

A PHYSICS-INFORMED NEURAL NETWORK APPROACH TO SIMULATING ICE DYNAMICS GOVERNED BY THE SHALLOW ICE APPROXIMATION

KAPIL CHAWLA AND WILLIAM HOLMES

Abstract. In this article we develop a Physics Informed Neural Network (PINN) approach to simulate ice sheet dynamics governed by the Shallow Ice Approximation. This problem takes the form of a time-dependent parabolic obstacle problem. Prior work has used this approach to address the stationary obstacle problem and here we extend it to the time dependent problem. Through comprehensive 1D and 2D simulations, we validate the model's effectiveness in capturing complex free-boundary conditions. By merging traditional mathematical modeling with cutting-edge deep learning methods, this approach provides a scalable and robust solution for predicting temporal variations in ice thickness. To illustrate this approach in a real world setting, we simulate the dynamics of the Devon Ice Cap, incorporating aerogeophysical data from 2000 and 2018.

Key words. Neural networks, ice thickness estimation, obstacle problems, feedforward neural networks, mathematical modeling, partial differential equations.

1. Introduction

Grounded ice thickness plays a critical role in understanding the behavior and stability of ice sheets, particularly in polar regions such as Greenland, Antarctica, and the Canadian Arctic. Ice sheet dynamics are governed by complex interactions between ice flow, surface accumulation, and bedrock topography, making the accurate modeling of these processes essential for predicting long-term ice sheet behavior and their contributions to global sea level rise [1, 2]. In particular, the Shallow Ice Approximation (SIA) provides a framework for modeling grounded ice, where ice flow is driven by internal deformation and the base is often assumed to be frozen, constraining the ice thickness by bedrock topography [24].

A key challenge in modeling grounded ice involves solving the partial differential equations (PDEs) that govern ice thickness evolution, while incorporating these constraints. This leads to a free boundary problem, where the ice thickness must remain non-negative and above the bedrock, giving rise to an obstacle problem [4, 27]. In previous work, we addressed the stationary obstacle problem using a neural network approach to solve for steady-state ice thickness [31, 32]. In this paper, we extend that framework to handle the time-dependent evolution of ice thickness, framing the grounded ice thickness problem as a parabolic obstacle problem.

In this work, we develop a machine learning framework to solve the grounded ice thickness problem using a Physics-Informed Neural Network (PINN). By framing the grounded ice problem as a parabolic obstacle problem, we leverage PINNs to approximate the solution of the ice thickness equation while ensuring that the obstacle and boundary conditions are satisfied [15, 14]. The neural network minimizes a composite loss function that includes terms for the PDE residual, obstacle constraint, and boundary conditions, providing a flexible and scalable method for solving this problem [19, 21].

We apply this framework to the Devon Ice Cap in the Canadian Arctic, utilizing ice thickness data from aerogeophysical surveys conducted in 2000 and 2018. By training the model on the 2000 data and validating it against the 2018 data, we demonstrate the model's ability to accurately predict the evolution of grounded ice thickness over time [23, 31].

This paper is organized as follows. Section 2 presents the time-dependent parabolic obstacle problem and the penalty method used to solve it [22]. This simplified problem shares structural similarities with the ice sheet model, which is our ultimate goal, and provides a more accessible setting to develop a computational solution approach. Section 3 describes the approximation of the solution using a Physics-Informed Neural Network (PINN) approach, including numerical results, optimization and training processes, and examples in both 1D and 2D. Section 4 introduces the mathematical model for ice sheet dynamics [9] and discusses the singularity, including that in the time derivative term. Section 5 presents numerical computations for the ice sheet model in 1D and 2D using manufactured solutions. Section 6 applies the developed techniques to the Devon Ice Cap data, demonstrating the model's effectiveness in real-world scenarios. Finally, Section 7 concludes the paper with key insights and outlines directions for future research.

2. Solving the Parabolic Obstacle Problem Using the Penalty Method

Here we formulate a simplified parabolic obstacle problem. This is similar in structure to the ice sheet model we consider in later sections. However it is simplified and thus provides a good beginning test bed to assess our solution method. Below we describe the variational formulation of this problem, the associated penalty method formulation, and the loss terms we will use in the formulation of our PINN problem.

2.1. Problem Setup. We consider the space-time domain $Q = \Omega \times (0, T)$, where $\Omega \subset \mathbb{R}^n$ is a bounded domain with smooth boundary $\partial\Omega$ (for this paper $n = 1, 2$). The goal is to find a solution $u(t, x)$ such that:

$$(1) \quad \begin{cases} \text{Find } u(t, x) \in K(\psi) \text{ such that:} \\ \int_Q (u_t(v - u) + \nabla u \cdot \nabla(v - u)) \, dx \, dt \geq \int_Q f(v - u) \, dx \, dt, \\ u \geq \psi \quad \text{in } Q, \\ u = g \quad \text{on } \partial\Omega \times (0, T), \\ u(0, x) = u_0(x) \quad \text{in } \Omega, \end{cases}$$

for all $v \in K(\psi)$, where $K(\psi) = \{v \in H_0^1(\Omega) : v \geq \psi\}$. The obstacle ψ , boundary condition g , and initial condition u_0 are given, and f represents a source term. The solution of the parabolic obstacle problem is expected to satisfy the constraint $u \geq \psi$ at all times, leading to the formation of active regions where the solution coincides with the obstacle and inactive regions where it evolves according to the parabolic operator. Away from the contact set, the solution is smooth in space and time, while reduced regularity may occur across the evolving free boundary. Due to the parabolic nature of the equation, the solution evolves continuously from the prescribed initial condition and exhibits temporal smoothing. These properties define the expected behavior of admissible solutions and are the features the PINN approximation is designed to reproduce.

2.2. Penalty Method Formulation. To solve the variational inequality, we approximate the obstacle constraint using a penalization approach. The penalized form of the problem seeks a weak solution to the following system of equations:

$$u_t^\epsilon - \Delta u^\epsilon - \frac{1}{\epsilon} \max(0, \psi - u^\epsilon) = f \quad \text{in } Q,$$

where $\epsilon > 0$ is the penalty parameter. Intuitively, the penalty term provides a soft enforcement of the obstacle constraint by penalizing violations whenever the solution drops below the obstacle. When $u^\epsilon \geq \psi$, the penalty term vanishes and the equation reduces to the unconstrained parabolic PDE, while for $u^\epsilon < \psi$ it becomes active and pushes the solution back into the admissible region. As the penalty parameter ϵ decreases, violations of the obstacle become increasingly suppressed, and the penalized solution approaches that of the original obstacle problem [3].

2.3. Solution Approximation Using Deep Neural Networks. Here we develop a PINN based approach to compute solutions to this time dependent obstacle problem with spatial dimensions $n = 1, 2$. We employ a fully connected feedforward neural network, denoted as $\mathbf{f} : \mathbb{R}^d \rightarrow \mathbb{R}$, to approximate the solution of the parabolic obstacle problem. Here the domain of \mathbf{f} is the time-space domain and its output is the elevation at that time and place: $\mathbf{f} : (t, x) \rightarrow u(t, x)$. The network parameters (weights and biases) are iteratively optimized through backpropagation to minimize a composite loss function that enforces the solution's adherence to the PDE, obstacle constraints, and boundary conditions.

To guide the neural network toward an accurate solution, we define a total loss function composed of several key components. In the below, (t_i, x_i) are co-location points in either the interior of the domain or its boundary.

- (1) **PDE Residual Loss:** This loss term ensures the solution satisfies the underlying PDE. Here the collocation points are in the interior of the domain.

$$L_{\text{PDE}}(u) = \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} (u_t(t_i, x_i) - \Delta u(t_i, x_i) - f(t_i, x_i))^2.$$

- (2) **Obstacle Constraint Loss:** This term penalizes violation of the obstacle condition. Smaller values of ϵ more strongly enforce the obstacle constraint. The collocation points here are in the interior of the domain.

$$L_{\text{penalty}}(u) = \frac{1}{\epsilon N_{\text{obs}}} \sum_{i=1}^{N_{\text{obs}}} \max(0, \psi(t_i, x_i) - u(t_i, x_i))^2.$$

- (3) **Boundary Condition Loss:** This term ensures the solution satisfies the boundary conditions. The collocation points here are on the domain boundary.

$$L_{\text{boundary}}(u) = \frac{1}{N_{\text{boundary}}} \sum_{i=1}^{N_{\text{boundary}}} (u(t_i, x_i) - g(t_i, x_i))^2.$$

- (4) **Initial Condition Loss:** This term ensures the solution satisfies the initial condition.

$$L_{\text{initial}}(u) = \frac{1}{N_{\text{init}}} \sum_{i=1}^{N_{\text{init}}} (u(0, x_i) - u_0(x_i))^2.$$

The total loss function is then defined as:

$$L_{\text{total}}(u) = L_{\text{PDE}}(u) + L_{\text{penalty}}(u) + L_{\text{boundary}}(u) + L_{\text{initial}}(u).$$

The obstacle loss is assigned a larger weight than the remaining loss terms because it represents a unilateral feasibility constraint. Empirically, we found that smaller obstacle weights allow violations of the obstacle condition near the free boundary, while excessively large weights can temporarily reduce PDE residual accuracy. The selected weighting reflects a balance identified through sensitivity testing and resulted in stable convergence across all numerical experiments.

This composite loss function ensures that the neural network approximates the solution to the PDE, while respecting the obstacle condition and satisfying both the initial and boundary conditions. The neural network is trained using the Adam optimizer, with a learning rate scheduler to adjust the learning rate dynamically during training. By iteratively minimizing the total loss function, the network parameters are refined to produce an accurate approximation of the solution. The specifics of the computational domain and the collocation points chosen within it are discussed in the 1D and 2D problem sections below.

2.4. Numerical Results. Before delving into the specific case studies, we first describe the numerical experiments and methods utilized in this research. The main objective is to approximate solutions to parabolic obstacle problems using a deep learning framework. To achieve this, we employ a fully connected neural network architecture designed to handle both 1D and 2D parabolic obstacle problems.

We begin by developing a manufactured solution to assess the precision of our method. This approach, called the *Method of Manufactured Solutions (MMS)*, is frequently employed in computational science to evaluate numerical techniques by generating an exact solution for a modified form of the PDE. In our case, we select a solution that satisfies both the obstacle and boundary conditions. The residuals, which result from this solution not fulfilling the original PDE, are used to adjust the PDE with an additional inhomogeneous term. The exact solution to this modified PDE serves as a reference against which the neural network approximation is compared. By using MMS in these initial experiments, we can thoroughly validate the reliability and accuracy of the deep learning method.

After verifying the approach through MMS, we move on to solving the parabolic obstacle problem directly. This problem involves a PDE with a challenging obstacle constraint. To fully evaluate the performance of the method, we first address a simpler PDE with an obstacle constraint, then tackle the more complex parabolic PDE of interest. These computations are performed on both 1D and 2D domains. Additionally, we apply the method to a practical problem by estimating the elevation profile of ice above the bedrock of the Devon Ice Cap.

