A PHYSICS-INFORMED NEURAL NETWORK FRAMEWORK FOR MODELING OBSTACLE-RELATED EQUATIONS

HAMID EL BAHJA, JAN C. HAUFFEN, PETER JUNG, BUBACARR BAH, AND ISSA KARAMBAL

Abstract. Deep learning has been highly successful in some applications. Nevertheless, its use for solving partial differential equations (PDEs) has only been of recent interest with current state-of-the-art machine learning libraries, e.g., TensorFlow or PyTorch. Physics-informed neural networks (PINNs) are an attractive tool for solving partial differential equations based on sparse and noisy data. Here extend PINNs to solve obstacle-related PDEs which present a great computational challenge because they necessitate numerical methods that can yield an accurate approximation of the solution that lies above a given obstacle. The performance of the proposed PINNs is demonstrated in multiple scenarios for linear and nonlinear PDEs subject to regular and irregular obstacles.

1. Introduction

The classical obstacle problem aims to describe the shape of an elastic membrane lying above an obstacle. Geometrically speaking, consider an elastic membrane that takes a specific shape under the application of a particular force. If an obstacle is close enough to the membrane, and it is in the direction of the applied force, the membrane will deform because of the obstacle. Obstacle problems have strong physical and engineering backgrounds. They emerge for instance, for the mathematical description of many physical systems including elastic-plastic torsion [38], phase transition [14], membrane-fluid interaction [20], shallow ice sheets [19, 37], etc., with typical applications varying from tumor growth modeling [8], to chemical vapor deposition [15], crystal growth and solidification in materials [25], semiconductor design [16], and option pricing [29].

Throughout the years, several numerical methods have been developed to solve different types of obstacle problems typically focusing on solving the variational inequality that appears from the first variation of the constrained problem. For instance, a finite difference scheme based on the variational inequality and the use of a multigrid algorithm to speed up computations was studied in [18]. In [2], the finite element formulation of the variational inequality is solved using the Schwarz domain decomposition method. The convergence of the Schwarz domain decomposition for nonlinear variational inequalities is established in [35, 1]. Alternative approaches use penalty formulations where the obstacle problem can also be relaxed to an unconstrained problem with the addition of a penalty term. In particular,

Hamid EL Bahja, AIMS, Cape Town, South Africa
Jan Christian Hauffen, Technical University Berlin, Germany
Peter Jung, Technical University Berlin and German Aerospace Center (DLR), Germany
Bubacarr Bah, AIMS, Cape Town, South Africa
Issa Karambal, Quantum leap Africa, Rwanda
E-mail addresses: helbahja@aims.ac.za, j.hauffen@tu-berlin.de, peter.jung@dlr.de, bubacarr@aims.ac.za, ikarambal@quantumleapafrica.org.

Key words and phrases. Physics-informed neural networks, Obstacle problems, Partial differential equations, Scientific machine learning.

in [36, 12, 33] the authors proposed an efficient numerical scheme for solving some obstacle problems based on a reformulation of the obstacle in terms of L^1 and L^2-like penalties on the variational problem. However, as expected the relaxed problem, using an L^1-penalty is non-differentiable, and the L^2-penalty which is parametrized with a coefficient that depends on a parameter ε requires that the parameter goes to infinity in order for the solution to be exact. Recently, in [9] the authors introduce a penalty method where a penalized weak formulation, in the sense of [36, 33], is minimized by using a deep neural network. Nevertheless, this method is tailored to each numerical example since the penalty parameters are tuned manually, which requires the knowledge of the exact solution a priori in order to have good approximations. For more numerical approaches see [43, 21, 26] and references therein.

Despite its effectiveness, each of the previously mentioned methods has its own limitations such as lack of convergence, non-differentiability, and domain discretization dependency. Also, in most cases, these methods have to be specifically tailored to a given problem setup and cannot be easily adapted to build a general framework for seamlessly tackling problems involving various types of partial differential equations related to an obstacle constraint. Furthermore, to the best of our knowledge, none of these existing techniques can be easily applied to data assimilation problems where only a finite number of sparse measurement points within the domain of interest are available. To overcome such setbacks, we propose a NN approaches due to the high expressive power of NN in function approximation [30, 28], recent advances in parallelized hardware, automatic differentiation [6, 5], and stochastic optimization [11]. In particular, we concentrate on physics-informed neural networks (PINNs) [31, 10, 27, 22]. This method has been used for numerically computing the solution of Schrödinger, Allen–Cahn, and Navier–Stokes equations [32, 31]. It has also been used for the approximation of the solution of high-dimensional stochastic PDEs [17]. As pointed out in [31], this approach can be considered as a class of reinforcement learning [23], where the learning is on maximizing an incentive or minimizing a loss rather than direct training on data. If the network prediction does not satisfy a governing equation, it will result in an increase in the cost and therefore the learning traverses a path that should minimize that cost. So far, most of the cases considered in the previous references are related to problems where the latent solution of the PDE has no constraints and no inter-facial phenomena occur which is the case for obstacle problems. To our knowledge, physically informed neural networks applied to obstacle problems have not yet been studied specially and systematically.

