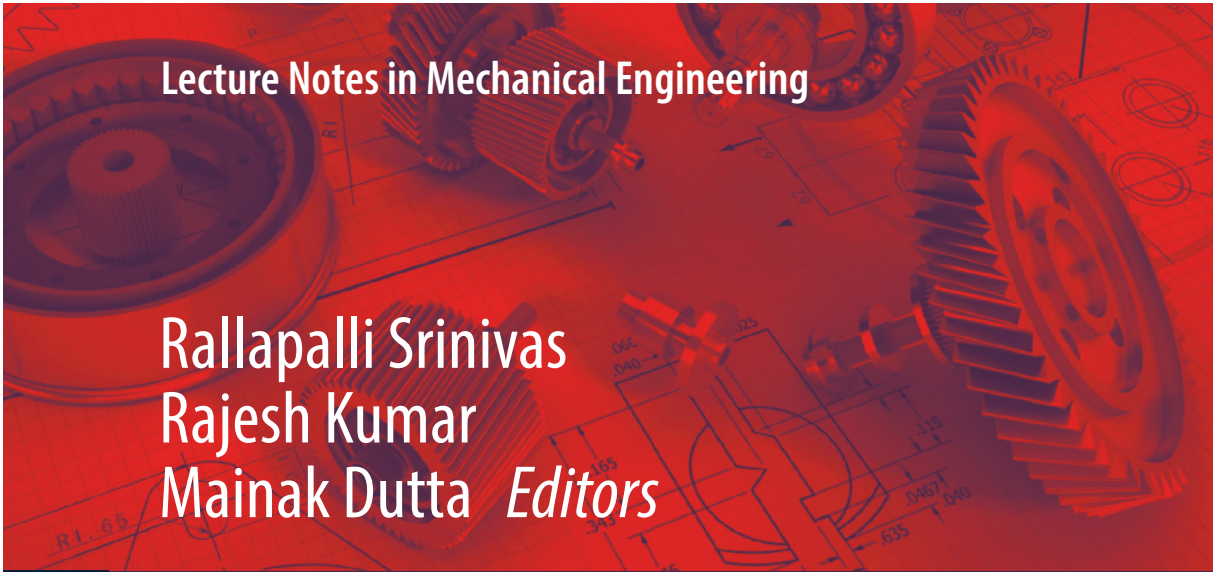




Lecture Notes in Mechanical Engineering

Rallapalli Srinivas  
Rajesh Kumar  
Mainak Dutta *Editors*

# Advances in Computational Modeling and Simulation

 Springer

# Contents

|                                                                                                                                                             |    |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| <b>Adsorption of Ammonia on p-doped Graphene Bilayer Surface; Energetics and Electronic Structure</b> .....                                                 | 1  |
| A. Sahithi and K. Sumithra                                                                                                                                  |    |
| <b>Design and Analysis of Complex Computer Models</b> .....                                                                                                 | 15 |
| Jeevan Jankar, Hongzhi Wang, Lauren Rose Wilkes, Qian Xiao, and Abhyuday Mandal                                                                             |    |
| <b>DNA Molecule Confined in a Cylindrical Shell: Effect of Partial Confinement</b> .....                                                                    | 31 |
| Neha Mathur, Arghya Maity, and Navin Singh                                                                                                                  |    |
| <b>Efficient Physics Informed Neural Networks Coupled with Domain Decomposition Methods for Solving Coupled Multi-physics Problems</b> .....                | 41 |
| Long Nguyen, Maziar Raissi, and Padmanabhan Seshaiyer                                                                                                       |    |
| <b>Estimation of Current Earthquake Hazard Through Nowcasting Method</b> .....                                                                              | 55 |
| Sumanta Pasari                                                                                                                                              |    |
| <b>Global Uniqueness Theorem for a Discrete Population Balance Model with Application in Astrophysics</b> .....                                             | 61 |
| Sonali Kaushik and Rajesh Kumar                                                                                                                             |    |
| <b>Hybrid Modeling of COVID-19 Spatial Propagation over an Island Country</b> .....                                                                         | 75 |
| Jayrold P. Arcede, Rachel C. Basañez, and Youcef Mammeri                                                                                                    |    |
| <b>Identification of Potential Inhibitors Against SARS-CoV-2 3CL<sup>pro</sup>, PL<sup>pro</sup>, and RdRP Proteins: An <i>In-Silico</i> Approach</b> ..... | 85 |
| Manju Nidagodu Jayakumar, Jisha Pillai U., Moksha Mehta, Karanveer Singh, Eldhose Iype, and Mainak Dutta                                                    |    |