For the neural network architecture, we utilize a fully connected design that includes at least five hidden layers, each containing 128 neurons. The *squared Rectified Linear Unit (ReLU²)* is chosen as the activation function due to its superior differentiability and strong empirical performance when addressing complex Partial Differential Equations (PDEs). This activation function enables the network to better capture nonlinear behaviors in the solution, promoting smoother and more efficient optimization [39].

2.5. Optimization and Training. For training of this solution neural network solution approximator, we use the Adam variant of stochastic gradient descent (SGD). The learning rate is initially set to 5×10^{-4} for the first 500 iterations. After that, the learning rate is reduced by half between iterations 500 and 750, and further

halved for the remaining iterations. This learning rate schedule was determined through trial and error, ensuring stable convergence of the neural network. The primary objective throughout the optimization process is the minimization of the loss functions.

For the implementation and training of our model, we rely on the *PyTorch* library, which provides the necessary tools for automatic differentiation and GPU acceleration. PyTorch's flexibility allows us to efficiently solve the parabolic obstacle problems and adjust the network architecture as needed.

2.6. 1D Example. We consider a 1D parabolic obstacle problem in the spatial domain $\Omega = [0, 1]$ and temporal domain $t \in [0, 1]$, where both the obstacle function $\psi(x, t)$ and the exact solution $u(x, t)$ are piecewise-defined functions of space and time. The obstacle function is defined as:

$$\psi(x, t) = \begin{cases} 100x^2, & 0 \leq x < 0.25, \\ 100x(1-x) - 12.5, & 0.25 \leq x < 0.5, \\ 100x(1-x) - 12.5, & 0.5 \leq x < 0.75, \\ 100(1-x)^2, & 0.75 \leq x \leq 1. \end{cases}$$

The exact solution to the problem is given by the following piecewise-defined function:

$$u(x, t) = \begin{cases} (100 - 50\sqrt{2})x \exp(-\gamma t), & 0 \leq x < \frac{1}{2\sqrt{2}}, \\ 100x(1-x) - 12.5 \exp(-\gamma t), & \frac{1}{2\sqrt{2}} \leq x < 0.5, \\ 100x(1-x) - 12.5 \exp(-\gamma t), & 0.5 \leq x < 1 - \frac{1}{2\sqrt{2}}, \\ (100 - 50\sqrt{2})(1-x) \exp(-\gamma t), & 1 - \frac{1}{2\sqrt{2}} \leq x \leq 1. \end{cases}$$

In the following experiments, we set $\gamma = 0.001$, which controls the rate of exponential decay over time. These functions ensure that the exact solution respects the obstacle constraint $u(x, t) \geq \psi(x, t)$.

The neural network is tasked with approximating the solution $u(x, t)$ while minimizing the total loss function, which is composed of the PDE residual loss, obstacle penalty loss, boundary condition loss, and initial condition loss. The network is trained over 5000 iterations using a penalty parameter $1/\epsilon = 1e - 5$, which controls the strength of the obstacle constraint and enforces $u(x, t) \geq \psi(x, t)$ via soft penalization.

During training, we use $N_{\text{PDE}} = 1000$ randomly sampled collocation points in the interior of the domain for the PDE residual and obstacle loss, $N_{\text{boundary}} = 1000$ points on the spatial boundary, and $N_{\text{initial}} = 1000$ points for the initial condition at $t = 0$. The interior and boundary collocation points are re-sampled randomly at each training iteration to promote generalization and robustness of the learned solution. This strategy avoids overfitting to specific point configurations and ensures better coverage of the domain over time.

The evolution of the loss components during training is shown in Figure 1, where the total loss, PDE loss, obstacle loss, and boundary condition loss are displayed in a log scale, demonstrating effective convergence. Note that the initial condition loss was not plotted separately, as it closely overlapped with the boundary condition loss at $t = 0$, and its values were negligible throughout training. Nevertheless, it was included in the total loss computation.

Additionally, the evolution of the L^1 error during training is shown in Figure 2, where the error is averaged over the entire spatial and temporal domain. The L^1 error is evaluated on an independently sampled set of space-time points that is

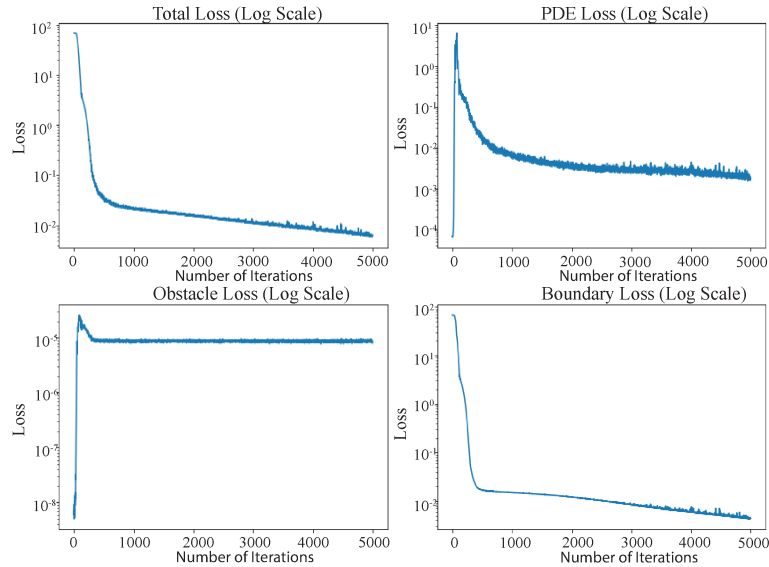


FIGURE 1. Loss components during training for the 1D parabolic obstacle problem. The figure shows the total loss, PDE loss, obstacle loss, and boundary condition loss.

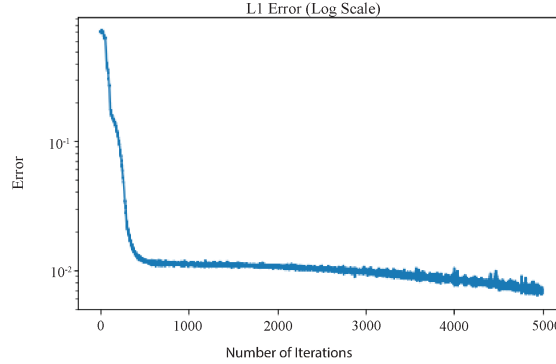


FIGURE 2. Evolution of the L^1 error for the 1D problem, showing the convergence of the neural network's approximation to the exact solution. The L^1 error is averaged over all sampled collocation points in both the spatial and temporal domains.

not used for training. This demonstrates the convergence of the neural network's approximation to the exact solution over the domain, rather than overfitting to the training collocation points.

Finally, Figures 3 compare the neural network's prediction, the exact solution, and the obstacle at different time steps $t = 0.0$, $t = 0.5$, and $t = 1.0$ for the 1D problem. These comparisons illustrate the accuracy of the neural network in approximating the exact solution over time, while adhering to the obstacle constraint.

2.7. Influence of Penalty Parameter $1/\epsilon$ on L^1 Error. The penalty parameter $1/\epsilon$, which governs the obstacle constraint, plays a significant role in controlling the

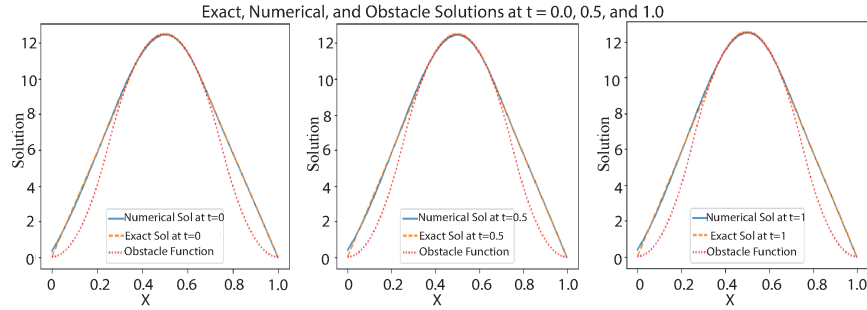


FIGURE 3. Comparison of neural network predictions with the exact and obstacle solutions for the 1D problem at $t = 0.0$, $t = 0.5$, and $t = 1.0$.

TABLE 1. L^1 errors corresponding to different values of the penalty parameter $1/\epsilon$.

| Penalty Parameter $1/\epsilon$ | L^1 Error |
|--------------------------------|-------------|
| $1e^{-1}$ | 0.40 |
| $1e^{-2}$ | 0.25 |
| $1e^{-4}$ | 0.35 |
| $1e^{-5}$ | 0.01 |
| $1e^{-8}$ | 0.33 |
| $1e^{-10}$ | 0.60 |

accuracy of the solution. The parameter ϵ is introduced in the obstacle constraint loss, and a smaller ϵ leads to a stronger penalty, enforcing the obstacle condition more strictly.

Table 1 shows the relative errors for a range of values of $1/\epsilon$. This analysis demonstrates that the penalty parameter plays a significant role in the accuracy of the model. When $1/\epsilon$ is too small, the obstacle constraint is not sufficiently enforced, while excessively large values of $1/\epsilon$ can temporarily reduce PDE residual accuracy during training. Thus, an appropriate balance is required.

3. 2D Example: Comparative Study of Two Obstacle Functions

In this section, we analyze a 2D parabolic obstacle problem over a radial spatial domain with a temporal domain $t \in [0, 1]$. Creating a non-trivial manufactured solution that temporally interacts with the obstacle is challenging. As such, we design two MMS test cases to examine the interaction between the analytical solution and the obstacle function.

(1). **Case 1:** The obstacle function remains strictly below the analytical solution, allowing free evolution over time. In this case the solution changes significantly with time but there is little interaction between the solution and the obstacle.

(2). **Case 2:** The obstacle function coincides with the analytical solution in specific regions, restricting movement. In this case the obstacle significantly influences the solution, but the temporal dynamics of that solution are more restricted.

The following subsections provide a detailed comparative analysis, including visualization, analytical formulation, and numerical approximation for both cases.

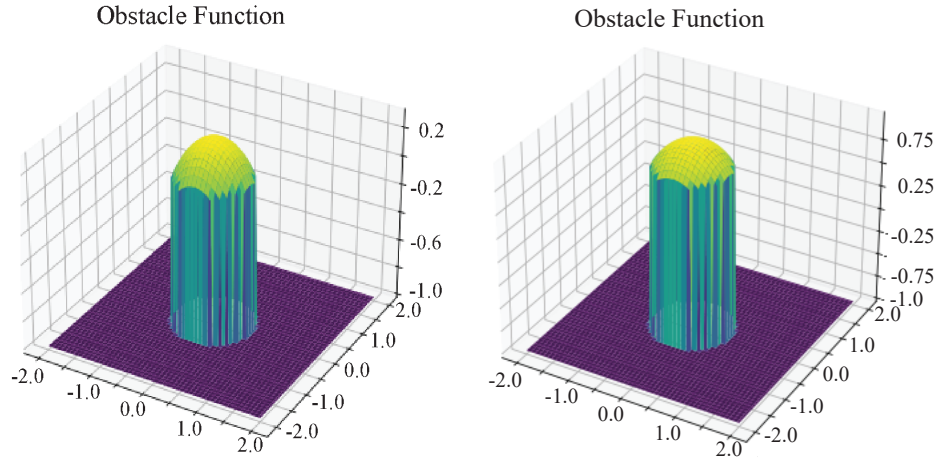


FIGURE 4. Obstacle functions for Case 1 (left) and Case 2 (right). In Case 1, the obstacle remains below the analytical solution, whereas in Case 2, it coincides with the analytical solution in some regions.

