Physics-Informed Neural Networks for 1-D Steady-State Diffusion-Advection-Reaction Equations

Physics-Informed Neural Networks for 1D Steady-State Diffusion–Advection–Reaction Equations

Laura Laghi,[*,a,b] Enrico Schiassi,[b] Mario De Florio,[b] Roberto Furfaro,[b,c] and Domiziano Mostacci[a]

[a] *Department of Industrial Engineering*
Alma Mater Studiorum - Università di Bologna
Via dei Colli 16, 40136, Bologna, Italy

[b] *Department of Systems and Industrial Engineering, The University of Arizona*
1127 E. James E. Rogers Way, Tucson, AZ 85721, USA

[c] *Department of Aerospace and Mechanical Engineering, The University of Arizona*
1130 N Mountain Ave, Tucson, AZ 85721, USA

[*]Email: laura.laghi6@unibo.it

Number of pages: 43
Number of tables: 35
Number of figures: 5

Abstract

This work aims to solve six problems with four different Physics-Informed Machine Learning frameworks and compare the results in terms of accuracy and computational cost. First, we considered the Diffusion-Advection-Reaction Equations, which are second-order linear differential equations with two boundary conditions. The first algorithm is the classic Physics-Informed Neural Networks (PINNs). The second one is Physics-Informed Extreme Learning Machine (PIELM). The third framework is Deep-Theory of Functional Connections (Deep-TFC), a multi-layer NN based on the solution approximation via a constrained expression that always analytically satisfies the boundary conditions. Finally, the last algorithm is the Extreme Theory of Functional Connections X-TFC, which combines TFC and shallow NN with random features (e.g., Extreme Learning Machine, ELM). The results show that, for these kinds of problems, ELM-based frameworks, especially X-TFC, overcome those using Deep NN both in terms of accuracy and computational time.

Keywords — Extreme Learning Machine, Functional Interpolation, Physics-Informed Neural Networks

I. INTRODUCTION

Ordinary and partial differential equations are usually solved and modeled using the Finite Elements Method (FEM). The domain is discretized via mesh generation, which depends on the geometry of the problem. The equation will be solved on the points that define each element of the mesh. If we need to know the solution in a different position instead, we can only obtain the answer via interpolation. This leads to a significantly increasing of computational times and in addition, to a lack of accuracy in the solution. Moreover, FEM techniques suffer from the curse of dimensionality, and they may lead to several numerical instabilities such as spurious oscillations even if the solution is regular.

Over the past few years, the development of machine learning techniques and artificial intelligence has led to a growing interest in neural networks, especially for their application in solving differential equations. The general term to refer to them is Artificial Neural Networks (ANN). The concept that lies behind ANN is to replicate the connections among brain neurons and create a network that, under a given input, brings out a reliable output. Physics-Informed Neural Networks (PINNs) are a class of ANN specifically applied to learn how to solve a differential equation following the physics behind the problem [1]. Once the training is done, the accuracy of the solution can be tested on test points. Test points evaluation is directly computed from the NN approximation and this avoids interpolation techniques and error propagation that derives from using them.

Nevertheless, the efficiency and applicability of NNs for the approximation of differential problems can present some drawbacks. For example, literature has often been reported a failure of learning machines techniques to deal with sharp gradient problems [2].

This work will consider the linear one-dimensional second-order Diffusion-Advection-Reaction Equations. The general formulation can describe, for example, the temperature behavior in Heat Transfer problems, and the pollutant concentration in Atmosphere Transport. Moreover, it can extensively represents the oil-water fraction in Reservoir Engineering, and also transport phenomena in porous media, such as membranes. The project aims to compare four PINNs frameworks to solve a class of linear 1-D steady-state Diffusion-Advection-Reaction Equations with constant coefficients where the solution is known analytically and it is regular. This class of problems is computational challenging because of the steep gradients that arise in the computation.

The report is organized as follows. Firstly, a brief introduction on the four frameworks:

Classic Physics-Informed Neural Networks (PINN) [3], Physics-Informed Extreme Learning Machine (PIELM) [4], Deep Theory of Functional Connections (Deep-TFC) [5], and Extreme Theory of Functional Connections (X-TFC)[6] are based on [7]. In the following section we present six problems we aim to solve and the characteristic parameters are here specified for each case. Then, we outline the results and highlight which algorithms are more accurate for each case. Finally, in the conclusions derived from the findings, the results reveal how shallow neural networks are a robust numerical approach to obtain accurate solutions with high competitive computational times for this kind of steep gradient solutions.

II. PINNS APPROACH TO SOLVING ODE BOUNDARY VALUE PROBLEMS

Computational techniques are widely used to solve various problems in engineering. The interest in involving neural networks for this purpose has recently grown. There are two main reasons why physics-informed machine learning techniques are becoming increasingly popular in the computational field. Firstly, they aim to solve supervised learning tasks while respecting any given laws of physics described by general differential equations. Secondly, due to the prohibitive cost of data acquisition, faster simulations and more accurate computational solutions are necessary to obtain valuable and reliable results.

If we consider the physical laws that govern the time-dependent dynamics of a system, this prior information can be used as a constraint that limits the space of admissible solutions to a manageable size. Such neural networks are built to respect every symmetry, invariance, or conservation principle originating from the physical laws governing the empirical data, modeled by general time-dependent and nonlinear partial differential equations. This simple yet powerful framework allows to deal with a wide range of problems in computational science. It introduces a potentially transformative technology leading to the development of new data-efficient and physics-informed learning machines, new classes of numerical solvers for ordinary or partial differential equations [8, 9, 10, 11, 12], as well as new data-driven approaches for model inversion and control optimization[13].

Parametric DEs are powerful tools used for mathematical modeling of various problems, which are present in various fields. There exist two types of DEs: parametric ordinary DEs (ODEs), which are univariate independent variable equations, and parametric partial DEs (PDEs),

which are multivariate independent variable equations. The solution of these equations can be used to simulate, identify, characterize, design, and verify the design of a variety of systems. In many practical problems, it is not trivial to find an analytical solution to these parametric DEs. Sometimes the analytical solution does not even exist. Thus, for these cases, it is preferred to solve these equations numerically. For the numerical solution of ODEs, various methods exist, with the most popular being based on the Runge-Kutta family. Other methods include finite difference, Chebyshev-Picard iteration, and pseudo-spectral methods [14]. This work aims to solve a set of problems with four different techniques and compare the results obtained in terms of accuracy and computational cost.

II.A. Standard PINN and PIELM

As already explained, numerical methods implemented through machine learning frameworks can be applied to a wide range of differential equations and systems of differential equations. Here, we will analyze a comparison between four different techniques used to solve seven physics problems subject to boundary conditions. The general equation that could represent all the different problems is better known as a one-dimensional (1D) second-order linear boundary value problem (BVP). The differential equation of a typical second-order BVP in the space domain can be expressed, in its implicit form, as follows:

$$\mathcal{F}\left(x, u(x), \dot{u}(x), \ddot{u}(x)\right) = f(x) \qquad \text{subject to} \qquad \begin{cases} u(x = x_0) = u_0 \\ u(x = x_f) = u_f \end{cases} \tag{1}$$

where the independent variable is the space $x \in [x_0, x_f]$, $u(x)$ is the unknown function, $\dot{u}(x), \ddot{u}(x)$ are its first and second derivatives, respectively, and $f(x)$ is the forcing term.

II.B. Standard Physics-Informed Neural Network

In the first place, we explain how Physics-Informed Neural Networks (PINN) can be used to solve the problem introduced in the previous section. According to the standard PINN framework, the function $u(x)$ could be approximated with a neural network, as $u(x) \approx u_{NN}\left(x, \theta\right)$ where θ are the NN parameters, w, b, and β, which will be trained via gradient based methods. To compute the derivatives of the NN, the Automatic Differentiation (AD) has been implemented in MATLAB,

obtaining $\dot{u}_{NN}(x,\theta)$ and $\ddot{u}_{NN}(x,\theta)$. The differential equation, in terms of NN approximation, has the form:

$$\begin{cases} \mathcal{F}\left(x, u_{NN}\left(x,\theta\right), \dot{u}_{NN}\left(x,\theta\right), \ddot{u}_{NN}\left(x,\theta\right)\right) = f(x) \\ u_{NN}\left(x_0,\theta\right) = u_{NN_0} \\ u_{NN}\left(x_f,\theta\right) = u_{NN_f} \end{cases} \tag{2}$$

Therefore, to train the network, the Mean Squared Error (MSE) has to be minimized with the respect to the NN parameters

$$\min_{\theta}(MSE), \quad \theta \in \mathcal{R} \tag{3}$$

where

$$MSE = MSE_{\mathcal{R}} + MSE_{u_0} + MSE_{u_f} \tag{4}$$

$MES_{\mathcal{R}}$ represents the MSE of the residual approximated with the neural network computed on the internal training points (5):

$$MSE_{\mathcal{R}} = \frac{1}{N_R}\sum_{i=1}^{N_R} \left|\mathcal{F}\left(x_i, u_{NN}\left(x_i,\theta\right), \dot{u}_{NN}\left(x_i,\theta\right), \ddot{u}_{NN}\left(x_i,\theta\right)\right) - f(x_i)\right|^2 \tag{5}$$

while MSE_{u_0} and MSE_{u_f} represent the mean squared errors of the residual approximated with the neural network computed on boundary points

$$MSE_{u_0} = |u_{NN_0} - u_0|^2 \tag{6}$$

$$MSE_{u_f} = |u_{NN_f} - u_f|^2 \tag{7}$$

Once the hyper parameters are computed, they are plugged into the NN approximation, tom obtain the numerical solution of the equation.

II.C. Physics-Informed Extreme Learning Machine

In this section we present how to apply PIELM to solve a generic second-order BVP in the space domain. The first step of the PIELM method is to expand the unknown solution with a shallow neural network:

$$u(x) \approx u_{ELM}(x,\beta) = \sum_{j=1}^{L} \beta_j\, \sigma(w_j x + b_j) \tag{8}$$

where L is the number of neurons, $w_j \in \mathbf{R}$ is the input weights vector connecting the j^{th} neuron with the input nodes, $\beta_j \in \mathbf{R}$ is the output weights vector that connecting the j^{th} neuron with the output node, b_j is the bias of the j^{th} neuron, and $\sigma(\cdot)$ are the activation functions, which are always chosen by the user. Using a shallow neural network means that we will train a one-layer NN, where input weights w_j and bias b_j are randomly selected and not tuned during the training. Thus, they are known parameters. Consequently, the output weights β_j are the only unknown variables and this offers the advantage to compute the derivatives analytically. Therefore, the possibility to avoid the computational cost of the automatic differentiation. The approximated function could be expressed as:

$$u_{ELM}\left(x,\beta\right) = \sum_{j=1}^{L} \beta_j \, \sigma(w_j x + b_j) = \sum_{j=1}^{L} \beta_j \, \sigma_j = \begin{bmatrix} \sigma_1 \\ \vdots \\ \sigma_L \end{bmatrix}^T \beta_j = \sigma^T(x)\, \beta \tag{9}$$

Replacing the unknown solutions and its derivatives with the neural network approximation, the BVP could be rewritten as:

$$\begin{cases} \mathcal{F}\left(x, u_{ELM}\left(x,\beta\right), \dot{u}_{ELM}\left(x,\beta\right), \ddot{u}_{ELM}\left(x,\beta\right)\right) = f(x) \\ u_{ELM}\left(x_0,\beta\right) = u_{ELM_0} = \sigma^T(x_0)\,\beta = \sigma_0^T\,\beta \\ u_{ELM}\left(x_f,\beta\right) = u_{ELM_f} = \sigma^T(x_f)\,\beta = \sigma_f^T\,\beta \end{cases} \tag{10}$$

The independent variable x is discretized in $N_{\mathcal{F}}$ points, and $x \in (x0, xf)$ – domain boundaries excluded. Then the implicit equation becomes

$$\mathcal{F}\left(x_i, u_{ELM}\left(x_i,\beta\right), \dot{u}_{ELM}\left(x_i,\beta\right), \ddot{u}_{ELM}\left(x_i,\beta\right)\right) = f(x_i); \qquad \forall i = 1, ..., N_{\mathcal{F}} \tag{11}$$

The only unknowns of this algorithm are the output weights β and, by rearranging the terms, the following system of linear algebraic equations is obtained:

$$\mathbf{A}_{\mathcal{F}}\,\beta = b_{\mathcal{F}} \tag{12}$$

By taking into account the boundary conditions, the system can be written as:

$$\mathbf{A}_{ELM}\,\beta = \mathbf{b}_{ELM} \tag{13}$$

where

$$\mathbf{A}_{ELM} = \begin{bmatrix} \mathbf{A}_{\mathcal{F}} & \sigma_0^T & \sigma_f^T \end{bmatrix} \quad \text{and} \quad b_{ELM} = \begin{bmatrix} \mathbf{b}_{\mathcal{F}} & u_0 & u_f \end{bmatrix} \tag{14}$$

Techniques similar to PIELM can be often found in literature, such as [15], where Dong et al. developed a technique based on deep neural networks and hidden-layer parameters that are randomly generated and optimized [16] for solving linear and non-linear PDEs. For the advection and the diffusion case showed in [15], they achieved good results compared to traditional methods such as FEM.