|                                                                                                                                      |     |
|--------------------------------------------------------------------------------------------------------------------------------------|-----|
| <b>Influences of Top-Surface Topography on Structural and Residual Trapping During Geological CO<sub>2</sub> Sequestration</b> ..... | 113 |
| Pradeep Reddy Punnam, Balaji Krishnamurthy,<br>and Vikranth Kumar Surasani                                                           |     |
| <b>Isotopes for Improving Hydrologic Modeling and Simulation of Watershed Processes</b> .....                                        | 123 |
| Joe Magner, Brajeswar Das, Rallapalli Srinivas, Anupam Singhal,<br>and Anu Sharma                                                    |     |
| <b>Kinetic Energy Correction Factor for a Converging–Diverging Nozzle</b> .....                                                      | 131 |
| Sadhya Gulati, Snehaunshu Chowdhury, and Eldhose Iype                                                                                |     |
| <b>Modeling of Fluid-Structure Interactions with Exact Interface Tracking Methods</b> .....                                          | 139 |
| Pardha S. Gurugubelli and Vaibhav Joshi                                                                                              |     |
| <b>Multi-Phase Fluid-Structure Interaction with Diffused Interface Capturing</b> .....                                               | 155 |
| Vaibhav Joshi and Pardha S. Gurugubelli                                                                                              |     |
| <b>Multiscale Modeling of Chromatin Considering the State and Shape of Molecules</b> .....                                           | 171 |
| Yuichi Togashi                                                                                                                       |     |
| <b>Numerical Methods for the Isoperimetric Problem on Surfaces</b> .....                                                             | 177 |
| Amit R. Singh                                                                                                                        |     |
| <b>QSAR—An Important In-Silico Tool in Drug Design and Discovery</b> ....                                                            | 191 |
| Ravichandran Veerasamy                                                                                                               |     |
| <b>Simulating Landscape Hydrologic Connectivity in a Precise Manner Using Hydro-Conditioning</b> .....                               | 209 |
| Rallapalli Srinivas, Matt Drewitz, Joe Magner, Ajit Pratap Singh,<br>Dhruv Kumar, and Yashwant Bhaskar Katpatal                      |     |
| <b>The Effect of Matrix Property Smoothing on the Reliability of Fibre-Reinforced Composites</b> .....                               | 217 |
| Sadik L. Omairey, Peter D. Dunning, and Srinivas Sriramula                                                                           |     |
| <b>Understanding the Binding Affinity and Specificity of miRNAs: A Molecular Dynamics Study</b> .....                                | 229 |
| Swarnima Kushwaha, Ayushi Mandloi, and Shibasish Chowdhury                                                                           |     |

# Efficient Physics Informed Neural Networks Coupled with Domain Decomposition Methods for Solving Coupled Multi-physics Problems



Long Nguyen, Maziar Raissi, and Padmanabhan Seshaiyer

**Abstract** In this work, we introduce a novel coupled methodology called PINNs-DDM that combines a physics informed neural networks (PINNs) approach with a domain decomposition method (DDM) approach to solve multi-physics problems. The coupled methodology is applied to a variety of benchmark problems and validated against their exact solutions. Motivated by the need to solve coupled problems in enclosed spaces, we consider an application of coupling scalar transport equations to fluid dynamics equations using PINNs-DDM. While the examples and benchmark problems used in this work are in lower dimensions, they provide the necessary insight into the efficiency of the coupled method. It was noted that one of the key applications of the method is its performance for problems with limited training data. The computational results suggest that the method is very robust and can be applied to study complex real-world applications.

## 1 Introduction

Research in computational mathematics, which comprises modeling, analysis, simulation, and computing has become the foundation for solving most multidisciplinary problems in science and engineer. These real-world problems often involve complex dynamic interactions of multiple physical processes which presents a significant challenge, both in representing the physics involved and in handling the resulting coupled behavior. If the desire to predict and learn from the system is added to the picture,

---

L. Nguyen · P. Seshaiyer (✉)  
George Mason University, Mathematical Sciences, Fairfax, USA  
e-mail: [pseshaiy@gmu.edu](mailto:pseshaiy@gmu.edu)

L. Nguyen  
e-mail: [lnguye33@gmu.edu](mailto:lnguye33@gmu.edu)

M. Raissi  
University of Colorado, Applied Mathematics, Boulder, USA  
e-mail: [Maziar.Raissi@colorado.edu](mailto:Maziar.Raissi@colorado.edu)

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2022  
R. Srinivas et al. (eds.), *Advances in Computational Modeling and Simulation*,  
Lecture Notes in Mechanical Engineering,  
[https://doi.org/10.1007/978-981-16-7857-8\\_4](https://doi.org/10.1007/978-981-16-7857-8_4)

then the complexity increases even further. Hence, to capture the complete nature of the solution to the problem, a coupled multidisciplinary approach is essential.

The efficient solution to a complex coupled system that consists of functionally distinct components is still a challenging problem in computational sciences research. Direct numerical solution of the highly non-linear equations governing even the most simplified models are often challenging. The past few decades have seen significant advances in algorithms for efficient solutions including finite element methods for solving coupled multi-physics problems [1, 3, 4, 18, 19], domain decomposition method (DDM) [10, 11, 29] and multi-fidelity methods and sampling methods for Partial Differential Equations [21, 22]. Recent advances in machine learning (ML) methods, automatic differentiation (AD) [7], and ML libraries such as TensorFlow [20] and Pytorch [17] have made them potentially powerful tools for parameter estimation and data assimilation in multi-physics problems. Recently, Physics-informed (deep) neural networks (PINNs) were used to learn solutions and parameters in partial and ordinary differential equations [24]. These methodologies have helped to make tremendous progress in the development, testing, analysis, implementation, and applications of computational mathematics for simulation, optimization, and control.

