeta\|_{0,\Omega}^2 \\ & + \|\tilde{\varepsilon} \nabla_\perp \psi \cdot \mathbf{n} + \nabla \|\psi \cdot \mathbf{n}\|_{-1/2, \Gamma_N}^2 + \|\xi \mathbf{b}^\perp \cdot \mathbf{n} + \zeta \mathbf{b} \cdot \mathbf{n}\|_{-1/2, \Gamma_N}^2, \end{aligned}$$

for all $\tilde{\varepsilon} \in \mathbb{R}^+$ and $(\psi, \xi, \zeta) \in (H^1(\Omega))^3$.

Observing from (20b), we can find two main differences from (16b). The first difference is that the Dirichlet boundary condition constraint does not exist in (20b), instead we add the observation constraint $\|\psi - \phi_e\|_{0,\Omega}^2$. The second one is that output of optimization problem consists of both a DNN and a parameter ε .

4.2. Numerical discretization. In this part, we will propose discrete functionals to the identification APFOS-LS functional (20b), which can also be made by adding *priori* information to (18b). Different from continuous case, there are two aspects should be taken into account: on the one hand, the solution ϕ_e is only sampled on certain points, and they can be noise free or with noise; on the other hand, the parameter ε should be guaranteed always non-negative.

Following the same principle, the corresponding identification APFOS-LS formulation reads

$$(21a) \quad \hat{\mathcal{G}}[\hat{\phi}, \hat{\tau}, \hat{\sigma}; \mathbf{f}](\varepsilon^*, \Theta) = \min_{\tilde{\varepsilon}^* \in \mathbb{R}, \tilde{\Theta} \in \mathbb{R}^N} \hat{\mathcal{G}}[\hat{\psi}, \hat{\xi}, \hat{\zeta}; \mathbf{f}](\tilde{\varepsilon}^*, \tilde{\Theta}),$$

where the corresponding discrete functional to (20b) is

$$(21b) \quad \begin{aligned} & \hat{\mathcal{G}}[\hat{\psi}, \hat{\xi}, \hat{\zeta}; \mathbf{f}](\tilde{\varepsilon}^*, \tilde{\Theta}) \\ = & \frac{\beta_e}{N_f} \sum_{i=1}^{N_f} (\hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) - \phi_e(\mathbf{x}_i))^2 \\ & + \frac{\beta_{f1}}{N_f} \sum_{i=1}^{N_f} (\nabla \cdot (\hat{\xi}(\mathbf{x}_i, \tilde{\Theta}) \mathbf{b}^\perp(\mathbf{x}_i)) + \nabla \cdot (\hat{\zeta}(\mathbf{x}_i, \tilde{\Theta}) \mathbf{b}(\mathbf{x}_i)) + f(\mathbf{x}_i))^2 \\ & + \frac{\beta_{f2}}{N_f} \sum_{i=1}^{N_f} (\nabla \hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) \cdot \mathbf{b}^\perp(\mathbf{x}_i) - \hat{\xi}(\mathbf{x}_i, \tilde{\Theta}))^2 + \frac{\beta_{f3}}{N_f} \sum_{i=1}^{N_f} (\nabla \hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) \cdot \mathbf{b}(\mathbf{x}_i) - \exp(\tilde{\varepsilon}^*) \hat{\zeta}(\mathbf{x}_i, \tilde{\Theta}))^2 \\ & + \frac{\beta_{N1}}{N_n} \sum_{k=1}^{N_n} (\exp(\tilde{\varepsilon}^*) \nabla_\perp \hat{\psi}(\mathbf{x}_k, \tilde{\Theta}) \cdot \mathbf{n}(\mathbf{x}_k) + \nabla \|\hat{\psi}(\mathbf{x}_k, \tilde{\Theta}) \cdot \mathbf{n}(\mathbf{x}_k)\|)^2 \\ & + \frac{\beta_{N2}}{N_n} \sum_{k=1}^{N_n} (\hat{\xi}(\mathbf{x}_k, \tilde{\Theta}) \mathbf{b}^\perp(\mathbf{x}_k) \cdot \mathbf{n}(\mathbf{x}_k) + \hat{\zeta}(\mathbf{x}_k, \tilde{\Theta}) \mathbf{b}(\mathbf{x}_k) \cdot \mathbf{n}(\mathbf{x}_k))^2, \end{aligned}$$

where the parameters N_f , N_d , N_n have same meanings as stated in Section 3.2. For the identification problem, we merge the Dirichlet boundary term to the observation term. Furthermore, the importance of each term needs to be changed. The weights are redefined as β_e , β_{f1} , β_{f2} , β_{f3} , β_{N1} , β_{N2} . It is important to notice that the identification APFOS-LS formulation (21a) gives a scalar number ε^* and a deep neural network at the same time. Moreover, we do not estimate directly

the anisotropic strength ε , instead we estimate ε^* such that $\varepsilon = \exp(\varepsilon^*)$. This change of variable can always make sure the estimation ε being non-negative. Once the estimation ε^* is obtained from (21a), we can recover ε by $\varepsilon = \exp(\varepsilon^*)$.

The non-AP LS scheme for identifying the anisotropic strength ε is stated in A.2. We may expect that for $\varepsilon \approx 1$, the identification scheme (34) and the AP identification scheme (21) provide similar results. While for $\varepsilon \ll 1$, the the identification scheme (34) will no longer work since its corresponding forward problem is ill-posed at the limit $\varepsilon \rightarrow 0$. In contrast, the AP identification scheme (21) will still provide accurate estimate of ε thanks to the well-posedness of APFOS scheme.

5. NUMERICAL INVESTIGATION

5.1. Setup for anisotropic elliptic equation. Two frameworks are proposed to investigate efficiency of the APFOS scheme.

Setup I. A 2D test case is manufactured thanks to definition of the B -field

$$(22) \quad \mathbf{B} = \begin{pmatrix} m\pi\theta(x^2 - x) \sin(m\pi z) \\ \pi + \theta(2x - 1) \cos(m\pi z) \end{pmatrix}$$

the associated solution ϕ is defined as

$$(23a) \quad \phi = \phi_0 + \varepsilon \cos(2\pi z) \sin(\pi x),$$

$$(23b) \quad \phi_0 = \sin(\omega(\pi x + \theta(x^2 - x) \cos(m\pi z))).$$

The magnetic field \mathbf{b} is represented as the normalization of B -field \mathbf{B} , namely $\mathbf{b} = \frac{\mathbf{B}}{|\mathbf{B}|}$.

Setup II. In a 3D case, the B -field is defined as

$$(24) \quad \mathbf{B} = \begin{pmatrix} 2\pi(x^2 - x) \sin(3\pi y) \sin(\pi z) \\ \pi(x^2 - x) \sin(3\pi y) \sin(\pi z) \\ 2\pi + 2(2x - 1) \sin(3\pi y) \cos(\pi z) + 3\pi(x^2 - x) \cos(3\pi y) \cos(\pi z) \end{pmatrix}$$

and the solution ϕ is given by

$$(25a) \quad \phi = \phi_0 + \varepsilon \cos(2\pi z) \sin(\pi x) \sin(\pi y),$$

$$(25b) \quad \phi_0 = \sin(\omega(\pi x + (x^2 - x) \sin(3\pi y) \cos(\pi z))).$$

For both cases, the right-hand side f is analytically computed via

$$(26) \quad f = -\Delta_{\perp} \phi - \frac{1}{\varepsilon} \Delta_{\parallel} \phi.$$

To verify the efficiency of the proposed deep neural network-based APFOS scheme (APFOS-DNN), three groups of test cases by different parameter settings are carried out in 2D and 3D frameworks, respectively. In real applications, these model parameters are of physical sense, such as the direction of the magnetic field. In this study, they are set by the following way:

Case 1. ($\theta = 0, m = 1, \omega = 2$);

Case 2. ($\theta = 2, m = 1, \omega = 2$);

Case 3. ($\theta = 10, m = 2, \omega = 2$).