II.D. Deep-TFC and X-TFC

The Theory of Functional Connections (TFC) is a mathematical tool developed by Professor Mortari in Ref.[7], which aims to solve numerically problems in computational science such as differential equations [17, 11, 18]. This new method consists in deriving analytical expressions, called *constrained expressions*, that can be used to represent functions subject to a set of linear constraints. We call $f(x)$ the resulting constrained expression. It is expressed as a function of $g(x)$ arbitrarily chosen. Regardless of how you choose $f(x)$, the constrained expression will always satisfy the set of linear constraints. TFC can be apply to solve a generic second-order BVP in the space domain using both shallow and deep neural networks [17]. To solve the problem expressed in Eq.(1), the unknown function is approximated by the so-called *constrained expression* (CE) [7], which can be expressed in its general form:

$$u(x) \approx u_{CE}\left(x, g(x)\right) = g(x) + \phi_1(x)\underbrace{\left(u_0 - g_0\right)}_{\rho_1} + \phi_2(x)\underbrace{\left(u_f - g_f\right)}_{\rho_2} \tag{15}$$

where $g(x)$ is the free chosen function, $\phi_{1,2}$ are the switching functions, and $\rho_{1,2}$ are the projection functions, which are functions that project the constrained expression on the boundary. The switching functions are built in such a way that the constrained expression can analytically satisfy the boundary conditions $\phi_1(u_0) = 1, \phi_2(u_0) = 0$ and $\phi_1(u_f) = 1, \phi_2(u_f) = 0$.

Then, the derivatives can be written as

$$\dot{u}_{CE}(x) = \dot{g}(x) + \dot{\phi}_1(x)\,\rho_1 + \dot{\phi}_2(x)\,\rho_2 \tag{16}$$

$$\ddot{u}_{CE}(x) = \ddot{g}(x) + \ddot{\phi}_1(x)\,\rho_1 + \ddot{\phi}_2(x)\,\rho_2 \tag{17}$$

Therefore, if we use a deep neural network as free chosen function, the technique will be called Deep Theory of Functional Connections (Deep-TFC). On the other hand, if $g(x)$ is a shallow neural network computed with ELM, the framework is known as Extreme Theory of Functional Connections (X-TFC)[6].

III. DIFFUSION-ADVECTION-REACTION EQUATIONS

The aim of this work is to solve the Diffusion - Advection - Reaction equation with the four frameworks presented in the previous section and compare the results. The general form of the problem is defined as follows:

$$\begin{cases} -\mu u''(x) + \gamma u'(x) + \lambda u(x) = f(x) \qquad \text{where} \qquad x \in [0,1] \\ \nu_0 u'(0) + \rho_0 u(0) = g_0 \\ \nu_1 u'(1) + \rho_1 u(1) = g_1 \end{cases} \tag{18}$$

where μ is the diffusion coefficient, γ is the advection coefficient and λ is the reaction one. Dirichlet and Neumann boundary conditions are derived by setting $\nu_i = 0$ or $\rho_i = 0$. The following problems are particular cases derived from Eq.(18).

III.A. Sinusoidal-bump Problem

The first case is a sinusoidal-bump 1D boundary value problem with homogeneous Dirichlet boundary conditions, where k represents the number of oscillations in the domain:

$$\begin{cases} u''(x) + (4k^2\pi^2 - 1)u(x) = 4k\pi e^x\, cos(2k\pi x) \qquad \text{where} \qquad x \in [0,1] \\ u(0) = 0 \\ u(1) = 0 \qquad \text{while} \qquad k \in \mathcal{N} \end{cases} \tag{19}$$

The DE above has exact solution $u(x) = e^x\, sin(2k\pi x)$. We will evaluate the results for $k = 1, 5, 50$, to consider equations more computationally challenging.

III.B. High-Order Polynomial

The second example is a boundary value problem containing a high-order polynomial:

$$\begin{cases} u''(x) = f(x) \qquad \text{where} \qquad x \in [0,1] \\ u(0) = 0 \\ u(1) = 0 \end{cases} \tag{20}$$

The forcing term is $f(x) = 2^{2p} p(1-x)^{p-2} x^{p-2} \left(-1 + 2x - 2x^2 + p(1 - 4x + 4x^2)\right), p \in \mathcal{N}$. The problem above has exact solution $u(x) = 2^{2p}\, x^p (1-x)^p$. We will analyze the results for: $p = 10, 100, 250$ to make it more computational challenging,while in Ref.[2] they only observed the result for $p = 10$ computed via ELM network to compare with a non-exact solution obtained with FDM.

III.C. Diffusion - Reaction Problem

The fourth problem is derived from Eq.(18), considering $\gamma = 0$, $\lambda > 0$ and $f(x) \equiv 0$ and imposing as Dirichlet boundary conditions $g_0 = 0$, $g_1 = 1$, we obtain a diffusion - reaction problem:

$$\begin{cases} -\mu u''(x) + \lambda u(x) = 0 \qquad\qquad x \in [0,1] \\ g_0 = 0 \\ g_1 = 1 \end{cases} \tag{21}$$

The analytical solution is $u(x) = \dfrac{sinh\,(\vartheta x)}{sinh(\vartheta)} \qquad \text{where} \qquad \vartheta = \sqrt{\dfrac{\lambda}{\mu}}$. We notice that also in this problem, the solution can present steep gradients if $\frac{\lambda}{\mu} \gg 1$.

III.D. Internal Layer Profile

The fifth problem is one of the high transient problems that lead to an internal layer:

$$\begin{cases} -u''(x) = \dfrac{2\alpha^3(x-x_0)}{(1+\alpha^2(x-x_0)^2)^2} & x \in [0,1] \\ u(0) = -\frac{3}{2} \\ u(1) = \frac{3}{2} \end{cases} \tag{22}$$

The exact solution of the problem is $u(x) = atan\left(\alpha(x-x_0)\right)$. In the test we run, we choose $\alpha = 1, 60, 1000$ and $x_0 = \dfrac{4}{9}$. Even if it is an internal layer problem, the kind of differential equation is the same as Eq.(20).

III.E. Internal Peak Profile

The sixth problem is an internal peak test, which is a rescaled sinusoidal problem:

$$\begin{cases} -u''(x) = \left(\dfrac{-4x^2}{\varepsilon^2} + \dfrac{2}{\varepsilon}\right) e^{-\dfrac{x^2}{\varepsilon}} & x \in [0,1] \\ u(0) = 0 \\ u(1) = 1 \end{cases} \tag{23}$$

The exact solution of the problem is $u(x) = e^{-\dfrac{x^2}{\varepsilon}}$. In the code we run, we choose three different values of $\varepsilon : 1, 10^{-3}, 10^{-6}$. The kind of differential equation is the same as Eq.(20).

III.F. Comb-like Profile

The last example is a more complex case, which has as exact solution a comb-like profile:

$$\begin{cases} -u''(x) = -2(\varepsilon+x)cos\left(\dfrac{1}{\varepsilon+x}\right) + \dfrac{sin\left(\dfrac{1}{\varepsilon+x}\right)}{(\varepsilon+x)^4} & x \in [0,1] \\ u(0) = sin\left(\dfrac{1}{\varepsilon}\right) \\ u(1) = sin\left(\dfrac{1}{\varepsilon+1}\right) \end{cases} \tag{24}$$

The analytical solution of the problem is $u(x) = sin\left(\dfrac{1}{\varepsilon+x}\right)$. We consider two cases: $\varepsilon = \dfrac{1}{\pi}, \dfrac{1}{10\pi}$, to have solutions with different oscillations in the domain. The kind of differential equation is the same as Eq.(20).

IV. RESULTS AND DISCUSSION

In this section, the hyperparameters for each case – unless otherwise specified – are reported in Table I.

TABLE I
Simulation Parameters

	PINN	**PIELM**	**Deep-TFC**	**X-TFC**
Activation function	tanh	tanh	tanh	tanh
Optimizer	ADAM	Least-Squares	L-BFGS	Least-Squares
Learning - rate	Adaptive	–	Adaptive	–
$N_{\mathcal{R}}$	10000	10000	10000	10000
N	2	2	0	0
Epochs	Problem Dependent	1	Problem Dependent	1
Mini - batch size	Problem Dependent	10000	10000	10000
Number of batch	Problem Dependent	1	1	1

The following Tables show the results of the problems for each case scenarios in terms of maximum error, mean error, Euclidean norm, standard deviation, training time and absolute errors on the boundary conditions. The absolute error is computed by comparing the numerical solution with the analytical solution on the test points.

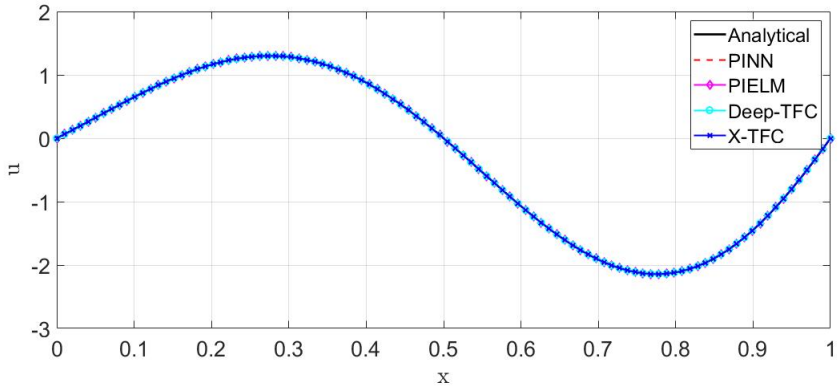
In Tables II - XXXV, the superscripts and subscripts reported next to each method's acronym have the following meanings:

- PINN $^{\text{epochs, number of batches}}_{\text{number of layers, number of neurons per layer}}$
- PIELM $_{\text{number of neurons}}$
- Deep-TFC $^{\text{epochs}}_{\text{number of layers, number of neurons per layer}}$
- X-TFC $_{\text{number of neurons}}$

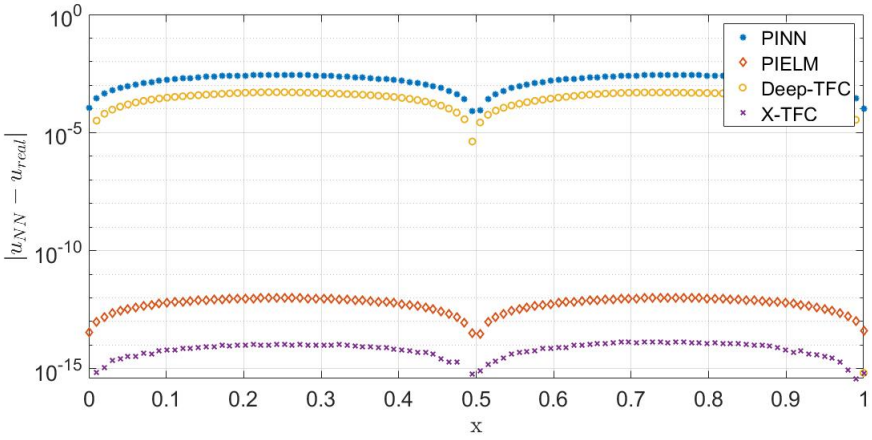
which characterize the architecture of the Neural Network at hand.

All the simulation run on a machine with an Intel Core i7 - 9700 CPU PC with 64 GB of RAM. For some cases, we presented also the plots of the solutions calculated with each methods and the relative absolute errors. The same graphs were presented during ICTT-27.

Sinusoidal-bump Problem - Case k = 1. From the results shown in Table II, we notice that shallow NNs obtain the solution with excellent accuracy and low computational cost. Generally,



(a)

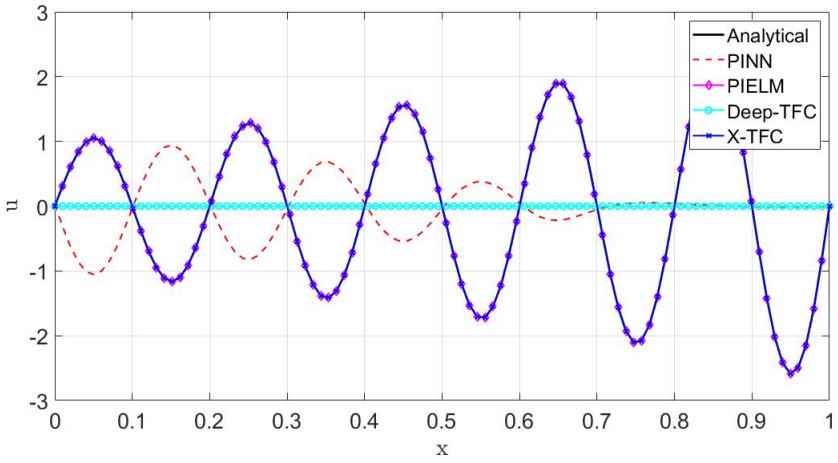


(b)

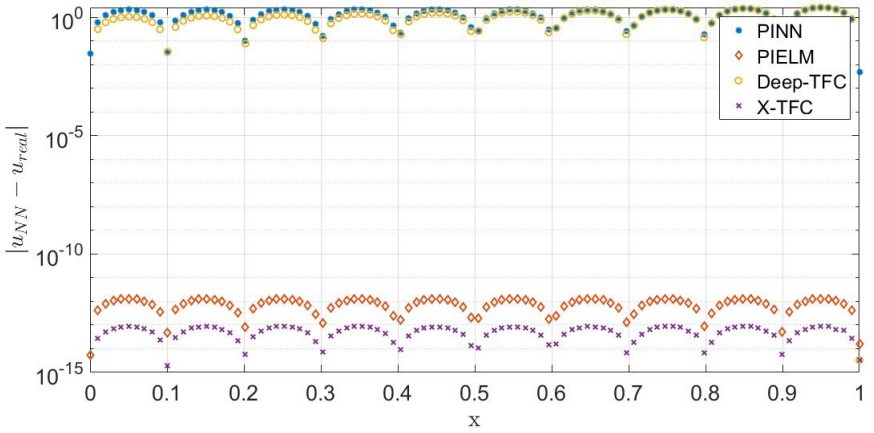
Fig. 1. Solution plots (a) and absolute errors (b) on test points for $k = 1$.

