Proceedings of the 12th International and 52nd National Conference on Fluid Mechanics and Fluid Power (FMFP)
December 19-21, 2025, Nirma University, Gujarat, India

FMFP2025–CFD–154

Depth-Averaged Flow Modelling of Snow Avalanches Using Physics-Informed Neural Networks: Height Field Prediction in an Idealised Inclined-Plane Setup

Pradyumn Singh Sikarwar[1], Atharva D. Joshi[2], Vishal Sharma[1] and Gaurav Bhutani[1,*]

[1]School of Mechanical and Materials Engineering, Indian Institute of Technology Mandi, Himachal Pradesh 175075, India
[2] Department of Mathematics, Indian Institute of Science Education and Research (IISER) Bhopal, Madhya Pradesh 462066, India

Emails: s24047@students.iitmandi.ac.in; atharvajoshi23@iiserb.ac.in; d22051@students.iitmandi.ac.in; gaurav@iitmandi.ac.in

ABSTRACT

This work applies Physics-Informed Neural Networks (PINNs) to model snow avalanche dynamics by solving depth-averaged equations for flow height (or depth) on an inclined plane. The PINN framework directly integrates the governing continuity equation, initial, and boundary conditions into the loss function, enabling a data-efficient, mesh-free solution. A systematic hyperparameter study was conducted to optimise model performance, examining loss-function weights, collocation density, and network architecture. The analysis highlights how appropriate weighting of the boundary conditions and a balanced network configuration significantly improve predictive accuracy and training stability. The results demonstrate that machine learning approaches can effectively capture the complex behaviour of depth-averaged flows when combined with physical constraints, offering a robust alternative to traditional numerical methods.

Keywords: Snow avalanche dynamics, Depth-averaged equations, Physics-informed neural networks, Machine Learning.

I. INTRODUCTION

Snow avalanches are highly destructive natural hazards that pose significant risks to human life and infrastructure. Over the past few decades, modelling has emerged as the preferred approach for predicting avalanche flows, primarily due to the considerable risks associated with direct experimentation or observation. However, modelling these flows is not straightforward, as avalanches are highly unsteady, have a heterogeneous composition, and involve strong coupling between fluid and granular components. As a result, numerous approaches have been proposed in the literature to capture their behaviour under varying conditions.

One widely used method for modelling avalanche flows is the depth-averaged approach, popularised in the field by Savage and Hutter (1989). This method is particularly effective for shallow flows, where the flow spreads predominantly in the longitudinal and lateral directions, and the vertical thickness is relatively small (Sharma and Bhutani, 2025). By averaging the governing equations over the flow depth, the problem is simplified from three dimensions to two, making it more tractable for analytical and numerical treatment.

Despite this simplification, the resulting coupled system of continuity and momentum equations remains nonlinear and hyperbolic, posing significant challenges for numerical solvers. Nonlinear terms can lead to the growth of instabilities, while the hyperbolic nature of the equations results in wave propagation, shock formation and discontinuities (LeVeque, 2002). These effects make standard numerical schemes insufficient and necessitate specialised shock-capturing methods. Furthermore, accurately predicting the height profile is critical, as it informs the development of mitigation strategies and the design of protective structures along avalanche paths (Boensch et al., 2015).

In recent years, Physics-Informed Neural Networks (PINNs) have emerged as a powerful alternative for solving partial differential equations (Ellath Meethal, 2023). PINNs can learn solutions that satisfy governing equations, initial conditions, and boundary conditions by embedding physical laws directly into the training process. This approach bypasses the need for traditional discretisation schemes, offering greater flexibility in handling complex domains, parameter estimation, and sparse or noisy data. Given these capabilities, PINNs provide a promising framework for studying avalanche dynamics, particularly in scenarios where conventional CFD-based methods encounter limitations (Zhao et al., 2024).

The present work modifies a two-dimensional avalanche flow problem over a simplified one-dimensional inclined surface. A parabolic initial snow mass is assumed to slide downslope under the influence of gravity. The continuity equation governing the evolution of snow height (or flow depth) is solved using PINNs, with the velocity field prescribed analytically. The predicted height profiles are compared against analytical solutions available in the literature, and the accuracy of the PINN-based solution is evaluated through quantitative error analysis. Furthermore, parametric studies are carried out to investigate the influence of various machine learning hyperparameters on the solution quality.