In this paper, by giving noisy measurements of the collocation points, we try to answer the following question: "What is the Physics-informed neural networks approach to finding the equilibrium position and contact area of an elastic membrane whose boundary is held fixed, and which is constrained to lie above a given obstacle?". To answer this question, we must solve the problem of computing data-driven solutions to the partial differential equations of the following general form

$$\begin{cases} \mathcal{N}[u](x) = f(x) & \text{in } \Omega, \\ u(x) = g(x) & \text{on } \partial\Omega, \\ u(x) \geq \varphi(x) & \text{in } \Omega, \end{cases} \tag{1.1}$$

where $\mathcal{N}$ is a differential operator, $u : \overline{\Omega} \longrightarrow \mathbb{R}$ is the latent solution, Ω is a bounded domain with Lipschitz regular boundary in $\mathbb{R}^N$, $\overline{\Omega}$ and $\partial\Omega$ are respectively the closure and the boundary of Ω, f and g are fixed values functions, and $\varphi : \overline{\Omega} \longrightarrow \mathbb{R}^N$ is a given obstacle. In this regard, our specific contributions can be summarized as follows:

- We present a PINNs framework that can be applied to various linear and nonlinear PDEs subject to regular and irregular obstacles.
- We show the effectiveness of our PINNs throughout many numerical experiments in solving various types of obstacle problems.

The paper is organized as follows. In Section 2, we give a mathematical and geometrical description of obstacle problems, In Section 3, we present a detailed description of our proposed PINNs. In Section 4, we demonstrate the effectiveness of our proposed PINNs through the lens of two representative case studies, including linear PDEs with regular and irregular obstacles, and nonlinear PDEs with regular and irregular obstacles. Finally, we conclude the paper in Section 5.

2. Mathematical overview

In this section, we take $\mathcal{N}[u] = \Delta u$ and $f = 0$. Therefore, (1.1) becomes a problem of minimization of the following energy functional

$$F(u) = \int_\Omega |\nabla u|^2 \, dx \tag{2.1}$$

among all functions u satisfying $u \geq \varphi$ in Ω, for a given obstacle φ. The Euler-Lagrange equation of such a minimization problem is

$$\begin{cases} u \geq \varphi & \text{in } \Omega, \\ \Delta u = 0 & \text{in } \{u \geq \varphi\}, \\ -\Delta u \geq 0 & \text{in } \Omega. \end{cases}$$

In other words, the solution u is above the obstacle φ, it is harmonic whenever it does not touch the obstacle, and it is superharmonic everywhere. The domain Ω will be split into two

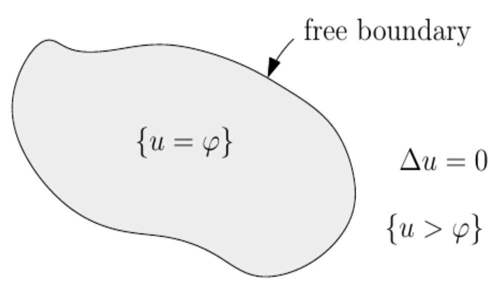

Figure 1. The contact set and the free boundary in the classical obstacle problem.

regions: one in which the solution u is harmonic, and one in which the solution equals the obstacle. The latter region is known as the contact set $\{u = \varphi\} \subset \Omega$. The interface that separates these two regions is the free boundary. For instance, in the context of financial mathematics, this type of problem appears as a model for pricing American options [29, 24]. The function φ represents the option's payoff, and the contact set is the exercise region. Notice that, in this context, the most important unknown to understand is the exercise region, i.e., one wants to find and/or understand the two regions $\{x \in \Omega : u(x) = \varphi(x)\}$ in which we should exercise the option, and $\{x \in \Omega : u(x) > \varphi(x)\}$ in which we should wait and not exercise the option yet. The free boundary is the separating interface between these two regions. It is worth noting that several studies guarantee the existence and the uniqueness of the solution to obstacle problems, see for example [3, 7] and references therein.

3. Physics-informed neural networks (PINNs)

In this section, we study Physics-informed neural networks of our obstacle problem. To this end, following the original work of Raissi et al [31], we assume that the solution $u(x)$ of (1.1) can be approximated by using a feed-forward α-layer neural network $u(x;\theta)$ where θ is a collection of all parameters in the network, such that

$$u(x;\theta) = \Sigma^{\alpha} \circ \Sigma^{\alpha-1} \circ ... \circ \Sigma^{1}(x),$$

where

$$\Sigma^{i}(z) = \sigma^{i}\left(W^{i}z + b^{i}\right), \text{ for } i = 1, .., \alpha.$$