Our work in this chapter is motivated by the following multi-physics application. Over the last two years, we have been faced with an unprecedented sequence of events due to COVID-19 that has impacted not only health but also economy, jobs, education, and many other sectors. As emergency efforts resume and continue in the coming weeks, thousands of personnel would need to be transported in passenger and cargo compartments. Many of these passengers would be infected or exposed to the virus and several agencies are already preparing to develop rapid solutions to study the speed of the contagion by understanding the airflow inside an aircraft that is transporting people. One of the principal dynamics involved in this process is the interplay between the flow from the air vents inside the cabins modeled via computational fluid dynamics and the scalar transport that models the concentration of the pathogen. This paper will attempt to create a simplistic model to simulate and predict the mechanisms of this coupled dynamics through a spatial and temporal distribution of airborne infection risk in an enclosed space. One of the new contributions in this work is to re-introduce PINNs and DDM methods and then develop a framework to solve multi-physics problems using a coupled PINNs-DDM methodology.

Our outline of the chapter is as follows. In Sect. 2, we will introduce the models, describe the methods and background. Specifically in Sect. 2.1, we recall the Schwarz Domain Decomposition method and apply it to the Poisson equation. Section 2.2 introduces the physics informed neural networks (PINNs) approach to solving PDEs and is applied to Burger's equation. In Sect. 2.3, we develop a new algorithm to couple PINNs with DDM to create a PINNs-DDM algorithm for multi-physics problems and apply it to various benchmark equations. Finally, in Sect. 3 we present discussion and future work.

## 2 Models, Methods and Background

In this section, we describe a multi-physics model that couples fluid dynamics that models the air velocity with scalar transport that models the concentration of a particle in an enclosed space such as an aircraft cabin. For simplicity, we will keep the exposition in the paper to one dimension that will help provide an insight into higher dimensions. After a brief introduction of these equations, we introduce DDM through a parallel Schwarz algorithm and apply it to the Poisson equation. Then, we will introduce PINNs and apply it as a forward solver to solve the Burgers equation.

The particle transport that models the concentration of a pathogen [12] that is guided by the air velocity may be modeled using the equation:

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x_i}(\rho \phi U_i) = \frac{\partial}{\partial x_i} \left( \Gamma_\phi \frac{\partial \phi}{\partial x_i} \right) + S_\phi \quad (1)$$

where  $\phi$  is contaminant concentration (for example, droplets from infected or exposed COVID-19 individuals),  $t$  is time,  $x_i$  is coordinate,  $\rho$  is air density,  $U_i$  is air velocity,  $\Gamma_\phi$  is the diffusion coefficient, and  $S_\phi$  is the mass flow rate of source per unit volume. In this work, we will consider a one-dimensional problem with the following equation for describing the

$$\frac{\partial V}{\partial t} + U \frac{\partial V}{\partial x} = \alpha_V(x) \frac{\partial^2 V}{\partial x^2} + S_V \quad (2)$$

where  $\alpha_V$  and  $S_V$  are the respective diffusion coefficient and external source term. Applications of fluid dynamics in studying airflow in ventilated enclosed spaces have been studied over last several decades [5, 14, 15]. While turbulence models building on three-dimensional Navier-Stokes equations are typically used to evaluate and design various air distributions, we will use the following one-dimensional viscous Burger's equation in this work for simplicity. This is given by

$$\frac{\partial U}{\partial t} + U \frac{\partial U}{\partial x} = \alpha_U(x) \frac{\partial^2 U}{\partial x^2} + S_U \quad (3)$$

Here,  $\alpha_U$  and  $S_U$  correspond to the viscosity of the flow and external source term, respectively. Often these partial differential equations (2) and (3) are discretized using appropriate numerical methods (such as, e.g., finite difference, finite elements, finite volumes) that involves solving a non-linear system of equations. Next, we describe a framework that can be used with any of these discretization methods to make the algebraic solution more efficient on parallel computer platforms.

## 2.1 Domain Decomposition Method (DDM)

Domain decomposition methods were introduced in the nineteenth century by German analyst Herman Schwarz as a way to reformulate and solve any given boundary-value problem on a computational domain that is partitioned into multiple subdomains [27]. This convenient framework and several variations over the past decades has allowed for efficient techniques for solving multi-physics problems that are governed by differential equations of various types in different subregions of the computational domain [2, 6, 8, 9, 26, 30].

To illustrate, DDM, let us consider the Poisson equation  $-\Delta u = f$  defined on the computational domain  $\Omega$  with  $u = g$  on the boundary  $\partial\Omega$ . Let  $\Omega$  be the union of a disk ( $\Omega_1$ ) and a rectangle ( $\Omega_2$ ) as shown in Fig. 1. The key idea behind the classical Schwarz algorithm is to iteratively solve alternating sub-problems in the domains  $\Omega_1$  and  $\Omega_2$  until the algorithm converges as follows:

$$\begin{aligned} -\Delta u_i^{n+1} &= f \quad \text{in } \Omega_i \\ u_i^{n+1} &= g \quad \text{on } \partial\Omega_i \cap \partial\Omega \setminus \partial\Omega_i \cap \overline{\Omega_{3-i}} \\ u_i^{n+1} &= u_{3-i}^n \quad \text{on } \partial\Omega_i \cap \overline{\Omega_{3-i}} \end{aligned}$$

for  $i = 1, 2$ . Schwarz proved the convergence of the algorithm and thus the well-posedness of the Poisson problem in complex geometries. If the algorithm converges, the solutions  $u_1^\infty = u_2^\infty$  in the intersection of the subdomains.