The paper is organised as follows. In Section II, governing depth-averaged equations, which are solved in this work, are presented, followed by the analytical solution for a parabolic cap case. The section also discusses physics-informed neural networks as implemented in this work. Next,

in Section III, PINNs-predicted results for the evolution of avalanche flow are presented and compared with the analytical solution. Additionally, a hyperparametric study is also discussed. Conclusions, limitations of the present work and scope of future work are given in Section IV, highlighting the application of physics-informed neural networks in engineering applications.

II. METHODOLOGY

A. Governing equations

This section presents the non-dimensionalised depth-averaged model equations for shallow flows obtained from integrating the flow variables along a direction normal to local bed topography. The scaling variables for non-dimensionalising the equation are considered as:

$$h = \frac{\bar{h}}{H}, \quad u' = \frac{\bar{u'}}{\sqrt{gL}}, \quad x = \frac{\bar{x}}{L}, \quad t = \frac{\bar{t}}{\sqrt{\frac{L}{g}}},$$

where the barred terms represent the dimensional quantities. The scaling quantities in x and y directions are taken differently: L is the typical spreading of the initial slide, and H is the typical depth. g represents acceleration due to gravity.

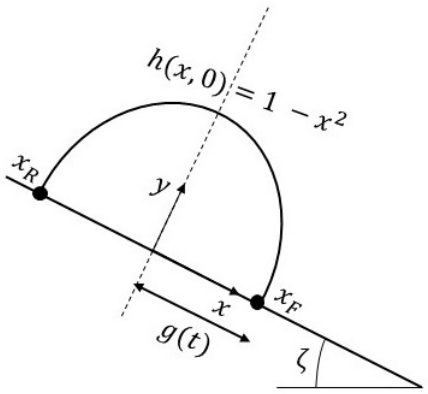

Figure 1: **Initial position of parabolic cap setup.**

The non-dimensionalised depth-averaged continuity equation in 2D is given as:

$$\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) = 0, \tag{1}$$

where $h(x,t)$ is the snow depth along the normal direction y in the local coordinate system. The local coordinate x is along the downslope direction, as shown in Figure 1. Here, $u(x,t)$ is depth-averaged velocity in x direction defined as $u = \int_0^h u' dy$.

The non-dimensionalised depth-averaged momentum equation in x direction is obtained as:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \sin\zeta - \tan\delta\cos\zeta - \beta\frac{\partial h}{\partial x}, \tag{2}$$

where ζ is the angle of inclination from horizontal, and δ is the bed friction angle. In the above equation β is defined as:

$$\beta = \epsilon k_{ap}\cos\zeta. \tag{3}$$

Here, ϵ is a non-dimensional number, representing the ratio of the height of the flow to its width in the flow plane, and k_{ap} is the earth pressure coefficient.

The above Equations (1) and (2) are solved for flow parameters $h(x,t)$ and $u(x,t)$ with a specified initial height profile depicting the initial mass and a zero initial velocity since an avalanche typically starts from rest. These are represented as:

$$h(x,0) = h_0(x), \qquad u(x,0) = 0.$$

Furthermore, the boundary conditions imposed in the governing system of equations are:

$$h(x,t) = h_F(t) = 0 \;\; at \;\; x = x_F,$$

and

$$h(x,t) = h_R(t) = 0 \;\; at \;\; x = x_R.$$

B. Analytical solution

The governing equations and the initial and boundary conditions discussed in the previous section were solved for a parabolic cap situation in this work. The initial snow profile in this case was given as $h(x,0) = 1 - x^2$, as shown in Figure 1. The resulting analytical expressions for $h(x,t)$ and $u(x,t)$, derived using the similarity method mentioned in Savage and Hutter (1989), are expressed as:

$$h(x,t) = \frac{1}{g(t)}\left[1 - \left(\frac{x - \frac{at^2}{2}}{g(t)}\right)^2\right], \tag{4}$$

and

$$u(x,t) = \left[\frac{2}{g(t)}\left(g(t) - 1\right)\right]^{1/2}\left(\frac{x - \frac{at^2}{2}}{g(t)}\right) + at. \tag{5}$$

Here, $g(t)$ represents one-half of the lateral spread of the snow avalanche heap at any given time, as shown in Figure 1, and a is a known value expressed as $a = \sin\zeta - \tan\delta\cos\zeta$.