X-TFC and PIELM architectures reach accuracy of 10^{-15} and 10^{-13} respectively, both with ≈ 2.0 s of training time. On the contrary, both Deep NNs fail to learn the solution in almost every scenario. PINN can only achieve the solution in the architecture that presents 5 layers and 50 neurons per layer with at least 50000 epochs and 8 hours of training. Deep-TFC is only successful in the architecture with one layer and 50 neurons per layer with computational time of ≈ 53.5 minutes. In Fig.1, we can graphically appreciate the differences in terms of absolute errors.

Sinusoidal-bump Problem - Case k = 5. From the results show in Table IV, we notice that deep NNs totally fail to learn the solution: there is no architecture capable of solving the problem. On the other hand, shallow NNs obtain high performances such as 10^{-14} for X-TFC and 10^{-13}



(a)



(b)

Fig. 2. Solution plots (a) and absolute errors (b) on test points for $k = 5$.

for PIELM, with a computational cost of $\approx$ 2.0 s each. In particular, X-TFC architecture with 1000 neurons gains the best results within 2.0 seconds of training. In Fig.2, we can graphically appreciate the differences in terms of absolute errors. In Fig.2, we can graphically appreciate the differences in terms of absolute errors.

Sinusoidal-bump Problem - Case k = 50. From the results show in Table VI, it can be seen how deep NNs totally fail in learning how to solve the equation. The number of oscillations increases significantly and also shallow NNs are less accurate than previous scenarios. In particular, X-TFC finds the solution in both architectures with 1000 and 10000 neurons with a mean error

of 10^{-07} – $\approx$ 2 seconds and 10^{-13} – $\approx$ 2.5 minutes, respectively. On the other hand, PIELM only achieves the solution in the order of 10^{05} with the 10000 neurons with a computational time of $\approx$ 2.5 minutes. The best absolute errors on the boundary conditions for PINN and PIELM are in the order of 10^{-03} and 10^{-08}, respectively,as it can be seen from Table VII. While, for the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied.

High-Order Polynomial - Case p = 10. From the results show in Tables VIII, we notice how shallow NNs can learn the solution with excellent accuracy and low computational cost. In particular, X-TFC and PIELM most performing architectures reach both accuracy of 10^{-16} with $\approx$ 2.0 seconds of training time. On the other hand, both Deep NNs need higher computational time to reach worse results, such as 10^{-08} for Deep-TFC, and 10^{-03} for PINN, with computational times of $\approx$ 14 hours 45 minutes and $\approx$ 30 minutes, respectively. The best absolute errors on the boundary conditions for PINN and PIELM are in the order of 10^{-03} and 10^{-16}, respectively, as it can be seen from Table IX. While, for the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied.

High-Order Polynomial - Case p = 100. Table X shows how shallow NNs can learn the solution with excellent accuracy and low computational cost. In particular, X-TFC and PIELM architectures reach accuracy of 10^{-15}, both with $\approx$ 2.0 seconds of computational time. On the contrary, Deep-TFC achieve significantly lower accuracy, such as 10^{-06} with training times of $\approx$ 3 hours. We notice that none of the PINN architectures can compute the solution. As can be seem from Table XI The absolute errors on the boundary conditions for and PIELM are in the order of 10^{-16}. For the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied.

High-Order Polynomial - Case p = 250 Table XII shows how shallow NNs can learn the solution with excellent accuracy and low computational cost. In particular, X-TFC and PIELM architectures both reach accuracy of 10^{-15}, within $\approx$ 2.0 seconds of computational time. On the contrary, Deep-TFC achieve significantly lower accuracy, such as 10^{-05} with training times of $\approx$ 18 minutes. We notice that none of the PINN architectures can compute the solution. As can be

seem from Table XIII The absolute errors on the boundary conditions for and PIELM are in the order of 10^{-17}. For the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied.

Diffusion-Reaction problem - Case $\lambda = 1$. From the results show in Tables XIV, it can be seen how shallow NNs can learn the solution with excellent accuracy and low computational cost. In particular, X-TFC and PIELM architectures reach accuracy of 10^{-17} and 10^{-15}, respectively, both with ≈ 2.0 seconds of computational time. Conversely, both Deep NNs achieve significantly lower accuracy, such as 10^{-06} for Deep-TFC, and 10^{-07} for PINN, with a computational time of ≈ 11.7 seconds and ≈ 20 days respectively. The best absolute errors on the boundary conditions for PINN and PIELM are in the order of 10^{-05} and 10^{-16}, respectively,as it can be seen from Table XV. While, for the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied.

Diffusion-Reaction problem - Case $\lambda = 300$. Table XVI shows how shallow NNs can learn the solution with excellent accuracy and low computational cost. In particular, X-TFC and PIELM architectures reach accuracy of 10^{-15} and 10^{-14}, respectively, with ≈ 2.5 and ≈ 2.7 minutesof computational time. On the contrary, we note that Deep NNs learn the solution with lower accuracy, such as 10^{-03} with a training time of ≈ 11.4 seconds for DEEP-TFC and ≈ 30 minutes for PINN. In particular, they can not follow the sharp gradient on the final boundary. The best absolute errors on the boundary conditions for PINN and PIELM are in the order of 10^{-04} and 10^{-15}, respectively,as it can be seen from Table XVII. While, for the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied.

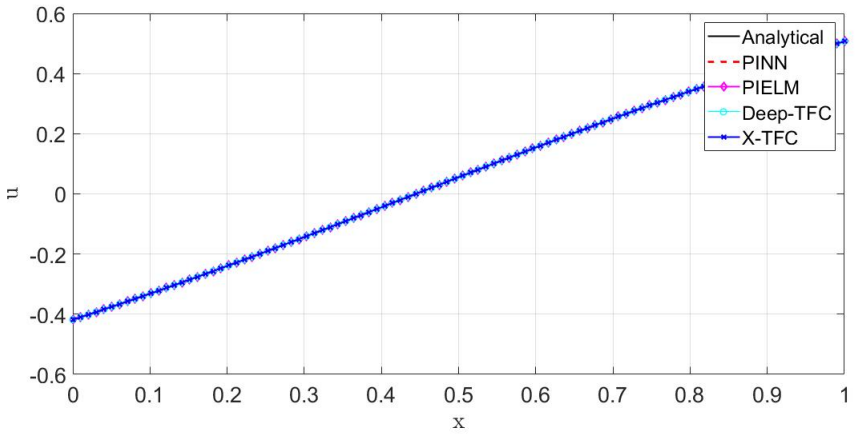
Diffusion-Reaction problem - Case $\lambda = 50000$. From the results show in Tables XVIII, it can be seen how shallow NNs can learn the solution with good accuracy and low computational cost. In particular, X-TFC and PIELM architectures reach accuracy of 10^{-14} and 10^{-09}, with computational time of ≈ 2.4 and ≈ 2.3 minutes, respectively. We note that Deep NNs learn the solution with lower accuracy, such as 10^{-02} with a training time of ≈ 13.5 seconds for DEEP-TFC and ≈ 30 minutes for PINN. In particular, they can not follow the sharp gradient on the final boundary. The best absolute errors on the boundary conditions for PIELM are in the order of

10^{-15},as it can be seen from Table XIX. While, for the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied.

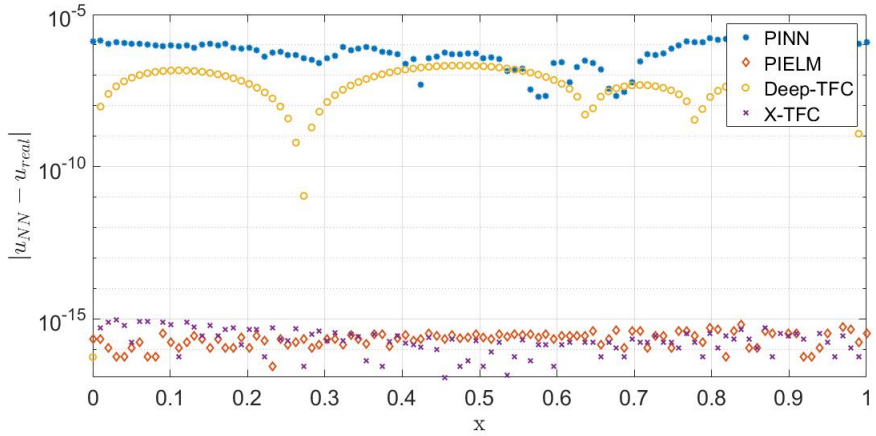
Internal Layer Profile - Case $\alpha = 1$. From the results show in Tables XX, it can be seen how shallow NNs can learn the solution with excellent accuracy and low computational cost. In particular, X-TFC and PIELM architectures reach accuracy of 10^{-17} and 10^{-16}, with computational time of ≈ 2.0 and ≈ 0.02 seconds, respectively. We note that both Deep NNs achieve significantly lower accuracy, such as 10^{-08} with a training time of ≈ 35.2 seconds for Deep-TFC and 294 hours for PINN. The best absolute errors on the boundary conditions for PIELM and PINN are in the order of 10^{-16} and 10^{-08} respectively,as it can be seen from Table XXI. While, for the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied. In Fig.3, we can graphically appreciate the differences in terms of absolute errors.

Internal Layer Profile - Case $\alpha = 60$. Table XXII shows how shallow NNs can learn the solution with excellent accuracy and low computational cost in almost every scenario. In particular, X-TFC and PIELM architectures can reach accuracies of 10^{-14} and 10^{-13} respectively, in ≈ 2.0 seconds On the contrary, Deep-TFC achieves significantly lower accuracy, such as 10^{-03}, with training times of ≈ 2.8 minutes. We notice that every PINN architecture fail to learn the solution. As can be seem from Table XXIII The absolute errors on the boundary conditions for and PIELM are in the order of 10^{-13}. For the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied. In Fig.4, we can graphically appreciate the differences in terms of absolute errors.

Internal Layer Profile - Case $\alpha = 1000$. In the most demanding case of this problem, we notice that also shallow NNs accuracies are lower than usual, but still with the lowest computationl costs. Table XXIV, shows that both X-TFC and PIELM architectures reach accuracy of 10^{-04} with ≈ 2.0 minutes of training time. On the contrary, both Deep NNs fail to learn the solution in every scenario. The best absolute errors on the boundary conditions for PIELM are in the order of 10^{-05}, as it can be seen from Table XXV. While, for the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied. In Fig.5, we can graphically appreciate the differences in terms of absolute errors.

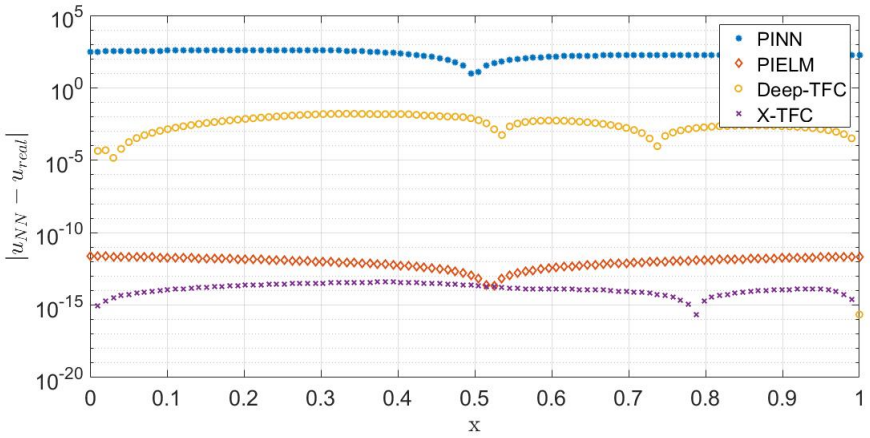


(a)

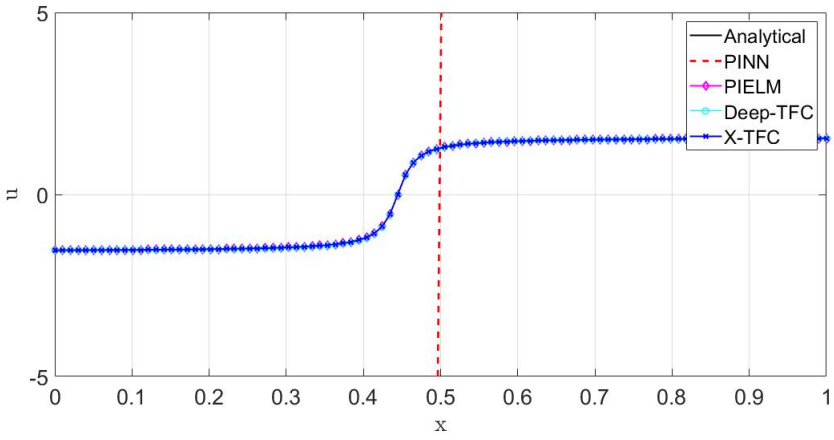


(b)

Fig. 3. Solution plots (a) and absolute errors (b) on test points for $\alpha = 1$.

