Coupling PIES and PINN for Solving Two-Dimensional Boundary Value Problems via Domain Decomposition

Krzysztof Szerszeń [0000-0001-9256-2622] and Eugeniusz Zieniuk [0000-0002-6395-5096]

University of Bialystok, Faculty of Computer Science
15-245 Białystok, Konstantego Ciołkowskiego 1M, Poland
k.szerszen@uwb.edu.pl, e.zieniuk@uwb.edu.pl

Abstract. The paper proposes coupling Parametric Integral Equation System (PIES) and Physics-Informed Neural Network (PINN) for solving two-dimensional potential boundary value problems defined by the Laplace equation. As a result, the computational domain can be decomposed into subdomains, where solutions are obtained independently using PIES and PINN while simultaneously satisfying interface connection conditions. The efficacy of this approach is validated through a numerical example.

Keywords: Parametric Integral Equation System (PIES), Physics-Informmed Neural Network (PINN), PIES-PINN coupling, 2D boundary value problems

1 Introduction

In recent years, the field of machine learning has attracted considerable attention within the scientific community, also due to its ability to solve problems formulated by partial differential equations (PDEs). Notably, the application of Physics-Informed Neural Networks (PINN) [1] has proven to be successful in addressing both forward and inverse problems [2]. PINN transforms a boundary value problem defined by PDE into an optimization problem, where the objective function can be directly defined by the PDE through automatic differentiation. The exponential increase in recent publications and the diverse range of applications position PINN as a viable alternative to established computational methods such as FEM, FDM, and meshless methods. While PINN shows promising potential, it faces several challenges, including the computational overhead linked with training neural networks, scalability concerns for complex geometries and the tendency of networks to learn functions with higher variability at a slower pace than those with simpler distributions. Enhancing the efficiency of PINN can be achieved by decomposing the computational domain into subdomains and employing a distinct neural network in each. Different PINN variants, such as cPINN [3], xPINN [4] as well as [5], have been introduced using this approach. Moreover, the possibility of accelerating computations through the utilization of multiple graphics processing units for network training in subdomains has been demonstrated [6].

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_11

This paper proposes coupling PINN with Parametric Integral Equation System (PIES) for solving two-dimensional boundary value problems. PIES enables a mathematical reduction of the dimension of the given boundary value problem by one. Consequently, the process of obtaining solutions within the computational domain relies on analyzing the solution of the problem at its boundary. In the proposed approach, the computational domain is decomposed into subdomains, where solutions are independently obtained using PIES and PINN while simultaneously satisfying compatibility conditions at the interfaces of these subdomains. This approach is facilitated by the fact that PIES does not require discretization of the domain, as is the case with the FEM, or just the boundary, as in the BEM. The boundary between PIES and PINN subdomains can be depicted using parametric curves, while the Chebyshev series can approximate the field and flux density functions

The proposed hybrid approach combines the advantages of domain-based methods, such as PINN, with methods based on the analysis of solutions at the boundary, such as PIES. PINN solves for unknowns within the domain, whereas PIES only deals with unknowns at the boundaries. Moreover, PIES is efficient and relatively straightforward to use in treating bounded or unbounded domains with linear material behavior. On the other hand, PINN is better suited for domains with inhomogeneities and nonlinearities. The experimental section provides a preliminary accuracy analysis of the proposed approach using a two-dimensional linear potential problem defined by Laplace's equation. To the best of the authors' knowledge, this represents the first known attempt to integrate PINN with existing computational methods for solving PDEs.

2 Problem statement

We consider a boundary value problem defined in the domain Ω bounded by the boundary Γ. As shown in Figure 1, this domain can be partitioned into subdomains: Ω_A where the solution is obtained using PINN and Ω_B, where the solution is determined using PIES.

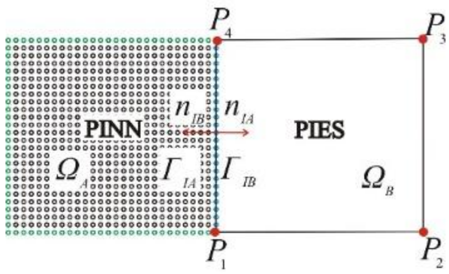

Fig. 1. Declaration of the subdomain Ω_A in PINN, the boundary of the subdomain Ω_B in PIES, with the normal vectors $\boldsymbol{n}_{IA}$ and $\boldsymbol{n}_{IB}$ to the boundary at the interface of the subdomains.