C. Physics-Informed Neural Networks

Physics-informed neural networks (PINNs) are a class of deep learning models, typically implemented as fully connected feed-forward neural networks, that solve partial differential equations (PDEs) by directly embedding physical constraints, such as governing equations, initial conditions, and boundary conditions, into the training process. Collocation points are selected throughout the spatial-temporal domain, where the residuals of the PDE and the deviation from the prescribed initial and boundary values are evaluated. Automatic differentiation is used to compute the required derivatives, ensuring physical consistency, see Fig. 2.

The PINN is trained by minimising a composite loss function, typically formulated as:

$$\mathcal{L}_{\text{total}} = w_{\text{pde}}\mathcal{L}_{\text{PDE}} + w_{\text{ic}}\mathcal{L}_{\text{IC}} + w_{\text{bc}}\mathcal{L}_{\text{BC}}, \tag{6}$$

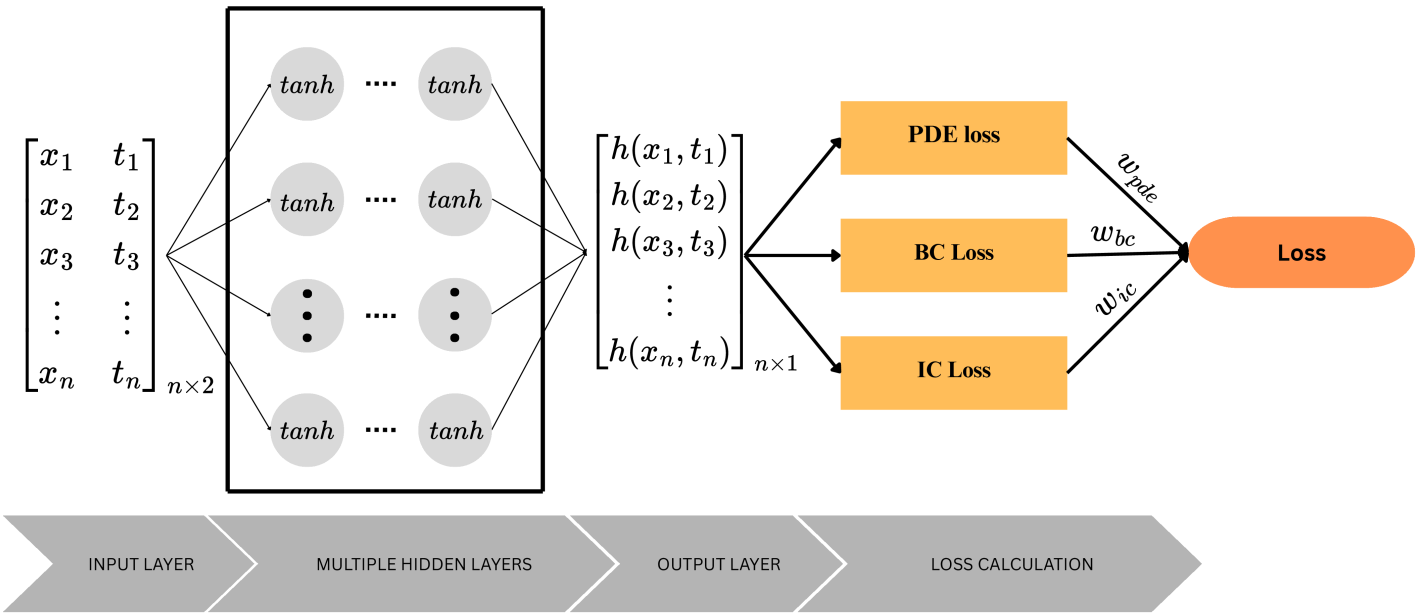

Figure 2: **PINN workflow used in this study: inputs (x,t) pass through multiple hidden layers to predict $h(x,t)$, with PDE, IC and BC losses combined into the total loss.**

where $\mathcal{L}_{\mathrm{PDE}}$, $\mathcal{L}_{\mathrm{IC}}$, and $\mathcal{L}_{\mathrm{BC}}$ denote the residual losses for the governing equations, initial, and boundary conditions, respectively, and w_{pde}, w_{ic}, and w_{bc} are the corresponding weighting coefficients.