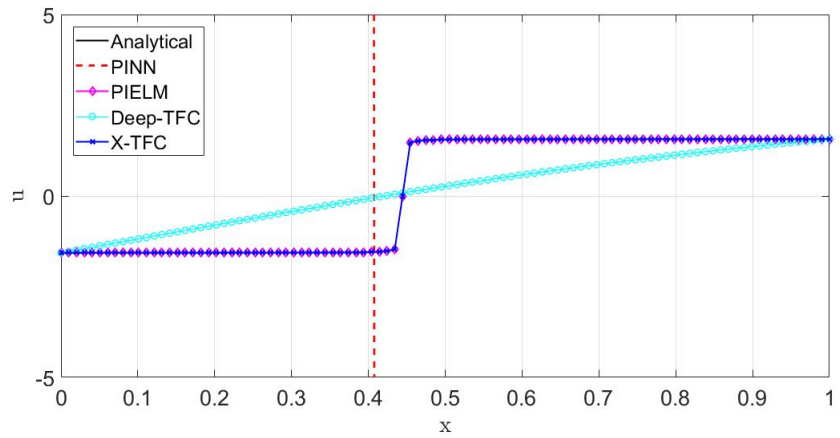


(a)

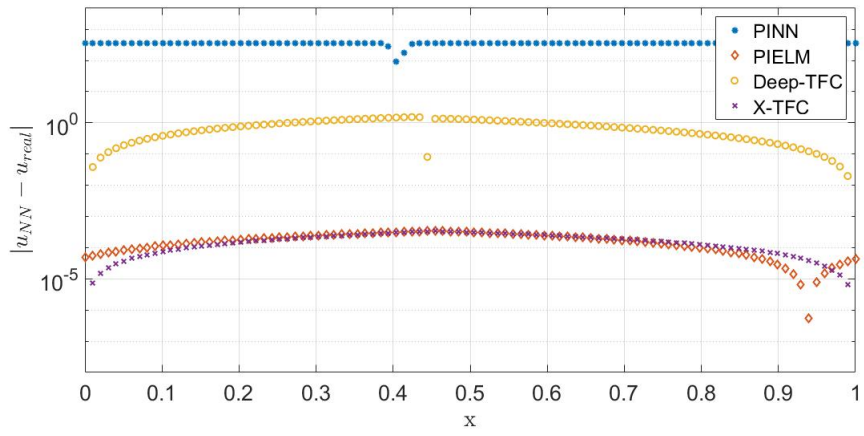


(b)

Fig. 4. Solution plots (a) and absolute errors (b) on test points for $\alpha = 60$.



(a)



(b)

Fig. 5. Solution plots (a) and absolute errors (b) on test points for $\alpha = 1000$.

Internal Peak Profile - Case $\varepsilon = 1$. From the results show in Table XXVI, it can be seen how shallow NNs can learn the solution with excellent accuracy and low computational cost. In particular, X-TFC and PIELM architectures reach accuracy of 10^{-16} and 10^{-15}, respectively, both with ≈ 0.02 seconds of computational time. Conversely, both Deep NNs achieve significantly lower accuracy, such as 10^{-06} for Deep-TFC, and 10^{-05} for PINN. We note that one of the two most performing PINN architectures is a one-hidden layer NNs. The computational time is ≈ 4.9 seconds for Deep-TFC and ≈ 26 minutes for PINN. The best absolute errors on the boundary conditions for PINN and PIELM are in the order of 10^{-06} and 10^{-15}, respectively,as it can be seen from Table XXVII. While, for the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied.

Internal Peak Profile - Case $\varepsilon = 10^{-03}$. From the results show in Table XXVIII, it can be seen how shallow NNs can learn the solution with excellent accuracy and low computational cost. In particular, X-TFC and PIELM architectures reach accuracy of 10^{-14} and 10^{-13}, respectively, both with ≈ 2.6 minutes of computational time. Conversely, both Deep NNs achieve significantly lower accuracy, such as 10^{-03} for Deep-TFC, and 10^{-05} for PINN with a training time of ≈ 1.2 minutes and ≈ 2.5 hours, respecitively. The best absolute errors on the boundary conditions for PINN and PIELM are in the order of 10^{-04} and 10^{-13}, respectively, as it can be seen from Table XXIX. While, for the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied.

Internal Peak Profile - Case $\varepsilon = 10^{-06}$. Table XXX shows how shallow NNs can learn the solution with gpod accuracy and low computational cost. In particular, X-TFC and PIELM architectures both reach accuracy of 10^{-12}, within ≈ 2.6 minutes of training. On the contrary, PINN reaches the solution only with two configurations and achieves significantly lower accuracy, such as 10^{-03} with computational time of ≈ 2.5 hours. We notice that Deep-TFC always fails to learn the solution. As can be seem from Table XXXI The absolute errors on the boundary conditions for and PIELM and PIELM are in the order of 10^{-12} and 10^{-02} respectively. For the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied.

Comb-like Profile - Case $\varepsilon = \frac{1}{\pi}$. Table XXXII shows how shallow NNs can learn the solution with excellent accuracy and low computational cost. In particular, X-TFC and PIELM architectures reach accuracy of 10^{-15} and 10^{-14}, respectively, both with ≈ 2.0 seconds of computational time. On the other hand both Deep NNs achieve significantly lower accuracy, such as 10^{-04} for Deep-TFC, and 10^{-05} for PINN, with computational times of ≈ 5.0 seconds and ≈ 2.0 hours, respectively. The best absolute errors on the boundary conditions for PINN and PIELM are in the order of 10^{-05} and 10^{-14}, respectively,as it can be seen from Table XXXIII. While, for the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied.

Comb-like Profile - Case $\varepsilon = \frac{1}{10\pi}$. In the most challenging case of this problem, Table XXXIV shows how shallow NNs can learn the solution with good accuracy and low computational cost in almost every scenario. In particular, X-TFC architectures reach accuracy of 10^{-13} in ≈ 2.6 minutes, while in the same training time PIELM can reach 10^{-12}. On the contrary, Deep fails to learn the solution with every architecture trained. PINN can achieve the solution only in one configuration, with accuracy of 10^{-02} and training times of ≈ 88.0 hours. As can be seem from Table XXXV The absolute errors on the boundary conditions for and PIELM and PINN are in the order of 10^{-12} and 10^{-02}, respectively. For the TFC-based frameworks, the absolute error is exactly 0, as the constraints are analytically satisfied.

V. CONCLUSION

This paper is a comparison of four different PINNs frameworks for solving physics problems arising from the second-order linear Diffusion-Advection-Reaction equation. The motivation behind the choice of these physics problems is the existence of sharp gradients in the solution, which makes it computationally challenging. We solved six different problems, varying the DEs' parameters in each of them, to create discontinuities in the solutions and increase the computational challenge. For each method, several architectures have been used to find the optimal set of hyperparameters and to get the best performances. All the problems we faced led us to draw the same qualitative conclusions.

The classic PINN framework is able to find the solutions for the simplest cases with tolerable

accuracy values but less acceptable computational costs (times never less than 20 minutes), while it fails for slightly more difficult cases. Deep-TFC proved to be slightly better in accuracy than the classic PINN by learning the solution for the cases where classic PINN starts to fail but fails in turn for subsequent cases. This small improvement is due to the introduction of the TFC constrained expression in the approximation of the solution since it solves the problem on the constraints analytically. PIELM framework was able to solve all the most challenging problems with steeper discontinuities, with great accuracy and highly performing computational times. This substantial difference from the previous two methods is due to the employment of Shallow NNs as an approximation of the solution instead of multiple hidden layers of neurons. Finally, by using the TFC constrained expression and a Shallow NN as a free function, the accuracy of the solutions of all the problems have been improved to the machine error accuracy, with computational times never longer than 3 minutes. This is the case of X-TFC, which is the most performing framework within this simulations.

In conclusion, we proved how Shallow NNs are more advised than Deep NNs in the machine learning community for solving 1D steady-state differential equations. Also, the biggest limitation of the physics-driven frameworks, i.e. the non-analytic satisfaction of the constraints, can be easily overcome by the use of constrained expression introduced by the TFC. Future works will focus on solving non-linear ODEs, and linear and non-linear PDEs to investigate the different performance of the frameworks and apply them to solve problems in several fields of physics and engineering, such as Radiative Transfer, Rarefied Gas Dynamics, Nuclear Reactor Dynamics, Chemical Kinetics, and many more.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] 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 (2019).

[2] F. Calabrò, G. Fabiani, and C. Siettos, "Extreme learning machine collocation for the numerical solution of elliptic PDEs with sharp gradients," Computer Methods in Applied Mechanics and Engineering, **387**, 114188 (2021).

[3] G. E. Karniadakis, I. G. Kevrekidis, L. Lu, P. Perdikaris, S. Wang, and L. Yang, "Physics-informed machine learning," Nature Reviews Physics, **3**, 6, 422 (2021).

[4] V. Dwivedi and B. Srinivasan, "Physics Informed Extreme Learning Machine (PIELM)–A rapid method for the numerical solution of partial differential equations," Neurocomputing, **391**, 96 (2020).

[5] C. Leake and D. Mortari, "Deep theory of functional connections: A new method for estimating the solutions of partial differential equations," Machine learning and knowledge extraction, **2**, 1, 37 (2020).

[6] E. Schiassi, R. Furfaro, C. Leake, M. De Florio, H. Johnston, and D. Mortari, "Extreme theory of functional connections: A fast physics-informed neural network method for solving ordinary and partial differential equations," Neurocomputing, **457**, 334 (2021).

[7] D. Mortari, "The theory of connections: connecting points," Mathematics, **5**, 4, 57 (2017).

[8] M. De Florio, E. Schiassi, R. Furfaro, B. D. Ganapol, and D. Mostacci, "Solutions of Chandrasekhar's basic problem in radiative transfer via theory of functional connections," Journal of Quantitative Spectroscopy and Radiative Transfer, **259**, 107384 (2021).

[9] M. De Florio, E. Schiassi, B. D. Ganapol, and R. Furfaro, "Physics-informed neural networks for rarefied-gas dynamics: Thermal creep flow in the Bhatnagar–Gross–Krook approximation," Physics of Fluids, **33**, 4, 047110 (2021).

[10] M. De Florio, E. Schiassi, B. D. Ganapol, and R. Furfaro, "Physics-Informed Neural Networks for rarefied-gas dynamics: Poiseuille flow in the BGK approximation," Zeitschrift für angewandte Mathematik und Physik, **73**, 3, 1 (2022).

[11] M. De Florio, E. Schiassi, A. D'Ambrosio, D. Mortari, and R. Furfaro, "Theory of Functional Connections Applied to Linear ODEs Subject to Integral Constraints and Linear Ordinary Integro-Differential Equations," Mathematical and Computational Applications, **26**, 3, 65 (2021).

[12] M. De Florio, E. Schiassi, and R. Furfaro, "Physics-informed neural networks and functional interpolation for stiff chemical kinetics," Chaos: An Interdisciplinary Journal of Nonlinear Science, **32**, 6, 063107 (2022).

[13] E. Schiassi, A. D'Ambrosio, A. Scorsoglio, R. Furfaro, and F. Curti, "CLASS OF OPTIMAL SPACE GUIDANCE PROBLEMS SOLVED VIA INDIRECT METHODS AND PHYSICS-INFORMED NEURAL NETWORKS," preprint (2021).

[14] E. Schiassi, M. De Florio, B. D. Ganapol, P. Picca, and R. Furfaro, "Physics-informed neural networks for the point kinetics equations for nuclear reactor dynamics," Annals of Nuclear Energy, **167**, 108833 (2022).

[15] S. Dong and Z. Li, "Local extreme learning machines and domain decomposition for solving linear and nonlinear partial differential equations," Computer Methods in Applied Mechanics and Engineering, **387**, 114129 (2021).

[16] S. Dong and J. Yang, "On computing the hyperparameter of extreme learning machines: Algorithm and application to computational PDEs, and comparison with classical and high-order finite elements," Journal of Computational Physics, 111290 (2022).

[17] D. Mortari, "Least-squares solution of linear differential equations," Mathematics, **5**, 4, 48 (2017).

[18] D. Mortari, H. Johnston, and L. Smith, "High accuracy least-squares solutions of nonlinear differential equations," Journal of Computational and Applied Mathematics, **352**, 293 (2019).