3.1. Comparison of Obstacle Functions. The obstacle functions for the two cases are defined as follows:

Case 1: No Contact Between Obstacle and Analytical Solution.

$$\psi_1(r, t) = \begin{cases} \sqrt{1 - r^2} - 0.7, & r \leq r_*, \\ -1, & r > r_*. \end{cases}$$

Case 2: Partial Contact Between Obstacle and Analytical Solution.

$$\psi_2(r, t) = \begin{cases} \sqrt{1 - r^2}, & r \leq r_*, \\ -1, & r > r_*. \end{cases}$$

The primary distinction lies in Case 2, where the obstacle function touches the analytical solution in some regions, limiting movement.

3.2. Comparison of Analytical Solutions. The analytical solution for both cases is given by:

$$u(r, t) = \begin{cases} \sqrt{1 - r^2} \exp(-\gamma t), & r \leq r_*, \\ -\frac{r_*^2 \log(r/2)}{\sqrt{1 - r_*^2}} \exp(-\gamma t), & r > r_*. \end{cases}$$

where $r_* = 0.6979651482$ and γ is a constant that is different for cases 1 and 2. This function evolves over time, constrained by the obstacle function in each case.

3.3. Numerical Comparison of Analytical and Numerical Solutions. The following figures compare the analytical and numerical solutions for both cases at different time steps ($t = 0.0, 0.5, 1.0$).

Figure 7 show one-dimensional slice plots along $y = 0$, illustrating the analytical solution, the PINN approximation, and the obstacle at selected time instances.

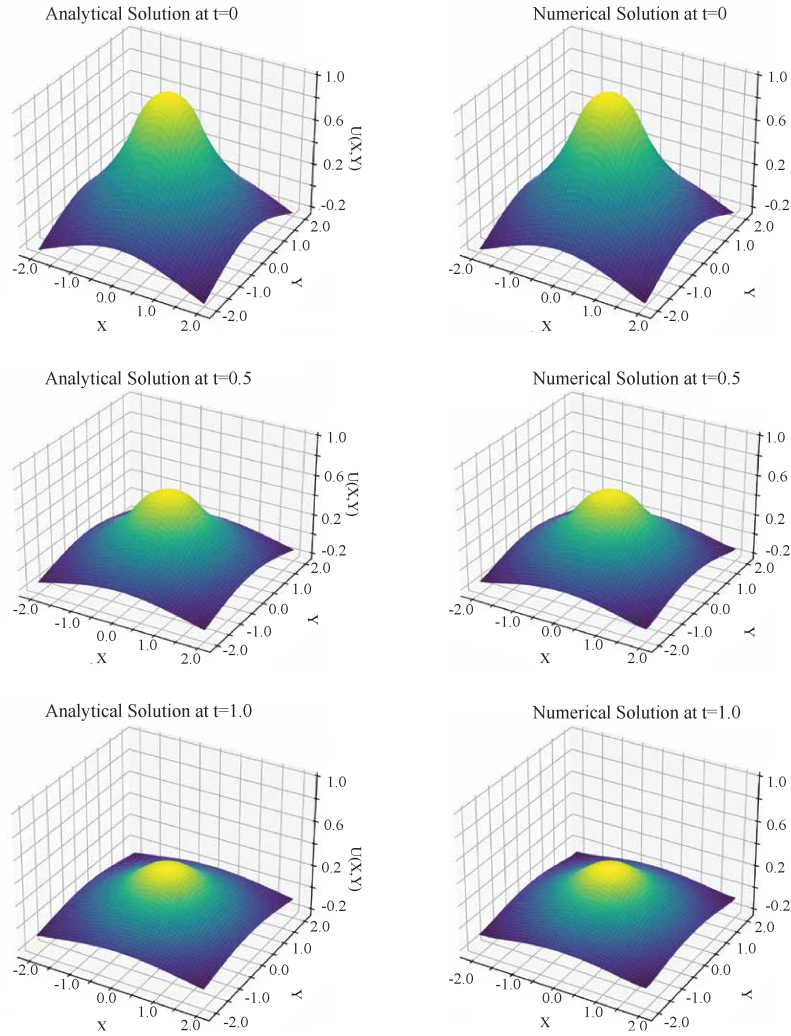


FIGURE 5. Comparison of analytical and numerical solutions for **Case 1** at different time steps. Top: $t = 0.0$. Middle: $t = 0.5$. Bottom: $t = 1.0$. Each row shows (from top to bottom) the analytical solution and the numerical solution for each time step.

3.4. Loss Components and L^1 Error Analysis. The neural network minimizes the total loss, which includes PDE loss, obstacle loss, and boundary condition loss. Figure 8 shows the evolution of these loss components over 2000 iterations for both Case 1 and Case 2, with different colors used to distinguish each case. The brief increase observed in the total, PDE and boundary losses for Case 1 is associated with collocation-point re-sampling and the learning-rate schedule, which can produce short-term fluctuations in individual loss components during training. The evolution of the L^1 error is presented in Figure 9, demonstrating the convergence of the numerical approximation to the analytical solution.

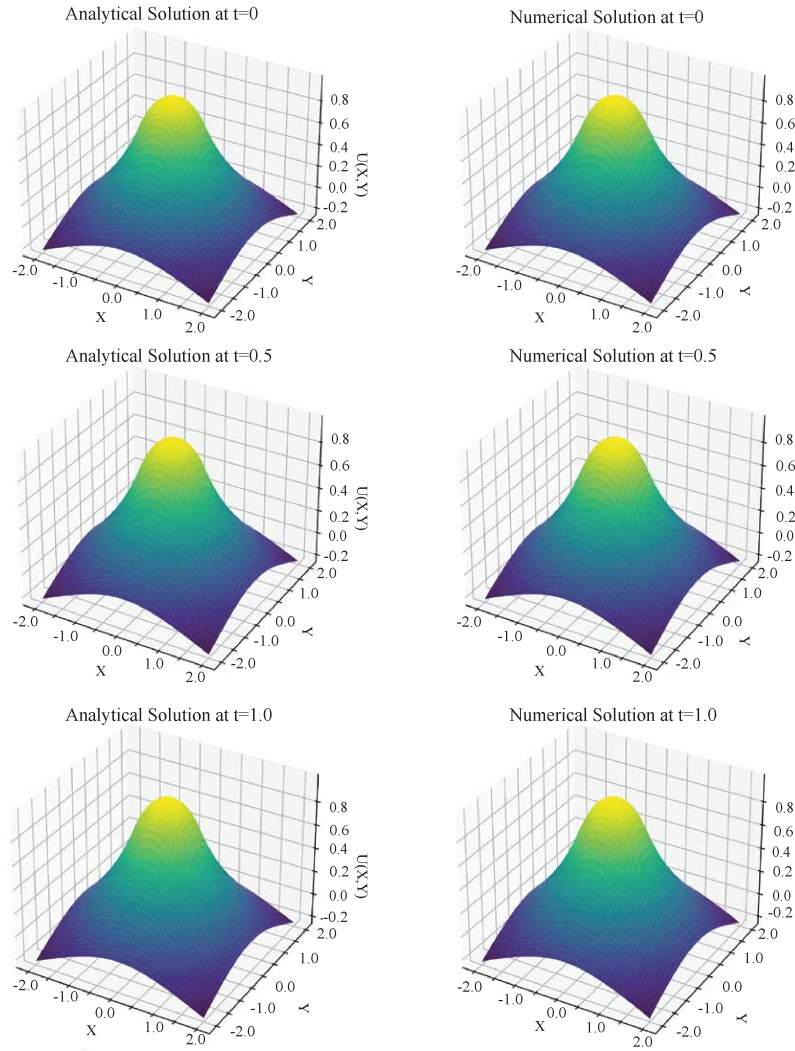


FIGURE 6. Comparison of analytical and numerical solutions for **Case 2** at different time steps. Top: $t = 0.0$. Middle: $t = 0.5$. Bottom: $t = 1.0$. Each row shows (from top to bottom) the analytical solution and the numerical solution for each time step. The obstacle function restricts movement in certain regions, impacting the solution's evolution.

To further assess the spatial accuracy of the numerical approximation, we examine the pointwise absolute error at the final time $t = 1.0$. Figure 10 shows its spatial distribution for both test cases.

These results demonstrate that the neural network effectively learns the behavior of the parabolic obstacle problem under different constraints, validating the numerical approach.

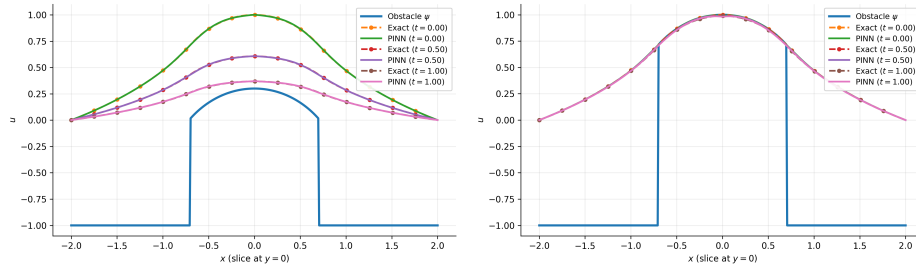


FIGURE 7. One-dimensional slice of the solution along $y = 0$ for the 2D problem. **Left:** Case 1, where the solution remains above the obstacle. **Right:** Case 2, where the obstacle constrains the solution over time.