Training is achieved by minimising the total loss using gradient-based optimisation algorithms. The common choice includes the Adam optimiser, which is well-suited for handling complex and high-dimensional loss.

D. Approach

The fluid height $h(x,t)$ in the present paper was obtained by enforcing the depth-averaged continuity equation:

$$\frac{\partial h}{\partial t} + \frac{\partial\big(u_{\mathrm{a}}(x,t)\,h\big)}{\partial x} = 0,$$

where $u_{\mathrm{a}}(x,t)$ is the known analytical velocity profile, given in Equation (5). By substituting u_{a} into the above PDE, PINNs only had to learn h. Automatic differentiation of the network output $h_\theta(x,t)$, where θ denotes the trainable parameters of the neural network, with respect to x and t, in this work, resulted in the residual of the continuity equation at each point.

To construct the training dataset for the PINNs framework, three sets of points in the space-time domain were uniformly sampled. These included the collocation points $\{(x_{\mathrm{col}},t_{\mathrm{col}})\}$ to enforce the PDE residual, the initial-condition points $\{(x_{\mathrm{ic}},0)\}$ for the prescribed $h(x,0)$ value, and the boundary-condition points $\{(x_{\mathrm{bc}},t)\}$ along the domain edges, enforcing constant boundary values.

In accordance with Equation (6), the total loss $\mathcal{L}$ is formulated as a weighted combination of the governing equation residual, the initial condition mismatch, and the boundary condition mismatch. Specifically, in this work, these contributions are defined as:

$$\mathcal{L}_{\mathrm{PDE}} = \frac{1}{N_{\mathrm{col}}}\sum_i \Big|\partial_t h_\theta(x_i,t_i) + \partial_x\big[u_{\mathrm{a}}(x_i,t_i)\,h_\theta(x_i,t_i)\big]\Big|^2, \tag{7}$$

$$\mathcal{L}_{\mathrm{IC}} = \frac{1}{N_{\mathrm{ic}}}\sum_j \big|h_\theta(x_j,0) - h_{\mathrm{true}}(x_j,0)\big|^2, \tag{8}$$

and

$$\mathcal{L}_{\mathrm{BC}} = \frac{1}{N_{\mathrm{bc}}}\sum_k \big|h_\theta(x_{\mathrm{bc}},t_k) - h_{\mathrm{bc}}(t_k)\big|^2. \tag{9}$$

The weights w_{pde}, w_{ic}, and w_{bc} act as tuning parameters to balance the relative importance of the three loss components.

III. RESULTS AND DISCUSSION

The sensitivity of the proposed model to its principal hyperparameters was first investigated. The objective was to determine how parameter variations influenced prediction accuracy and the computational cost. Accuracy was quantified using the root-mean-square (RMS) error between the predicted non-dimensional height $h(x,t)$ and the analytical solution, computed over a uniform 400×100 space–time grid covering the entire domain, which represents the overall deviation of the PINN prediction from the analytical height across all grid points. The training was repeated five times for each configuration, and the RMS values reported are the averages of these runs to ensure consistency. Training times were recorded as wall–clock seconds for 10,000 steps on an NVIDIA GeForce RTX 3050 GPU.

After the optimum set of hyperparameters was identified, the performance of the PINN model was evaluated. Predicted

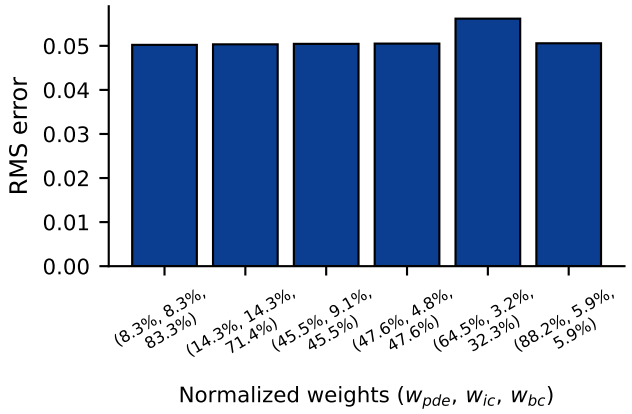

Figure 3: **Variation of RMS error for different loss–weight combinations** $(w_{\mathrm{pde}}, w_{\mathrm{ic}}, w_{\mathrm{bc}})$**, expressed as percentages of the total loss weight, after** $10{,}000$ **training steps.**