TABLE II
Performance summary in terms of absolute errors for the case $k = 1$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{150000,5}_{1,50}$	2.81	1.79	0.20	0.86	$\approx$ 51 h
$\mathbf{PINN}^{15000,5}_{1,100}$	2.83	1.80	0.20	0.87	$\approx$ 59 h
$\mathbf{PINN}^{15000,5}_{5,50}$	1.39	0.89	0.10	0.43	$\approx$ 3.2 h
$\mathbf{PINN}^{100000,5}_{5,50}$	1.32e-01	8.39e-02	9.30e-03	4.04e-02	$\approx$ 15.7 h
$\mathbf{PINN}^{500000,5}_{5,50}$	2.72e-03	1.74e-03	1.93e-04	8.38e-04	$\approx$ 42.3 h
$\mathbf{PINN}^{15000,5}_{5,100}$	1.65	1.05	0.117	0.508	$\approx$ 2.3 h
$\mathbf{PINN}^{15000,5}_{10,50}$	1.86	1.19	0.132	0.572	$\approx$ 3 h
$\mathbf{PINN}^{15000,5}_{10,100}$	2.23	1.42	0.158	0.687	$\approx$ 3.5 h
$\mathbf{PIELM}_{100}$	9.45e-12	5.97e-12	6.62e-13	2.87e-12	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	9.89e-13	6.30e-13	6.99e-14	3.04e-13	$\approx$ 2.0 s
$\mathbf{PIELM}^{*}_{10000}$	2.76e-10	1.76e-10	1.96e-11	8.50e-11	$\approx$ 2.6 min
$\mathbf{Deep\text{-}TFC}^{3000}_{1,50}$	3.95e-09	1.56e-09	1.89e-10	1.06e-09	$\approx$ 53.5 min
$\mathbf{Deep\text{-}TFC}^{3000}_{1,100}$	3.62	1.52	0.20	1.31	$\approx$ 5.9 min
$\mathbf{Deep\text{-}TFC}^{3000}_{5,50}$	3.62	1.52	0.20	1.31	$\approx$ 16.7 min
$\mathbf{Deep\text{-}TFC}^{3000}_{5,100}$	3.62	1.52	0.20	1.31	$\approx$ 51.4 min
$\mathbf{Deep\text{-}TFC}^{3000}_{10,50}$	3.62	1.52	0.20	1.31	$\approx$ 49.8 min
$\mathbf{Deep\text{-}TFC}^{3000}_{10,100}$	3.62	1.52	0.20	1.31	$\approx$ 1.3 h
$\mathbf{X\text{-}TFC}_{100}$	2.96e-12	8.22e-14	1.03e-14	6.32e-14	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	1.38e-14	7.48e-15	8.47e-16	8.47e-16	$\approx$ 2.0 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	1.47e-13	7.83e-14	9.14e-15	4.74e-14	$\approx$ 2.6 min

TABLE III
Absolute errors on the boundary conditions for the case $k = 1$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{150000,5}_{1,50}$	1.11e-01	1.11e-01
$\mathbf{PINN}^{15000,5}_{1,100}$	1.11e-01	1.20e-01
$\mathbf{PINN}^{15000,5}_{5,50}$	5.54e-02	5.61e-02
$\mathbf{PINN}^{50000,5}_{5,50}$	1.40e-02	1.42e-02
$\mathbf{PINN}^{100000,5}_{5,50}$	5.23e-03	2.24e-03
$\mathbf{PINN}^{500000,5}_{5,50}$	1.12e-04	1.03e-04
$\mathbf{PINN}^{15000,5}_{5,100}$	6.62e-02	6.62e-02
$\mathbf{PINN}^{15000,5}_{10,50}$	7.43e-02	7.43e-02
$\mathbf{PINN}^{3000,5}_{10,100}$	8.86e-02	8.95e-02
$\mathbf{PIELM}_{100}$	3.76e-13	4.24e-13
$\mathbf{PIELM}_{1000}$	3.61e-14	4.00e-14
$\mathbf{PIELM}_{10000}$	9.76e-12	1.23e-11
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE IV
Performance summary in terms of absolute errors for the case $k = 5$.

max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**	
$\mathbf{PINN}^{15000,5}_{1,50}$	2.60	1.09	0.12	0.64	$\approx$ 1.8 h min
$\mathbf{PINN}^{3000,5}_{1,1000}$	2.60	1.08	0.12	0.64	$\approx$ 23 min
$\mathbf{PINN}^{3000,5}_{1,10000}$	2.60	1.08	0.13	0.64	$\approx$ 31 min
$\mathbf{PINN}^{3000,5}_{5,50}$	2.60	1.08	0.13	0.64	$\approx$ 28 min
$\mathbf{PINN}^{3000,5}_{5,100}$	2.60	1.26	0.14	0.65	$\approx$ 25 min
$\mathbf{PINN}^{3000,5}_{10,50}$	2.57	1.38	0.154	0.696	$\approx$ 29 min
$\mathbf{PINN}^{3000,5}_{10,100}$	1.83	1.15	0.13	0.57	$\approx$ 36 min
$\mathbf{PIELM}^{*}_{100}$	1.78e-05	1.13e-05	1.26e-06	5.60e-06	$\approx$ 0.02 s
$\mathbf{PIELM}^{*}_{1000}$	1.29e-12	8.14e-13	9.08e-14	4.04e-13	$\approx$ 2.0 s
$\mathbf{PIELM}^{*}_{10000}$	7.43e-08	4.69e-08	5.23e-09	2.33e-08	$\approx$ 2.7 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	2.42	1.12	0.12	0.66	$\approx$ 25.8 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	1.95	1.12	0.12	0.56	$\approx$ 19.5 min
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	1.73	1.06	0.12	0.55	$\approx$ 2.7 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	1.73	1.06	0.12	0.55	$\approx$ 4.8 s
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	1.73	1.09	0.12	0.54	$\approx$ 17.1 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	1.71	1.09	0.12	0.54	$\approx$ 55 min
$\mathbf{X\text{-}TFC}^{*}_{100}$	1.35e-10	4.33e-11	5.48e-12	3.39e-11	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}^{*}_{1000}$	8.93e-14	5.51e-14	6.15e-15	2.74e-14	$\approx$ 2.0 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	2.50e-12	1.57e-12	1.75e-13	7.82e-13	$\approx$ 2.7 min

TABLE V
Absolute errors on the boundary conditions for the case $k = 5$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{15000,5}_{1,50}$	2.60e-02	3.70e-02
$\mathbf{PINN}^{3000,5}_{1,1000}$	2.80e-02	2.37e-01
$\mathbf{PINN}^{3000,5}_{1,10000}$	2.30e-02	2.11e-02
$\mathbf{PINN}^{3000,5}_{5,50}$	2.96e-02	4.90e-02
$\mathbf{PINN}^{3000,5}_{5,100}$	3.77e-02	3.13e-02
$\mathbf{PINN}^{3000,5}_{10,50}$	1.51e-03	3.41e-04
$\mathbf{PINN}^{3000,5}_{10,100}$	1.89e-01	2.25e-01
$\mathbf{PIELM}^{*}_{100}$	1.39e-07	1.45e-07
$\mathbf{PIELM}^{*}_{1000}$	5.55e-15	1.63e-14
$\mathbf{PIELM}^{*}_{10000}$	5.98e-10	5.85e-10
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE VI
Performance summary in terms of absolute errors for the case $k = 50$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	1.73	1.06	0.119	0.554	$\approx$ 23 min
$\mathbf{PINN}^{3000,5}_{1,100}$	1.73	1.06	0.119	0.554	$\approx$ 23 min
$\mathbf{PINN}^{3000,5}_{5,50}$	1.73	1.06	0.119	0.554	$\approx$ 28 min
$\mathbf{PINN}^{3000,5}_{5,100}$	1.73	1.06	0.119	0.554	$\approx$ 32 min
$\mathbf{PINN}^{3000,5}_{10,50}$	1.73	1.06	0.119	0.554	$\approx$ 37 min
$\mathbf{PINN}^{3000,5}_{10,100}$	1.73	1.06	0.119	0.554	$\approx$ 38 min
$\mathbf{PINN}^{50000,5}_{1,1000}$	1.73	1.06	0.119	0.554	$\approx$ 6.5 h
$\mathbf{PIELM}^{*}_{100}$	1.73	1.06	1.20e-01	5.55e-01	$\approx$ 0.02 s
$\mathbf{PIELM}^{*}_{1000}$	1.06	6.70e-01	7.48e-02	3.33e-01	$\approx$ 2.0 s
$\mathbf{PIELM}^{*}_{10000}$	5.16e-05	3.28e-05	8.15e-07	1.59e-05	$\approx$ 2.5 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	1.73	1.06	0.12	0.55	$\approx$ 1.31 s
$\mathbf{Deep\text{-}TFC}^{6e04}_{1,100}$	1.73	1.06	0.12	0.55	$\approx$ 2.26 s
$\mathbf{Deep\text{-}TFC}^{6e04}_{5,50}$	1.73	1.06	0.12	0.55	$\approx$ 1.85 s
$\mathbf{Deep\text{-}TFC}^{6e04}_{5,100}$	1.73	1.06	0.12	0.55	$\approx$ 4.67 s
$\mathbf{Deep\text{-}TFC}^{6e04}_{10,50}$	1.73	1.06	0.12	0.55	$\approx$ 4.04 s
$\mathbf{Deep\text{-}TFC}^{6e04}_{10,100}$	1.73	1.06	0.12	0.55	$\approx$ 5.76 s
$\mathbf{X\text{-}TFC}^{*}_{100}$	1.73	1.06	1.20e-01	5.55e-01	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}^{*}_{1000}$	9.97e-07	6.28e-07	7.02e-08	3.14e-07	$\approx$ 2.0 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	4.22e-13	2.46e-13	6.11e-15	1.19e-13	$\approx$ 2.5 min

TABLE VII
Absolute errors on the boundary conditions for the case $k = 50$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	1.90e-03	1.20e-03
$\mathbf{PINN}^{3000,5}_{1,100}$	2.50e-02	1.81e-01
$\mathbf{PINN}^{3000,5}_{5,50}$	3.02e-03	4.00e-04
$\mathbf{PINN}^{3000,5}_{5,100}$	2.74e-03	1.59e-03
$\mathbf{PINN}^{3000,5}_{10,50}$	1.50e-03	3.41e-04
$\mathbf{PINN}^{3000,5}_{10,100}$	2.04e-03	4.28e-03
$\mathbf{PINN}^{50000,5}_{1,1000}$	1.14e-04	2.44e-04
$\mathbf{PIELM}^{*}_{100}$	6.78e-04	8.46e-04
$\mathbf{PIELM}^{*}_{1000}$	7.57e-04	9.34e-04
$\mathbf{PIELM}^{*}_{10000}$	4.11e-08	4.10e-08
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE VIII
Performance summary in terms of absolute errors for the case $p = 10$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	1.20e-02	6.11e-03	7.06e-04	3.56e-03	$\approx$ 25 min
$\mathbf{PINN}^{3000,5}_{1,100}$	1.06e-02	5.42e-03	6.28e-04	3.19e-03	$\approx$ 25 min
$\mathbf{PINN}^{3000,5}_{5,50}$	6.49e-03	3.56e-03	3.76e-04	1.20e-03	$\approx$ 31 min
$\mathbf{PINN}^{3000,5}_{5,100}$	7.94e-03	3.98e-03	4.61e-04	2.33e-03	$\approx$ 32 min
$\mathbf{PINN}^{3000,5}_{10,50}$	7.18e-03	3.66e-03	4.20e-04	2.07e-03	$\approx$ 40 min
$\mathbf{PINN}^{3000,5}_{10,100}$	2.85e-02	1.29e-02	1.56e-03	8.70e-03	$\approx$ 40 min
$\mathbf{PIELM}_{100}$	1.77e-13	3.14e-14	5.11e-15	4.054e-14	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	2.22e-15	6.69e-16	8.10e-17	4.58e-16	$\approx$ 2.0 s
$\mathbf{PIELM}^{*}_{10000}$	2.09e-13	1.04e-13	1.20 e-14	6.00e-14	$\approx$ 2.6 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	4.30e-05	1.10e-05	9.09e-06	1.42e-06	$\approx$ 2 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	1.06-04	2.10e-05	2.41e-05	3.18e-06	$\approx$ 3 min
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	1.68e-06	4.66e-07	3.49e-07	5.81e-08	$\approx$ 27 min
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	9.36e-07	3.03e-07	2.25e-07	3.77e-08	$\approx$ 2.8 h
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	1.08e-06	6.26e-07	3.10e-07	6.98e-08	$\approx$ 3.7 h
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	1.67e-07	5.80e-08	4.30e-08	7.21e-09	$\approx$ 14.8 h
$\mathbf{X\text{-}TFC}_{100}$	1.81e-13	3.02e-14	5.05e-15	4.07e-14	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	1.11e-15	1.55e-16	2.70e-17	2.22e-16	$\approx$ 2.0 s
$\mathbf{X\text{-}TFC}_{10000}$	4.80e-14	2.92e-14	3.28e-15	1.51e-14	$\approx$ 2.6 min

TABLE IX
Absolute errors on the boundary conditions for the case $p = 10$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	1.20e-02	6.11e-03
$\mathbf{PINN}^{3000,5}_{1,100}$	1.06e-02	1.06e-02
$\mathbf{PINN}^{3000,5}_{5,50}$	3.17e-03	3.49e-03
$\mathbf{PINN}^{3000,5}_{5,100}$	7.93e-03	7.94e-03
$\mathbf{PINN}^{3000,5}_{10,50}$	7.10e-03	7.18e-03
$\mathbf{PINN}^{3000,5}_{10,100}$	2.68e-02	2.85e-02
$\mathbf{PIELM}_{100}$	1.04e-09	9.48e-10
$\mathbf{PIELM}_{1000}$	5.41e-16	7.22e-16
$\mathbf{PIELM}_{10000}$	2.09e-13	2.00e-13
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE X
Performance summary in terms of absolute errors for the case $p = 100$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	NaN	NaN	NaN	NaN	$\approx$ 25 min
$\mathbf{PINN}^{3000,5}_{1,100}$	NaN	NaN	NaN	NaN	$\approx$ 25 min
$\mathbf{PINN}^{3000,5}_{5,50}$	NaN	NaN	NaN	NaN	$\approx$ 31 min
$\mathbf{PINN}^{3000,5}_{5,100}$	NaN	NaN	NaN	NaN	$\approx$ 32 min
$\mathbf{PINN}^{3000,5}_{10,50}$	NaN	NaN	NaN	NaN	$\approx$ 40 min
$\mathbf{PINN}^{3000,5}_{10,100}$	NaN	NaN	NaN	NaN	$\approx$ 40 min
$\mathbf{PIELM}_{100}$	1.23e-09	5.51e-10	6.39e-11	3.25e-11	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	1.13e-14	1.59e-15	2.14e-16	1.43e-15	$\approx$ 2 s
$\mathbf{PIELM}^{*}_{10000}$	6.82e-14	3.13e-14	3.72e-15	2.01e-14	$\approx$ 2.6 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	6.44e-04	1.98e-04	1.45e-04	2.45e-05	$\approx$ 6.5 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	6.62e-04	1.90e-04	1.83e-04	2.63e-05	$\approx$ 4.7 min
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	9.98e-05	1.97e-05	2.67e-05	3.30e-06	$\approx$ 46 min
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	1.12e-05	3.60e-06	2.38e-06	4.31e-07	$\approx$ 3 h
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	1.58e-04	1.00e-04	4.87e-05	1.11e-05	$\approx$ 2.7 h
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	1.15e-05	5.40e-06	2.72e-06	6.04e-07	$\approx$ 11 h
$\mathbf{X\text{-}TFC}_{100}$	7.38e-10	1.81e-10	2.37e-11	1.53e-10	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	1.18e-14	1.56e-15	2.26e-16	1.65e-15	$\approx$ 2 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	1.15e-14	4.01e-15	5.41e-16	3.66e-15	$\approx$ 2.6 min