In the above, the symbol $\circ$ denotes composition of functions, i is the layer number, x is the input to the network, Σ^{α} is the output layer of the network, and W^i and b^i are respectively the weight matrices and bias vectors of layer i, all collected in $\theta = \{W^i, b^i\}_{i=1}^{\alpha}$. σ^i is the (point-wise) activation function $i = 1, .., \alpha - 1$. In our numerical experiments below, for all the hidden layers, the hyperbolic-tangent function is used which is a preferable activation function due to its smoothness and non-zero derivative. Consequently, the parameters θ of $u(x,\theta)$ can be learned by minimizing the following composite loss function

$$\mathcal{L}(\theta) = \mathcal{L}_r(\theta) + \mathcal{L}_b(\theta), \tag{3.1}$$

where

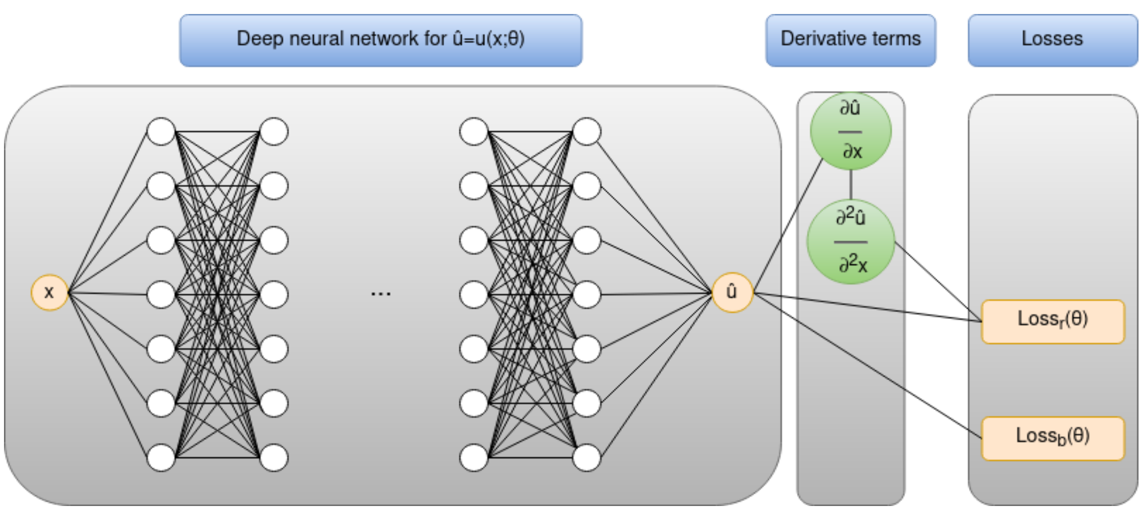

Figure 2. A schematic of network architecture used for constructing a PINNs-based obstacle model.

$$\mathcal{L}_r(\theta) = \frac{1}{N_r}\sum_{i=1}^{N_r}\left|H(u(x_r^i;\theta) - \varphi(x_r^i)) \cdot R(x_r^i;\theta) + \text{ReLu}(\varphi(x_r^i) - u(x_r^i;\theta))\right|^2, \tag{3.2}$$

$$\mathcal{L}_b(\theta) = \frac{\lambda_b}{N_b}\sum_{i=1}^{N_b}\left|u(x_b^i,\theta) - g(x_b^i)\right|^2, \tag{3.3}$$

such that $R(x, \theta)$ is the PDE residual of (1.1) defined as

$$R(x;\theta) = \mathcal{N}[u](x,\theta) - f(x),$$

H is the Heaviside step function defined as

$$H(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{otherwise,} \end{cases}$$

N_b and N_r denote the batch sizes for the training data $\{x_b^i, g(x_b^i)\}_{i=1}^{N_b}$ and $\{x_r^i, f(x_r^i)\}_{i=1}^{N_r}$ respectively, which can be randomly sampled at each iteration of a gradient descent algorithm, and the parameters λ_b denote weight coefficients in the loss function, and can effectively assign a different weighting to each individual loss term. The parameters λ_b, may be user-specified or tuned manually or automatically. In this work, the weight coefficient λ_b is auto-tuned by using the back-propagated gradient statistics during training [39]. The residual loss function $\mathcal{L}_r(\theta)$ includes derivatives of the neural network approximation function, it is possible to compute these derivatives using automatic differentiation [5] at any point in Ω without the need for manual computation.

Simultaneous training of the residual and the obstacle constraint separately tend to fail to converge in our case. In order to overcome this difficulty, we propose $\mathcal{L}_r(\theta)$ as introduced in (2.2) such that, we first optimize the network parameters of $|R(x_r^i;\theta)|^2$ with those of $|\text{ReLu}(\varphi(x_r^i) - u(x_r^i;\theta))|^2 = 0$ since $H(u(x_r^i;\theta) - \varphi(x_r^i)) = 1$, then vice versa for all $i = 1, .., N_r$, and the process is iterated until the desired tolerance is achieved.

We can summarize our PINNs method in Figure 2 and the following algorithm:

Algorithm 1: PINNs method for obstacle problems

Input: $\{x_r\}_{i=1}^{N_r}$, $\{x_b\}_{i=1}^{N_b}$, $\lambda_b = 1$, $tol > 0$
```
Initialize to create the neural network in Ω
Predict the PINNs solution u(·, θ).
Define the residual R(·, θ).
Define the ReLu(φ(·) − u(·, θ)) to penalize u(·, θ) that are under the obstacle φ.
while (1e − 2) × |R(·, θ)|^2 + |ReLu(φ(·) − u(·, θ))|^2 > tol do
    Compute loss L(θ) = L_r(θ) + L_b(θ).
    Update the weight coefficients λ_b using [39].
    Train the loss L(θ) by using Adam's optimizer.
    Updates weights and biases.
```
Output: $R(u(x_r;\theta)) \approx 0$, $u(x_b;\theta) \approx g(x_b)$, and $u(\cdot,\theta) >= \varphi(\cdot)$

4. Numerical results

In this section, we apply our PINNs approach introduced above to various numerical examples. Throughout all case studies we will use fully-connected neural networks to approximate the latent functions representing PDE solutions and unknown boundaries where the needed hyper-parameters for each obstacle are summarized in Table 1.