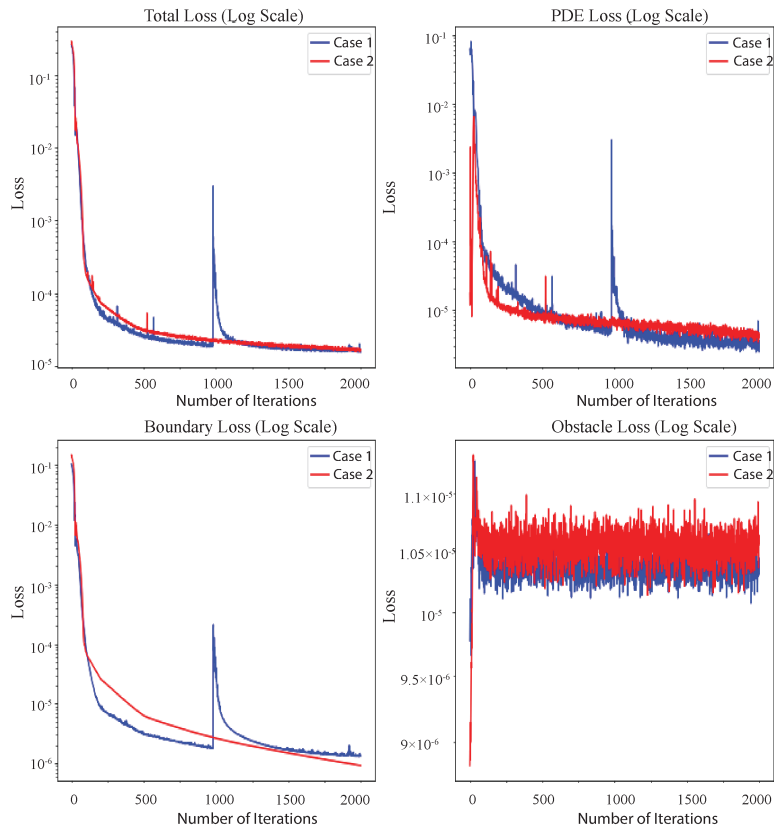


FIGURE 8. Loss components during training for the 2D problem. The figure presents the total loss, PDE loss, obstacle loss, and boundary condition loss in logarithmic scale over 2000 iterations. Different colors indicate Case 1 and Case 2, showing how the loss components evolve distinctly for each scenario.

4. Mathematical Model of Ice Sheet Dynamics

In this section, we present the mathematical formulation of the model, adapted from [1, 2], and describe the evolution of ice sheet dynamics. This model considers

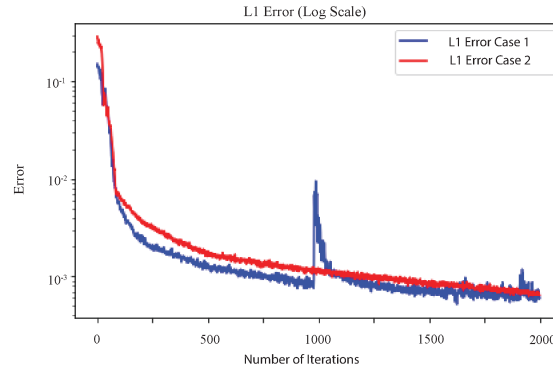


FIGURE 9. Evolution of the L^1 error for the 2D problem. The plot shows the convergence of the numerical solution towards the analytical solution, with separate curves for Case 1 and Case 2.

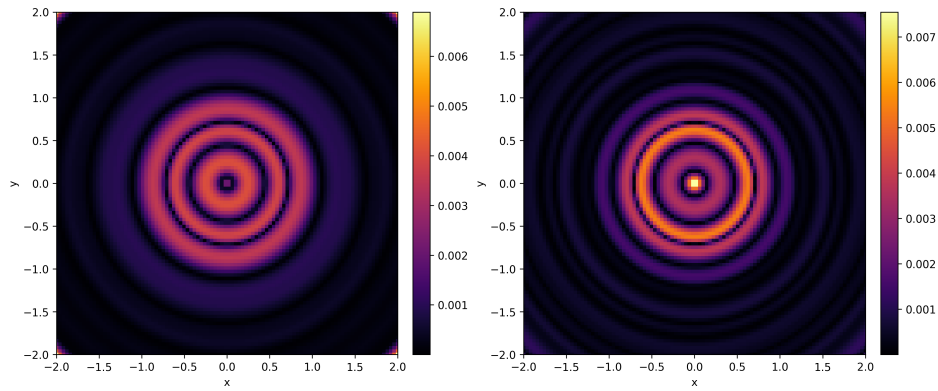


FIGURE 10. Pointwise absolute error distribution at the final time $t = 1.0$ for the 2D parabolic obstacle problem. **Left:** Case 1 (no contact between obstacle and analytical solution). **Right:** Case 2 (partial contact between obstacle and analytical solution).

the surface elevation of the ice sheet relative to the underlying lithosphere and the evolution of ice thickness.

4.1. Model Assumptions. Before we proceed with the governing equations, we outline the key assumptions of the model:

- The ice sheet's surface elevation $h(t, x)$ is constrained to remain above the lithosphere elevation (the obstacle) $b(x)$, for all $x \in \Omega$ and for all times $t \in [0, T]$.
- The ice thickness $H(t, x) = h(t, x) - b(x)$ is constrained to be non-negative, resulting in an obstacle problem where the lithosphere elevation $b(x)$ serves as the lower bound (obstacle) for the surface elevation.
- The basal sliding velocity $U_b = 0$ when the base of the ice is frozen.
- The surface mass balance term $a(t, x)$ depends on time, space, and the surface elevation $h(t, x)$, while basal melting is neglected ($a_b = 0$).

4.2. Governing Equations. Let Ω be a two-dimensional bounded domain in \mathbb{R}^2 , representing the spatial extent of the ice sheet. For any point $x = (x_1, x_2) \in \bar{\Omega}$, the surface elevation of the ice sheet is represented by $h(t, x)$, where $t \in [0, T]$.

The evolution of the ice sheet's surface elevation, $h(t, x)$, is subject to the constraint that the surface remains above the lithosphere elevation, denoted by $b(x)$:

$$(2) \quad h(t, x) \geq b(x), \quad \forall x \in \Omega, t \in [0, T].$$

Here, $b(x)$ describes the topography of the lithosphere, with $b(x) > 0$ representing elevations above sea level and $b(x) < 0$ representing those below.

The ice thickness $H(t, x)$ is defined as the difference between the surface elevation and the lithosphere:

$$(3) \quad H(t, x) = h(t, x) - b(x), \quad H(t, x) \geq 0.$$

This formulation introduces an obstacle problem where the obstacle is the lithosphere elevation $b(x)$.

At each time $t \in (0, T)$, the region covered by ice is defined as:

$$(4) \quad \Omega_t^+ = \{x \in \Omega \mid h(t, x) > b(x)\} = \{x \in \Omega \mid H(t, x) > 0\}.$$

The corresponding free boundary [35], denoted by $\Gamma_{f,t}$, represents the boundary between the ice-covered region Ω_t^+ and the region devoid of ice $\Omega_t^- = \Omega \setminus \Omega_t^+$. The free boundary is mathematically expressed as:

$$(5) \quad \Gamma_{f,t} = \Omega \cap \partial\Omega_t^+.$$

The evolution of the ice sheet is governed by the conservation of mass, influenced by two primary sources: the surface mass balance, a_s , and the basal melting rate, a_b . When the basal temperature is below the melting point, we assume the basal melting rate is negligible ($a_b = 0$), so the dynamics focus on the surface mass balance term, $a(t, x) = a_s(t, x, h)$.

The total ice flux, $Q(t, x)$, which represents the horizontal volume of ice moving across the surface, is given by:

$$(6) \quad Q(t, x) = \int_b^h U(t, x, z) dz,$$

where $U(t, x, z)$ denotes the horizontal ice velocity at elevation z , and the ice velocity is primarily driven by the surface gradient ∇h . Since the base is frozen [36] ($U_b = 0$), the velocity field is determined purely by internal deformation of the ice.

We also assume that there is no ice flux across the free boundary [1]:

$$(7) \quad Q \cdot \nu = 0 \quad \text{on} \quad \Gamma_{f,t},$$

where ν is the outward normal vector to the boundary. Moreover, since there is no ice in the region not covered by the ice sheet, we set:

$$(8) \quad Q = 0 \quad \text{in} \quad \Omega_t^-.$$

Additionally, the ice thickness satisfies:

$$(9) \quad H = 0 \quad \text{on} \quad \Gamma_{f,t}, \quad \text{and} \quad H = 0 \quad \text{on} \quad \partial\Omega.$$

4.3. Ice Thickness Equation. The dynamics of ice thickness $H(t, x)$ are governed by the balance between the flux divergence and the surface mass balance [36]:

$$(10) \quad \frac{\partial H}{\partial t} = -\nabla \cdot Q + a(t, x).$$

This equation captures how the ice thickness evolves in response to horizontal ice fluxes and surface accumulation or ablation.

4.4. Variational Formulation. To solve the ice thickness equation, we seek a variational formulation, which provides a more flexible framework for weak solutions, allowing us to handle the free boundary effectively. Let $v = v(t, x)$ be any test function such that $v(t, x) \geq b(x)$ for almost every $(t, x) \in (0, T) \times \Omega$. Multiplying the ice thickness equation by $(v - h)$ and integrating over Ω , we get:

$$(11) \quad \int_{\Omega} \frac{\partial H}{\partial t} (v - h) \, dx + \int_{\Omega} \nabla \cdot Q (v - h) \, dx = \int_{\Omega} a (v - h) \, dx.$$

Applying the Gauss-Green theorem simplifies this to:

$$(12) \quad \int_{\Omega_t^+} \left(\frac{\partial H}{\partial t} + \nabla \cdot Q - a \right) (v - h) \, dx = 0 \quad \text{for all } t \in (0, T).$$

In regions where $H = 0$, the equation becomes:

$$(13) \quad \frac{\partial H}{\partial t} \geq 0, \quad a \leq 0.$$

4.5. Boundary Value Problem. The full boundary value problem is now described as finding $H \geq 0$ such that:

$$(14) \quad \frac{\partial H}{\partial t} + \nabla \cdot Q - a = 0 \quad \text{in } \Omega_t^+,$$

and

$$(15) \quad \frac{\partial H}{\partial t} + \nabla \cdot Q - a \geq 0 \quad \text{in } \Omega_t^-,$$

with the boundary condition $H(t, \cdot) = 0$ on $\partial\Omega$ and the initial condition $H(0, \cdot) = H_0 \geq 0$. Combining equations (14) and (15) leads to the following free boundary value problem:

$$(16) \quad \begin{cases} \frac{\partial H}{\partial t} + \nabla \cdot Q - a = 0, & \text{in } \Omega_t^+, \\ \frac{\partial H}{\partial t} + \nabla \cdot Q - a \geq 0, & \text{in } \Omega_t^-, \\ H(t, \cdot) = 0, & \text{on } \partial\Omega, \\ H(0, \cdot) = H_0 \geq 0. \end{cases}$$

The ice thickness dynamics described by equation (16) outline the evolution of the ice sheet in both the ice-covered and ice-free regions.

4.6. Ice Velocity and Flux. To model the horizontal ice flow velocity U , we rely on Glen’s power law, which provides a relationship between the velocity and the surface elevation h . The horizontal ice flow velocity U can be expressed as:

$$(17) \quad U(\cdot, \cdot, z) = -2(\rho g)^{p-1} \left(\int_z^b A(s)(h-s)^{p-1} ds \right) |\nabla h|^{p-2} \nabla h + U_b.$$

This formulation assumes a frozen base where $U_b = 0$ [36], and the velocity field is determined by internal deformation of the ice. Substituting this into the equation for ice flux Q gives:

$$(18) \quad Q(\cdot, \cdot, z) = -2(\rho g)^{p-1} \left(\int_h^b \int_z^b A(s)(h-s)^{p-1} ds dz \right) |\nabla h|^{p-2} \nabla h + (h-b)U_b,$$

where the second term represents basal sliding and the first term accounts for internal deformation. As the surface elevation h approaches the free boundary, the gradient of the surface norm tends to infinity, causing challenges in modeling near the boundary.