We can extend the Schwarz method to a general differential operator  $\mathcal{L}(u) = f$  in  $\Omega$  with boundary condition  $\mathcal{B}(u) = g$  on  $\partial\Omega$  as follows:

---

**Algorithm 1:** A parallel Domain Decomposition Method for Two Sub-domains  $\Omega_1, \Omega_2$

---

```

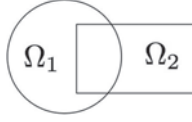
Result: Give initial guess  $u_1^0$  on  $\partial\Omega_1 \cap \overline{\Omega_2}$  and  $u_2^0$  on  $\partial\Omega_2 \cap \overline{\Omega_1}$ 
initialization;
for  $n=1$  to  $maxIter$  do
    Solve for  $u_i^n$  ( $i = 1, 2$ ):
         $\mathcal{L}(u_i^n) = f$  in  $\Omega_i$ 
         $\mathcal{B}(u_i^n) = g$  on  $\partial\Omega_i \cap \partial\Omega \setminus \partial\Omega_i \cap \overline{\Omega_{3-i}}$ 
         $u_i^n = u_{3-i}^{n-1}$  on  $\partial\Omega_i \cap \overline{\Omega_{3-i}}$ 

    if  $\|u_i^n - u_{3-i}^n\| \leq Tol$  then
        | STOP;
    end
end

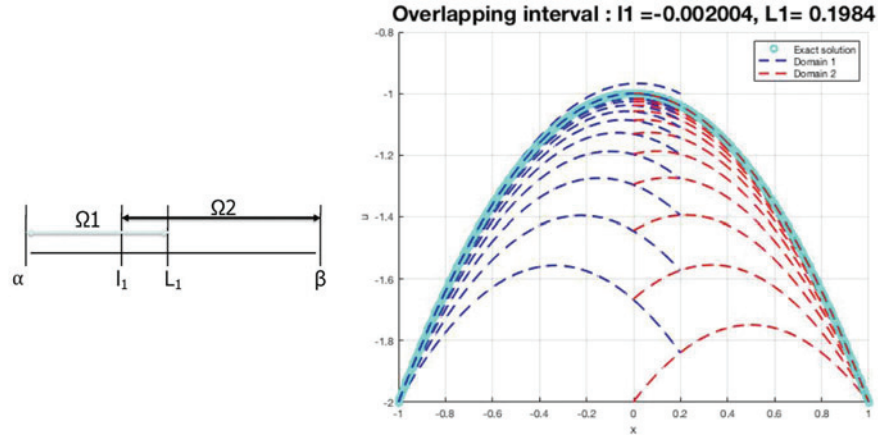
```

---

To demonstrate the performance of the Schwarz DDM, we consider the Poisson equation in 1-dimension  $-\frac{d^2u}{dx^2} = 2$  on the domain  $(\alpha, \beta)$  with  $u = -2$  at  $\alpha = -1$  and  $\beta = 1$ . The exact solution is  $u(x) = -x^2 - 1$  which is plotted as the bold line in Fig. 2 which shows the convergence of solutions in the two subdomains  $\Omega_1, \Omega_2$ .



**Fig. 1**  $\Omega = \Omega_1 \cup \Omega_2$



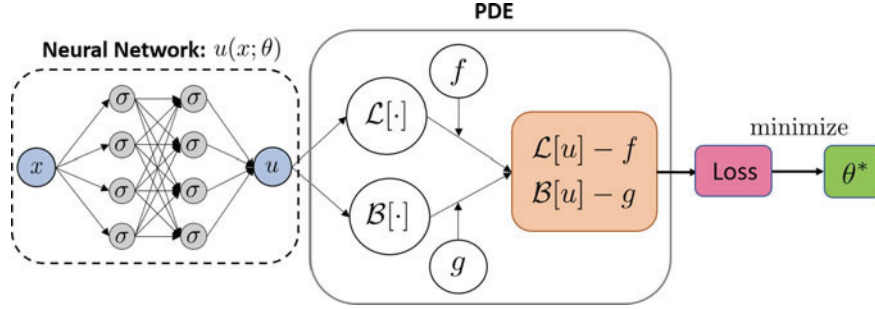
**Fig. 2** Computational domain decomposed into two overlapping subdomains (left) and the Schwarz convergence of the computational solution (right)

Next we introduce a framework around probabilistic machine learning to discover governing equations expressed by parametric linear operators.

## 2.2 Physics Informed Neural Network (PINNs)

Physics informed neural networks (PINNs) are deep learning-based techniques [22–24] for solving equations describing multi-physics including ordinary and partial differential, integro-differential and fractional order operators. One of the tools that makes these deep learning methods successful is the use of neural networks which is a system of decisions modeled after the human brain [16].

Consider the illustration shown in Fig. 3. The first layer of perceptrons first weighs and biases the input  $x$ . The next layer then will make more complex decisions based on those inputs, until the final decision layer is reached which generates the outputs  $u$ . The left part of the figure visualizes a standard neural network parameterized by  $\theta$ . The middle part in the figure applies the given physical laws to the network.  $\mathcal{L}$  and  $\mathcal{B}$  are the differential and the boundary operators, respectively. The ODE/PDE data  $(f, g)$  are obtained from random sample points. The loss function is computed by evaluating  $\mathcal{L}[u]$  and  $\mathcal{B}[u]$  on the sample points, which can be done efficiently



**Fig. 3** Illustration of the physics informed neural network approach [24, 28]

through automatic differentiation. Minimizing the loss with respect to the network's parameters  $\theta$  produces a PINNs  $u(x; \theta^*)$ , which serves as an approximation to the solution to the ODE/PDE.