Obstacle	N_b	N_r	Layers	Nodes	*tol*
φ_1	200	5000	6	24	5e-4
φ_2	100	10000	8	30	1e-3
φ_3	10	5000	4	24	4e-5
φ_4	10	5000	4	24	2e-6
φ_5	50	10000	3	24	3e-6
φ_6	50	14000	4	20	3.7e-4

TABLE 1. Hyper-parameter settings employed throughout all numerical experiments presented in this work.

4.1. **One-dimensional linear obstacle problems.** To check the performance of our proposed PINNs, we first restrict ourselves to studying a variety of one-dimension regular obstacle problems which was previously considered in [14, 34]. Therefore, we have the following Poisson's equation

$$-\frac{\partial^2 u}{\partial^2 x} = 0, \text{ for } x \in \Omega = (0,1), \tag{4.1}$$

subject to Dirichlet boundary condition

$$u(1) = u(0) = 0, \tag{4.2}$$

and to the following smooth one-dimension obstacle

$$\varphi_1(x) = \begin{cases} 100x^2 & \text{for } 0 \leq x \leq 0.25, \\ 100x(1-x) - 12.5 & \text{for } 0.25 \leq x \leq 0.75, \\ 100(1-x)^2 & \text{for } 0.75 \leq x \leq 1, \end{cases} \tag{4.3}$$

such that $u \geq \varphi_1$ over $\Omega = (0,1)$. The correspondent analytic solution of (4.1) under constraints (4.2) and (4.3) is

$$u(x) = \begin{cases} (100 - 50\sqrt{2})x & \text{for } 0 \leq x \leq \frac{1}{2\sqrt{2}}, \\ 100x(1-x) - 12,5 & \text{for } \frac{1}{2\sqrt{2}} \leq x \leq 1 - \frac{1}{2\sqrt{2}}, \\ (100 - 50\sqrt{2})(1-x) & \text{for } 1 - \frac{1}{2\sqrt{2}} \leq x \leq 1. \end{cases} \tag{4.4}$$

Recall that Poisson's equation is one of the pivotal parts of electrostatics where the solution is the potential field caused by a given electric charge or mass density distribution. To this end, we will show that our deep neural network $u(x;\theta)$ approximates the latent solution $u(x)$ of (4.1) and satisfies the Dirichlet boundary condition and the obstacle constraint.

Subsequently, we can demonstrate that our PINNs also give a good approximation to the solution of the Poisson equation (4.1) under another smooth obstacle φ_2 defined as follows

$$\varphi_2(x) = \begin{cases} 10\sin(2\pi x) & \text{for } 0 \leq x \leq 0.25, \\ 5\cos(\pi(4x-1)) + 5 & \text{for } 0.25 \leq x \leq 0.75, \\ 10\sin(2\pi(1-x)) & \text{for } 0.75 \leq x \leq 1. \end{cases} \tag{4.5}$$

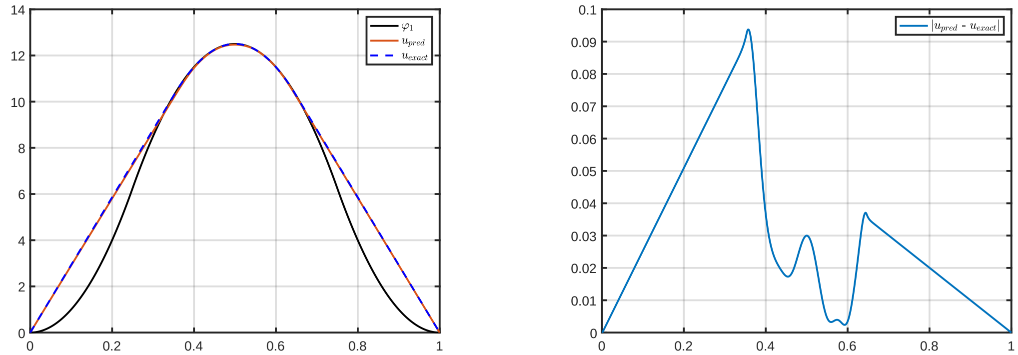

FIGURE 3. One-dimensional φ_1-obstacle Poisson's equation: (left) The predicted solution against the exact solution (4.4). (right) A plot of the pointwise L^∞-error estimation.