The viscosity of the ice is described in terms of the Glen power law with ice softness coefficient $A(x, z)$ and exponent $2.8 \leq p \leq 5$, with values for p suggested by laboratory experiments [38].

4.7. Regularization of the Gradient. To address the issue of gradient blow-up near the boundary, we introduce the following transformation, as proposed in [37], which simplifies the analysis by regularizing the gradient near the boundary. Let:

$$(19) \quad H := u^{\frac{p-1}{2p}}.$$

By using this transformation, H remains positive in $[0, T] \times \Omega$ if and only if $u > 0$. Additionally, $H(t, x) = 0$ if and only if $u(t, x) = 0$, for all $x \in \Omega$ and $t \in [0, T]$. This transformation allows us to express the governing equations in a more tractable form.

Firstly, the time derivative of H becomes:

$$(20) \quad \frac{\partial H}{\partial t} = \frac{\partial}{\partial t} \left(|u|^{\frac{3p-1}{2p}} - 2u \right),$$

which provides the rate of change of ice thickness in terms of u . Secondly, the integral for ice softness $A(s)$ is transformed as:

$$(21) \quad \int_b^h A(s)(h-s)^p ds = u^{\frac{(p+1)(p-1)}{2p}} \int_0^1 A(x, b + u^{\frac{p-1}{2p}} s') (1-s')^p ds'.$$

This simplifies the relationship between u and the ice softness. Thirdly, we transform the gradient term:

$$(22) \quad |\nabla h|^{p-2} \nabla h = \left(\frac{p-1}{2p} \right)^{p-1} u^{-\frac{(p+1)(p-2)}{2p}} |\nabla u + u^{\frac{p+1}{2p}} \nabla b|^{p-2} u^{-\frac{p+1}{2p}} (\nabla u + u^{\frac{p+1}{2p}} \nabla b).$$

This reformulation regularizes the gradient of the ice surface elevation h in terms of u .

Now, for all $x \in \Omega$ and for $u = u(t, x) \in \mathbb{R}$, we define the vector field $\Phi : \Omega \times \mathbb{R} \rightarrow \mathbb{R}^2$ by:

$$(23) \quad \Phi(x, u) := -\frac{2p}{p-1} u^{\frac{p+1}{2p}} \nabla b.$$

Plugging Φ into the previous equation for the gradient term, we get:

$$(24) \quad |\nabla h|^{p-2} \nabla h = \left(\frac{p-1}{2p} \right)^{p-1} u^{-\frac{(p+1)(p-2)}{2p}} |\nabla u - \Phi|^{p-2} (\nabla u - \Phi).$$

For all $x \in \Omega$ and for $u \in \mathbb{R}$, we define another vector field $\Psi : \Omega \times \mathbb{R} \rightarrow \mathbb{R}^2$ by:

$$(25) \quad \Psi(x, u) := (h - b)U_b.$$

Additionally, we introduce the function $\tilde{a} : [0, T] \times \Omega \times \mathbb{R} \rightarrow \mathbb{R}$, which modifies the surface mass balance term $a(t, x)$ as:

$$(26) \quad \tilde{a}(t, x, u) := a(t, x, b + u^{\frac{p-1}{2p}}).$$

Now, substituting the transformed expressions into the ice flux equation, we obtain:

$$(27) \quad -Q = 2 \left(\rho g \frac{p-1}{2p} \right)^{p-1} \left[\int_0^1 A(x, b + u^{\frac{p-1}{2p}} s') (1-s')^p ds' \right] \\ \times |\nabla u - \Phi|^{p-2} (\nabla u - \Phi) - \Psi.$$

For simplicity, we define the function $\mu : \Omega \times \mathbb{R} \rightarrow \mathbb{R}$ as:

$$(28) \quad \mu(x, u) := 2 \left(\rho g \frac{p-1}{2p} \right)^{p-1} \left[\int_0^1 A(x, b + u^{\frac{p-1}{2p}} s') (1-s')^p ds' \right].$$

Substituting $\mu(x, u)$ into the flux equation yields:

$$(29) \quad -Q = \mu(x, u) |\nabla u - \Phi|^{p-2} (\nabla u - \Phi) - \Psi.$$

Finally, the ice thickness dynamics are described by the following unilateral boundary value problem:

$$(30) \quad \begin{cases} \frac{\partial}{\partial t} \left(|u|^{\frac{3p-1}{2p}-2} u \right) - \nabla \cdot (\mu(x, u) |\nabla u - \Phi|^{p-2} (\nabla u - \Phi) - \Psi) = \tilde{a}, & \text{in } \Omega_t^+, \\ \frac{\partial}{\partial t} \left(|u|^{\frac{3p-1}{2p}-2} u \right) - \nabla \cdot (\mu(x, u) |\nabla u - \Phi|^{p-2} (\nabla u - \Phi) - \Psi) \geq \tilde{a}, & \text{in } \Omega_t^-, \\ u(t, \cdot) = 0, & \text{on } \partial\Omega, \\ u(0, \cdot) = u_0 := H_0^{\frac{2p}{p-1}} \geq 0. \end{cases}$$

The terms within this formulation capture the key dynamics of ice thickness evolution. However, they also introduce complexities due to the power-law dependence on u , especially when u approaches zero.

4.8. Singularity in the Time Derivative Term. In this regularized model, the time derivative involves terms of the form:

$$\frac{\partial}{\partial t} \left(|u|^{\frac{3p-1}{2p}-2} u \right),$$

where p lies within the range $2.8 \leq p \leq 5$ [38]. For these values of p , this term can exhibit singularities when u becomes small. The power-law nature of this expression causes the time derivative to grow rapidly as u approaches zero, leading to potential numerical instability or non-physical oscillations in the solution. This sensitivity becomes particularly significant in physical regions where u is close to zero, making the equation challenging to solve accurately.

4.8.1. Epsilon-Regularization to Manage the Singularity. To address the singular behavior in the time derivative, we employ an *epsilon-regularization* approach [29]. This method ensures that the time derivative remains well-defined, even as u approaches zero. The regularized time derivative term is modified as follows:

$$\frac{\partial}{\partial t} \left((|u|^2 + \tilde{\epsilon})^{\frac{3p-1}{4p}-1} u \right),$$

where $\tilde{\epsilon} > 0$ is a small regularization parameter. This adjustment ensures that the term remains smooth and finite as $u \rightarrow 0$, thus preventing abrupt changes or blow-ups in the solution. The regularization technique allows the model to maintain numerical stability while capturing the essential dynamics of the system for $2.8 \leq p \leq 5$ [38].

With the mathematical model now stabilized through epsilon-regularization, we proceed to compute numerical solutions.

5. Numerical Computation

In this section, we use the method of manufactured to test the efficacy of this PINN-based approach for the 2D ice sheet problem with a known analytic solution to compare results against. The goal is to evaluate the accuracy and robustness of the proposed PINN-based solver using a known analytical solution that satisfies the governing PDE and obstacle constraint in Section 4.

We consider a radially symmetric, time-dependent manufactured solution defined as:

$$u_{\text{exact}}(r, t) = u_0(r) \cdot e^{-\gamma t},$$

where $u_0(r)$ is the following spatial profile:

$$u_0(r) = \begin{cases} 1 - (F(r) - G(r) + 1 - E(r)), & \text{for } r \leq r^*, \\ -\left(\frac{p}{p-1}\right) r^{*\frac{1}{p-1}} + (1 - r^*)^{1/(p-1)-1} (r - r^*) \\ \quad + 1 - r^{*\frac{p}{p-1}} - (1 - r^*)^{\frac{p}{p-1}} + 1 - E(r^*), & \text{for } r > r^*, \end{cases}$$

with $r = \sqrt{x^2 + y^2}$, $r^* = 0.75$, and the auxiliary functions defined as:

$$F(r) = r^{\frac{p}{p-1}}, \quad G(r) = (1 - r)^{\frac{p}{p-1}}, \quad E(r) = \frac{p}{p-1} r.$$

This solution is constructed to satisfy the obstacle condition $u \geq b(r)$, where the obstacle function $b(r)$ is given by:

$$b(r) = \begin{cases} 1 - \left(r^{\frac{p}{p-1}} - (1 - r)^{\frac{p}{p-1}} + 1 - \frac{p}{p-1} r \right), & \text{for } r \leq r^*, \\ 0, & \text{for } r > r^*. \end{cases}$$

The time decay $e^{-\gamma t}$ introduces temporal dynamics while preserving the structure of the solution. The initial and boundary conditions are constructed to match this exact solution.

We use this setup to evaluate the performance of the PINN model for different values of the nonlinearity parameter p . Accuracy is measured using the L^1 error between the predicted and exact solutions, and convergence is assessed via the total loss and its components. The optimization procedure minimizes the total loss, which consists of the following components:

1. **PDE Residual Loss:** This term penalizes deviations from the governing ice thickness equation, ensuring that the neural networks solution satisfies the PDE at

a set of discrete points. The discrete version of the loss is defined as:

$$(31) \quad \mathcal{L}_{\text{PDE}} = \frac{1}{N} \sum_{i=1}^N \left| \frac{\partial}{\partial t} \left((|u(x_i, t_i)|^2 + \tilde{\epsilon})^{\frac{3p-1}{4p}-1} u(x_i, t_i) \right) - \nabla \cdot (\mu(x_i, u(x_i, t_i)) |\nabla u(x_i, t_i) - \Phi|^{p-2} (\nabla u(x_i, t_i) - \Phi) - \Psi) \right|^2$$

where N is the total number of uniformly sampled discrete points (x_i, t_i) in the spatial-temporal domain $\Omega \times (0, T)$, and $\tilde{\epsilon}$ is a regularization parameter introduced to manage potential singularities.

2. **Obstacle Loss:** At discrete points, this loss becomes:

$$(32) \quad \mathcal{L}_{\text{obstacle}} = \frac{1}{N} \sum_{i=1}^N (\max(0, -u(x_i, t_i)))^2,$$

where N is the total number of sampled points (x_i, t_i) in the spatial domain Ω , ensuring the constraint $u \geq 0$ by penalizing values of u that fall below zero.

3. **Boundary Condition Loss:** For boundary points, this loss is computed as:

$$(33) \quad \mathcal{L}_{\text{boundary}} = \frac{1}{N_{\text{boundary}}} \sum_{i=1}^{N_{\text{boundary}}} (u(x_i, t_i) - g(x_i, t_i))^2,$$

where N_{boundary} is the total number of sampled points (x_i, t_i) on the boundary $\partial\Omega$, enforcing $u = 0$ on the boundary. In the PINN formulation, the obstacle loss is defined in unscaled form, while the strength of the penalty is controlled through the weighting parameter β in the total loss. This weighting plays the role of the penalty parameter $1/\varepsilon$ in the penalized PDE formulation.