In this paper, we implement a physics informed neural network-based approach (PINNs) which makes decisions based on appropriate activation functions depending on the computed bias and weights. The network then seeks to minimize the mean squared error of the regression with respect to the weights and biases by utilizing gradient descent type methods used in conjunction with software such as tensorflow. To demonstrate the performance of the PINNs method, we consider Burger's equation given by for  $x \in [0, \pi]$  and  $t \in [0, 10]$ ,

$$u_t = \frac{1}{5}u_{xx} - uu_x + e^{-\frac{2t}{5}} \sin x \cos x \quad u(t, 0) = u(t, \pi) = 0 \quad u(0, x) = \sin x$$

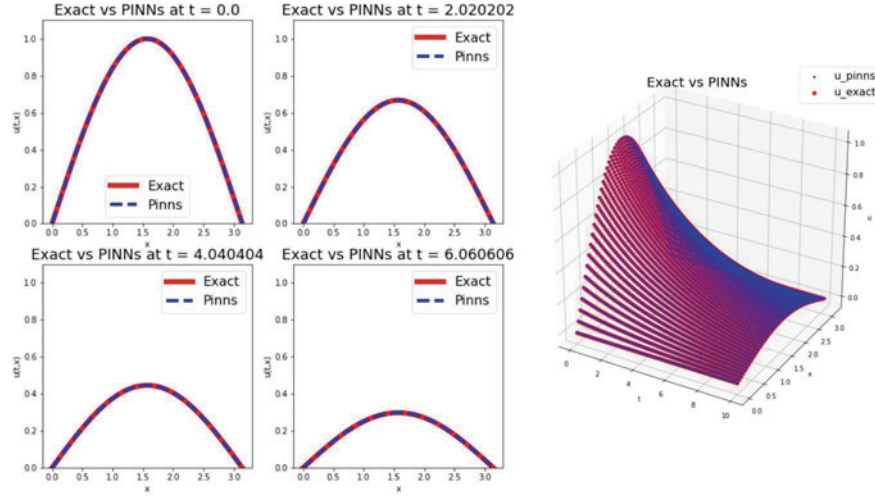
In order to create data for our simulation, we use exact solution given by  $u(t, x) = e^{-\frac{t}{5}} \sin x$ . For our PINNs implementation, we consider 3 hidden layers with size 50 for the neural network. We first choose 200 random data points  $N_u$  to estimate  $u$  by neural network denoted as  $u_{pred}$ . We also choose 200 random data points  $N_f$  to estimate PINNs residual. Defining this by,

$$f(x, t) = u_t - \frac{1}{5}u_{xx} + uu_x - e^{-\frac{2t}{5}} \sin x \cos x$$

we create a Loss function defined by

$$\text{Loss} = \frac{1}{N_u} \sum_{i=1}^{N_u} (u_{train} - u_{pred})^2 + \frac{1}{N_f} \sum_{i=1}^{N_f} (f(x, t))^2 \quad (4)$$

We train the model via 30,000 iterations using Adam [13] to minimize the Loss given by Eq. (4). Figure 4 shows that the solutions from PINNs taken at various times match



**Fig. 4** Comparison of PINNs versus exact solution at different times (left) and for all times and space (right)

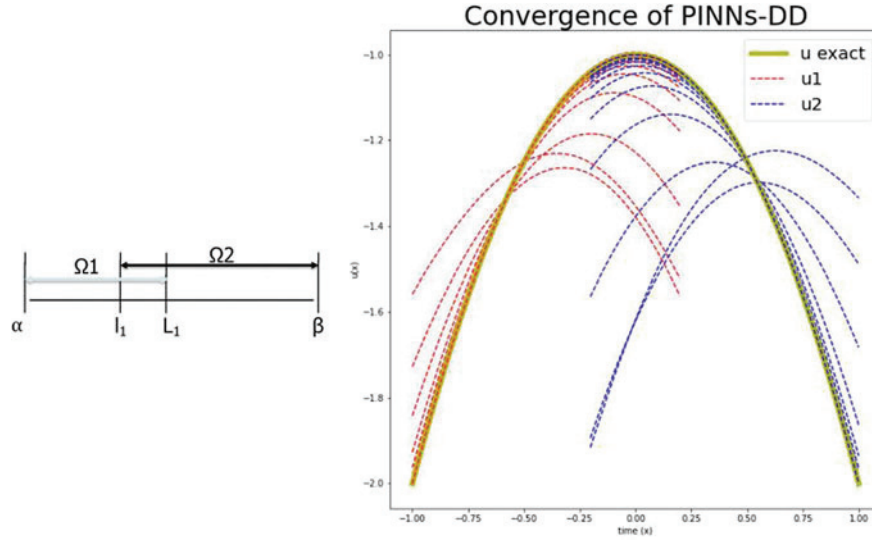
very well with the exact solution. The actual relative error in this case is  $2 \times 10^{-4}$ , and the value of the Loss function is  $5 \times 10^{-7}$ .