In the forward modeling, anisotropic strength ε is set to 1, $1e - 02$ and $1e - 20$ at each *Case* for testing the ability of improving the well-posedness. Fig. 1 illustrates the exact solution ϕ_e for *Setup I* with $\varepsilon = 1e - 20$, where ϕ_e is almost constant along with the direction of the magnetic field \mathbf{b} . More precisely, for *Case 1*, the exact solution $\phi_e(x, z) \approx \phi_e(x)$, while for *Case 2* and *Case 3*, after coordinate transformation, $\phi_e(X, Z) \approx \phi_e(X)$. Fig. 2 shows the exact solution ϕ_e for *Setup II* with $\varepsilon = 1e - 20$.

To describe the numerical performance, the following three types of errors are adopted

$$(27) \quad E_1 = \frac{\|\phi - \phi_e\|_1}{\|\phi_e\|_1}$$

$$(28) \quad E_2 = \frac{\|\phi - \phi_e\|_2}{\|\phi_e\|_2}$$

$$(29) \quad E_\infty = \frac{\|\phi - \phi_e\|_\infty}{\|\phi_e\|_\infty}$$

where ϕ_e denotes the exact solution.

Detailed description on the numerical implementation of the APFOS-DNN method are available in Alg. 1 and 2, forward modeling and identification problem, respectively.

Algorithm 1 APFOS-DNN method for forward modeling

Architecture: Number of discretized grid, hidden layer, neuron, and collocation points of boundary condition and model; activation function; initial weight and bias Θ^0 ; stopping criterion tol and iteration number Iter.

Input of network: Location of grid points $\mathbf{x} \in \mathbb{R}^2$ or $\mathbf{x} \in \mathbb{R}^3$.

Output of network: Prediction of solution $(\hat{\phi}, \hat{\tau}, \hat{\sigma})$ or $(\hat{\phi}, \hat{\tau}, \hat{\chi}, \hat{\sigma})$ of 2D/3D.

- 1: **while** $\|\hat{\mathcal{G}}(\Theta^k) - \hat{\mathcal{G}}(\Theta^{k-1})\| > \text{tol}$ and $k \leq \text{Iter}$ **do**
 - 2: Compute the gradient of loss function $\hat{\mathcal{G}}$ defined by (18b) or (19b) (2D/3D) with respect to Θ^k ;
 - 3: Update network parameters Θ^k by the descent optimization algorithm (i.e., L-BFGS);
 - 4: Compute the value of loss function $\hat{\mathcal{G}}(\Theta^k)$;
 - 5: $k = k + 1$.
 - 6: Perform the prediction by network $(\hat{\phi}, \hat{\tau}, \hat{\sigma})$ or $(\hat{\phi}, \hat{\tau}, \hat{\chi}, \hat{\sigma}) := \mathcal{N}(\mathbf{x}; \Theta)$ to obtain the numerical solution, where Θ denotes the optimal parameters by the iteration;
-

5.2. Numerical implementation and numerical results. Herein a numerical scheme based on the deep learning approach is implemented to obtain the numerical solution of anisotropic elliptic equation (1). Our goal aims at finding the optimal solution of the least-squares problem described by (18), (19) and (21). Therefore, the first-order derivatives with respect to the network parameters

Algorithm 2 APFOS-DNN method for identification problem

Architecture: Number of discretized grid, hidden layer, neuron, and collocation points of boundary condition, model and available observation; activation function; initial weight and bias Θ^0 ; stopping criterion tol and iteration number Iter1, 2.

Input: Initial model coefficient $\varepsilon^{*,0}$ (first guess).

Output: Optimal coefficient ε_{train}^* .

Training step: coefficient updation

- 1: **while** $\|\hat{\mathcal{G}}(\varepsilon^{*,k}) - \hat{\mathcal{G}}(\varepsilon^{*,k-1})\| > \text{tol}$ and $k \leq \text{Iter1}$ **do**
- 2: Compute the gradient of loss function $\hat{\mathcal{G}}$ defined by (21b) with respect to $\varepsilon^{*,k}$;
- 3: Update network parameters $\varepsilon^{*,k}$ by the descent optimization algorithm (i.e., Adam algorithm);
- 4: Compute the value of loss function $\hat{\mathcal{G}}(\varepsilon^{*,k})$;
- 5: $k = k + 1$.

Input: Location of grid points \mathbf{x} and initial coefficient $\varepsilon^{*,0} := \varepsilon_{train}^*$.

Output: Optimal coefficient ε and numerical solution $(\hat{\phi}, \hat{\tau}, \hat{\sigma})$.

Network optimization step with coefficient updation

- 1: **while** $\|\hat{\mathcal{G}}(\Theta^k; \varepsilon^{*,k}) - \hat{\mathcal{G}}(\Theta^{k-1}; \varepsilon^{*,k-1})\| > \text{tol}$ and $k \leq \text{Iter2}$ **do**
 - 2: Compute the gradient of loss function $\hat{\mathcal{G}}$ with respect to $(\Theta^k; \varepsilon^{*,k})$;
 - 3: Update network parameters $(\Theta^k; \varepsilon^{*,k})$ by the descent optimization algorithm (i.e., L-BFGS);
 - 4: Compute the value of loss function $\hat{\mathcal{G}}(\Theta^k; \varepsilon^{*,k})$;
 - 5: $k = k + 1$.
 - 6: Perform the prediction by network $(\hat{\phi}, \hat{\tau}, \hat{\sigma}) := \mathcal{N}(\mathbf{x}; (\Theta; \varepsilon))$ to obtain the numerical solution, where $\varepsilon := \exp(\varepsilon^*)$ and $(\Theta; \varepsilon^*)$ denotes the optimal parameters by the iteration.
-

Θ (i.e., weight \mathcal{W} and bias \mathcal{B}), need to be computed. In addition, we observe that the PDE-based network involves differential operators from the forward model, such as the first-order derivatives with respect to the coordinates, $\partial_x(\cdot)$ and $\partial_y(\cdot)$. Herein, the dominant automatic differentiation (AD) technique is employed that allows efficient computation of the derivatives. The so-called back-propagation algorithm belonging to AD has been popular among large scaled inverse problems, especially for deep learning [1]. L-BFGS [19] and stochastic gradient descent (SGD) algorithm (i.e., Adam algorithm) [17] for large data-sets are used for updating the network parameters.

In the next, we focus on the settings of the architecture of the deep neural network in 2D/3D forward modeling and identification test cases and numerical results.