4. **Initial Condition Loss:** This term enforces the correct initial profile $u(x, 0) = u_0(x)$ across the domain at time $t = 0$:

$$(34) \quad \mathcal{L}_{\text{initial}} = \frac{1}{N_{\text{initial}}} \sum_{i=1}^{N_{\text{initial}}} (u(x_i, 0) - u_0(x_i))^2,$$

where N_{initial} is the number of sampled points in the spatial domain Ω at the initial time $t = 0$. This term ensures that the network prediction aligns with the known initial condition of the system.

The total loss is then defined as a weighted sum of these individual losses:

$$\mathcal{L}_{\text{total}} = \alpha \mathcal{L}_{\text{PDE}} + \beta \mathcal{L}_{\text{obstacle}} + \gamma \mathcal{L}_{\text{boundary}} + \delta \mathcal{L}_{\text{initial}},$$

where α , β , γ , and δ are hyperparameters that control the relative importance of each loss term during training. In our experiments, we set $\alpha = \gamma = \delta = 1.0$, assigning equal weight to the PDE, boundary, and initial condition losses, while choosing a larger value $\beta = 4000.0$ to strongly enforce the obstacle constraint. This weighting reflects the unilateral nature of the obstacle condition and was found to produce stable convergence across all experiments.

In this section, we demonstrate the performance of the method for different values of the parameter p , which controls the nonlinearity in the ice thickness equation. Specifically, we present results for $p = 2.8$ and $p = 4.0$, focusing on the convergence of the total loss and the accuracy of the solution approximation.

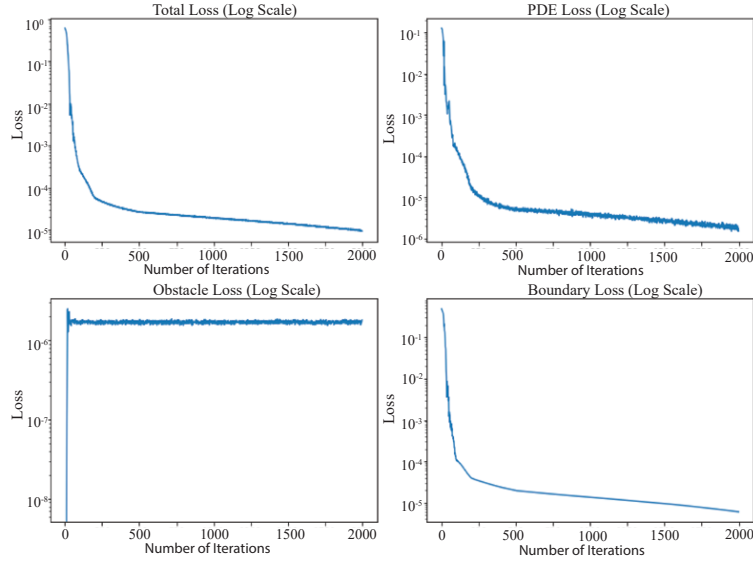


FIGURE 11. Loss components during training for $p = 2.8$. The total loss, PDE loss, obstacle loss and boundary condition loss are shown in log scale.

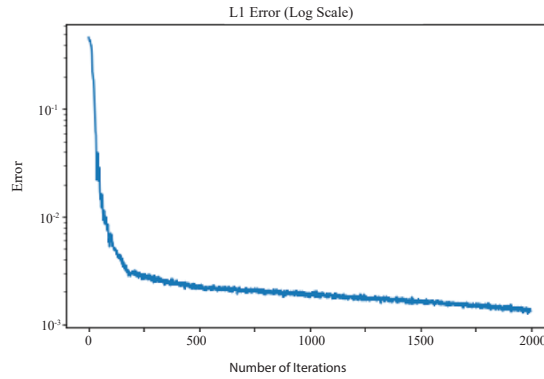


FIGURE 12. L^1 error during training for $p = 2.8$, shown in log scale.

5.1. Results for $p = 2.8$. Figure 11 shows the evolution of the total loss, PDE loss, obstacle loss, and boundary condition loss during the training process for $p = 2.8$. The results are displayed on a log scale over 2000 iterations. The rapid decrease in the total loss and the individual loss components indicates that the network efficiently approximates the solution while satisfying the obstacle and boundary conditions.

Figure 12 presents the evolution of the L^1 error during training, which measures the discrepancy between the neural network's predicted solution and the exact solution. The L^1 error decreases rapidly during the initial iterations and continues to improve gradually, demonstrating the accuracy of the method.

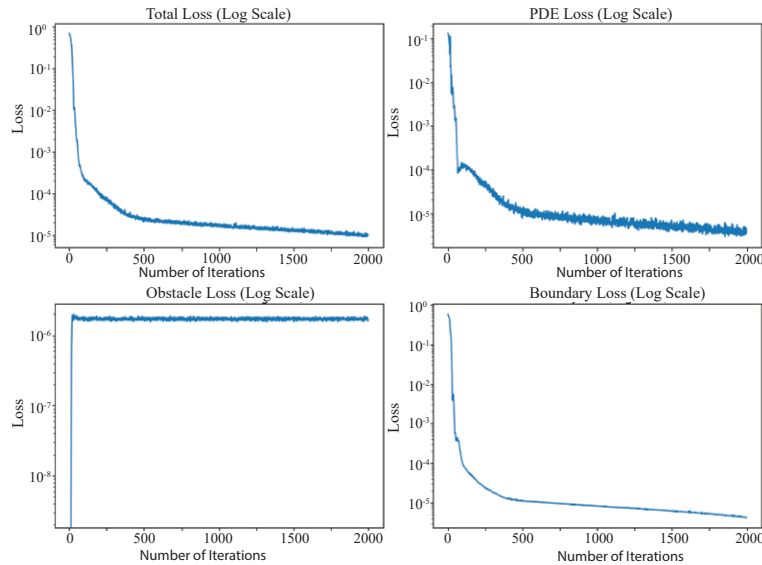


FIGURE 13. Loss components during training for $p = 4.0$. The total loss, PDE loss, obstacle loss, and boundary condition loss are shown in log scale.

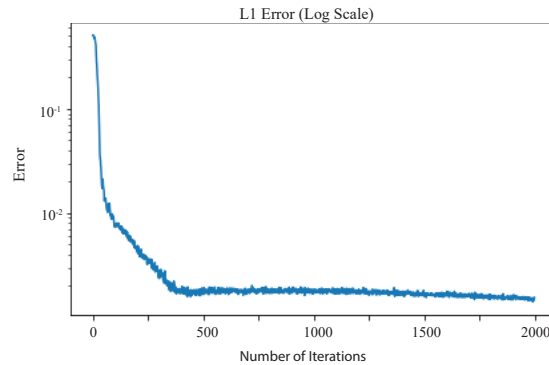


FIGURE 14. L^1 error during training for $p = 4.0$, shown in log scale.

5.2. Results for $p = 4.0$. For $p = 4.0$, Figure 13 illustrates the evolution of the total loss and the individual loss components over 2000 iterations. The loss curves are again shown in log scale. The network successfully minimizes the total loss, demonstrating stable convergence.

The evolution of the L^1 error for $p = 4.0$ is shown in Figure 14. The error decreases steadily, indicating that the network is learning an accurate approximation of the solution as the iterations progress.

5.3. Solution Comparison for $p = 4.0$. Figure 15 shows a comparison between the analytical solution and the numerical solution at three different time stamps: $t = 0.0$, $t = 0.5$, and $t = 1.0$, for $p = 4.0$.

At $t = 0.0$, the numerical solution closely matches the analytical solution, indicating that the initial condition is well captured by the model. As the system

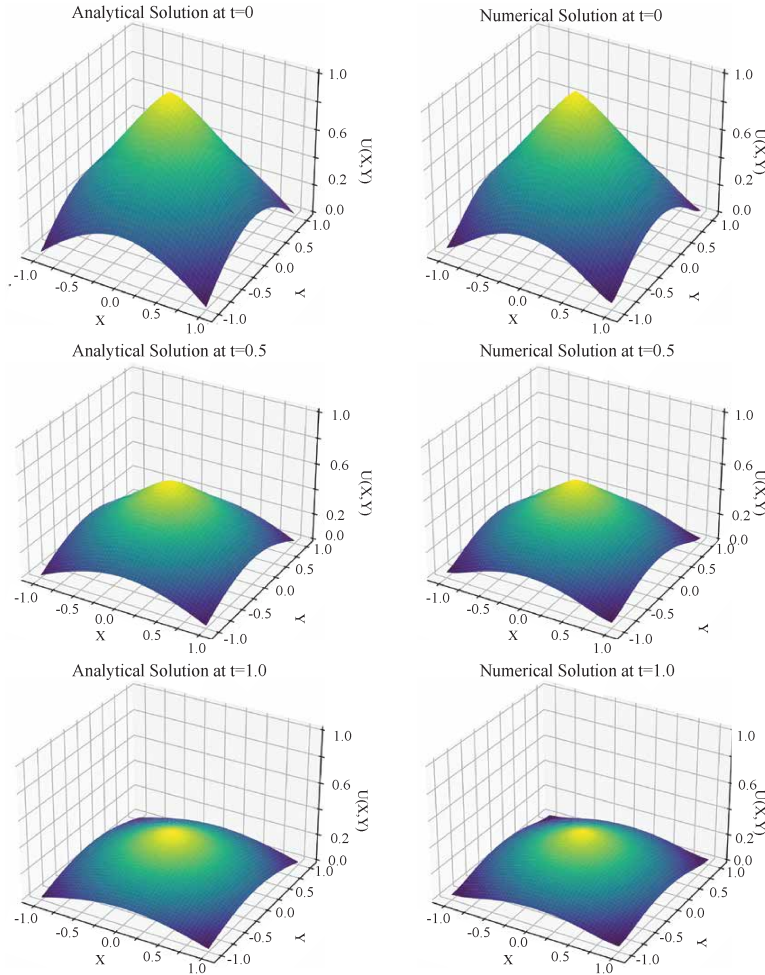


FIGURE 15. Comparison of analytical and numerical solutions at different time steps for $p = 4.0$. Top: $t = 0.0$, Middle: $t = 0.5$, Bottom: $t = 1.0$. Each row shows the analytical solution and the corresponding numerical approximation.

evolves to $t = 0.5$, the numerical solution continues to align well with the analytical solution, demonstrating the model's ability to follow the time-dependent dynamics. By $t = 1.0$, the numerical solution remains in good agreement with the analytical one, reflecting the robustness of the method throughout the temporal evolution.

6. Application to Devon Ice Cap Data

In this section, we apply the techniques developed earlier to a dataset derived from the Devon Ice Cap in the Canadian Arctic. The Devon Ice Cap, located on Devon Island, is one of the largest ice caps in the region, covering an area of approximately 14,010 km. Figure 16 shows a map of the Canadian Arctic, highlighting the location of the Devon Ice Cap.