### 2.3 A Novel PINNS-DDM Approach

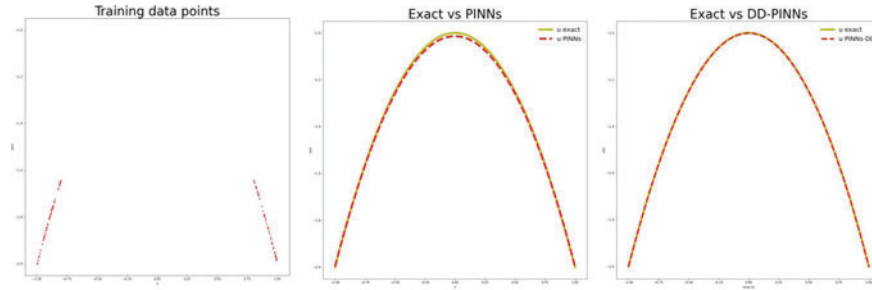
In this section, we will discuss how one can efficiently combine PINNs with DDM which is the novelty of this work, to improve accuracy in the solution methodology.

As described, PINNs is an effective method to solve multi-physics real-world applications modeled via ODE/PDE especially with good amount of training data. However in real-world applications, we often get limited data and DDM combined with PINNs can be an efficient way to solve such problems. The algorithm that will be described next was applied to the benchmark Poisson equation in 1-dimension  $-u_{xx} = 2$  on the domain  $(-1, 1)$  with  $u(-1) = u(1) = -2$  (See Fig. 5).

The result illustrated in Fig. 6 was generated using the following PINNs-DDM algorithm. Specifically, both Neural Networks were chosen to have 3 hidden layers with size of 50 nodes. Choosing random training data points  $N_{u_i} = N_{f_i} = 50$  for  $i = 1, 2$  as well as  $N_{12} = 50$  input points in overlap for training. Running the algorithm



**Fig. 5** Performance of PINNs-DDM algorithm



**Fig. 6** Training data limited to each sub-domain (left); Approximation using PINNs (middle); Approximation using PINNs-DDM (right)

for maximum number of iterations of 30, 000, we obtained an error in sub-domain one to be  $8.1 \times 10^{-5}$  and in sub-domain two to be  $3.8 \times 10^{-5}$ . Note that the PINNs-DDM is able to reproduce the same solution as in Fig. 2.

**Algorithm 2:** A novel PINNS-DDM Method for Sub-domains  $\Omega_i, i = 1, 2$ **Result:** Give data and the PDE  $\mathcal{L}(u) = f$  with boundary conditions  $\mathcal{B}(u) = g$ 

- 1 Initialize to create Neural Networks  $\mathcal{N}_i$  in  $\Omega_i \setminus \Omega_1 \cap \Omega_2$
- 2 Input the training Data  $u_{i,t}^j, j = 1..N_{u_i}$
- 3 Predict the solution  $u_{i,p}^j, j = 1..N_{u_i}$  using the Neural Network  $\mathcal{N}_i$  on  $\Omega_i \setminus \Omega_1 \cap \Omega_2$
- 4 Define the residuals  $f_i^j = f_i - \mathcal{L}(u_i^j)$  for  $j = 1..N_{f_i}$
- 5 **while**  $n \leq \text{maxiter}$  **do**

- Define the loss functions  $\text{Loss}_1$  and  $\text{Loss}_2$  as follows:

$$\text{Loss}_1 = \frac{1}{N_{u_1}} \sum_{j=1}^{N_{u_1}} (u_{1,t}^{j,n} - u_{1,p}^{j,n})^2 + \frac{1}{N_{f_1}} \sum_{j=1}^{N_{f_1}} (f_1^{j,n})^2$$

$$\text{Loss}_2 = \frac{1}{N_{u_2}} \sum_{j=1}^{N_{u_2}} (u_{2,t}^{j,n} - u_{2,p}^{j,n})^2 + \frac{1}{N_{f_2}} \sum_{j=1}^{N_{f_2}} (f_2^{j,n})^2$$

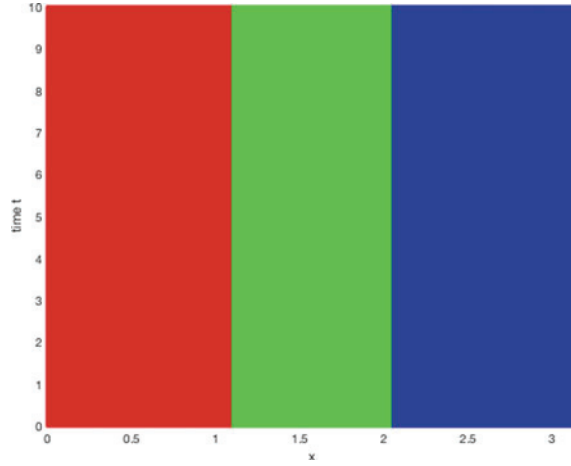
- Define the loss function  $\text{Loss}_{12}$  corresponding to the overlap in the domain decomposition using predicted functions  $u_{12,i,p}^{j,n}$  for  $j = 1..N_{u_{12}}$

$$\begin{aligned} \text{Loss}_{12} = & \frac{1}{N_{u_{12}}} \sum_{j=1}^{N_{u_{12}}} (u_{12,1,p}^{j,n} - u_{12,2,p}^{j,n-1})^2 + \frac{1}{N_{u_{12}}} \sum_{j=1}^{N_{u_{12}}} (u_{12,1,p}^{j,n-1} - u_{12,2,p}^{j,n})^2 \\ & + \frac{1}{N_{f_{12}}} \sum_{j=1}^{N_{f_{12}}} [(f_{12,1}^{j,n})^2 + (f_{12,2}^{j,n})^2] \end{aligned}$$