The corresponding exact solution such that $u \geq \varphi_2$ is

$$u(x) = \begin{cases} 10\sin(2\pi x) & \text{for } 0 \leq x \leq 0.25, \\ 10 & \text{for } 0.25 \leq x \leq 0.75, \\ 10\sin(2\pi(1-x)) & \text{for } 0.75 \leq x \leq 1. \end{cases} \tag{4.6}$$

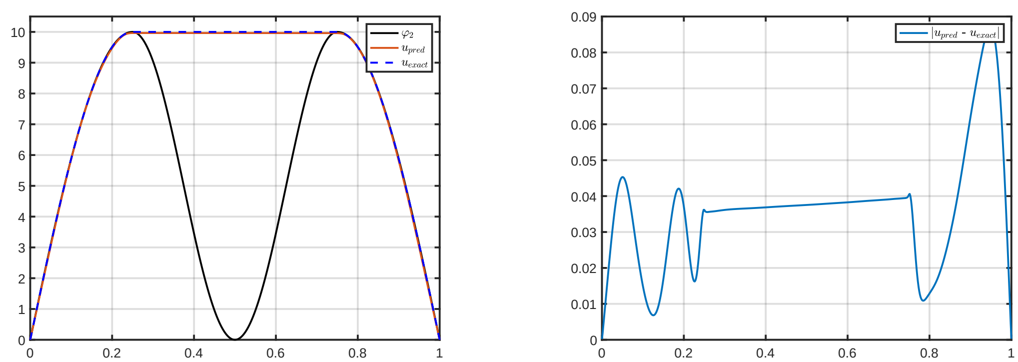

FIGURE 4. One-dimensional φ_2-obstacle Poisson's equation: (left) The predicted solution against the exact solution (4.6). (right) A plot of the pointwise L^∞-error estimation.

Figures 3 and 4 present visual comparisons with the exact solution u for obstacles φ_1 and φ_2 and the Dirichlet boundary condition (4.2). As can be seen, the approximations are in good agreement with the exact solutions. These figures indicate that our framework is able to obtain accurate predictions with a relative L^∞-error.

For the sake of completeness, we also considered an example with the same Poisson's equation with Dirichlet boundary and an irregular obstacle φ_3 defined as follows

$$\varphi_3(x) = \begin{cases} 5x - 0.75 & \text{for } 0.15 \leq x < 0.2, \\ 1 & \text{for } 0.2 \leq x \leq 0.8, \\ -5x + 4.25 & \text{for } 0.8 < x \leq 0.85, \\ 0 & \text{for else} \end{cases} \tag{4.7}$$

This obstacle has been used and can be traced back to [34], albeit with a typo in its definition.

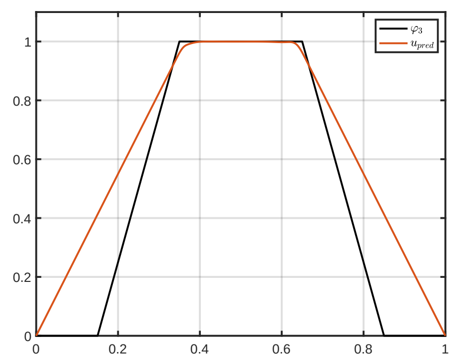

FIGURE 5. One-dimensional φ_3-obstacle Poisson's equation: The predicted PINNs solution.

Therefore, applying Algorithm 1 to equation (4.1) with Dirichlet boundary (4.2) and under the irregular obstacle φ_3 gives a PINNs solution $u(x, \theta)$ that fulfills the boundary condition and above the obstacle with some error in the rough corners of the obstacle. Therefore we cannot expect a very high accuracy similar to regular obstacles φ_1 and φ_2. This result is similar to the one given by the penalty method presented in [34].

4.2. **One-dimensional nonlinear obstacle problem.** For our next case, we intend to demonstrate further that the proposed PINNs can also be applied to non-linear problems. Accordingly, we study a classical model of stretching a non-linear elastic membrane over a fixed obstacle defined as

$$-\frac{\partial}{\partial x}\left(\frac{\frac{\partial u}{\partial x}}{\sqrt{(1 + \left|\frac{\partial u}{\partial x}\right|^2)}}\right) = 0, \text{ for } x \in \Omega = (0, 1), \tag{4.8}$$

subject to non-homogeneous Dirichlet boundary condition

$$u(0) = 5, \; u(1) = 10, \tag{4.9}$$

and to an obstacle φ_4 which is defined by the following oscillatory function

$$\varphi_4(x) = 10 \sin^2 \pi (x + 1)^2 \tag{4.10}$$

such that $u \geq \varphi_4$ over $\Omega = (0, 1)$. Therefore, by using Algorithm 1 for equation (4.8) according to constraints (4.9) and (4.10), the PINNs solution is identical to the obstacle on the contact set, and straight lines away from it as can be seen in Figure 6 which is similar to the results appeared in [36, 43].

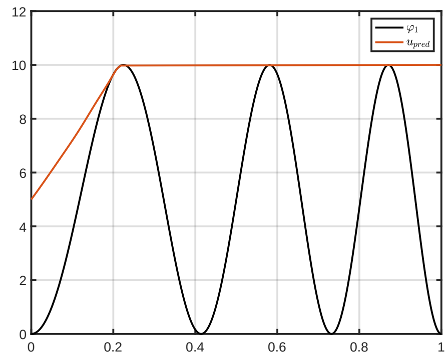

FIGURE 6. The PINNs solution of the nonlinear obstacle problem (4.10)-(4.12)