5.2.1. *Data-driven solutions in 2D.* It is well known that the selection of the hyper-parameters (i.e., the number of layer and neurons) will be crucial to the DNN-based scheme. In general, it could have an influence on the accuracy of the numerical solution and the convergence of the algorithm. In this study, the hyper-parameters could be derived from a twin experiment [25], in which the results by the setting of 3, 4, 5 hidden layers and 40, 60, 80 neurons are compared. It can be concluded from Table 1, when the network contains 4/4 layers and 40/60 neurons, the best results

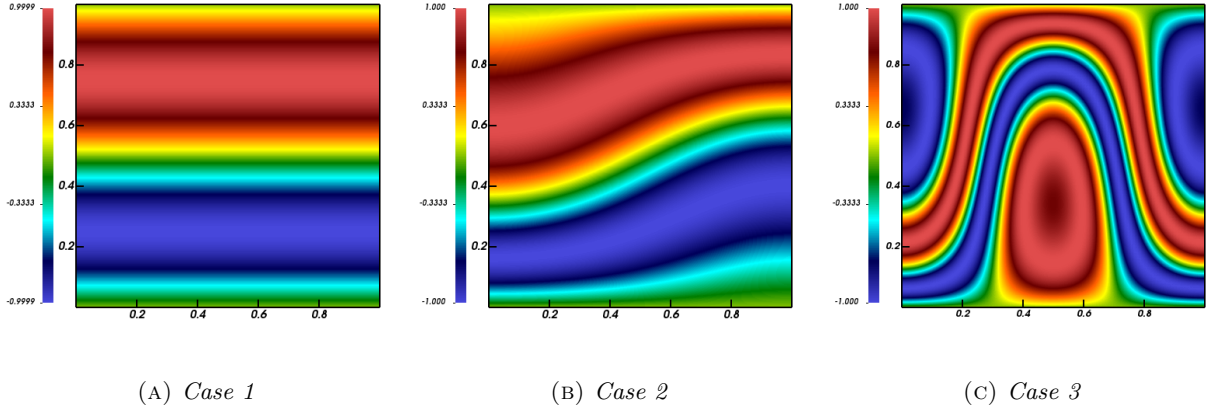


FIGURE 1. Exact solution ϕ_e in three 2D cases with $\varepsilon = 1e - 20$. From left to right: magnetic field \mathbf{b} aligned with z -axis, not aligned with z -axis and closed magnetic field.

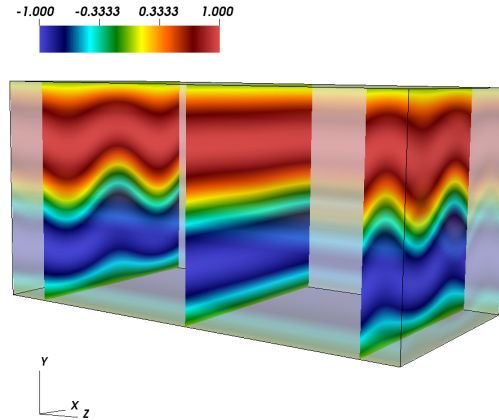


FIGURE 2. Exact solution ϕ_e in 3D case (*Case 2* with $\varepsilon = 1e - 20$).

can be obtained by non-AP-DNN and APFOS-DNN methods, respectively. Therefore, in 2D test cases, such robust hyper-parameters are adopted.

The solution is discretized at a square domain $\Omega = [0, 1] \times [0, 1]$ with $N \times N = 100 \times 100$ grids. Both the numbers of the collocation points at the Dirichlet boundary $\Gamma_D = \{0, 1\} \times [0, 1]$ and Neumann boundary $\Gamma_N = [0, 1] \times \{0, 1\}$ are set to $N_d = N_n = 100$, as well as $N_f = 4000$ inside the domain. According to the definition in [23], collocation points are selected randomly and only the information on such points are available. We take the weights $(\beta_D, \beta_N) = (N_d, 1)$ and $(\alpha_D, \alpha_N) = (N_d, 1)$ in (18b) and (32) respectively.

Table 2 shows the errors between the numerical solution obtained by the DNN-based method and the exact one. Meanwhile, the results at 1000 and 10000 iteration numbers are also recorded for analyzing the convergence (Normal and bold fonts). In test *Case 1*, it can be seen that the E_2 errors easily reach at the magnitude of $1e-04$ and $1e-03$ by using the non-AP-DNN approach when $\varepsilon = 1$ and $1e-02$, respectively. The APFOS-DNN method outperforms the non-AP-based one for different ε , specially for $\varepsilon \ll 1$ and the E_2 error could be less than $1e-04$. More clearly, the numerical solutions at $x = 0.5$ are displayed by Fig. 3a-3c. Those numerical results demonstrate that the accuracy of the APFOS scheme is not relevant to anisotropic strength ε . For the non-AP scheme, the numerical error will be involved when $\varepsilon \ll 1$ and it shows the non-AP scheme diverges for $\varepsilon = 1e-20$.

In test *Case 2*, it could be found from Table 2 that the solution with desired accuracy could not be reached by the non-AP-DNN method in the case of $\varepsilon = 1e-02$, which can be explained as numerical pollution as mentioned in [29]. The algorithm converges at around 5000 iterations and the value of loss function is $8.8e-05$. In the contrary, there is no doubt that the proposed APFOS-DNN approach is as effective as ever. The E_2 error could reach at the magnitude of $1e-04$. The difference could be found in Fig 3d and 3e. Non-AP method cannot approximate the peaks and troughs of the solution well. Fig 3e and 3f demonstrate that combining with the AP scheme, neural network is capable of maintaining the accuracy uniformly with respect to ε .

Additionally, we focus on a complex example (*Case 3*) that contains the closed magnetic field. The difficulty of such problem lies in the singularity phenomenon appears in the center of closed magnetic field. There are lots of singularity points at which the gradient of the field \mathbf{b} tends to infinity. Furthermore, the exact solution covers steep gradient characteristics around closed magnetic field (as shown in Fig. 1c). Therefore, it is necessary to make appropriate adjustments on the architecture of the neural network. Particularly, in the case of $\varepsilon = 1$, herein a neural network covering four hidden layers with 80 neurons is used in APFOS-DNN method. Meanwhile, the numbers of the collocation points inside the domain are changed to $N_f = 5000$.

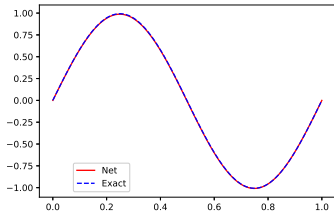
It can be seen from Table 2 that for $\varepsilon = 1$, non-AP-DNN method performs well. The E_2 error reaches at the magnitude of $1e-03$. However, in the case of $\varepsilon = 1e-02$, the E_2 error stops at the magnitude of $1e-01$. Fig. 4c suggests the numerical solution could not approximate the exact one well near the boundary. It can be concluded from Table 2 and Fig. 4d-4e that the APFOS-DNN method could achieve the numerical approximation with expected accuracy, whether inside the domain of closed magnetic field or at the location where the solution contains steep gradient characteristics. Compared with previous examples, the E_∞ errors of *Case 3* (as shown in Table 2) imply that approximating the solution at the singularity point will inevitably cause larger errors. In conclusion, compared with the non-AP-DNN method, the APFOS-DNN method is capable of overcoming numerical pollution of the anisotropic elliptic equation when ε is small, thus it is uniformly accurate with respect to ε . In the next section, we mainly care about its ability of dealing with 3D problem.