TABLE XI
Absolute errors on the boundary conditions for the case $p = 100$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	NaN	NaN
$\mathbf{PINN}^{3000,5}_{1,100}$	NaN	NaN
$\mathbf{PINN}^{3000,5}_{5,50}$	NaN	NaN
$\mathbf{PINN}^{3000,5}_{5,100}$	NaN	NaN
$\mathbf{PINN}^{3000,5}_{10,50}$	NaN	NaN
$\mathbf{PINN}^{3000,5}_{10,100}$	NaN	NaN
$\mathbf{PIELM}_{100}$	1.04e-09	9.48e-10
$\mathbf{PIELM}_{1000}$	1.05e-15	1.03e-15
$\mathbf{PIELM}^{*}_{10000}$	6.82e-14	6.54e-15
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE XII
Performance summary in terms of absolute errors for the case $p = 250$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	NaN	NaN	NaN	NaN	$\approx$ 25 min
$\mathbf{PINN}^{3000,5}_{1,100}$	NaN	NaN	NaN	NaN	$\approx$ 25 min
$\mathbf{PINN}^{3000,5}_{5,50}$	NaN	NaN	NaN	NaN	$\approx$ 31 min
$\mathbf{PINN}^{3000,5}_{5,100}$	NaN	NaN	NaN	NaN	$\approx$ 32 min
$\mathbf{PINN}^{3000,5}_{10,50}$	NaN	NaN	NaN	NaN	$\approx$ 40 min
$\mathbf{PINN}^{3000,5}_{10,100}$	NaN	NaN	NaN	NaN	$\approx$ 40 min
$\mathbf{PIELM}_{100}$	8.77e-06	2.81e-06	3.46e-07	2.03e-06	0.02 s
$\mathbf{PIELM}_{1000}$	2.71e-14	7.47e-15	8.97e-16	4.99e-15	$\approx$ 2 s
$\mathbf{PIELM}^{*}_{10000}$	4.27e-14	2.10e-14	2.41e-15	1.19e-14	$\approx$ 2.7 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	4.82e-02	1.05e-02	1.20e-02	1.59e-03	$\approx$ 7 sec
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	2.28e-04	6.96e-05	6.06e-05	9.21e-06	$\approx$ 18 min
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	4.70e-03	3.24e-03	1.41e-03	3.53e-04	$\approx$ 41 min
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	5.97e-04	4.16e-04	1.40e-04	4.39e-05	$\approx$ 1.3 h
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	8.86e-04	6.30e-04	2.84e-04	6.91e-05	$\approx$ 2.2 h
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	7.35e-04	4.80e-04	2.24-04	5.30e-05	$\approx$ 5.7 h
$\mathbf{X\text{-}TFC}_{100}$	9.17e-06	1.99e-06	2.94e-07	2.17e-06	0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	2.69e-14	7.61e-15	8.96e-16	4.74e-15	$\approx$ 2 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	2.03e-14	3.57e-15	4.80e-16	3.07e-15	$\approx$ 2.6 min

TABLE XIII
Absolute errors on the boundary conditions for the case $p = 250$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	NaN	NaN
$\mathbf{PINN}^{3000,5}_{1,100}$	NaN	NaN
$\mathbf{PINN}^{3000,5}_{5,50}$	NaN	NaN
$\mathbf{PINN}^{3000,5}_{5,100}$	NaN	NaN
$\mathbf{PINN}^{3000,5}_{10,50}$	NaN	NaN
$\mathbf{PINN}^{3000,5}_{10,100}$	NaN	NaN
$\mathbf{PIELM}_{100}$	3.83e-06	2.84e-06
$\mathbf{PIELM}_{1000}$	2.78e-17	4.99e-16
$\mathbf{PIELM}^{*}_{10000}$	4.27e-14	4.07e-14
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE XIV
Performance summary in terms of absolute errors for the case $\lambda = 1$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	5.28e-06	2.36e-06	2.79e-07	1.49e-06	$\approx$ 25 min
$\mathbf{PINN}^{3000,5}_{1,100}$	3.78e-04	2.52e-04	2.70e-05	9.77e-05	$\approx$ 22.5 min
$\mathbf{PINN}^{3000,5}_{5,50}$	3.21e-04	1.30e-04	1.61e-05	9.53e-05	$\approx$ 27 min
$\mathbf{PINN}^{3000,5}_{5,100}$	4.34e-04	1.89e-04	2.30e-05	1.31e-04	$\approx$ 27.5 min
$\mathbf{PINN}^{3000,5}_{10,50}$	7.47e-03	3.31e-03	3.91e-04	2.09e-03	$\approx$ 37 min
$\mathbf{PINN}^{3000,5}_{10,100}$	7.07e-01	2.76e-01	3.36e-02	1.94e-01	$\approx$ 34 min
$\mathbf{PINN}^{1e5,5}_{10,100}$	2.82e-03	1.23e-03	1.49e-04	8.43e-04	$\approx$ 43 h
$\mathbf{PINN}^{1e6,5}_{10,100}$	8.95e-07	1.94e-07	2.56e-08	1.68e-07	$\approx$ 20 days
$\mathbf{PIELM}_{100}$	1.23e-14	3.47e-15	4.34e-16	2.62e-15	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	1.08e-14	7.26e-15	7.64e-16	2.40e-15	$\approx$ 2.0 s
$\mathbf{PIELM}^{*}_{10000}$	1.52e-12	7.31e-13	8.49e-14	4.33e-13	$\approx$ 2.6 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	1.99e-05	1.04e-05	5.99e-06	1.19e-06	$\approx$ 6.5 s
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	7.41e-06	3.69e-06	2.20e-06	4.29e-07	$\approx$ 11.7 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	4.15e-05	1.69e-05	1.23e-05	2.09e-06	$\approx$ 42 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	7.15e-05	3.06e-05	2.36e-05	3.86e-06	$\approx$ 1.6 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	2.19e-03	1.09e-03	7.94e-04	1.35e-04	$\approx$ 1.5 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	9.09e-05	4.13e-05	3.02e-05	5.1e-06	$\approx$ 3.5 min
$\mathbf{X\text{-}TFC}_{100}$	4.22e-15	9.34e-16	1.37e-16	1.00e-15	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	2.22e-16	3.52e-17	6.63e-18	5.65e-17	$\approx$ 2.0 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	1.89e-15	1.12e-15	1.22e-16	4.97e-16	$\approx$ 2.5 min

TABLE XV
Absolute errors on the boundary conditions for the case $\lambda = 1$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	0.50	0.50
$\mathbf{PINN}^{3000,5}_{1,100}$	0.50	0.50
$\mathbf{PINN}^{3000,5}_{5,50}$	0.50	0.50
$\mathbf{PINN}^{3000,5}_{5,100}$	0.50	0.50
$\mathbf{PINN}^{3000,5}_{10,50}$	0.50	0.50
$\mathbf{PINN}^{3000,5}_{10,100}$	0.50	0.50
$\mathbf{PIELM}_{100}$	0.50	0.50
$\mathbf{PIELM}_{1000}$	0.50	0.50
$\mathbf{PIELM}^{*}_{10000}$	8.33e-08	8.33e-08
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE XVI
Absolute errors on the boundary conditions for the case $\lambda = 300$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	1.00	6.23e-01	1.84e-02	1.74e-01	$\approx$ 22 min
$\mathbf{PINN}^{3000,5}_{1,100}$	1.00	6.23e-01	1.84e-02	1.74e-01	$\approx$ 22 min
$\mathbf{PINN}^{3000,5}_{5,50}$	1.00	6.93e-01	1.85e-02	1.73e-01	$\approx$ 28 min
$\mathbf{PINN}^{3000,5}_{5,100}$	1.00	6.93e-01	1.85e-02	1.73e-01	$\approx$ 43 min
$\mathbf{PINN}^{3000,5}_{10,50}$	1.00	6.23e-01	1.84e-02	1.74e-01	$\approx$ 38 min
$\mathbf{PINN}^{3000,5}_{10,100}$	1.00	6.23e-01	1.84e-02	1.74e-01	$\approx$ 22 min
$\mathbf{PIELM}_{100}$	3.39e-09	4.01e-09	8.36e-11	7.37e-10	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	1.06e-12	7.29e-14	1.96e-14	1.83e-13	$\approx$ 1.8 s
$\mathbf{PIELM}^{*}_{10000}$	4.57e-13	2.85e-14	8.34e-15	7.87e-14	$\approx$ 2.7 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	8.68e-03	3.55e-03	2.42e-03	4.29e-04	$\approx$ 6.3 s
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	1.81e-02	6.25e-03	4.7 e-03	7.79e-04	$\approx$ 11.4 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	6.73e-03	2.57e-03	1.65e-03	3.05e-04	$\approx$ 49 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	3.56e-02	1.25e-02	8.57e-03	1.51e-03	$\approx$ 1.5 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	3.56e-02	1.25e-02	8.57e-03	1.51e-03	$\approx$ 1.5 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	4.70e-03	1.94e-03	1.33e-03	2.34e-04	$\approx$ 3.8 min
$\mathbf{X\text{-}TFC}_{100}$	2.92e-11	5.69e-12	9.04e-13	7.05e-12	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	8.77e-15	3.06e-15	3.78e-16	2.23e-15	$\approx$ 1.9 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	2.59e-14	3.05e-15	6.90e-16	6.22e-14	$\approx$ 2.5 min

TABLE XVII
Absolute errors on the boundary conditions for the case $\lambda = 300$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	1.44e-04	1.00
$\mathbf{PINN}^{3000,5}_{1,100}$	1.35e-03	0.99
$\mathbf{PINN}^{3000,5}_{5,50}$	1.07e-01	1.00
$\mathbf{PINN}^{3000,5}_{5,100}$	1.07e-01	1.00
$\mathbf{PINN}^{3000,5}_{10,50}$	0.0012	0.999
$\mathbf{PINN}^{3000,5}_{10,100}$	0.0012	0.999
$\mathbf{PIELM}_{100}$	3.29e-09	3.13e-09
$\mathbf{PIELM}_{1000}$	1.06e-12	1.11e-13
$\mathbf{PIELM}^{*}_{10000}$	3.26e-15	4.57e-15
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE XVIII
Performance summary in terms of absolute errors for the case $\lambda = 500000$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	1.00	1.01e-02	1e-03	1e-01	$\approx$ 23 min
$\mathbf{PINN}^{3000,5}_{1,100}$	1.00	1.02e-02	1e-03	1e-01	$\approx$ 23 min
$\mathbf{PINN}^{3000,5}_{5,50}$	1.00	1.02e-02	1e-02	1e-01	$\approx$ 28 min
$\mathbf{PINN}^{3000,5}_{5,100}$	1.00	1.02e-02	1e-02	1e-01	$\approx$ 33 min
$\mathbf{PINN}^{3000,5}_{10,50}$	1.00	1.02e-02	1e-02	1e-01	$\approx$ 36 min
$\mathbf{PINN}^{3000,5}_{10,100}$	1.00	1.02e-02	1e-02	1e-01	$\approx$ 37 min
$\mathbf{PIELM}_{100}$	1.00	1.00 e-01	1.00e-02	1.00e-02	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	0.99	1.00e-02	9.99e-03	9.99e-02	$\approx$ 2.0 s
$\mathbf{PIELM}^{*}_{10000}$	3.20e-07	3.54e-09	3.22e-19	3.21e-08	$\approx$ 2.3 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	9.89e-01	4.90e-01	2.93e-01	5.70e-02	$\approx$ 0.36 s
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	8.56e-01	9.55e-02	1.35e-01	1.65e-02	$\approx$ 13.5 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	9.89e-01	4.90e-01	2.93e-01	5.70e-02	$\approx$ 2.4 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	8.35e-01	8.36e-02	1.24e-01	1.49e-02	$\approx$ 2.1 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	9.11e-01	1.10e-01	1.72e-01	2.03e-04	$\approx$ 1.9 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	9.89e-01	4.90e-01	2.93e-01	5.70e-02	$\approx$ 47 s
$\mathbf{X\text{-}TFC}_{100}$	4.51e-02	6.03e-03	1.00e-03	8.04e-3	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	1.46e-06	5.74e-08	2.14e-08	2.07e-07	$\approx$ 2.0 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	1.43e-13	3.75e-14	4.56e-15	2.57e-14	$\approx$ 2.4 min