avalanche flow heights were compared against the corresponding analytical profiles, and their discrepancies were quantified. Once the most favourable training configuration was established, this allowed for a direct assessment of the model's predictive capability. In addition, the evolution of the training loss with respect to training steps was also analysed

A. **Hyperparameter Tuning**

The investigation commenced with a variation of the loss–function weights $(w_{\mathrm{pde}}, w_{\mathrm{ic}}, w_{\mathrm{bc}})$, which control the relative emphasis on the governing equation residual, initial condition, and boundary condition terms. As shown in Fig. 3, the variation in RMS error across different weight combinations was not substantial, indicating that the model achieved comparable accuracy over a range of settings. However, it was observed during training that assigning a relatively higher weight to the PDE term caused the network to focus intensely on satisfying the governing equation in the domain interior, leading to slower error reduction at the boundaries. This imbalance gradually propagated into the solution, causing the RMS error to rise with iterations. Based on these observations, the final weights were chosen as $(8.3\%, 8.3\%, 83.4\%)$, providing a balanced emphasis that ensured accurate enforcement of boundary conditions while maintaining consistency in the domain.

The influence of the number of collocation points, N_{col}, was assessed regarding prediction accuracy and computational cost. As illustrated in Fig. 4, the RMS error remained nearly constant across the tested range, indicating that increasing collocation density did not lead to significant gains in accuracy. A similar trend was observed for the training time, showing slight variation with N_{col}. These results suggest that the model performance is largely insensitive to further increases in collocation density, and a moderate choice of N_{col} is sufficient to achieve reliable predictions without added computational effort.

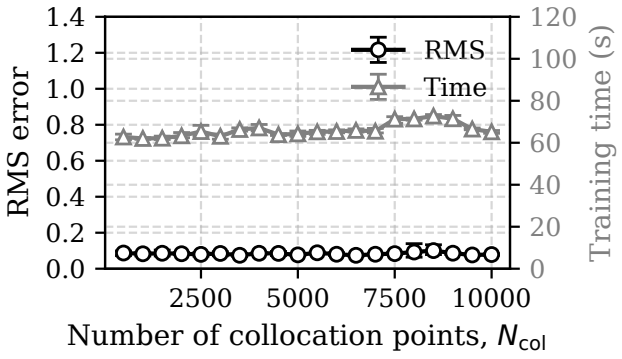

Figure 4: **RMS error and training time versus collocation points** N_{col}**. Mean** $\pm$ **min–max over five runs.**

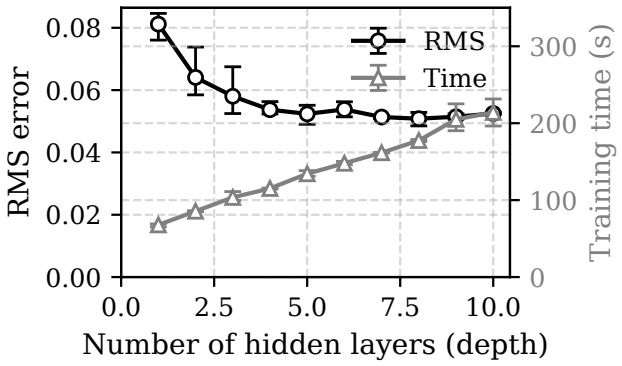

Figure 5: **RMS error and training time versus number of hidden layers. Mean** $\pm$ **min–max over five runs (10,000 steps).**

The effect of the number of hidden layers was then examined. As shown in Fig. 5, increasing the depth from a very shallow architecture to a moderate size substantially improved accuracy, with the error reducing steadily up to around five layers. Beyond this point, however, the gains became negligible, and in some cases, training stability slightly deteriorated, indicating that additional layers offered little benefit while increasing the risk of over-parameterisation. The computational cost grew nearly linearly with depth. This reflects the extra burden of propagating gradients through larger networks, making excessively deep architectures inefficient for this problem. A network with five hidden layers provided the most balanced configuration, delivering strong accuracy improvements without unnecessary computational expense.