In this section, we present a concise overview of these methods and their integration at the subdomain interface. The practical aspects of the presented approach are illustrated through the analysis of a stationary temperature field problem, mathematically described by the Laplace equation.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_11

2.1 Parametric Integral Equation System

Parametric Integral Equation Systems (PIES) is a computational method designed for solving boundary value problems. It eliminates the need for discretization of both the domain and boundary into elements. The PIES formulation for the 2D problem considered in this work, described by Laplace's equation, is as follows [7]:

$$0.5u_l(\bar{s}) = \sum_{j=1}^{n} \int_{s_{j-1}}^{s_j} \{\bar{U}_{lj}^{*}(\bar{s}, s) p_j(s) - \bar{P}_{lj}^{*}(\bar{s}, s) u_j(s)\} J_j(s) ds, \quad (1)$$

$$l = 1,2, \dots, n, s_{l-1} \le \bar{s} \le s_l, s_{j-1} \le s \le s_j.$$

As depicted in Figure 1, to obtain solutions, it is necessary to specify solely the boundary of the domain Ω, whose shape is analytically embedded in the subintegral functions $\bar{U}_{lj}^{*}(\bar{s}, s)$ and $\bar{P}_{lj}^{*}(\bar{s}, s)$ defined as follows:

$$\bar{U}_{lj}^{*}(\bar{s}, s) = ln \frac{1}{(\eta_1^2 + \eta_2^2)^{0.5}}, \ \bar{P}_{lj}^{*}(\bar{s}, s) = \frac{\eta_1 n_j^{(1)}(s) + \eta_2 n_j^{(2)}(s)}{\eta_1^2 + \eta_2^2}, \quad (2a,b)$$

$$\eta_1 = \Gamma_l^{(1)}(\bar{s}) - \Gamma_j^{(1)}(s), \eta_2 = \Gamma_l^{(2)}(\bar{s}) - \Gamma_j^{(2)}(s), \quad (3)$$

where $\Gamma_l^{(1)}$ $\Gamma_l^{(2)}$ are components of Bézier curves that we use to define the boundary, $n_j^{(1)}$, $n_j^{(2)}$ denote the the normals to the boundary. Solving the boundary value problem in PIES entails determining the field and flux distribution on the boundary, expressed in formula (1) through functions $u_j(s)$ and $p_j(s)$ defined on the j-th Bézier curve. Based on these functions in the second stage of the analysis, the solution can be derived at any point within the domian. The methodology for obtaining the solution to the boundary value problem in PIES, both on the boundary and within the domain have been presented in several earlier works, including [7]. In this study, $u_j(s)$ and $p_j(s)$ are defined through Chebyshev series in the following manner:

$$u_j(s) = \sum_{k=0}^{K-1} u_j^{(k)} T_j^{(k)}(s), \ p_j(s) = \sum_{k=0}^{K-1} p_j^{(k)} T_j^{(k)}(s), \quad (4a,b)$$

where $u_j^{(k)}$, $p_j^{(k)}$ are the coefficients of the series associated with the basis functions $T_j^{(k)}(s)$, representing Chebyshev polynomials of the first kind, and K is the number of terms in these series. The functions (4a,b) will be used to integrate with PINN, as described in section 2.3.

2.2 Physics-Informed Neural Network

PINN is a machine learning technique that utilizes neural networks to obtain approximate solutions to PDEs. The fundamental idea of PINN is to determine the parameters of the neural network, denoted as $\boldsymbol{\Theta}$, in such a way that it can approximate the solution $u(\boldsymbol{x}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\Theta})$, satisfying the given PDE for points $\boldsymbol{x} \in \Omega$ and the prescribed boundary conditions for points $\boldsymbol{x} \in \Gamma$. This can be expressed as [1]:

$$\boldsymbol{D}[u(\boldsymbol{x})] = f(\boldsymbol{x}), \boldsymbol{x} \in \Omega, \quad \boldsymbol{B}[u(\boldsymbol{x})] = g(\boldsymbol{x}), x \in \Gamma, \quad (5a,b)$$