The dataset used in this study includes ice thickness and surface elevation measurements from two different sources. The initial data comes from a 2000 airborne



FIGURE 16. Map of the Canadian Arctic region showing the location of the Devon Ice Cap on Devon Island, along with other significant ice caps and glaciers. The map provides geographical context for the study area. Source: Natural Resources Canada, <https://natural-resources.canada.ca/simple-science/making-for-lost-time-canadas-far-north/24224>.

radar survey, which provided a comprehensive view of the ice cap’s morphology [33]. An updated aerogeophysical survey, conducted in 2018, offers new insights into changes in ice thickness and underlying topography over the 18-year period [34].

6.1. Dataset Overview. The Devon Ice Cap spans approximately 14,010 km, with the main ice cap covering 12,050 km. The initial data for this study comes from a year 2000 airborne radar survey, which recorded maximum ice thicknesses of up to 750 meters [33]. In 2018, a follow-up aerogeophysical survey collected 4415 km of profile lines with line spacing ranging from 1.25 to 5 km [34]. We use the radar-derived digital elevation models (DEMs) to provide an initial condition for the ice thickness and the obstacle profile. We then train a PINN with this initial condition and obstacle to solve the PDE forward in time. Since μ is an unknown parameter in the model, we train multiple PINNs for different values of μ . These solution approximations are then used to propagate the initial condition forward in time for each value of μ and compare the predicted solution 18 years in the future to the 2018 data.

6.2. Numerical Solution Consistency and Smooth Transition. In this work, we developed a Physics-Informed Neural Network (PINN) model to solve the time-dependent PDEs governing ice thickness evolution. The 2000 ice thickness data [33] serves as the problems initial condition, and the target is the 2018 ice thickness data [34], obtained from aerogeophysical surveys. The network is trained to propagate the ice dynamics forward in time with the specific initial and bedrock data for the Devon Ice Cap.

In addition to ice thickness data, bedrock elevation data is incorporated as the obstacle condition for the model. This ensures that the predicted ice thickness does not fall below the physical constraints imposed by the underlying bedrock

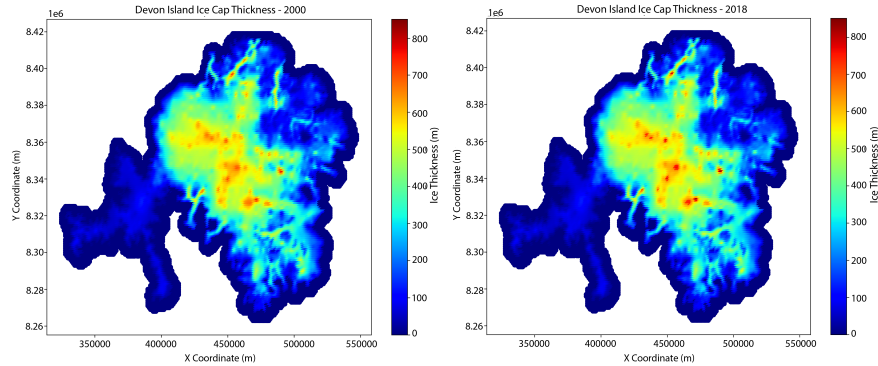


FIGURE 17. Comparison of Devon Ice Cap Ice Thickness: (Left) 2000 data used for initialization, (Right) 2018 data from aerogeophysical surveys. Both plots are scaled consistently for comparison.

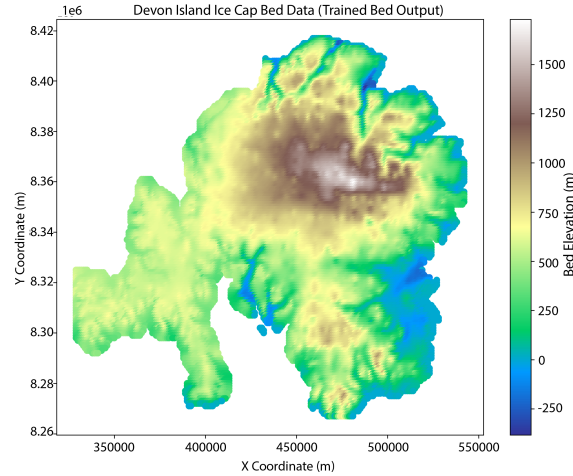


FIGURE 18. Bedrock elevation data for the Devon Ice Cap. This data is used as a constraint in the model, ensuring that the ice thickness evolution remains consistent with the physical boundaries of the bedrock.

topography, providing a more realistic simulation of the ice sheet's interaction with the terrain.

6.3. Neural Network Initialization and Training. Based on our prior work [31], random initialization of parameters in a PINN to solve this type of problem with complex bedrock topography can lead to poor training performance. To address this, we utilize a two-step training process for our PINN:

1) **Initialization:** The network is initially trained to match the 2000 ice thickness data (17). During this stage, the time is fixed at $t = 0$, and the network learns $u(x, y, 0)$, the ice thickness at the initial time. The same network architecture, which accepts (x, y, t) as inputs, is used throughout both phases. During the initialization phase, t is fixed at $t = 0$, effectively treating the problem as time-independent. This step allows the network to establish a stable and physically realistic initial solution. By doing so, we avoid the challenges associated with random initialization, such as

TABLE 2. Table of μ values and their associated L^1 errors for the 2018 Ice Thickness data at $p = 2.8$ and $p = 4$. The error trends help identify the optimal μ and p combination. For subsequent analysis, we will use $\mu = 5 \times 10^{-4}$ and $p = 4$.

| $\mu(x, u)$ | L^1 Error (p = 2.8) | L^1 Error (p = 4) |
|--------------------|-----------------------|---------------------|
| 5×10^{-6} | 0.0042 | 0.0031 |
| 5×10^{-5} | 0.0037 | 0.0024 |
| 5×10^{-4} | 0.00015 | 0.00001 |
| 0.005 | 0.0021 | 0.0015 |
| 0.05 | 0.0024 | 0.0019 |
| 0.5 | 0.0068 | 0.0051 |
| 1.0 | 0.014 | 0.011 |
| 2.0 | 0.53 | 0.41 |

erratic behavior and high error rates. Crucially, in this stage, the network is not solving the full time-dependent PDE; instead, it is focused on fitting the observed 2000 ice thickness data.

2) **Training for the Obstacle-PDE Problem:** Once the network has learned $u(x, y, 0)$, the network parameters are used as initial states for training the full time-dependent problem. The network then trains to solve the full obstacle-PDE, evolving the ice thickness over time based on the governing physics. This initialization ensures that the network begins the time-dependent training with a physically meaningful starting point, leading to faster convergence and reduced errors compared to random initialization.

6.4. Training Process and Solution Approximation. For the training process, we set the penalty parameters as $\alpha = 1.0$, $\beta = 4000$, $\gamma = 1.0$, and $\delta = 1.0$, determined by trial and error to balance the different components of the loss function. The relatively large value of β strongly enforces the obstacle constraint, while α , γ , and δ maintain the importance of the PDE residual, boundary, and initial condition losses, respectively.

The neural network architecture consisted of 15 hidden layers, each with 320 neurons, and was trained for 20,000 iterations to ensure accurate convergence. The training was conducted using the Adam optimizer with an initial learning rate of 5×10^{-3} . A multi-step learning rate scheduler was employed to gradually decrease the learning rate during training, which helped stabilize convergence and improve the optimization of the loss components. After training, we use the resulting PINN to predict the ice thickness in 2018. Since μ is a critical unknown parameter, we train PINNs to solve the PDE with different values of μ and compare the predicted solution in 2018 to find the value that best captures the progression of the thickness over time.

6.5. Analysis of Optimal μ Values and Corresponding L^1 Errors. Table 2 illustrates how the values of the parameters μ and p influence the level of agreement between the model's 2018 predictions and the 2018 data. Here, we quantify the L^1 error (at the fixed time 2018) between the predicted solution and the observed data, using $p = 2.8$ and $p = 4$. From this analysis we identify the best values of μ and p for future use (see the table caption).

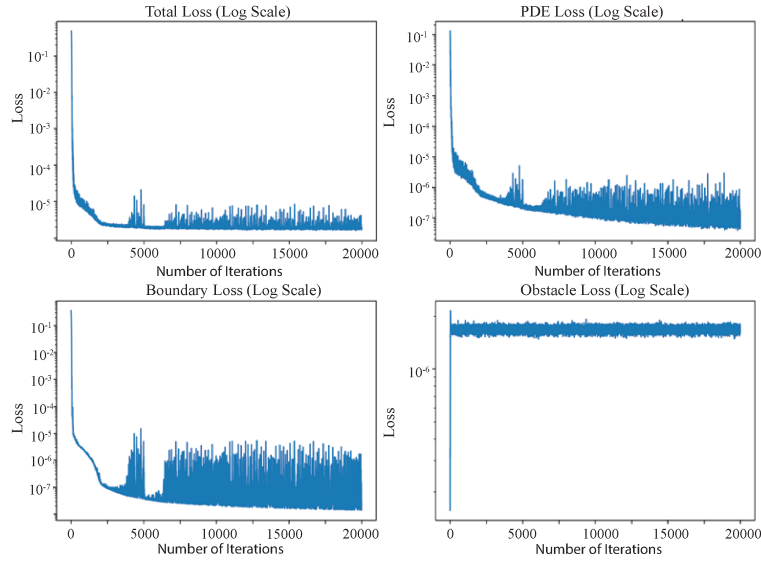


FIGURE 19. Loss Analysis for $p = 4$, using the optimal value $\mu = 5 \times 10^{-4}$ identified from Table 2. (a) Total Loss, (b) PDE Loss, (c) Boundary Condition Loss, and (d) Obstacle Loss over the training iterations (log scale).

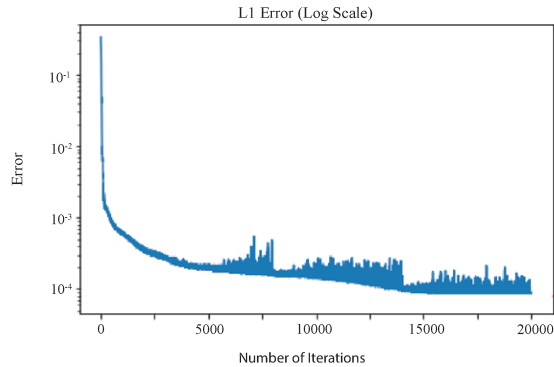


FIGURE 20. L^1 Error Analysis for the 2018 Dataset. The plot illustrates the discrepancy between the model's predictions and the observed ice thickness data for 2018, using the optimal values $\mu = 5 \cdot 10^{-4}$ and $p = 4$, providing a measure of the model's accuracy at this specific time point.