5.2.2. Data-driven solutions in 3D. In real applications, the 3D anisotropic elliptic equation could better describe the physical phenomenon. Therefore, herein we mainly care about the numerical implementation and performance of the proposed APFOS-DNN approach in 3D cases. The 3D numerical example established by *Setup II* with parameter setting of *Case 2* is investigated.

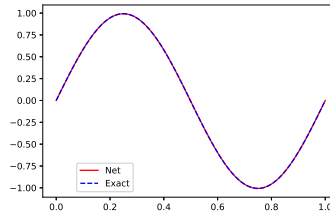
The 3D anisotropic elliptic equation is solved numerically at a cubic domain $\Omega = [0, 1] \times [0, 1] \times [0, 1]$ with $N \times N \times N = 50 \times 50 \times 50$ grids. The neural network covering four hidden layers with 60 neurons at each layer is used in the non-AP-DNN and APFOS-DNN methods. The numbers of the collocation

TABLE 1. Comparison of E_2 errors by using different numbers of neurons and hidden layers. Normal and bold fonts represent the E_2 errors of non-AP-DNN method at *Case 1* with $\varepsilon = 1e-02$ and APFOS-DNN method at *Case 2* with $\varepsilon = 1e-20$.

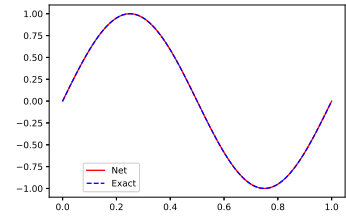
Error \ Layer	Neuron		
	3	4	5
40	8.97e-03	<u>5.25e-03</u>	7.18e-03
	1.36e-03	1.69e-03	1.01e-03
60	8.27e-03	1.62e-02	1.17e-02
	1.47e-03	<u>7.19e-04</u>	1.02e-03
80	3.77e-02	5.41e-02	2.58e-02
	8.03e-04	9.43e-04	8.21e-04



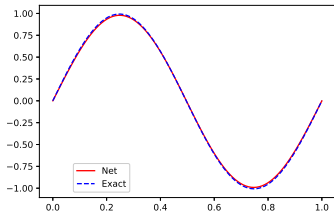
(A) Non-AP-DNN, *Case 1* with $\varepsilon = 1e-02$



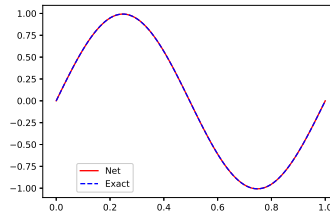
(B) APFOS-DNN, *Case 1* with $\varepsilon = 1e-02$



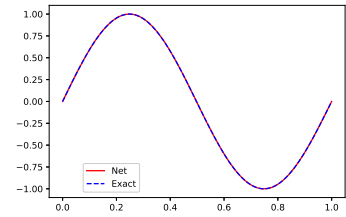
(C) APFOS-DNN, *Case 1* with $\varepsilon = 1e-20$



(D) Non-AP-DNN, *Case 2* with $\varepsilon = 1e-02$



(E) APFOS-DNN, *Case 2* with $\varepsilon = 1e-02$



(F) APFOS-DNN, *Case 2* with $\varepsilon = 1e-20$

FIGURE 3. Comparison of numerical solution in 2D case obtained by non-AP-DNN and APFOS-DNN methods (*Case 1* and *2*, $x = 0.5$).

TABLE 2. Comparison of errors in 2D case. Normal and bold fonts represent the errors at 1000 and 10000 iterations. From top to bottom for each *Case*: E_1 , E_2 and E_∞ .

Error \ Epsilon	Scheme	Case	1		1e-02		1e-20	
Non-AP-DNN	<i>Case 1</i>		2.53e-03	3.44e-04	8.06e-02	5.07e-03	-	-
			2.96e-03	4.30e-04	7.83e-02	5.25e-03	-	-
			1.01e-02	1.29e-03	7.83e-02	6.29e-03	-	-
	<i>Case 2</i>		2.59e-03	4.04e-04	8.53e-01	1.68e-02	-	-
			2.61e-03	4.90e-04	9.07e-01	1.67e-02	-	-
			8.92e-03	1.42e-03	1.16	1.91e-02	-	-
	<i>Case 3</i>		2.10e-01	2.66e-03	9.95e-01	9.55e-02	-	-
			1.98e-01	3.60e-03	9.98e-01	2.34e-01	-	-
			2.80e-01	1.37e-02	1.45	9.82e-01	-	-
APFOS-DNN	<i>Case 1</i>		1.14e-03	9.80e-05	1.96e-03	3.16e-04	2.20e-03	4.74e-04
			1.17e-03	1.06e-04	2.27e-03	3.55e-04	2.50e-03	5.56e-04
			2.46e-03	2.02e-04	5.93e-03	9.90e-04	7.49e-03	1.59e-03
	<i>Case 2</i>		2.06e-03	1.52e-04	3.54e-03	6.71e-04	7.43e-03	6.30e-04
			2.08e-03	1.60e-04	3.95e-03	7.65e-04	8.96e-03	7.19e-04
			3.03e-03	3.09e-04	8.48e-03	1.80e-03	1.91e-02	1.95e-03
	<i>Case 3</i>		1.32e-01	8.01e-03	7.73e-02	2.95e-02	1.23e-01	2.06e-02
			1.51e-01	8.07e-03	8.83e-02	3.67e-02	1.51e-01	2.57e-02
			3.25e-01	1.84e-02	2.29e-01	1.09e-01	4.35e-01	8.20e-02

points at the Dirichlet boundary $\Gamma_D = \{\{0, 1\} \times [0, 1] \times [0, 1], [0, 1] \times \{0, 1\} \times [0, 1]\}$ and Neumann boundary $\Gamma_N = [0, 1] \times [0, 1] \times \{0, 1\}$ are all set to $N_d = N_n = 1000$, as well as $N_f = 10000$ inside the domain. In the 3D case, we take the weights $(\beta_D, \beta_N) = (N_d, 1)$ in (19b).