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_11

where $\boldsymbol{D}[\cdot]$ is the differential operator acting on the function $u(\boldsymbol{x})$, and $\boldsymbol{B}[\cdot]$ is the boundary operator. The terms $f(\boldsymbol{x})$ and $g(\boldsymbol{x})$ specify forcing and boundary conditions, respectively. In the context of the Laplace's equation examined in this study, the operators take the following form: $\boldsymbol{D}[\cdot] = \frac{\partial^2}{\partial^2 \boldsymbol{x}}$, $\boldsymbol{B}[\cdot] = \frac{\partial}{\partial \boldsymbol{x}}$, with $f(\boldsymbol{x})$ being set to zero. Finding neural network parameters $\boldsymbol{\Theta}$ requires minimizing the loss functions linked to the partial differential equation and boundary conditions, formulated as:

$$\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{L}}_{PDE} + \boldsymbol{\mathcal{L}}_{BC}$$

$$= \frac{1}{N_{PDE}} \sum_{i=1}^{N_{PDE}} |\boldsymbol{D}[\mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\Theta})] - f(\boldsymbol{x}_i)|^2 + \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} |\boldsymbol{B}[\mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\Theta})] - g(\boldsymbol{x}_i)|^2. \quad (6)$$

In practical implementation, this is achieved through the iterative training of the network using a set of N_{PDE} points within the domain and N_{BC} on the boundary of the problem, also referred to as collocation points.

2.3 Coupling PIES and PINN

To couple PIES and PINN, it is necessary to take into account compatibility conditions at the interface of the Γ_{IA} and Γ_{IB} subdomains, as illustrated in Figure 1. These conditions can be formulated as follows:

$$u(\boldsymbol{x})|_{\Gamma_{IA}} = u(\boldsymbol{x})|_{\Gamma_{IB}}, \nabla u(\boldsymbol{x})|_{\Gamma_{IA}} \boldsymbol{n}_{IA} = -\nabla u(\boldsymbol{x})|_{\Gamma_{IB}} \boldsymbol{n}_{IB}, \quad (7a,b)$$

where $\boldsymbol{n}_{IA}$ and $\boldsymbol{n}_{IB}$ are normal vectors to the boundary at the interface. It is worth noting the distinct representation and obtaining of $u(\boldsymbol{x})$ and $\nabla u(\boldsymbol{x})$ for PIES and PINN. In the case of PIES, they are defined as $u(\boldsymbol{x})|_{\Gamma_{IB}} = u_{j=IB}(s)$ and $\nabla u(\boldsymbol{x})|_{\Gamma_{IB}} \boldsymbol{n}_{IB} = p_{j=IB}(s)$ using Chebyshev series (4a) and (4b), and their values depend on the coefficients $u_{j=IB}^{(k)}$, $p_{j=IB}^{(k)}$, where IB is the index of the Bézier curve defining the boundary in PIES at the interface with the PINN subdomain. On the other hand, in PINN, an iterative procedure is required to determine $u(\boldsymbol{x})|_{\Gamma_{IA}} = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\Theta})$ and $\nabla u(\boldsymbol{x})|_{\Gamma_{IA}} \boldsymbol{n}_{IA} = \boldsymbol{B}[\mathcal{N}(\boldsymbol{x}; \boldsymbol{\Theta})]$ for $\boldsymbol{x} \in \Gamma_{IA}$. As a result, solving the boundary problem in PIES and PINN will be carried out through an iterative procedure using a modified form of the objective function compared to (6), given by:

$$\boldsymbol{\mathcal{L}} = \boldsymbol{\mathcal{L}}_{PDE} + \boldsymbol{\mathcal{L}}_{BC} + \boldsymbol{\mathcal{L}}_{I}\ , \quad (8)$$

$$\boldsymbol{\mathcal{L}}_{I} = \frac{1}{N_I} \sum_{i=1}^{N_I} \left| \boldsymbol{B}[\mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\Theta})] - p_{j=IB}(s_i) \right|^2. \quad (9)$$

The last term $p_{j=IB}(s_i)$ in formula (9) represents the Chebyshev series (4b) approximating the value of the flux on the boundary Γ_{IB}, where s_i denotes the parameter values at points corresponding to $\boldsymbol{x} \in \Gamma_{IA}$ on the common interface. To determine the coefficients $p_{j=IB}^{(k)}$ of this series, we need to solve PIES, employing the collocation method as described in the context of PIES in [7,8]. By expressing formula (1) at collocation points identified with $\bar{s}$, we obtain a system of algebraic equations that approximate PIES, presented in the following matrix formula:

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_11

$$[H]\begin{Bmatrix} p \\ p_{IB} \end{Bmatrix} = [G]\begin{Bmatrix} u \\ u_{IB} \end{Bmatrix}. \tag{10}$$

The generation of the $[H]$ and $[G]$ matrix elements involves calculating the regular and singular integrals, with detailed information provided in [8]. In turn, u represents the set of coefficients of the Chebyshev series in the function $u_j(s)$ (4a) on the outer boundary of the problem, while u_{IB} denotes the set of coefficients on the interface Γ_{IB}. Similarly, p and p_{IB} are sets of coefficients of Chebyshev series approximating the flux $p_j(s)$ (4b) on the outer boundary and the interface Γ_{IB}, respectively. The coefficients, denoted by u in (10), are determined using the least squares method for applied boundary conditions specifed on the outer boundary. The coefficients u_{IB} are calculated utilizing the same least squares method, but this time relying on the current solution $u(\boldsymbol{x})|_{\Gamma_A}$ obtained from the PINN in each iteration. In the following step, we can derive p_{IB} from (10). It should be emphasized that both the matrices $[H]$ and $[G]$, as well as u, are determined only once at the beginning of the iterative procedure. The schematic of the algorithm cooupling PIES and PINN is presented in Figure 2.

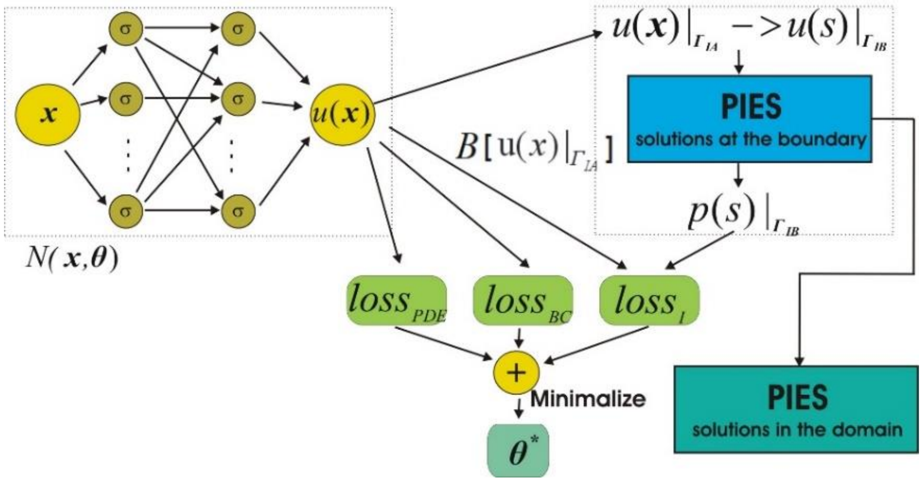

Fig. 2. A schematic diagram illustrating the coupling of PIES and PINN.

After the iterative procedure, we obtain solutions within both the domain and at the boundary in PINN, as well as solutions at the boundary in the case of PIES. The final step involves determining the solution within the domain Ω_B using the integral identity in PIES, as detailed in [9].

3 Numerical example

Below, we present the preliminary studies on the PIES-PINN coupling. We examine a two-dimensional boundary value problem defined by Laplace's equation within subdomains Ω_A and Ω_B, as illustrated in Figure 1. The following assumptions are considered:

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_11

- The square boundary of subdomain Ω_B in PIES is defined using 4 first-degree Bézier segments determined by 4 corner points P_1, P_2, P_3, P_4, with one of these segments used to define the interface Γ_{IB};
- In the square subdomain Ω_A, 100×100 collocation points are uniformly declared, with 100 points on $\boldsymbol{x} \in \Gamma_{IA}$;
- The functions $u(\boldsymbol{x})|_{\Gamma_{IB}}$ and $\nabla u(\boldsymbol{x})|_{\Gamma_{IB}} \boldsymbol{n}_{IB}$ are approximated using the first 5 terms of the Chebyshev series (4a) and (4b), respectively;
- The initial analysis presented here utilizes a fully connected neural network with 7 hidden layers, each comprising 100 neurons and employing Gaussian error linear unit (GELU) activations. Future research is planned to explore how different neural network architectures affect calculation accuracy. The network takes a two-element vector $\boldsymbol{x} = \{x_1, x_2\}$ as the input and produces a scalar value $u(\boldsymbol{x})$ at the output, representing the approximated pointwise field distribution within the domain and on the boundary;
- Dirichlet boundary conditions are imposed on the outer boundaries of the considered subdomains for two distinct functions that satisfy Laplace's equation. These functions serve as the expected analytical solutions within the problem domain:

$$u(x_1, x_2) = x_1^2 - x_2^2, \tag{11}$$

$$u(x_1, x_2) = e^{x_1} \cos x_2 + x_1. \tag{12}$$

The algorithm is implemented using the PyTorch library. In Figure 3, we present sample solutions and errors from the PIES-PINN, compared to (11) and (12), after 10000 iterations in the optimization process controlled by the ADAM optimizer.

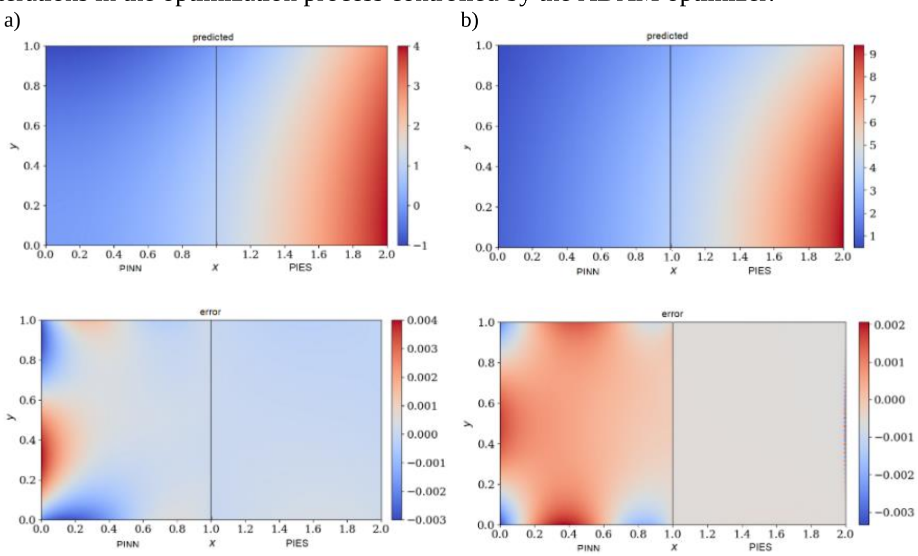

Fig. 3. PIES-PINN solutions and errors compared to (11) (a) and (12) (b).

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_11

As shown in Figure 3, the approximation error of PINN reaches the value of 1-3e within the subdomain Ω_A. Meanwhile, the error for PIES is even lower, reaching the value of 1-5e within the subdomain Ω_B. It should be noted that the solution in PIES near the boundary is inherently subject to an error due to the singular nature of the integral singularity used for this purpose, known as the boundary layer effect. To eliminate this error, the algorithm presented in [9] is applied.

Furthermore, in Figure 5, plots of the individual components of the loss function (8) are depicted for successive iterations.

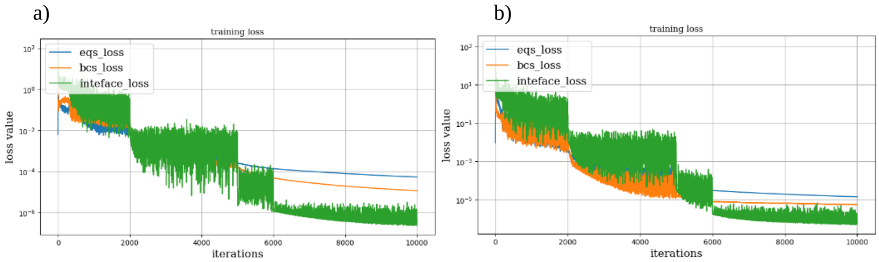

Fig. 4. Evolution of losses $\mathcal{L}_{PDE}, \mathcal{L}_{BC}, \mathcal{L}_I$ with iterations for (11) (a) and (12) (b).