TABLE XIX
Absolute errors on the boundary conditions for the case $\lambda = 500000$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	1.27e-04	1.00
$\mathbf{PINN}^{3000,5}_{1,100}$	7.30e-04	1.00
$\mathbf{PINN}^{3000,5}_{5,50}$	6.60e-04	1.00
$\mathbf{PINN}^{3000,5}_{5,100}$	5.40e-04	1.00
$\mathbf{PINN}^{3000,5}_{10,50}$	3.70e-04	1.00
$\mathbf{PINN}^{3000,5}_{10,100}$	1.16e-02	1.00
$\mathbf{PIELM}_{100}$	1.29e14	1
$\mathbf{PIELM}_{1000}$	5.88e-07	0.99
$\mathbf{PIELM}^{*}_{10000}$	3.34e-08	3.20e-07
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE XX
Performance summary in terms of absolute errors for the case $\alpha = 1$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	1.64e-05	6.81e-06	8.69e-07	5.42e-06	$\approx$ 20.3 min
$\mathbf{PINN}^{3000,5}_{1,100}$	2.58e-05	1.54e-05	1.62e-06	5.31e-06	$\approx$ 20.2 min
$\mathbf{PINN}^{3000,5}_{5,50}$	9.96e-05	5.30e-05	2.19e-06	3.20e-05	$\approx$ 28.2 min
$\mathbf{PINN}^{1e6,5}_{5,50}$	3.32e-07	8.21e-08	1.05e-06	6.54e-08	$\approx$ 294 h
$\mathbf{PINN}^{3000,5}_{5,100}$	6.31e-04	3.03e-04	3.50e-05	1.78e-04	$\approx$ 28.2 min
$\mathbf{PINN}^{3000,5}_{10,50}$	2.76e-03	1.40e-03	1.62e-04	8.18e-04	$\approx$ 36.2 min
$\mathbf{PINN}^{3000,5}_{10,100}$	8.75e-04	4.21e-04	4.87e-05	2.46e-04	$\approx$ 36.0 min
$\mathbf{PIELM}_{100}$	6.66e-16	2.32e-16	2.62e-17	1.22e-16	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	2.61e-15	1.11e-15	1.30e-16	3.79e-16	$\approx$ 2.0 s
$\mathbf{PIELM}^{*}_{10000}$	9.46e-13	4.67e-13	5.39e-14	2.72e-13	$\approx$ 2.7 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	5.10e-06	2.43e-06	2.93e-07	1.64e-06	$\approx$ 4.8 s
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	5.25e-07	2.26e-07	2.78-08	1.63e-07	$\approx$ 8.2 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	2.10e-07	7.69e-08	9.88e-09	6.24e-08	$\approx$ 35.2 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	2.64e-06	1.03e-06	1.31e-07	8.18e-07	$\approx$ 1.3 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	6.97e-08	3.34e-08	4.01e-09	2.22e-08	$\approx$ 1.3 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	1.24e-07	4.20e-08	5.62e-09	3.75e-08	$\approx$ 2.8 min
$\mathbf{X\text{-}TFC}_{100}$	9.44e-16	2.44e-16	3.24e-17	2.14e-16	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	5.55e-17	2.20e-17	3.10e-18	2.20e-17	$\approx$ 2.0 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	6.11e-16	1.18e-16	2.11e-17	1.75e-16	$\approx$ 2.6 min

TABLE XXI
Absolute errors on the boundary conditions for the case $\alpha = 1$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	9.86e-06	1.04e-05
$\mathbf{PINN}^{3000,5}_{1,100}$	6.46e-06	2.22e-05
$\mathbf{PINN}^{3000,5}_{5,50}$	9.54e-05	9.96e-05
$\mathbf{PINN}^{1e6,5}_{5,50}$	5.45e-08	1.70e-07
$\mathbf{PINN}^{3000,5}_{5,100}$	4.00e-04	6.31e-04
$\mathbf{PINN}^{3000,5}_{10,50}$	2.76e-03	2.73e-03
$\mathbf{PINN}^{3000,5}_{10,100}$	8.75e-04	8.42e-04
$\mathbf{PIELM}_{100}$	2.22e-16	3.33e-16
$\mathbf{PIELM}_{1000}$	2.30e-15	1.11e-16
$\mathbf{PIELM}^{*}_{10000}$	9.46e-13	9.04e-13
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE XXII
Performance summary in terms of absolute errors for the case $\alpha = 60$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	411.5	240.8	26.7	115.4	$\approx$ 21 min
$\mathbf{PINN}^{3000,5}_{1,100}$	475.1	351.2	37.7	119.6	$\approx$ 22 min
$\mathbf{PINN}^{3000,5}_{5,50}$	300.4	235.6	24.5	68.6	$\approx$ 25 min
$\mathbf{PINN}^{3000,5}_{5,100}$	315.4	153.6	18.55	104.6	$\approx$ 28 min
$\mathbf{PINN}^{3000,5}_{10,50}$	192.6	112.7	12.7	20.2	$\approx$ 36 min
$\mathbf{PINN}^{3000,5}_{10,100}$	59.00	12.2	106.8	198.7	$\approx$ 37 min
$\mathbf{PIELM}_{100}$	9.23e-03	1.22e-03	2.18e-04	1.82e-03	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	1.56e-13	1.21e-13	1.24e-14	2.77e-14	$\approx$ 2.0 s
$\mathbf{PIELM}^{*}_{10000}$	2.34e-12	1.16e-12	1.34e-13	6.71e-13	$\approx$ 2.6 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	1.40	4.43e-01	5.41e-02	3.78e-01	$\approx$ 0.48 s
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	1.13	6.30e-01	7.22e-02	3.55e-01	$\approx$ 0.44 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	2.17e-01	1.27e-01	1.43e-02	6.79e-02	$\approx$ 39.1 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	4.59e-01	3.05e-01	3.34e-02	1.35e-01	$\approx$ 1.5 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	1.08	7.07e-01	7.82e-02	3.37e-01	$\approx$ 1.5 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	1.63e-02	5.45e-03	7.58e-04	5.29e-03	$\approx$ 2.8 min
$\mathbf{X\text{-}TFC}_{100}$	9.11e-03	1.11e-02	2.16e-04	1.86e-03	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	1.02e-13	4.38e-14	5.14e-15	2.69e-14	$\approx$ 2.0 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	3.91e-14	1.65e-14	2.00e-15	1.14e-14	$\approx$ 2.6 min

TABLE XXIII
Absolute errors on the boundary conditions for the case $\alpha = 60$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	311.6	187.2
$\mathbf{PINN}^{3000,5}_{1,100}$	469.7	398.1
$\mathbf{PINN}^{3000,5}_{5,50}$	196.1	242.7
$\mathbf{PINN}^{3000,5}_{5,100}$	85.1	283.2
$\mathbf{PINN}^{3000,5}_{10,50}$	20.2	152.0
$\mathbf{PINN}^{3000,5}_{10,100}$	63.3	161.2
$\mathbf{PIELM}_{100}$	4.82e-04	4.89e-04
$\mathbf{PIELM}_{1000}$	1.28e-13	1.27e-13
$\mathbf{PIELM}^{*}_{10000}$	2.34e-12	2.24e-12
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE XXIV
Performance summary in terms of absolute errors for the case $\alpha = 1000$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	363.7	124.5	1485.5	81.4	$\approx$ 23 min
$\mathbf{PINN}^{3000,5}_{1,100}$	759.2	462.2	4789.1	126.2	$\approx$ 23 min
$\mathbf{PINN}^{3000,5}_{5,50}$	286.3	282.2	2883.4	25.6	$\approx$ 29 min
$\mathbf{PINN}^{3000,5}_{5,100}$	423.6	400.8	4067.4	69.8	$\approx$ 29 min
$\mathbf{PINN}^{3000,5}_{10,50}$	358.3	352.8	35.4	33.2	$\approx$ 28 min
$\mathbf{PINN}^{3000,5}_{10,100}$	160.7	152.8	15.4	22.5	$\approx$ 36 min
$\mathbf{PIELM}_{100}$	2.15e-01	2.13e-02	3.96e-03	3.35e-02	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	1.10e-02	3.12e-03	3.58e-04	1.77e-03	$\approx$ 2.0 s
$\mathbf{PIELM}^{*}_{10000}$	3.45e-05	1.74e-04	2.0e-05	1.00e-04	$\approx$ 2.7 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	1.53	7.19e-01	8.50e-02	4.55e-01	$\approx$ 0.26 s
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	1.48	7.27e-01	8.55e-02	4.52e-01	$\approx$ 0.40 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	1.37	3.84e-01	5.18e-02	3.49e-01	$\approx$ 3.1 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	1.87	9.69e-01	1.11e-01	5.51e-01	$\approx$ 9.7 s
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	2.05	8.07e-01	1.02e-01	6.19e-01	$\approx$ 2.8 s
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	1.63	7.65e-01	8.94e-02	4.65e-01	$\approx$ 3.7 s
$\mathbf{X\text{-}TFC}_{100}$	2.16e-01	2.01e-02	3.94e-03	3.40e-02	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	1.22e-02	8.61e-04	1.84e-04	1.63e-03	$\approx$ 2.0 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	3.38e-04	1.68e-04	1.95e-05	9.83e-05	$\approx$ 2.6 min

TABLE XXV
Absolute errors on the boundary conditions for the case $\alpha = 1000$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	92.8	92.8
$\mathbf{PINN}^{3000,5}_{1,100}$	460.7	460.7
$\mathbf{PINN}^{3000,5}_{5,50}$	286.3	286.3
$\mathbf{PINN}^{3000,5}_{5,100}$	418.5	423.5
$\mathbf{PINN}^{3000,5}_{10,50}$	358.3	358.3
$\mathbf{PINN}^{3000,5}_{10,100}$	160.7	156.6
$\mathbf{PIELM}_{100}$	7.27e-03	6.63e-03
$\mathbf{PIELM}_{1000}$	4.40e-03	2.90e-03
$\mathbf{PIELM}^{*}_{10000}$	4.98e-05	4.39e-05
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE XXVI
Performance summary in terms of absolute errors for the case $\varepsilon = 1$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	5.09e-05	3.27e-05	3.59e-06	1.49e-05	$\approx$ 25 min
$\mathbf{PINN}^{3000,5}_{1,100}$	8.89e-05	5.92e-05	6.24e-06	2.00e-05	$\approx$ 26 min
$\mathbf{PINN}^{3000,5}_{5,50}$	7.30e-04	3.40e-04	4.04e-05	2.19e-04	$\approx$ 31 min
$\mathbf{PINN}^{3000,5}_{5,100}$	2.60e-04	1.26e-04	1.47e-05	7.52e-05	$\approx$ 32 min
$\mathbf{PINN}^{3000,5}_{10,50}$	1.50e-01	7.46e-02	8.6e-03	4.32e-02	$\approx$ 39 min
$\mathbf{PINN}^{3000,5}_{10,100}$	3.30e-03	1.66e-03	1.91e-04	9.52e-04	$\approx$ 40 min
$\mathbf{PIELM}_{100}$	5.11e-15	1.17e-15	1.63e-16	1.14e-15	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	7.77e-15	1.72e-15	2.60-16	1.96e-15	$\approx$ 2.1 s
$\mathbf{PIELM}^{*}_{10000}$	1.03e-13	5.22e-14	5.00e-15	2.97e-14	$\approx$ 2.7 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	1.26e-05	7.62e-06	8.89e-07	4.62e-06	$\approx$ 4.9 s
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	9.60e-06	4.67e-06	5.62e-07	3.15e-06	$\approx$ 8.1 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	7.71e-06	3.82e-06	4.59e-07	2.56e-06	$\approx$ 33.6 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	3.76e-05	1.38e-05	1.73e-06	1.05e-05	$\approx$ 74.6 s
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	4.71e-06	1.88e-06	2.47e-07	1.61e-06	$\approx$ 2.6 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	3.00e-06	1.29e-06	1.55e-07	8.55e-07	$\approx$ 2.8 min
$\mathbf{X\text{-}TFC}_{100}$	3.33e-15	7.91e-16	1.13e-16	8.07e-16	$\approx$ 0.2 s
$\mathbf{X\text{-}TFC}_{1000}$	4.44e-16	2.19e-16	2.51e-17	1.23e-16	$\approx$ 1.9 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	3.55e-15	2.45e-15	2.66e-16	1.05e-15	$\approx$ 2.7 min

TABLE XXVII
Absolute errors on the boundary conditions for the case $\varepsilon = 1$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	1.73e-06	2.10e-05
$\mathbf{PINN}^{3000,5}_{1,100}$	1.54e-05	8.89e-05
$\mathbf{PINN}^{3000,5}_{5,50}$	4.24e-04	3.59e-04
$\mathbf{PINN}^{3000,5}_{5,100}$	2.60e-04	2.52e-04
$\mathbf{PINN}^{3000,5}_{10,50}$	1.5e-01	1.5e-01
$\mathbf{PINN}^{3000,5}_{10,100}$	3.30e-03	3.30e-04
$\mathbf{PIELM}_{100}$	2.66e-15	1.55e-15
$\mathbf{PIELM}_{1000}$	6.66e-15	2.78e-16
$\mathbf{PIELM}^{*}_{10000}$	1.023e-13	1.00e-13
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE XXVIII
Performance summary in terms of absolute errors for the case $\varepsilon = 10^{-03}$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	2.37	2.01	0.20	0.19	$\approx$ 26 min
$\mathbf{PINN}^{3000,5}_{1,100}$	0.14	5.13e-02	6.26e-03	3.61e-02	$\approx$ 26 min
$\mathbf{PINN}^{3000,5}_{1,1000}$	3.15e-02	1.21e-02	1.31e-03	5.15e-03	$\approx$ 30 min
$\mathbf{PINN}^{10000,5}_{1,5000}$	1.67e-02	4.15e-03	5.23e-04	3.20e-03	$\approx$ 2.1 h
$\mathbf{PINN}^{3000,5}_{5,50}$	1.57e-02	7.81e-03	9.06e-04	4.62e-03	$\approx$ 30 min
$\mathbf{PINN}^{15000,5}_{5,50}$	3.33e-04	1.76e-04	1.97e-05	8.97e-05	$\approx$ 2.7 h
$\mathbf{PINN}^{15000,5}_{5,100}$	2.05e-04	7.23e-05	8.89e-06	5.20e-05	$\approx$ 2.5 h
$\mathbf{PINN}^{15000,5}_{10,50}$	5.77e-03	2.84e-04	3.29e-04	1.67e-03	$\approx$ 3.3 h
$\mathbf{PINN}^{15000,5}_{10,100}$	2.98e-03	1.54e-03	1.76e-04	8.60e-04	$\approx$ 3 h
$\mathbf{PIELM}_{100}$	8.32e-06	1.26e-06	2.30e-07	1.94e-06	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	4.95e-13	3.81e-13	3.86e-14	6.33e-14	$\approx$ 2.0 s
$\mathbf{PIELM}^{*}_{10000}$	2.07e-13	1.02e-13	1.18e-14	6.00e-14	$\approx$ 2.6 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	1.42e-01	4.99e-02	7.17e-03	5.18e-02	$\approx$ 5.2 s
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	3.80e-03	1.30e-03	1.83e-04	1.29e-03	$\approx$ 8.5 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	7.34e-03	3.70e-03	4.21e-04	2.01e-03	$\approx$ 38.3 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	2.44e-02	1.22e-02	1.46e-03	8.21e-03	$\approx$ 1.3 s
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	2.04e-02	9.44e-03	1.22e-03	7.70e-03	$\approx$ 1.2 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	2.16e-02	1.34e-02	1.47e-03	6.07e-03	$\approx$ 2.7 min
$\mathbf{X\text{-}TFC}_{100}$	8.21e-06	1.25e-06	2.31e-07	1.95e-07	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	7.22e-14	2.19e-14	2.80e-15	1.76e-14	$\approx$ 2.0 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	1.77e-14	1.29e-14	1.39e-15	5.38e-15	$\approx$ 2.6 min