The effect of network width, expressed as the number of neurons per layer, is shown in Fig. 6. The RMS error drops only at very small widths, after which it flattens out and remains almost unchanged beyond four neurons. This shows that increasing the width has little influence on the model accuracy for the present problem. The training time stays nearly constant across the tested range, with only

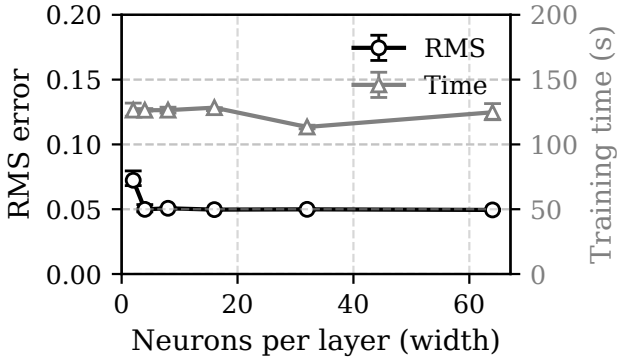

Figure 6: **RMS error and training time versus neurons per layer. Mean ± min–max over five runs.**

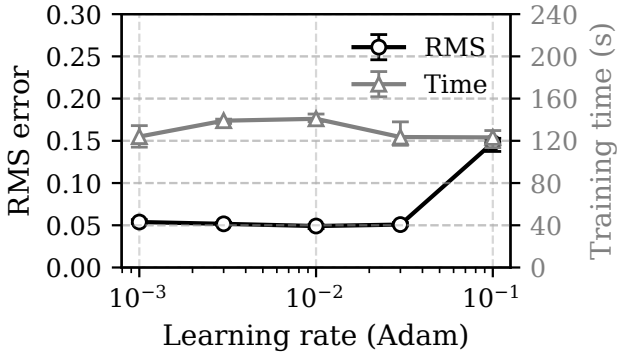

Figure 7: **RMS error and training time versus learning rate (Adam). Mean ± min–max over five runs (10,000 steps).**

minor variations. Hence, a modest width can achieve stable performance without adding computational cost.

The effect of the learning rate on training performance was then explored. Figure 7 shows that the model performed best at an intermediate learning rate. Very low values led to slower convergence and slightly higher final errors, while excessively high values caused instability and a greater loss of accuracy. The optimal range, around 10^{-2}, provided the most reliable results for achieving low errors. Interestingly, the training time did not vary significantly across the stable range of learning rates, indicating that the primary influence of this hyperparameter lies in determining whether the optimiser converges smoothly rather than in reducing computational cost.

The influence of the number of training steps was then analysed to determine when the model reaches a satisfactory level of convergence. Figure 8 shows that the RMS error decreased steadily in the initial training phase, achieving most accuracy gains within the first few thousand steps. Beyond this point, improvements became marginal, and in some cases, the error slightly fluctuated, suggesting that further training provided little additional benefit. The training

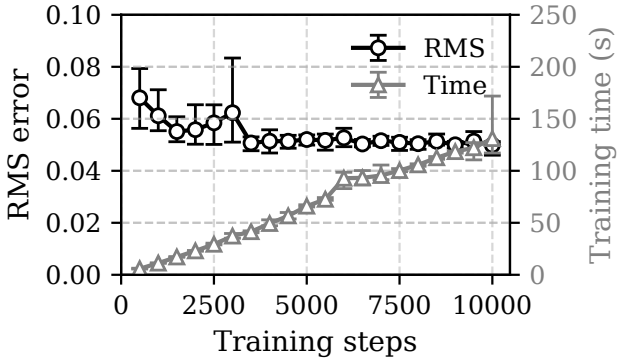

Figure 8: **RMS error and training time versus training steps. Mean ± min–max over five runs.**

time, presented in Fig. 8, increased linearly with the number of steps, as expected. These results indicate that sufficient iterations are required for the model to converge, but extending training far beyond this point offers slight improvement in accuracy while unnecessarily increasing computational cost.