We can observe fluctuations in the interface loss $\mathcal{L}_I$ (9) during iterations for both (11) and (12) boundary conditions. However, as depicted in Figure 5, the approximated values of $u(\boldsymbol{x})|_{\Gamma_{IB}}$ and $\nabla u(\boldsymbol{x})|_{\Gamma_{IB}}\boldsymbol{n}_{IB}$ at the interface Γ_{IB} closely match the analytical solutions, which is crucial for the accuracy of coupling PINN and PIES.

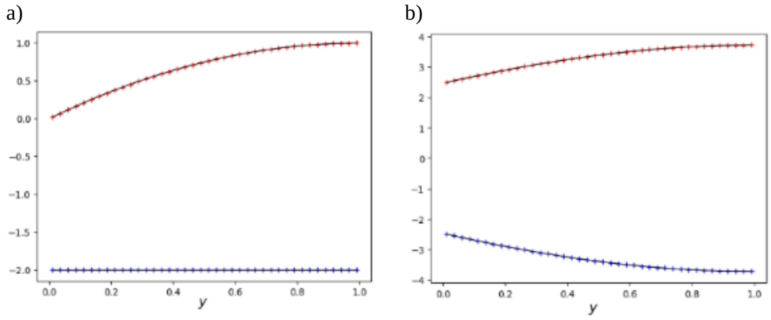

Fig. 5. The exact (black solid line) and predicted $u(\boldsymbol{x})|_{\Gamma_{IB}}$(red dashed line)**,** $\nabla u(\boldsymbol{x})|_{\Gamma_{IB}}\boldsymbol{n}_{IB}$ (blue dashed line) solutions comparison at the interface Γ_{IB} in PIES for (11) (a) and (12) (b).

Here, $u(\boldsymbol{x})|_{\Gamma_{IB}}$ and $\nabla u(\boldsymbol{x})|_{\Gamma_{IB}}\boldsymbol{n}_{IB}$ are approximated using the first 5 terms of the Chebyshev series. To enhance the accuracy of this approximation, we can increase the number of terms K in (4a,b). This is the subject of further research.

4 Conclusions

The paper proposes the coupling PIES and PINN to solve two-dimensional boundary value problems, using Laplace's equation as an example. This is advantageous because in PIES, it is only necessary to define the boundary of the domain, while concurrently

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_11

separating the representation of such a shape into Bézier curves from the approximation of solutions on the boundary in the form of Chebyshev series. In the upcoming research stage, we aim to assess the proposed initial concept with more complex shapes consisting of numerous sub-domains. Additionally, there are plans to broaden the concept to address other problems modeled by different PDEs, such as the Navier-Lamé equation. Additionally, there are intentions to apply higher-degree Bézier curves to describe domains with curved boundary shapes.

References

1. Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). 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.
2. Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific machine learning through physics–informed neural networks: Where we are and what's next. Journal of Scientific Computing, 92(3), 88.
3. Jagtap, A. D., Kharazmi, E., & Karniadakis, G. E. (2020). Conservative physics-informed neural networks on discrete domains for conservation laws: Applications to forward and inverse problems. Computer Methods in Applied Mechanics and Engineering, 365, 113028.
4. Jagtap, A. D., & Karniadakis, G. E. (2020). Extended physics-informed neural networks (XPINNs): A generalized space-time domain decomposition based deep learning framework for nonlinear partial differential equations. Communications in Computational Physics, 28(5).
5. Zhang, B., Wu, G., Gu, Y., Wang, X., & Wang, F. (2022). Multi-domain physics-informed neural network for solving forward and inverse problems of steady-state heat conduction in multilayer media. Physics of Fluids, 34(11).
6. Shukla, K., Jagtap, A. D., & Karniadakis, G. E. (2021). Parallel physics-informed neural networks via domain decomposition. Journal of Computational Physics, 447, 110683.
7. Zieniuk, E. (2003). Bézier curves in the modification of boundary integral equations (BIE) for potential boundary-values problems. International Journal of Solids and Structures, 40(9), 2301-2320.
8. Zieniuk, E., & Szerszeń, K. (2022). A regularization of the parametric integral equation system applied to 2D boundary problems for Laplace's equation with stability evaluation. Journal of Computational Science, 61, 101658.
9. Zieniuk, E., Szerszeń, K., & Bołtuć, A. (2023). A novel strategy for eliminating the boundary layer effect in the regularized integral identity in PIES for 2D potential problem. International Journal of Computational Methods, 20(03), 2250053.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63759-9_11