4.3. **Two-dimensional obstacle.** Our next numerical example is a two-dimensional (2D) problem defined on the domain $\Omega = [-2,2] \times [-2,2]$ such that

$$-\frac{\partial^2 u}{\partial^2 x} - \frac{\partial^2 u}{\partial^2 y} = 0, \text{ on } \Omega, \tag{4.11}$$

with the following obstacle

$$\varphi_5(x,y) = \begin{cases} \sqrt{1-x^2-y^2}, & \text{for } x^2+y^2 \leq 1, \\ -1, & \text{otherwise.} \end{cases} \tag{4.12}$$

This obstacle-related equation has been widely used by many authors to show the accuracy of their proposed method [36, 43]. Since φ_5 is a radially-symmetric obstacle, the analytical solution of (4.11) subject to (4.12) is also radially-symmetric such that

$$u(x,y) = \begin{cases} \sqrt{1-x^2-y^2}, & \text{for } x^2+y^2 \leq \beta, \\ -\beta^2 \dfrac{\log\left(\frac{\sqrt{x^2+y^2}}{2}\right)}{\sqrt{1-\beta^2}} & \text{otherwise,} \end{cases} \tag{4.13}$$

where $\beta = 0.6979651482...$ which satisfies $\beta^2\left(1-\log\left(\frac{\beta}{2}\right)\right) = 1$. Our PINNs solution and the L^∞-pointwise difference with the exact solution are presented in Figure 7. It can be seen that the error is focused as peaks near the free boundary, where the function $u(x,y)$ is no longer C^2 (two times differentiable on Ω)and is relatively small elsewhere.

For our final numerical example, we introduce a nonlinear non-harmonic equation of the form

$$\begin{cases} -\text{div}\left(|\nabla u|^{p-2}\nabla u\right) + 1 = 0, & \text{in } \Omega, \\ u = 0, & \text{on } \partial\Omega, \end{cases}$$

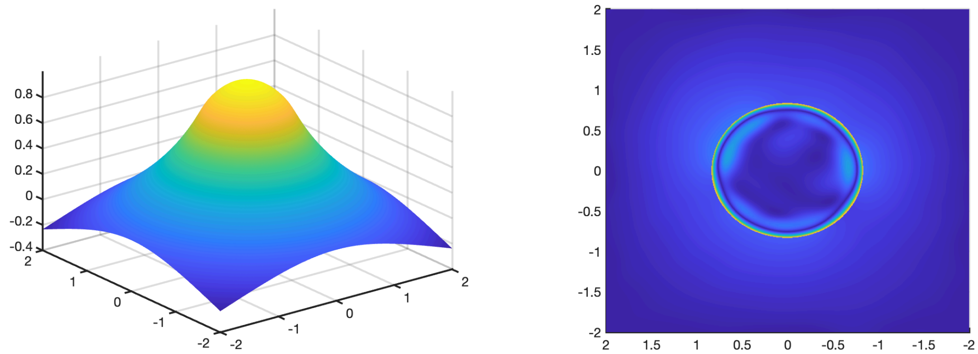

FIGURE 7. Radially symmetric 2D obstacle problem: (left) The predicted PINNs solution. (right) The relative pointwise L^∞-error map.

which is known as the anisotropic p-Laplacian equation, which is related to many physical and biological models [3, 4]. We test our PINNs framework for $(p = 4)$-Laplacian over the domain $\Omega = [0, 2] \times [0, 2]$ subject to the following irregular obstacle

$$\varphi_6(x, y) = \begin{cases} 1, & \text{for } 0.5 \le x \le 1.5, \\ 0, & \text{otherwise.} \end{cases}$$

As shown in [33], the related exact solution is the following

$$u(x, y) = \begin{cases} \frac{3}{4}|x + 7.75086|^{\frac{4}{3}} - 11.50434, & \text{if } x < 0.5, \\ 1, & \text{if } 0.5 \le x \le 1.5, \\ \frac{3}{4}|-x + 9.75086|^{\frac{4}{3}} - 11.50434, & \text{if } x > 1.5 \end{cases}$$

Our numerical solution and the pointwise L^∞-error between exact and PINNs solution are

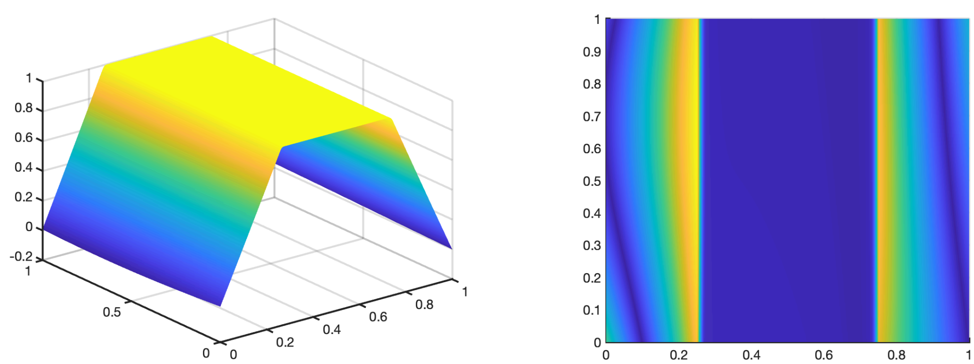

FIGURE 8. Quasi-linear 2D p-Laplacian obstacle problem: (left) The predicted PINNs solution. (right) The relative pointwise L^∞-error map.

presented in Figure 8. Despite the discontinuity of the obstacle which affects the regularity of the solution, our PINNs solution is smooth away from the obstacle and agrees well with the obstacle on its support. Also, it can be seen that the error is concentrated on the boundary of the obstacle, where the function $u(x, y)$ is discontinuous and is relatively small elsewhere.