B. Model Performance

Based on the hyperparameter studies discussed above, the final configuration of the PINNs was selected to balance accuracy and computational cost. The number of collocation points was set to $N_{\mathrm{pde}} = 2000$ for the PDE residual and $N_{\mathrm{ic}} = N_{\mathrm{bc}} = 1000$ for the initial and boundary conditions. The loss-function weights percentage were chosen as $(w_{\mathrm{pde}}, w_{\mathrm{ic}}, w_{\mathrm{bc}}) = (8.3\%, 8.3\%, 83.4\%)$, with five hidden layers of eight neurons each, an Adam learning rate of 0.01 and a total of 5000 training steps.

Figure 9 shows the evolution of the total training loss and its components over the training steps. The steady reduction of the PDE, IC, and BC losses demonstrates the network's ability to satisfy the governing equations simultaneously and impose conditions, achieving convergence within the prescribed number of iterations.

The predictive performance of the trained model is illustrated in Fig. 10, which compares the PINN-predicted height profiles $h(x,t)$ against the analytical solution at several time instances. The model captured the temporal evolution of the profile with high fidelity, correctly reproducing the propagation and decay of the initial condition over the domain. These results confirm that the optimised PINN can robustly approximate the solution of the target PDE under the given setup.

The accuracy of the trained model was evaluated over time by computing the RMS error between the PINN predictions and the analytical solution, as shown in Fig. 11. The error is lowest at $t = 0$, when the initial condition dominates. It then rises to a peak near $t \approx 0.5$ as the profile evolves, before steadily decreasing as the network tracks the PDE dynamics. Later times, the error stays low with only a slight increase, likely due to minor boundary effects.

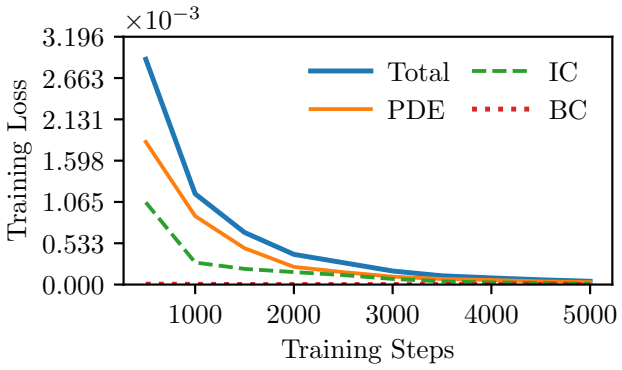

Figure 9: **Training loss versus training steps for the optimised PINN configuration. The total loss and its PDE, IC, and BC components decrease consistently, indicating successful convergence.**

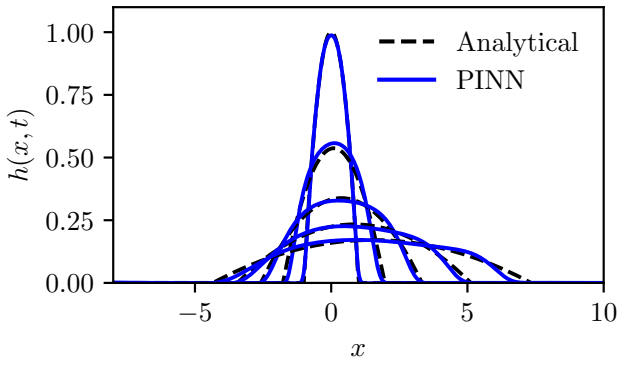

Figure 10: **Comparison of PINN-predicted height profiles $h(x,t)$ with the analytical solution at different time instances. The model accurately reproduces the transient behaviour of the solution.**

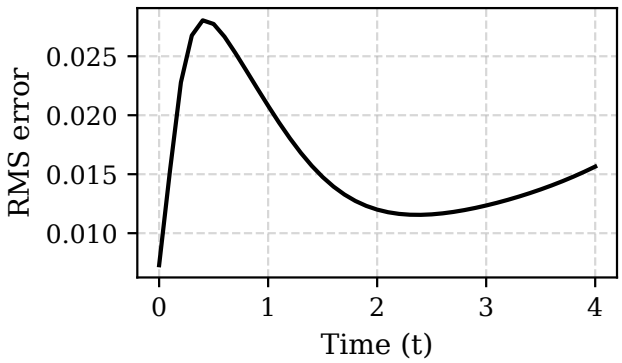

Figure 11: **RMS error between PINN predictions and the analytical solution over time.**