As expected, some impressive numerical results could be obtained by the neural network-based method in 3D case. Table 3 records three types of errors at 5000 and 10000 iterations (Normal and bold fonts) for different ε and methods. It can be observed that both approaches could produce the solution with E_2 errors of the magnitude of $1e - 03$, except the non-AP-DNN method in the case of $\varepsilon = 1e - 02$ and $1e - 20$. Because of the well-posedness of the APFOS scheme, the value of

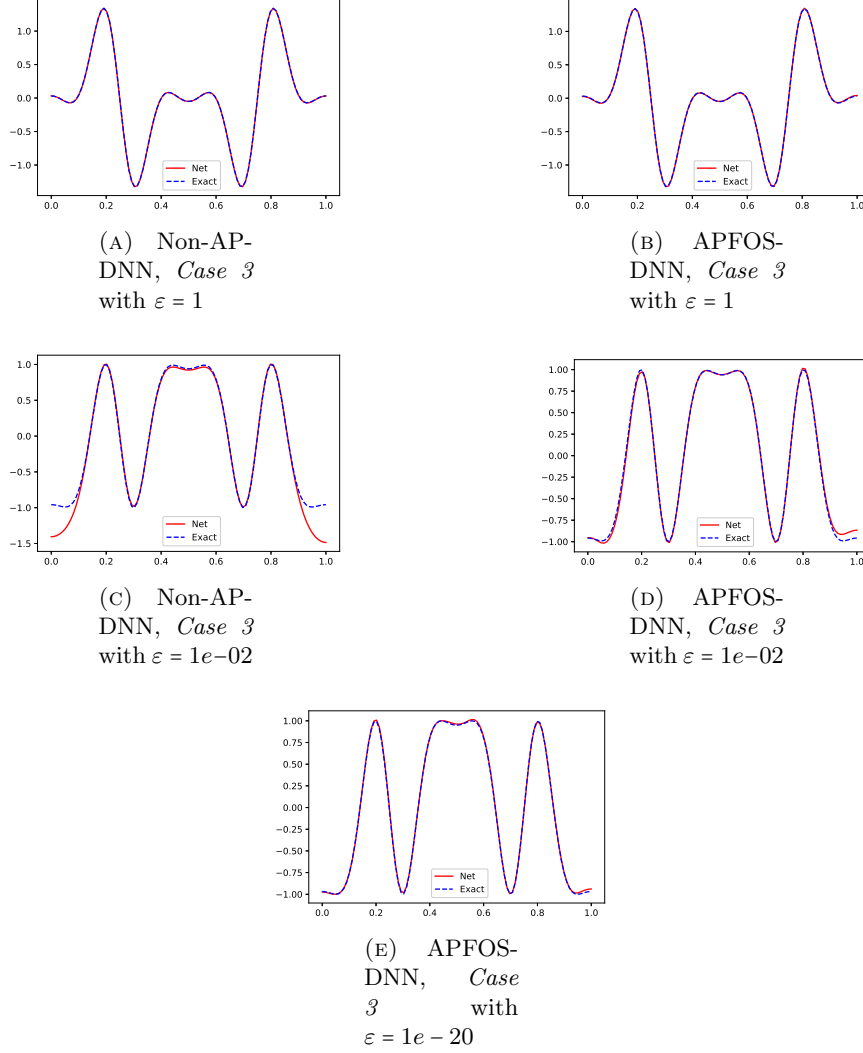


FIGURE 4. Comparison of numerical solution in 2D case obtained by non-AP-DNN and APFOS-DNN methods (*Case 3*, $x = 0.5$).

loss function $\hat{\mathcal{G}}$ defined by (19b) could drop to 0.007 after 10000 iterations when $\varepsilon = 1e - 20$. In addition, an obvious advantage over the classical methods is that the APFOS-DNN method could work well when the numbers of collocation points are far less than the discretized grids used for the classical methods. Notice that the numbers of collocation points are 4000 in 2D cases (40% of uniform grid), and they are only 10000 in 3D cases (8% of uniform grid).

Fig. 5 displays the corresponding numerical results at the location $(x = 0.5, z = 0)$ (Left column), $(x = 0.5, z = 0.5)$ (Middle column), and $(x = 0.5, z = 1)$ (Right column), respectively. The unsatisfied results shown in Fig. 5a-5c indicate the ineffectiveness of the non-AP-DNN method in the case of small ε . The well-posed problem by using the APFOS scheme yields promising results (as shown in Fig. 5d-5i). As it is said in [23], the DNN-based method could achieve good prediction accuracy

TABLE 3. Comparison of errors in 3D case. Normal and bold fonts represent the errors at 5000 and 10000 iterations. From top to bottom for each method: E_1 , E_2 and E_∞ .

Error \ Epsilon	1		1e-02		1e-20	
	Scheme					
Non-AP-DNN	2.78e-03	1.75e-03	9.98e-01	1.01	-	-
	3.28e-03	1.99e-03	1.14	1.14	-	-
	1.26e-02	6.97e-03	1.79	1.75	-	-
APFOS-DNN	4.07e-03	2.08e-03	9.24e-03	5.95e-03	9.81e-03	6.57e-03
	4.46e-03	2.28e-03	1.20e-02	7.30e-03	1.23e-02	8.32e-03
	8.73e-03	4.71e-03	3.79e-02	2.01e-02	4.86e-02	2.65e-02

given a sufficiently expressive neural network architecture and a sufficient number of collocation points if the partial differential equation is well-posed. However, it can be found from Table 3 that the E_2 errors are about 10 times larger than those in 2D case. A reasonable explanation is that the random selection of the collocation points leads to lose the location where the exact solution contains complex structures. In the future work, we will pay more attention to the adaptive grids method that takes the structure of the solution into consideration.

As a result, owing to the nice performance on 3D problem, the neural network-based APFOS method will be a good choice of strategy in real applications.

5.2.3. Data-driven discovery. A reliable numerical prediction not only needs a good knowledge of physical model with initial/boundary conditions and an efficient numerical method, but the physical coefficient in the model. As mentioned in 2D and 3D forward modeling, the anisotropic strength ε in (1) leads to solution with different characteristics, even makes the problem ill-posed. In this scenario, it is necessary to identify the coefficient ε before forward modeling. Herein, we attempt to use the neural network-based approach to retrieve ε by providing limited information of the solution, called observation in the field of inverse problem. The following architecture of neural network is established.

2D examples established by *Setup I* with three cases of parameter settings are investigated, i.e., *Case 1*, *2* and *3*. The identification problem is conducted at a square domain $\Omega = [0, 1] \times [0, 1]$ with $N \times N = 100 \times 100$ grids. A neural network covering four hidden layers with 40 and 60 neurons per layer is adopted in the non-AP-DNN method and APFOS-DNN method. The numbers of the collocation points at the Neumann boundary $\Gamma_N = [0, 1] \times \{0, 1\}$ are set to $N_n = 100$. The exact solution ϕ_e available at $N_f = 3000$ points inside domain is taken as observation. Notice that the DNN-based identification problem includes additional training step in which only the model coefficient ε is updated by the available observation (referring to Alg. 2). The Adam algorithm is employed in the training step and set to 5000 iterations. It is empirically observed that due to a