5. Conclusions and Outlook

In summary, we present a PINNs framework for modeling obstacle-related PDEs. The distinguishing feature of our proposed PINNs is the ability to train simultaneously the residual of our PINNs solution to be close to zero if the obstacle constraint is satisfied and to train the PINNs solution to be above the obstacle if else, which gives us the ability to predict the equilibrium position of the solution whose boundary is fixed and which is constrained to lie above the given obstacle. Furthermore, we have tested the effectiveness of our PINNs across a series of numerical case studies involving different formulations of the classical one- and two-dimension PDEs with regular and irregular obstacles. As demonstrated by the numerical experiments presented here, the proposed computational framework is general and flexible in the sense that it requires minimal implementation effort in order to be adapted to different kinds of obstacle problems. The proposed PINNs do not require a large amount of labeled data generated in advance using high-fidelity simulation tools. It also does not rely on domain discretization which is typically done in the classical numerical methods.

Despite the success exhibited by the proposed PINNs, it still has some limitations such as the computational cost being generally much larger than that of the methods mentioned in the introductory section since a large number of optimization iterations are required but this can be alleviated via offline training [40, 41]. We could also change the NN architecture including the activation function type, NN width/depth, and connections between different hidden layers such as cutting and adding certain connections. We can tune these attributes of NN architecture automatically by leveraging meta-learning techniques [13, 42]. For long-time integration, one can also use time-parallel methods to simultaneously compute on multiple GPUs for shorter time domains. Another challenging question pertains to handling irregular obstacles with a complicated geometry that may include sharp cusps, mushy regions, or discontinuities which we are trying to solve in our future work. Moreover, although we are using the strong form of PDEs which is easy to execute by automatic differentiation, other weak/variational forms can also be effective, although they require the use of quadrature grids. More generally, here we have to admit that we are still in the very early stages of rigorously understanding the capabilities and limitations of PINNs. In future works, we will investigate the form of the loss function in order to avoid excessive local minima and give pretty neat approximations to more complicated obstacle cases. On the other hand, we will investigate also some interesting related problems, in particular, the double obstacle problems where the function is constrained to lie above one obstacle function and below another, level surfaces, and free boundary problems.

Funding

This work was partially supported by DAAD grants 57417688 and 57512510.

References

[1] L. Badea, Convergence rate of a multiplicative Schwarz method for strongly nonlinear variational inequalities, in Analysis and Optimization of Differential Systems, Springer, Boston, 31–41, 2003.

[2] L. Badea, X. C. Tai, and J. Wang, Convergence rate analysis of a multiplicative Schwarz method for variational inequalities, SIAM J. Numer. Anal., 41, 1052–1073, 2003.
[3] H. El Bahja: Obstacle problem of a nonlinear anisotropic parabolic equation. Ricerche mat (2021). https://doi.org/10.100/s11587-021-00559-3
[4] H. El Bahja: Bounded nonnegative weak solutions to anisotropic parabolic double phase problems with variable growth.Applicable Analysis, 1-14, 2021. https://doi.org/10.1080/00036811.2021.2021191
[5] A.G. Baydin, B. A. Pearlmutter, A.A. Radul, J.M. Siskind, Automatic differentiation in machine learning: A survey, J. Mach. Learn. Res. 18 (1),5595–5637, 2017.
[6] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, Y. Bengio, Theano: A CPU and GPU math expression compiler, in: Proceedings of the Python for Scientific Computing Conference, SciPy, Austin, TX, 1–7, 2010.
[7] V. Bögelein, F. Duzaar, G. Mingione: Degenerate problems with irregular obstacles. J. Reine Angew. Math. 650, 107–160, 2011.
[8] M. A. Chaplain, A. Stuart, A mathematical model for the diffusion of tumor angiogenesis factor into the surrounding host tissue, Mathematical Medicine and Biology: A Journal of the IMA 8 (3), 191-220, 1991.
[9] X. Cheng, X. Shen, X. Wang, K. Liang, A deep neural network-based method for solving obstacle problems, Nonlinear Analysis: Real World Applications, 72, 2023.
[10] M. Dissanayake and N. Phan-Thien, Neural-network-based approximations for solving partial differential equations, Commun. Numer. Methods Eng., 10, 195-201, 1994.
[11] Y.M. Ermoliev, R.J.-B Wets, Numerical Techniques for Stochastic Optimization, Springer-Verlag, 1988.
[12] D. A. French, S. Larsson, and R. H. Nochetto, Pointwise a posteriori error analysis for an adaptive penalty finite element method for the obstacle problem, Comput. Methods Appl. Math., 1(1):18-38, 2001.
[13] C. Finn, P. Abbeel, and S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in Proceedings of the 34th International Conference on Machine Learning, 1126-1135, 2017.
[14] A. Friedman, Variational Principles and Free-boundary Problems, Courier Dover Publications, 2010.
[15] A. Friedman, Free boundary problems in science and technology, Notices of the AMS 47 (8), 854–861, 2000.Pitman Research Notes in Mathematics Series
[16] A. Friedman, B. Hu, Stefan problem with kinetic condition arising in semiconductor processing, Pitman Research Notes in Mathematics Series, 121–121, 1995.
[17] J. Han, A. Jentzen, W. E, Solving high-dimensional partial differential equations using deep learning, Proc. Natl. Acad. Sci. 115 (34), 8505–8510, 2018.
[18] R.H.W. Hoppe, Multigrid algorithms for variational inequalities, SIAM J. Numer. Anal., 24, 1046–1065, 1987.
[19] G. Jouvet, E. Bueler, Steady, shallow ice sheets as obstacle problems: well-posedness and finite element approximation, SIAM J. Appl. Math., 72(4):1292-1314, 2012.
[20] D. Kinderlehrer, G. Stampacchia, An introduction to Variational Inequalities and their Applications, Classics in Applied Mathematics, SIAM, 31, 2000.
[21] R. Kornhuber and Q. Zou, Efficient and reliable hierarchical error estimates for the discretization error of elliptic obstacle problems, Math. Comput., 80(273):69–88, 2011.
[22] I. E. Lagaris, A. Likas, and D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Trans. Neural Networks, 9, 987-1000, 1998.
[23] S. Lange, T. Gabel, M. Riedmiller, Reinforcement learning, in: M. Wiering, M. van Otterlo (Eds.), Adaptation, Learning, and Optimization, in: Adaptation, Learning, and Optimization, 12, Springer Berlin Heidelberg, Berlin, Heidelberg, 45–73, 2012.
[24] P. Laurence, S. Salsa, Regularity of the free boundary of an American option on several assets, Comm. Pure Appl. Math. 62, 969-994, 2009.
[25] J. Madejski, Solidification of droplets on a cold surface, International Journal of Heat and Mass Transfer 19 (9), 1009–1013, 1976.
[26] K. Majava and X.-C. Tai, A level set method for solving free boundary problems associated with obstacles, Int. J. Numer. Anal. Model., 1(2):157–171, 2004.
[27] B. P. van Milligen, V. Tribaldos, and J. Jimenez, Neural network differential equation and plasma equilibrium solver, Phys. Rev. Lett., 75, 3594-3597, 1995.