TABLE XXIX
Absolute errors on the boundary conditions for the case $\varepsilon = 10^{-03}$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	1.71	2.37
$\mathbf{PINN}^{3000,5}_{1,100}$	5.97e-02	1.38e-01
$\mathbf{PINN}^{3000,5}_{5,50}$	1.57e-02	1.57e-02
$\mathbf{PINN}^{15000,5}_{5,50}$	1.18e-04	3.33e-04
$\mathbf{PINN}^{15000,5}_{5,100}$	1.68e-04	2.05e-04
$\mathbf{PINN}^{15000,5}_{10,50}$	5.65e-03	5.77e-03
$\mathbf{PINN}^{15000,5}_{10,100}$	2.98e-03	2.96e-03
$\mathbf{PIELM}_{100}$	1.36e-07	1.48e-07
$\mathbf{PIELM}_{1000}$	4.09e-13	2.48e-13
$\mathbf{PIELM}^{*}_{10000}$	2.07e-13	2.00e-13
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE XXX
Performance summary in terms of absolute errors for the case $\varepsilon = 10^{-06}$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{15000,5}_{1,50}$	135.78	56.00	6.84	40.91	$\approx$ 2 h
$\mathbf{PINN}^{15000,5}_{1,100}$	212.8	133.9	14.0	41.5	$\approx$ 1.9 h
$\mathbf{PINN}^{15000,5}_{5,50}$	2.04e-02	1.99e-02	2.00e-03	2.27e-03	$\approx$ 2.3 h
$\mathbf{PINN}^{15000,5}_{5,100}$	1.53e-02	9.82e-03	1.01e-03	2.74e-03	$\approx$ 2.5 h
$\mathbf{PINN}^{15000,5}_{10,50}$	1.83	1.81	0.18	0.0035	$\approx$ 3 h
$\mathbf{PINN}^{15000,5}_{10,100}$	3.84	2.13	0.25	0.73	$\approx$ 3 h
$\mathbf{PIELM}_{100}$	5.80e-01	7.06e-02	1.30e-02	1.10e-01	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	1.49e-01	9.69e-02	9.96e-03	2.32e-02	$\approx$ 2.1 s
$\mathbf{PIELM}^{*}_{10000}$	1.13e-11	9.39e-12	9.62e-13	3.22e-12	$\approx$ 2.6 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	9.90e-01	4.91e-01	5.71e-02	2.93e-01	$\approx$ 0.26 s
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	9.90e-01	4.95e-01	5.74e-02	2.93e-01	$\approx$ 0.32 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	9.90e-01	4.90e-01	5.70e-01	2.93e-01	$\approx$ 1.0 s
$\mathbf{Deep\text{-}TF0C}^{60000}_{5,100}$	9.90e-01	4.90e-01	5.70e-01	2.93e-01	$\approx$ 1.6 s
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	9.90e-01	4.90e-01	5.70e-01	2.93e-01	$\approx$ 2.1 s
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	9.90e-01	4.90e-01	5.70e-01	2.93e-01	$\approx$ 3.7 s
$\mathbf{X\text{-}TFC}_{100}$	5.90e-01	6.86e-02	1.30e-02	1.11e-01	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	1.53e-02	4.11e-03	5.19e-04	3.18e-03	$\approx$ 2.0 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	1.15e-11	8.46e-12	9.04e-13	3.22e-12	$\approx$ 2.6 min

TABLE XXXI
Absolute errors on the boundary conditions for the case $\varepsilon = 10^{-06}$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{15000,5}_{1,50}$	212.8	71.7
$\mathbf{PINN}^{15000,5}_{1,100}$	218.7	52.5
$\mathbf{PINN}^{15000,5}_{5,50}$	2.03e-02	2.04e-02
$\mathbf{PINN}^{15000,5}_{5,100}$	1.00e-02	1.00e-02
$\mathbf{PINN}^{15000,5}_{10,50}$	1.83	1.81
$\mathbf{PINN}^{15000,5}_{10,100}$	3.84	1.46
$\mathbf{PIELM}_{100}$	1.27e-02	1.67e-02
$\mathbf{PIELM}_{1000}$	1.40e-01	5.79e-02
$\mathbf{PIELM}^{*}_{10000}$	9.82e-12	1.13e-11
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE XXXII
Performance summary in terms of absolute errors for the case $\varepsilon = \frac{1}{\pi}$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	7.16e-04	2.78e-04	3.48e-05	2.11e-04	$\approx$ 25 min
$\mathbf{PINN}^{15000,5}_{1,50}$	1.11e-04	6.47e-05	6.96e-06	2.58e-05	$\approx$ 2.0 h
$\mathbf{PINN}^{3000,5}_{1,100}$	1.16e-03	7.19e-04	7.68e-05	2.69e-04	$\approx$ 25.5 min
$\mathbf{PINN}^{3000,5}_{5,50}$	1.76e-04	8.23e-03	9.63e-06	5.04e-05	$\approx$ 32 min
$\mathbf{PINN}^{3000,5}_{5,100}$	1.14e-03	5.77e-04	6.68e-05	3.38e-04	$\approx$ 30 min
$\mathbf{PINN}^{3000,5}_{10,50}$	4.55e-02	2.27e-02	2.63e-03	1.33e-02	$\approx$ 40 min
$\mathbf{PINN}^{3000,5}_{10,100}$	1.53e-02	7.64e-03	8.81e-04	4.41e-03	$\approx$ 40 min
$\mathbf{PIELM}_{100}$	9.69e-11	1.63e-11	2.75e-12	2.23e-11	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	3.81e-14	2.85e-14	2.89e-15	4.63e-15	$\approx$ 2.0 s
$\mathbf{PIELM}^{*}_{10000}$	6.16e-13	3.03e-13	3.50e-14	1.77e-13	$\approx$ 2.6 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	2.52e-03	9.96e-04	1.25e-04	7.62e-04	$\approx$ 5.0 s
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	1.25e-03	4.35e-04	5.76e-05	3.80e-04	$\approx$ 8.5 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	1.367e-03	5.28e-04	7.26e-05	5.00-04	$\approx$ 34.5 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	8.31e-04	4.43e-04	5.09e-05	2.52e-04	$\approx$ 1.3 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	9.24e-03	3.77e-03	5.09e-04	3.43e-03	$\approx$ 1.3 min
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	6.12e-04	3.51e-04	3.89e-05	1.69e-04	$\approx$ 3.0 min
$\mathbf{X\text{-}TFC}_{100}$	9.00e-11	1.51e-11	2.64e-12	2.17e-11	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	3.89e-15	2.66e-15	2.79e-16	8.32e-16	$\approx$ 1.9 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	2.40e-14	1.73e-14	1.88e-15	7.41e-15	$\approx$ 2.6 min

TABLE XXXIII
Performance summary in terms of absolute errors for the case $\varepsilon = \frac{1}{\pi}$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	5.64e-04	1.90e-04
$\mathbf{PINN}^{15000,5}_{1,50}$	1.43e-05	7.99e-05
$\mathbf{PINN}^{3000,5}_{1,100}$	2.00e-04	8.14e-04
$\mathbf{PINN}^{3000,5}_{5,50}$	1.76e-04	1.46e-04
$\mathbf{PINN}^{3000,5}_{5,100}$	1.14e-03	1.14e-03
$\mathbf{PINN}^{3000,5}_{10,50}$	4.55e-02	4.54e-02
$\mathbf{PINN}^{3000,5}_{10,100}$	1.53e-02	1.52e-02
$\mathbf{PIELM}_{100}$	4.48e-12	4.32e-12
$\mathbf{PIELM}_{1000}$	3.21e-14	2.25e-14
$\mathbf{PIELM}^{*}_{10000}$	6.16e-13	5.89e-13
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0

TABLE XXXIV
Absolute errors on the boundary conditions for the case $\varepsilon = \frac{1}{10\pi}$.

	max(error)	**mean(error)**	**norm(error)**	**std(error)**	**training time**
$\mathbf{PINN}^{3000,5}_{1,50}$	221.4	64.75	8.00	47.4	$\approx$ 24.3 min
$\mathbf{PINN}^{3000,5}_{1,100}$	450.5	162.9	196.4	109.9	$\approx$ 24 min
$\mathbf{PINN}^{15000,5}_{5,50}$	2.14	1.67	0.177	0.60	$\approx$ 2.7 h
$\mathbf{PINN}^{15000,5}_{5,100}$	1.91	0.46	0.0747	0.592	$\approx$ 2.8 h
$\mathbf{PINN}^{500000,5}_{5,100}$	3.03-02	1.52e-02	1.75e-03	8.84e-03	$\approx$ 88.0 h
$\mathbf{PINN}^{15000,5}_{10,50}$	4.584	3.782	0.393	1.063	$\approx$ 3.2 h
$\mathbf{PINN}^{15000,5}_{10,100}$	3.314	2.066	0.229	1.004	$\approx$ 3.2 h
$\mathbf{PINN}^{200000,5}_{10,100}$	0.161	0.0684	0.00837	0.0485	$\approx$ 43.3 h
$\mathbf{PIELM}_{100}$	1.45	7.69e-02	2.07e-02	1.93e-01	$\approx$ 0.02 s
$\mathbf{PIELM}_{1000}$	1.32e-04	8.56e-05	8.79e-06	2.01e-05	$\approx$ 2.0 s
$\mathbf{PIELM}^{*}_{10000}$	3.18e-12	1.53e-12	1.77e-13	9.01e-13	$\approx$ 2.6 min
$\mathbf{Deep\text{-}TFC}^{60000}_{1,50}$	1.15	4.27e-01	5.10e-02	2.80e-01	$\approx$ 0.28 s
$\mathbf{Deep\text{-}TFC}^{60000}_{1,100}$	1.15	4.27e-01	5.10e-02	2.80e-01	$\approx$ 0.33 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,50}$	1.15	4.27e-01	5.10e-02	2.80e-01	$\approx$ 0.74 s
$\mathbf{Deep\text{-}TFC}^{60000}_{5,100}$	1.15	4.27e-01	5.10e-02	2.80e-01	$\approx$ 0.74 s
$\mathbf{Deep\text{-}TFC}^{60000}_{10,50}$	1.15	4.27e-01	5.10e-02	2.80e-01	$\approx$ 0.63 s
$\mathbf{Deep\text{-}TFC}^{60000}_{10,100}$	1.15	4.27e-01	5.10e-02	2.80e-01	$\approx$ 0.95 s
$\mathbf{X\text{-}TFC}_{100}$	1.44	7.47e-02	2.06e-02	1.93e-01	$\approx$ 0.02 s
$\mathbf{X\text{-}TFC}_{1000}$	4.31e-05	1.26e-05	1.39e-06	5.78e-06	$\approx$ 2.0 s
$\mathbf{X\text{-}TFC}^{*}_{10000}$	2.46e-13	2.02e-13	2.08e-14	4.70e-14	$\approx$ 2.6 min

TABLE XXXV
Absolute errors on the boundary conditions for the case $\varepsilon = \frac{1}{10\pi}$.

	$\lvert u(0) - u_{NN}(0)\rvert$	$\lvert u(1) - u_{NN}(1)\rvert$
$\mathbf{PINN}^{3000,5}_{1,50}$	221.4	38
$\mathbf{PINN}^{3000,5}_{1,100}$	450.5	162.9
$\mathbf{PINN}^{15000,5}_{5,50}$	1.963	1.964
$\mathbf{PINN}^{15000,5}_{5,100}$	1.15e-02	8.84e-03
$\mathbf{PINN}^{500000,5}_{5,100}$	3.02e-02	3.03e-02
$\mathbf{PINN}^{15000,5}_{10,50}$	1.82	4.42
$\mathbf{PINN}^{15000,5}_{10,100}$	3.314	3.305
$\mathbf{PINN}^{200000,5}_{10,100}$	1.60e-01	1.60e-01
$\mathbf{PIELM}_{100}$	1.13e-02	1.33e-02
$\mathbf{PIELM}_{1000}$	1.28e-04	5.25e-05
$\mathbf{PIELM}^{*}_{10000}$	3.18e-12	3.07e-12
Deep-TFC	0.0	0.0
X-TFC	0.0	0.0