6.6. Loss and Error Analysis for $p = 4$. Here we analyze the training metrics for the $p = 4$ case. The loss components - total loss, PDE loss, obstacle loss, and boundary loss are illustrated in Figure 19. Figure 20 further illustrates the L^1 error between computed solution and the data at 2018 as a function of training iteration. This metric provides a direct measure of the accuracy of the model at this time point, demonstrating the effectiveness of the model in capturing the observed ice thickness distribution at a future time point that is unseen during the training.

The L^1 error for the 2018 dataset, as shown in Figure 20, reflects the precision of the model in predicting the observed ice thickness. This error metric, combined with the smooth convergence of all loss terms, further validates the robustness and precision of the model in capturing the dynamics of the Devon Ice Cap, particularly for the 2018 data.

7. Conclusion

This study presents a machine learning framework for solving parabolic obstacle problems in ice sheet modeling, with a focus on the Devon Ice Cap. By integrating traditional mathematical models with Physics-Informed Neural Networks (PINNs), our approach effectively captures the complex free-boundary conditions of grounded ice thickness dynamics. The framework was validated against data from 2000 and 2018, demonstrating its predictive capability for temporal changes in ice thickness and its scalability for different parameter settings.

Our results highlight the utility of PINNs in simulating ice sheet evolution under non-linear conditions, especially when data availability is limited. However, the models accuracy depends on parameter tuning and the quality of input data, such as the bedrock topography and surface elevation, which influences the network's ability to handle sharp gradients near the free boundary.

Future research could focus on enhancing the model by incorporating additional climate-related factors, such as temperature-dependent melting and basal sliding mechanisms. Additionally, applying this framework to other regions or ice caps would further validate its robustness and adaptability, potentially contributing to global climate models that forecast ice sheet contributions to sea level rise.

Overall, this study underscores the potential of machine learning approaches in geophysical modeling, bridging the gap between data-driven methods and traditional numerical modeling to address complex environmental challenges.

8. Acknowledgments

The authors acknowledge valuable discussions with Professor Ed Bueler, whose guidance was crucial in implementing the Shallow Ice Approximation (SIA) model on the Devon Ice Cap. This work was supported in part by the Research Fund of Indiana University.

Appendix A. Variational Inequality Formulation

For completeness, we present the variational inequality formulation associated with the ice thickness dynamics model. This formulation provides a framework to account for the free boundary dynamics of the ice-covered and ice-free regions.

The variational inequality formulation corresponding to the unilateral boundary value problem in Section 4 takes the following form: find $u \geq 0$ such that:

$$\begin{aligned} & \int_0^T \int_{\Omega} \left(\frac{\partial}{\partial t} \left(|u|^{\frac{3p-1}{2p}-2} u \right) \right) (v - u) \, dx \, dt \\ & - \int_0^T \int_{\Omega} \mu(x, u) |\nabla u - \Phi|^{p-2} (\nabla u - \Phi) \cdot \nabla (v - u) \, dx \, dt \\ & \geq \int_0^T \int_{\Omega} \tilde{a}(t, x, u) (v - u) \, dx \, dt + \int_0^T \int_{\Omega} \Psi(x, u) \cdot \nabla (v - u) \, dx \, dt, \end{aligned}$$

for all test functions $v \geq 0$ such that $v(t, x) \geq 0$ for almost every $(t, x) \in (0, T) \times \Omega$.

The boundary conditions are:

$$u(t, \cdot) = 0 \quad \text{on} \quad \partial\Omega, \quad \forall t \in (0, T),$$

with the initial condition:

$$u(0, \cdot) = u_0 = H_0^{\frac{2p}{p-1}} \geq 0.$$

In this formulation, the time-dependent term captures the evolution of ice thickness u , while the other terms account for internal deformation, advection, and basal sliding effects. This framework ensures that weak solutions can address the irregular boundary characteristics inherent in the ice sheet dynamics model.

Acknowledgments

This research was supported by Indiana University.

References

- [1] G. Jovuet and E. D. Bueler, Steady, shallow ice sheets as obstacle problems: Well-posedness and finite element approximation, *SIAM J. Appl. Math.*, 72 (2012), 1292–1314.
- [2] P. Piersanti and R. Temam, On the dynamics of grounded shallow ice sheets: Modeling and analysis, *Adv. Nonlinear Anal.*, 12 (2023), 20220280.
- [3] B. Geshkovski, *Obstacle Problems: Theory and Applications*, Master’s Thesis, DeustoTech, Bilbao, 2024.
- [4] J. F. Rodrigues, *Obstacle Problems in Mathematical Physics*, Elsevier, Amsterdam, 1987.
- [5] I. Harari and T. J. R. Hughes, Galerkin/least-squares finite element methods for the reduced wave equation with non-reflecting boundary conditions in unbounded domains, *Comput. Methods Appl. Mech. Engrg.*, 98 (1992), 411–454.
- [6] K. Xia and H. Yao, A Galerkin/least-squares finite element formulation for nearly incompressible elasticity and Stokes flow, *Appl. Math. Model.*, 31 (2007), 513–529.
- [7] E. Burman, P. Hansbo, M. G. Larson, and R. Stenberg, Galerkin least squares finite element method for the obstacle problem, *Comput. Methods Appl. Mech. Engrg.*, 313 (2017), 362–374.
- [8] R. Kornhuber, Monotone multigrid methods for elliptic variational inequalities II, *Numer. Math.*, 72 (1996), 481–499.
- [9] R. Kornhuber, Monotone multigrid methods for elliptic variational inequalities I, *Numer. Math.*, 69 (1994), 167–184.
- [10] D. Yuan and X. Cheng, An iterative algorithm based on the piecewise linear system for solving bilateral obstacle problems, *Int. J. Comput. Math.*, 89 (2012), 2374–2384.
- [11] T. Führer, First-order least-squares method for the obstacle problem, *Numer. Math.*, 144 (2020), 55–88.
- [12] K. Majava and X.-C. Tai, A level set method for solving free boundary problems associated with obstacles, *Int. J. Numer. Anal. Model.*, 1 (2004), 157–172.
- [13] Q. Ran, X. Cheng, and S. Abide, A dynamical method for solving the obstacle problem, *Numer. Math. Theory Methods Appl.*, 13 (2020), 353–371.
- [14] S. H. Rudy, J. N. Kutz, and S. L. Brunton, Deep learning of dynamics and signal-noise decomposition with time-stepping constraints, *J. Comput. Phys.*, 396 (2019), 483–506.
- [15] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.*, 378 (2019), 686–707.
- [16] W. E and B. Yu, The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems, *Commun. Math. Stat.*, 6 (2018), 1–12.
- [17] N. Winovich, K. Ramani, and G. Lin, ConvPDE-UQ: Convolutional neural networks with quantified uncertainty for heterogeneous elliptic partial differential equations, *J. Comput. Phys.*, 394 (2019), 263–279.
- [18] R. K. Tripathy and I. Bilionis, Deep UQ: Learning deep neural network surrogate models for high dimensional uncertainty quantification, *J. Comput. Phys.*, 375 (2018), 565–588.
- [19] Y. Khoo and L. Ying, SwitchNet: A neural network model for forward and inverse scattering problems, *SIAM J. Sci. Comput.*, 41 (2019), A3182–A3201.
- [20] J. Huang, C. Wang, and H. Wang, Adaptive learning on the grids for elliptic hemivariational inequalities, arXiv:2104.04881, 2021.

- [21] X. Cheng, S. Xing, X. Wang, and K. Liang, A deep neural network-based method for solving obstacle problems, *Nonlinear Anal. Real World Appl.*, 72 (2023), 103864.
- [22] R. Scholz, Numerical solution of the obstacle problem by the penalty method, *Computing*, 32 (1984), 297–306.
- [23] J. Bamber, Greenland 5 km DEM, ice thickness, and bedrock elevation grids, NASA NSIDC DAAC, Boulder, CO, 2001.
- [24] R. Greve and H. Blatter, *Dynamics of Ice Sheets and Glaciers*, Springer-Verlag, Berlin, 2009.
- [25] G. Jouvet, J. Rappaz, E. Bueler, and H. Blatter, Existence and stability of steady-state solutions of the shallow-ice-sheet equation by an energy-minimization approach, *J. Glaciol.*, 57 (2011), 345–354.
- [26] A. Paszke et al., PyTorch: An imperative style, high-performance deep learning library, *Adv. Neural Inf. Process. Syst.*, 32 (2019).
- [27] N. Calvo et al., On a doubly nonlinear parabolic obstacle problem modelling ice sheet dynamics, *SIAM J. Appl. Math.*, 63 (2002), 683–707.
- [28] D. Kinderlehrer and G. Stampacchia, *An Introduction to Variational Inequalities and Their Applications*, SIAM, Philadelphia, 2000.
- [29] L. C. Evans, *Partial Differential Equations*, 2nd ed., AMS, Providence, RI, 2010.
- [30] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, Adaptive activation functions accelerate convergence in physics-informed neural networks, *J. Comput. Phys.*, 404 (2020), 109136.
- [31] K. Chawla, W. Holmes, and R. Temam, From obstacle problems to neural insights: Feed-forward neural network modeling of ice thickness, *Int. J. Numer. Anal. Model.*, 22 (2025), 1–20.
- [32] K. Chawla, W. Holmes, and R. Temam, From obstacle problems to neural insights: Feedforward neural network modeling of ice thickness, *arXiv:2407.19529*, 2024.
- [33] J. Dowdeswell, T. Benham, M. Gorman, D. Burgess, and M. Sharp, Form and flow of the Devon Island Ice Cap, Canadian Arctic, *J. Geophys. Res.*, 109 (2004).
- [34] A. Rutishauser, D. D. Blankenship, D. A. Young, N. S. Wolfenbarger, L. H. Beem, M. L. Skidmore, A. Dubnick, and A. S. Criscitiello, Radar sounding survey over Devon Ice Cap indicates the potential for a diverse hypersaline subglacial hydrological environment, *The Cryosphere*, 16 (2022), 379–395.
- [35] H. Brézis, Problèmes unilatéraux, *J. Math. Pures Appl.* (9), 51 (1972), 1–168.
- [36] R. Greve and H. Blatter, *Dynamics of Ice Sheets and Glaciers*, Springer-Verlag, Berlin Heidelberg, 2009.
- [37] P. A. Raviart, Sur la résolution et l’approximation de certaines équations paraboliques non linéaires dégénérées, *Arch. Ration. Mech. Anal.*, 25 (1967), 64–80.
- [38] D. L. Goldsby and D. L. Kohlstedt, Superplastic deformation of ice: Experimental observations, *J. Geophys. Res.*, 106 (2001), 11017–11030.
- [39] A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, Adaptive activation functions accelerate convergence in physics-informed neural networks, *J. Comput. Phys.*, 404 (2020), 109136.

Department of Mathematics and the Institute for Scientific Computing and Applied Mathematics, Indiana University, Indiana 47405, USA

E-mail: kapchaw@iu.edu

Department of Mathematics and Cognitive Science Program, Indiana University, Indiana 47405, USA

E-mail: wrholmes@iu.edu

URL: <https://math.indiana.edu/about/faculty/holmes-william.html>