IV. CONCLUSIONS

This study applied PINNs to predict avalanche flow height's spatio–temporal evolution on an inclined plane. The model accurately reproduces analytical profiles by using the depth-averaged continuity equation. The hyperparameter study showed that loss-function weights, number of hidden layers, learning rate, and training steps most influenced model accuracy. In contrast, collocation density and number of neurons per layer had comparatively little effect. With a balanced configuration, the network achieved stable training and reliable predictions. A key limitation of the present work is that the velocity field was prescribed analytically rather than learned, restricting the ability to capture coupled dynamics. Future work will address this by enforcing continuity and momentum equations so that height and velocity can be predicted together. Additional improvements will include adaptive loss weighting, refined collocation near sharp gradients, and validation against numerical solvers on more realistic topographies.

ACKNOWLEDGEMENTS

PSS and VS gratefully acknowledge the Ministry of Education (MoE), Government of India, for the scholarship support during their postgraduate studies.

NOMENCLATURE

a	Non-dimensional acceleration parameter	–
g	Acceleration due to gravity	m/s^2
$g(t)$	Time-dependent lateral spread function	–
h	Non-dimensional flow depth	–
$\bar{h}$	Dimensional flow depth	m
$h_0(x)$	Initial snow height profile	–
$h_F(t)$	Forward boundary height	–
$h_R(t)$	Rear boundary height	–
k_{ap}	Earth pressure coefficient	–
L	Reference length scale	m
N_{col}	Number of collocation points	–
N_{ic}	Number of IC points	–
N_{bc}	Number of BC points	–
t	Non-dimensional time	–
$\bar{t}$	Dimensional time	s
u	Depth-averaged velocity	–
$\bar{u}'$	Dimensional velocity	m/s
u'	Non-dimensional velocity	–
$u_{\text{a}}(x,t)$	Analytical velocity profile	–
w_{pde}	PDE loss weight	–
w_{ic}	IC loss weight	–
w_{bc}	BC loss weight	–
x	Non-dimensional downslope coordinate	–
$\bar{x}$	Dimensional downslope coordinate	m
x_F	Forward boundary location	–
x_R	Rear boundary location	–
δ	Bed friction angle	rad

ϵ	Aspect ratio (depth/width)	–
ζ	Inclination angle	rad
β	Pressure-gradient coefficient	–
$\mathcal{L}_{\mathrm{total}}$	Total loss	–
$\mathcal{L}_{\mathrm{PDE}}$	PDE loss	–
$\mathcal{L}_{\mathrm{IC}}$	IC loss	–
$\mathcal{L}_{\mathrm{BC}}$	BC loss	–

REFERENCES

M. Boensch, F. Rudolf-Miklau, S. Sauermoser, and A. Mears. *The technical avalanche protection handbook*. John Wiley & Sons, 2015.

R. Ellath Meethal. *Hybrid modelling and simulation approaches for the solution of forward and inverse problems in engineering by combining finite element methods and neural networks*. PhD thesis, Technische Universität München, 2023.

R. J. LeVeque. *Finite volume methods for hyperbolic problems*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.

S. B. Savage and K. Hutter. The motion of a finite mass of granular material down a rough incline. *Journal of fluid mechanics*, 199:177–215, 1989.

V. Sharma and G. Bhutani. Computational modeling of snow avalanche dynamics: A case study of teling nala near atal tunnel. In Hardik Kothadia, K. R. Arun, G. Rajesh, and Jaywant H. Arakeri, editors, *Proceedings of Fluid Mechanics and Fluid Power (FMFP) 2023, Vol. 2*, pages 469–481, Singapore, 2025. Springer Nature Singapore. ISBN 978-981-97-6783-0.

C. Zhao, F. Zhang, W. Lou, X. Wang, and J. Yang. A comprehensive review of advances in physics-informed neural networks and their applications in complex fluid dynamics. *Physics of Fluids*, 36(10), 2024.