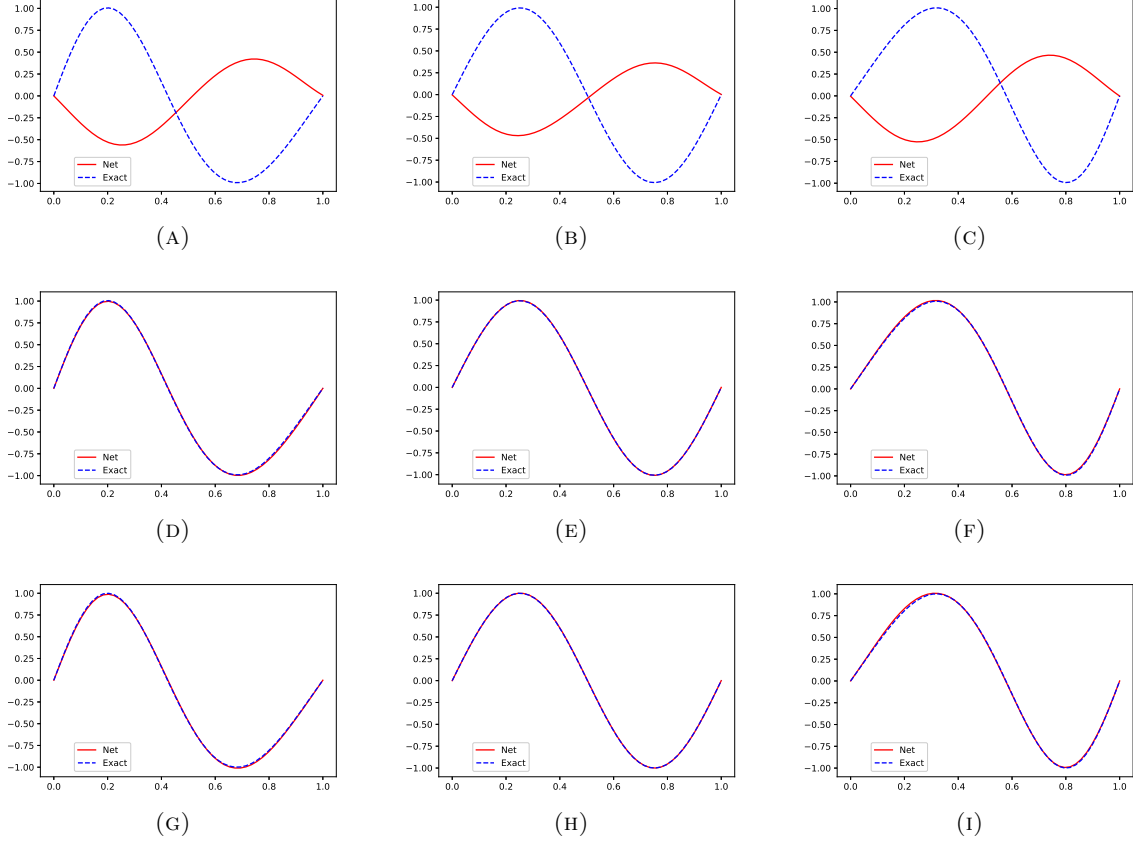


FIGURE 5. Comparison of numerical solution in 3D case obtained by non-AP-DNN and APFOS-DNN methods (*Case 2*). From top to bottom: non-AP-DNN with $\varepsilon = 1e-02$; APFOS-DNN with $\varepsilon = 1e-02$; APFOS-DNN with $\varepsilon = 1e-20$. From left to right: location $(x = 0.5, z = 0)$, $(x = 0.5, z = 0.5)$, and $(x = 0.5, z = 1)$.

reliable estimation on coefficient via such training step, the following simultaneous identification of coefficient ε and optimization of the network parameters Θ could converge fast.

For identifying the anisotropic strength ε , empirically, in the loss function (21b) and (34b) the weights in the term with respect to ε should be changed larger for enhancing the importance of such terms. Herein, we take the weights $(\beta_e, \beta_{f_1}, \beta_{f_2}, \beta_{f_3}, \beta_{N_1}, \beta_{N_2}) = (N_f, 1, 1, N_f, N_n, 1)$ and $(\alpha_e, \alpha_f, \alpha_N) = (N_f, N_f, N_n)$.

Table 4 displays the results of the identification problem obtained by DNN-based non-AP and APFOS methods, the E_2 errors and estimated coefficient ε . The promising results of APFOS-DNN approach indicate its capacity of dealing with the forward and inverse problem. In the contrary, the non-AP-DNN one could not yield results with desired accuracy and particularly it does not work in *Case 3*. Finally, to show the convergence of the proposed method, the changing of estimated coefficient ε with the increasing iterations is provided by Fig. 6a-6c. It can be seen that the APFOS-DNN method converges fast and stably both in the training and prediction steps. Additionally,

TABLE 4. Comparison of identification results. Normal and bold fonts represent estimated ε and E_2 errors of the numerical solution.

Scheme \ Exact ε	Case	1	1e-02	1e-20	
		<hr/>			
Non-AP-DNN	<i>Case 1</i>	9.96e-01	9.97e-03	4.16e-05	
		3.24e-03	1.22e-04	1.40e-04	
	<i>Case 2</i>	9.97e-01	8.20e-03	1.54e-03	
		3.55e-03	1.70e-03	1.51e-03	
	<i>Case 3</i>	4.38e-03	2.26e-02	2.17e-02	
		3.58e-01	2.23e-02	2.26e-02	
	<hr/>				
	APFOS-DNN	<i>Case 1</i>	1.00	1.00e-02	4.58e-07
			4.34e-04	1.21e-04	3.33e-05
<i>Case 2</i>		1.00	1.00e-02	1.42e-05	
		4.99e-04	8.29e-05	2.73e-05	
<i>Case 3</i>		9.71e-01	1.02e-02	4.11e-03	
		7.36e-03	2.25e-03	1.93e-03	

some test cases including observation with gaussian noise of different levels are taken into account. The results indicate that the APFOS-DNN method is robust to gaussian noise. As a result, compared with the non-AP-DNN method, the proposed APFOS-DNN method is more suitable to identify any anisotropic strength $\varepsilon \geq 0$.

6. CONCLUSION AND PERSPECTIVES

In this paper, an asymptotic preserving scheme based on first-order system (APFOS) for solving the anisotropic elliptic equations is proposed. The APFOS scheme is well-posed with respect to all anisotropic strength $\varepsilon \geq 0$. We also rewrite the APFOS scheme into least-squares formulations. Then the APFOS scheme is numerically solved in deep neural network (DNN) framework. Comparing to non-AP method, our APFOS scheme is uniformly accurate with respect to all ε . It can treat various 2D/3D anisotropic elliptic equations with aligned, non-aligned, closed magnetic field. Finally, the APFOS scheme is applied to identify the anisotropic strength ε . Similarly, the APFOS scheme can predict accurately ε in all cases, whereas the non-AP scheme does not work if we want to identify $\varepsilon \ll 1$.

The further coming work is to develop adaptive DNN for the APFOS scheme. Indeed, in current numerical performance of the APFOS scheme, we first select hyper parameters, such as number of layers, neurons, collocation points *etc.*, from a simple twin experiment, then all tests will use those hyper parameters. However, due to the complexity of certain tests, such as 3D tests or closed