- Compute total loss  $\text{Loss} = \text{Loss}_1 + \text{Loss}_2 + \text{Loss}_{12}$
- Train the  $\text{Loss}$  function using Adam's method [13]
- Update weights and biases
- Compute the solution  $u_{12,i}$  in the  $\Omega_1 \cap \Omega_2$  with the respective Neural Networks  $\mathcal{N}_i$
- **if**  $\|u_{12,1}^n - u_{12,2}^n\| \leq \text{Tot}$  **then**  
| STOP;  
| **end**

**end**

Suppose we are only provided training data with each of the subdomains but no data in the overlap  $\Omega_1 \cap \Omega_2$ . This is shown in the left panel in Fig. 6 where the training data set is not complete. Using PINNS, the idea would be to solve for the whole domain by creating two different Neural Networks to solve  $u_i$  in  $\Omega_i \setminus (\Omega_1 \cap \Omega_2)$  by PINNs. This is shown in the middle panel in Fig. 6. Employing only PINNS, we are able to generate a reasonable estimate of the exact solution with an L2-error of 0.42. Finally, to improve the accuracy, we employed coupled PINNs-DDM over the entire domain. This is shown in the last panel in Fig. 6. We noted that the coupled method is able to approximate the exact solution very well with an L2-error of 0.042. For generating these solutions 30,000 iterations for both PINNs and PINNs-DDM were employed.

**Fig. 7**  $\Omega = \Omega_1 \cup \Omega_2$ 

To show the application of PINNs-DDM algorithm created as a part of this work for multi-physics problems, we apply the method to the coupled system (2)–(3). Note that we use this system for simplicity of presentation, but one can extend this to other complex multi-physics systems as well. For this computation, we use

$$\alpha_V(x) = \frac{1}{20}, \quad \alpha_U(x) = \frac{1}{5}, \quad S_U = e^{-\frac{2}{5}t} \sin x \cos x, \quad S_V = 2e^{1-\frac{2}{5}t} \sin x \cos 2x$$

for  $x \in [0, \pi]$  and  $t \in [0, 10]$ .

In order to validate our PINNs-DDM method, we evaluate the error against the exact solution for this system given by  $V(x, t) = e^{1-\frac{2}{5}t} \sin 2x$  and  $U(x, t) = e^{-\frac{2}{5}t} \sin x$ .

For the computational domain shown in Fig. 7 where  $\Omega = [0, \pi] \times [0, 10]$ , we assume that we have data in sub-domain one (red rectangle) and in sub-domain 2 (blue rectangle), but no training data in the overlap (green rectangle). As prescribed by Algorithm 2, we create two Neural Networks that both have both spatial and temporal inputs  $(x, t)$ , 3 hidden layers with size of 50 nodes and 2 outputs  $V$  and  $U$  that are coupled. Also, we choose randomly 500 training data points for each domain as well in the overlap, along with 500 input solutions at the points  $(x, t)$  for training.

Figure 8 shows the plots for errors between the PINNs-DDM solution and the exact solution in each sub-domain after 50,000 iterations of training. The plots on the left denote the approximation for  $U$  and on the right for  $V$ . Clearly, the errors indicate superior performance of the PINNs-DDM method.

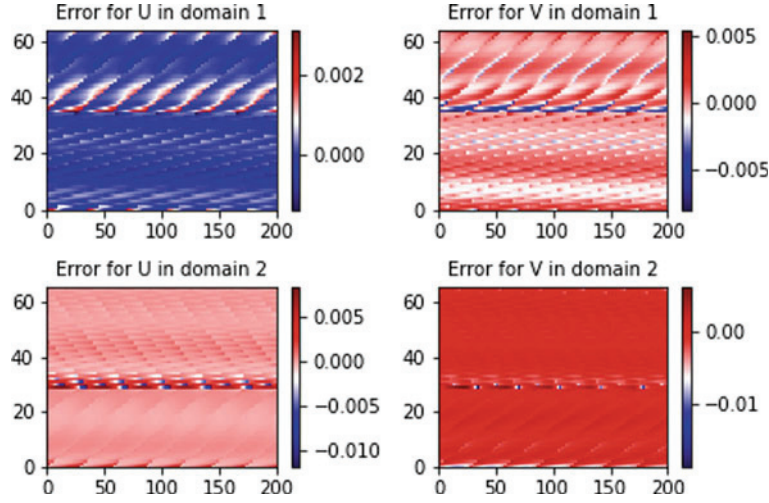


Fig. 8 Performance of PINNs-DDM for a coupled multi-physics problem

### 3 Discussion and Future Work

In this work, we combine a physics informed neural networks (PINNs) approach with a domain decomposition method (DDM) approach to yield a highly efficient methodology called PINNs-DDM for solving multi-physics problems is developed. While the examples and benchmark problems used in this work are not in higher dimensions, they provide the necessary insight into the efficiency of the method. One of the interesting findings from this work includes the performance of PINNs-DDM when only limited data is available to train. Our results suggest that PINNs-DDM is a robust candidate for solving complex system of PDEs motivated by real-world applications.