[28] G. Pang, L. Yang, and G. E. Karniadakis, Neural-Net-Induced Gaussian Process Regression for Function Approximation and PDE Solution, preprint, arXiv:1806.11187, 2018.

[29] H. Pham, Optimal stopping, free boundary, and American option in a jump-diffusion model, Appl. Math. Optim. 35, 145-164, 1997.

[30] M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein, On the Expressive Power of Deep Neural Networks, preprint, arXiv:1606.05336, 2016.

[31] M. Raissi, P. Perdikaris, and G. Karniadakis, 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, 2019.

[32] S. Rudy, A. Alla, S.L. Brunton, J.N. Kutz, Data-driven identification of parametric partial differential equations, SIAM J. Appl. Dyn. Syst. 18 (2), 643–660, 2019.

[33] H. Schaeffer, A penalty method for some nonlinear variational obstacle problems." Communications in Mathematical Sciences 16.7, 1757-1777, 2018.

[34] R. Scholz: Numerical solution of the obstacle problem by the penalty method. Computing 32, no. 4, 297-306, 1984.

[35] X.C. Tai, Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities, Numer. Math., 93, 755–786, 2003.

[36] G. Tran, H. Schaeffer, W.M. Feldman, S.J. Osher, An L^1 penalty method for general obstacle problems. SIAM J. Appl. Math. 75(4), 1424-1444, 2015.

[37] C. Vazquez, E. Schiavi, J. Durany, J. I. Diaz, and N. Calvo, On a doubly nonlinear parabolic obstacle problem modeling ice sheet dynamics, SIAM J. Appl. Math., 63(2):683-707, 2003.

[38] F.Wang, X.L. Cheng, An algorithm for solving the double obstacle problems. Appl. Math. Comput. 201(1-2), 221-228, 2008.

[39] S. Wang, T. Yujun, and P. Perdikaris: Understanding and mitigating gradient flow pathologies in physics-informed neural networks." SIAM Journal on Scientific Computing 43, no. 5, A3055-A3081, 2021.

[40] N. Winovich, K. Ramani, G. Lin: ConvPDE-UQ: Convolutional neural networks with quantified uncertainty for heterogeneous elliptic partial differential equations on varied domains, Journal of Computational Physics, 394, 263-279, 2019.

[41] Y. Zhu, N. Zabaras, P.-S. Koutsourelakis, P. Perdikaris: Physics-constrained deep learning for high-dimensional surrogate modeling and uncertainty quantification without labeled data, arXiv preprint arXiv:1901.06314, 2019.

[42] B. Zoph and Q. V. Le, Neural architecture search with reinforcement learning, arXiv preprint arXiv:1611.01578, 2016.

[43] D. Zosso, B. Osting, M. M. Xia, and S. J. Osher, An efficient primal-dual method for the obstacle problem, J. Sci. Comput., 73(1):416–437, 2017.