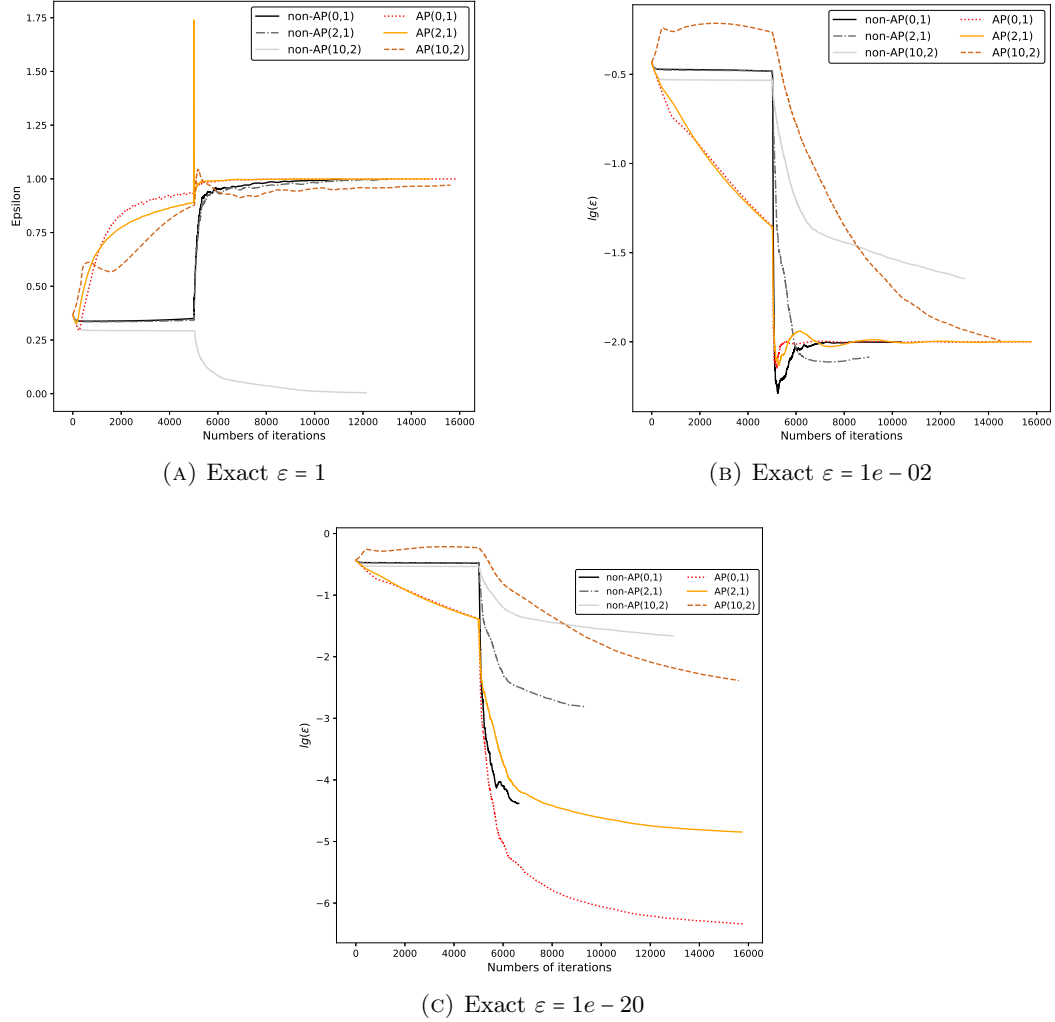


FIGURE 6. Comparison of identification results. Non-AP/AP(θ, m) in the legend represents the non-AP-DNN/APFOS-DNN method with parameters (θ, m).

magnetic field tests, the accuracy seems not to be optimal. As pointed in [5], an adaptive DNN will be needed, where least-squares functional can be used as *a posteriori* estimator, thus the parameters of DNN will be selected in iterations.

7. ACKNOWLEDGEMENTS

This work has been supported by Heilongjiang Natural Science Foundation (LH2019A013).

APPENDIX A. THE NON-AP LEAST SQUARES SCHEMES FOR THE ANISOTROPIC EQUATION (11a)

A.1. **The non-AP LS scheme for solving the anisotropic equation** (11a). For the anisotropic elliptic equation (10), the LS formulations is to find $\phi \in H^2(\Omega)$ such that

$$(30) \quad \mathcal{L}(\phi; \mathbf{f}) = \min_{\psi \in H^2(\Omega)} \mathcal{L}(\psi; \mathbf{f}).$$

LS functional is inspired from the balanced least-squares (BLS) functional of [5], in which they point out that usual norm on boundary condition is weaker than that for the equation, thus a balanced LS functional is adopted as follows

$$(31) \quad \mathcal{L}(\psi; \mathbf{f}) = \|\varepsilon \Delta_{\perp} \psi + \Delta_{\parallel} \psi + \varepsilon f\|_{0,\Omega}^2 + \|\psi - g\|_{3/2,\Gamma_D}^2 + \|\varepsilon \nabla_{\perp} \psi \cdot \mathbf{n} + \nabla_{\parallel} \psi \cdot \mathbf{n}\|_{1/2,\Gamma_N}^2.$$

Then we discretize (30)-(31) as follows. The discrete LS formulation reads

$$\hat{\mathcal{L}}[\hat{\phi}; \mathbf{f}](\Theta) = \min_{\tilde{\Theta} \in \mathbb{R}^N} \hat{\mathcal{L}}[\hat{\psi}; \mathbf{f}](\tilde{\Theta}),$$

where $\hat{\phi} = \hat{\phi}(\mathbf{x}, \Theta)$ represents an approximation of solution of anisotropic equation (10), depending on the location $\mathbf{x} \in \mathbb{R}^2$ and the weights and biases denoted as $\Theta \in \mathbb{R}^N$. The corresponding discrete LS functional is

$$(32) \quad \begin{aligned} \hat{\mathcal{L}}[\hat{\psi}; \mathbf{f}](\tilde{\Theta}) = & \frac{1}{N_f} \sum_{i=1}^{N_f} (\varepsilon \Delta_{\perp} \hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) + \Delta_{\parallel} \hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) + \varepsilon f(\mathbf{x}_i))^2 + \frac{\alpha_D}{N_d} \sum_{j=1}^{N_d} (\hat{\psi}(\mathbf{x}_j, \tilde{\Theta}) - g(\mathbf{x}_j))^2 \\ & + \frac{\alpha_N}{N_n} \sum_{k=1}^{N_n} (\varepsilon \nabla_{\perp} \hat{\psi}(\mathbf{x}_k, \tilde{\Theta}) \cdot \mathbf{n}(\mathbf{x}_k) + \nabla_{\parallel} \hat{\psi}(\mathbf{x}_k, \tilde{\Theta}) \cdot \mathbf{n}(\mathbf{x}_k))^2. \end{aligned}$$

Notice that the discrete LS functional (32) is an approximation of (31), where $H^{3/2}$ and $H^{1/2}$ norms are replaced by weighted L^2 norms. The weights α_D and α_N are specified in Section 5.

A.2. **The non-AP LS scheme for identifying the anisotropic strength ε .** Let us now consider a non-AP identification scheme for the anisotropic problem (10). The identification LS formulations is to find $\varepsilon \in \mathbb{R}^+$ and $\phi \in H^2(\Omega)$ such that

$$\mathcal{L}(\varepsilon, \phi; \mathbf{f}) = \min_{\tilde{\varepsilon} \in \mathbb{R}^+, \psi \in H^2(\Omega)} \mathcal{L}(\tilde{\varepsilon}, \psi; \mathbf{f}).$$

The corresponding LS functional, by adding *priori* information ϕ_e to (31), reads

$$(33) \quad \begin{aligned} \mathcal{L}(\tilde{\varepsilon}, \psi; \mathbf{f}) = & \|\psi - \phi_e\|_{0,\Omega}^2 \\ & + \|\tilde{\varepsilon} \Delta_{\perp} \psi + \Delta_{\parallel} \psi + \tilde{\varepsilon} f\|_{0,\Omega}^2 + \|\psi - g\|_{3/2,\Gamma_D}^2 + \|\tilde{\varepsilon} \nabla_{\perp} \psi \cdot \mathbf{n} + \nabla_{\parallel} \psi \cdot \mathbf{n}\|_{1/2,\Gamma_N}^2, \end{aligned}$$

where $\mathbf{f} = (f, g)$, $\tilde{\varepsilon} \in \mathbb{R}^+$ and $\psi \in H^2(\Omega)$.