While this work has helped us to develop and design an efficient PINNs-DDM algorithm, there is still work that needs to be done to understand the convergence mathematically. As motivated in the introduction, the reason for exploring the system (2)–(3) is to understand the spread of droplet concentration (for example, from COVID-19) in the presence of airflow in enclosed spaces (such as aircraft cabins). Coupled with these are models that involve epidemiological equations that model the spread of a disease such as COVID-19. We hope to apply PINNs-DD to such an application in higher dimensions as well as equations that admit discontinuous coefficients  $\alpha_V$  and  $\alpha_U$  in a forthcoming paper. Another interesting aspect is to employ PINNs-DDM as an inverse approach to conduct parameter identification which is another aspect that will be investigated in the future.

**Acknowledgements** This work is supported in part by the Computational Mathematics program at the National Science Foundation through grant DMS 2031027 and DMS 2031029.

## References

1. Aulisa E, Manservigi S, Seshaiyer P (2005) A non-conforming computational methodology for modeling coupled problems. *Nonlinear Anal: Theory, Methods Appl* 63(5–7):e1445–e1454
2. Aulisa E, Manservigi S, Seshaiyer P (2006) A computational multilevel approach for solving 2D Navier-Stokes equations over non-matching grids. *Comput Methods Appl Mech Eng* 195(33–36):4604–4616
3. Aulisa E, Manservigi S, Seshaiyer P (2008) A multilevel domain decomposition approach to solving coupled applications in computational fluid dynamics. *Int J Numer Methods Fluids* 56(8):1139–1145
4. Aulisa E, Cervone A, Manservigi S, Seshaiyer P (2009) A multilevel domain decomposition approach for studying coupled flow applications. *Commun Comput Phys* 6(2):319
5. Baker AJ, Taylor MB, Winowich NS, Heller MR (2000) Prediction of the distribution of indoor air quality and comfort in aircraft cabins using computational fluid dynamics (CFD). In: *Air quality and comfort in airliner cabins*, ASTM international
6. Bernardi C, Maday Y, Patera AT (1993) Domain decomposition by the mortar element method. In: *Asymptotic and numerical methods for partial differential equations with critical parameters*. Springer, Dordrecht, pp 269–286
7. Baydin AG, Pearlmutter BA, Radul AA, Siskind JM (2018) Automatic differentiation in machine learning: a survey. *J Mach Learn Res* 18
8. Cao Y, Gunzburger M, He X, Wang X (2014) Parallel, non-iterative, multi-physics domain decomposition methods for time-dependent Stokes-Darcy systems. *Math Comput* 83(288):1617–1644
9. Farhat C, Mandel J, Roux FX (1994) Optimal convergence properties of the FETI domain decomposition method. *Comput Methods Appl Mech Eng* 115(3–4):365–385
10. Gander MJ, Halpern L, Nataf F (2000) Optimized Schwarz methods. In: *12th international conference on domain decomposition methods*, pp 15–27
11. Gander MJ, Hairer E (2008) Domain decomposition methods in science and engineering XVII
12. Hathway EA, Noakes CJ, Sleigh PA, Fletcher LA (2011) CFD simulation of airborne pathogen transport due to human activities. *Build Environ* 46(12):2500–2511
13. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. [arXiv:1412.6980](https://arxiv.org/abs/1412.6980)
14. Nielsen PV (1974) Flow in air conditioned rooms: model experiments and numerical solution of the flow equations
15. Lo LM (1997) Numerical studies of airflow movement and contaminant transport in hospital operating rooms. University of Minnesota, M.Sc. Thesis
16. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
17. Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, Lin Z, Desmaison A, Antiga L, Lerer A (2017) Automatic differentiation in Pytorch
18. Qi W, Seshaiyer P, Wang J (2021) A four-field mixed finite element method for Biot’s consolidation problems. *Electron Res Arch* 29(3):2517
19. Qi W, Seshaiyer P, Wang J (2021) Finite element method with the total stress variable for Biot’s consolidation model. *Numer Methods PartL Differ Equ* 37(3):2409–2428
20. Ramsundar B, Zadeh RB (2018) TensorFlow for deep learning: from linear regression to reinforcement learning. O’Reilly Media, Inc.
21. Raissi M, Seshaiyer P (2014) A multi-fidelity stochastic collocation method for parabolic partial differential equations with random input data. *Int J Uncertain Quantif* 4(3)
22. Raissi M, Seshaiyer P (2018) Application of local improvements to reduced-order models to sampling methods for nonlinear PDEs with noise. *Int J Comput Math* 95(5):870–880
23. Raissi M, Karniadakis GE (2018) Hidden physics models: machine learning of nonlinear partial differential equations. *J Comput Phys* 357:125–141
24. Raissi M, Perdikaris P, Karniadakis GE (2019) Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J Comput Phys* 378:686–707

25. Seshaiyer P, Smith P (2003) A non-conforming finite element method for sub-meshing. *Appl Math Comput* 139(1):85–100
26. Seshaiyer P, Suri M (2000) Uniform HP-convergence results for the mortar finite element method. *Math Comput* 69(230):521–546
27. Schwarz HA (1870) Ueber einen Grenzübergang durch alternirendes Verfahren. *Zürcher u, Furrer*
28. Shin Y, Darbon J, Karniadakis GE (2020) On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs. [arXiv:2004.01806](https://arxiv.org/abs/2004.01806)
29. Smith BF (1997) Domain decomposition methods for partial differential equations. In: *Parallel numerical algorithms*. Springer, Dordrecht, pp 225–243
30. Swim EW, Seshaiyer P (2006) A nonconforming finite element method for fluid-structure interaction problems. *Comput Methods Appl Mech Eng* 195(17–18):2088–2099