The discrete LS formulation reads

$$(34a) \quad \hat{\mathcal{L}}[\hat{\phi}; \mathbf{f}](\varepsilon^*, \Theta) = \min_{\tilde{\varepsilon}^* \in \mathbb{R}, \tilde{\Theta} \in \mathbb{R}^N} \hat{\mathcal{L}}[\hat{\psi}; \mathbf{f}](\tilde{\varepsilon}^*, \tilde{\Theta}).$$

where the corresponding discrete LS functional to (33) is

$$\begin{aligned}
 \hat{\mathcal{L}}[\hat{\psi}; \mathbf{f}](\tilde{\varepsilon}^*, \tilde{\Theta}) = & \frac{\alpha_e}{N_f} \sum_{i=1}^{N_f} \left(\hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) - \phi_e(\mathbf{x}_i) \right)^2 \\
 (34b) \quad & + \frac{\alpha_f}{N_f} \sum_{i=1}^{N_f} \left(\exp(\tilde{\varepsilon}^*) \Delta_{\perp} \hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) + \Delta_{\parallel} \hat{\psi}(\mathbf{x}_i, \tilde{\Theta}) + \exp(\tilde{\varepsilon}^*) f(\mathbf{x}_i) \right)^2 \\
 & + \frac{\alpha_N}{N_n} \sum_{k=1}^{N_n} \left(\exp(\tilde{\varepsilon}^*) \nabla_{\perp} \hat{\psi}(\mathbf{x}_k, \tilde{\Theta}) \cdot \mathbf{n}(\mathbf{x}_k) + \nabla_{\parallel} \hat{\psi}(\mathbf{x}_k, \tilde{\Theta}) \cdot \mathbf{n}(\mathbf{x}_k) \right)^2.
 \end{aligned}$$

REFERENCES

- [1] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18(153):1–43, 2018.
- [2] S. Bernard, V. Monteiller, D. Komatitsch, and P. L. Lasaygues. Ultrasonic computed tomography based on full-waveform inversion for bone quantitative imaging. *Physics in Medicine & Biology*, 2017.
- [3] C. Besse, P. Degond, F. Deluzet, J. Claudel, G. Gallice, and C. Tessieras. A model hierarchy for ionospheric plasma modeling. *Mathematical Models & Methods in Applied Sciences*, 14(03):393–415, 2004.
- [4] C. Besse, F. Deluzet, C. Negulescu, and C. Yang. Efficient numerical methods for strongly anisotropic elliptic equations. *Journal of Scientific Computing*, 55(1):231–254, 2013.
- [5] Z. Cai, J. Chen, M. Liu, and X. Liu. Deep least-squares methods: an unsupervised learning-based numerical method for solving elliptic pdes. *Journal of Computational Physics*, 420:109707, 2020.
- [6] Z. Cai, R. Lazarov, and T. A. M. F. McCormick. First-order system least squares for second-order partial differential equations: part i. *SIAM Journal on Numerical Analysis*, 31(6):1785–1799, 1994.
- [7] Q. Chan-Wai-Nam, J. Mikael, and X. Warin. Machine learning for semi linear pdes. *Journal of Scientific Computing*, 79(3):1667–1712, 2019.
- [8] P. Degond and F. Deluzet. Asymptotic-preserving methods and multiscale models for plasma physics. *Journal of Computational Physics*, 336:429–457, 2017.
- [9] P. Degond, F. Deluzet, A. Lozinski, J. Narski, and C. Negulescu. Duality-based asymptotic-preserving method for highly anisotropic diffusion equations. *Communications in Mathematical Sciences*, 10(1):1–31, 2012.
- [10] P. Degond, F. Deluzet, and C. Negulescu. An asymptotic preserving scheme for strongly anisotropic elliptic problems. *Multiscale Modeling & Simulation*, 8(2):645–666, 2009.
- [11] P. Degond, A. Lozinski, J. Narski, and C. Negulescu. An asymptotic-preserving method for highly anisotropic elliptic equations based on a micro-macro decomposition. *Journal of Computational Physics*, 231(7):2724–2740, 2012.
- [12] F. Deluzet and J. Narski. A two field iterated asymptotic-preserving method for highly anisotropic elliptic equations. *Multiscale Modeling & Simulation*, pages 434–459, 2019.
- [13] L. C. Evans. *Partial Differential Equations*. American Mathematical Society, 2010.
- [14] J. Han, M. Nica, and A. R. Stinchcombe. A derivative-free method for solving elliptic partial differential equations with deep neural networks. *Journal of Computational Physics*, 419:109672, 2020.
- [15] J. Huang, H. Wang, and H. Yang. Int-deep: a deep learning initialized iterative method for nonlinear problems. *Journal of Computational Physics*, 419:109675, 2020.
- [16] C. Huré, H. Pham, and X. Warin. Deep backward schemes for high-dimensional nonlinear pdes. *Mathematics of Computation*, 89(324):1, 2020.
- [17] D. Kingma and J. Ba. Adam: a method for stochastic optimization. *Computer Science*, 2014.
- [18] F. X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations: theoretical aspects. *Tellus Series A-Dynamic Meteorology and Oceanography*, 38(2):97–110, 1986.
- [19] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.
- [20] E. T. Meier. *Modeling plasmas with strong anisotropy, neutral fluid effects, and open boundaries*. PhD thesis, University of Washington, 2011.
- [21] J. D. Meiss. *Plasma confinement*. Dover Publications, 2003.
- [22] R.-E. Plessix. A review of the adjoint-state method for computing the gradient of a functional with geophysical applications. *Geophysical Journal International*, 167(2):495–503, 2006.

- [23] M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.
- [24] M. Tang and Y. Wang. An asymptotic preserving method for strongly anisotropic diffusion equations based on field line integration. *Journal of Computational Physics*, 330:735–748, 2017.
- [25] O. Titaud, A. Vidard, I. Souopgui, and F. X. Le Dimet. Assimilation of image sequences in numerical models. *Tellus Series A-dynamic Meteorology & Oceanography*, 62(1):30–47, 2010.
- [26] Y. Wang, W. Ying, and M. Tang. Uniformly convergent scheme for strongly anisotropic diffusion equations with closed field lines. *SIAM Journal on Scientific Computing*, pages B1253–B1276, 2018.
- [27] C. Yang, J. Claustre, and F. Deluzet. Iterative solvers for elliptic problems with arbitrary anisotropy strengths. *Multiscale Modeling & Simulation*, 16(4):1795–1823, 2018.
- [28] C. Yang, F. Deluzet, and J. Narski. On the numerical resolution of anisotropic equations with high order differential operators arising in plasma physics. *Journal of Computational Physics*, 386:502–523, 2019.
- [29] C. Yang, F. Deluzet, and J. Narski. Preserving the accuracy of numerical methods discretizing anisotropic elliptic problems. submitted.

Email address: long.li@hit.edu.cn

Email address: yangchang@hit.edu.cn