PHYSICS-INFORMED NEURAL NETWORKS

Using a data-driven and physics-informed framework for solving real-world complex problems

Shivprasad Kathane

Prof. Shyamprasad Karagadde

Introduction

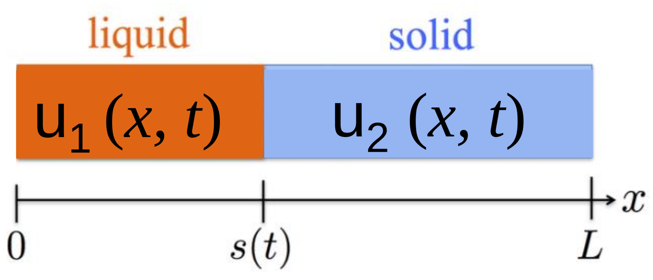
- Physics-Informed Neural Network (PINN) is a deep-learning framework which incorporates process-information [1]
- It enforces constraints corresponding to the partial differential equations (PDEs), initial and boundary conditions while learning from the observed data
- This could be useful in solving real-world complex problems with imperfect data, missing conditions, shocks, steep gradients, multiphysics prediction etc.

Objective

1. To solve a two-phase problem involving a moving interface, under missing conditions
2. To solve a multiphysics problem involving prediction of a complex composition profile, temperature profile and interface movement using clever strategies with PINN framework

Problem Definition

(1) Predicting temperature (u) and interface position (s) for ice-melting with missing boundary conditions [3]



$$k_1 = 2, k_2 = 1, \alpha_1 = -2, \alpha_2 = 1, s_0 = \tfrac{1}{2}, u^* = 0$$

$$\frac{\partial u_i}{\partial t} = k_i \frac{\partial^2 u_i}{\partial x^2}, \quad (x,t) \in \Omega_i, \quad i = 1,2$$

$$\Omega_1 = \{(x,t) \in \Omega : 0 < x < s(t), t \in (0,T]\}$$

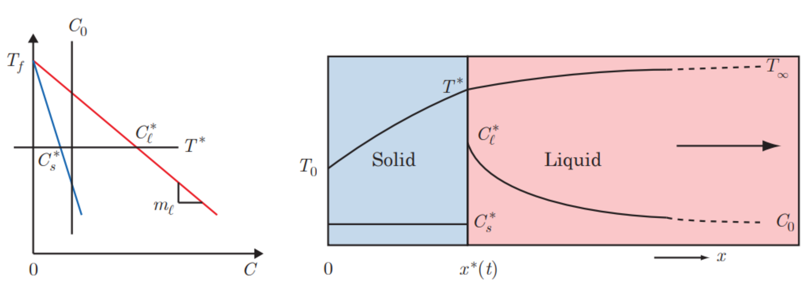
$$\Omega_2 = \{(x,t) \in \Omega : s(t) < x < L, t \in (0,T]\}$$

$$u_1(s(t),t) = u_2(s(t),t) = u^*, \quad t \in [0,1]$$

$$s'(t) = \alpha_1 \frac{\partial u_1}{\partial x}(s(t),t) + \alpha_2 \frac{\partial u_2}{\partial x}(s(t),t), \quad t \in [0,1]$$

$$s(0) = s_0$$

(2) Phase-change in a binary system [6]: Predicting temperature, composition (C) and interface position



Phase diagram (L) & expected temperature/composition profiles (R)

$C_0 = 0.05$, $m_l = -81.1$, $m_s = -478$, $c_p = 820$, $\rho = 7000$, $L_f = 276000$, $T_f = 1538$, $T_\infty = 1540$, $T_0 = 1510$, $k_0 = 0.17$, $Le = 300$

Domain $\epsilon \in [0, \epsilon^*]$

$C_s = k_0 C_l^*$

PDE1: $\frac{\partial \theta_s}{\partial \tau} = \frac{\partial^2 \theta_s}{\partial \epsilon^2}$

Domain $\epsilon \in [\epsilon^*, 1]$

PDE2: $\frac{\partial \theta_l}{\partial \tau} = \frac{\partial^2 \theta_l}{\partial \epsilon^2}$

PDE3: $\frac{\partial C_l}{\partial \tau} = \frac{1}{Le}\frac{\partial^2 C_l}{\partial \epsilon^2}$

Interface ($\epsilon = \epsilon^*$)

PDE4: $\frac{1}{Ste}\frac{\partial \epsilon^*}{\partial \tau} = \frac{\partial \theta_s}{\partial \epsilon} - \frac{\partial \theta_l}{\partial \epsilon}$

PDE5: $\frac{-1}{Le}\frac{\partial C_l}{\partial \epsilon} = C_l^*(1 - k_0)\frac{d\epsilon^*}{d\tau}$

Initial ($\tau = 0$)

$\theta_l = 1\,;\, C_l = C_0\,;\, \epsilon^* = 0$

Interface ($\epsilon = \epsilon^*$)

$\theta_s = \theta_l = \theta^*$

$C_l^* = (T_0 - T_\infty)(\theta_f - \theta^*)/m_l$

Left BC ($\epsilon = 0$)

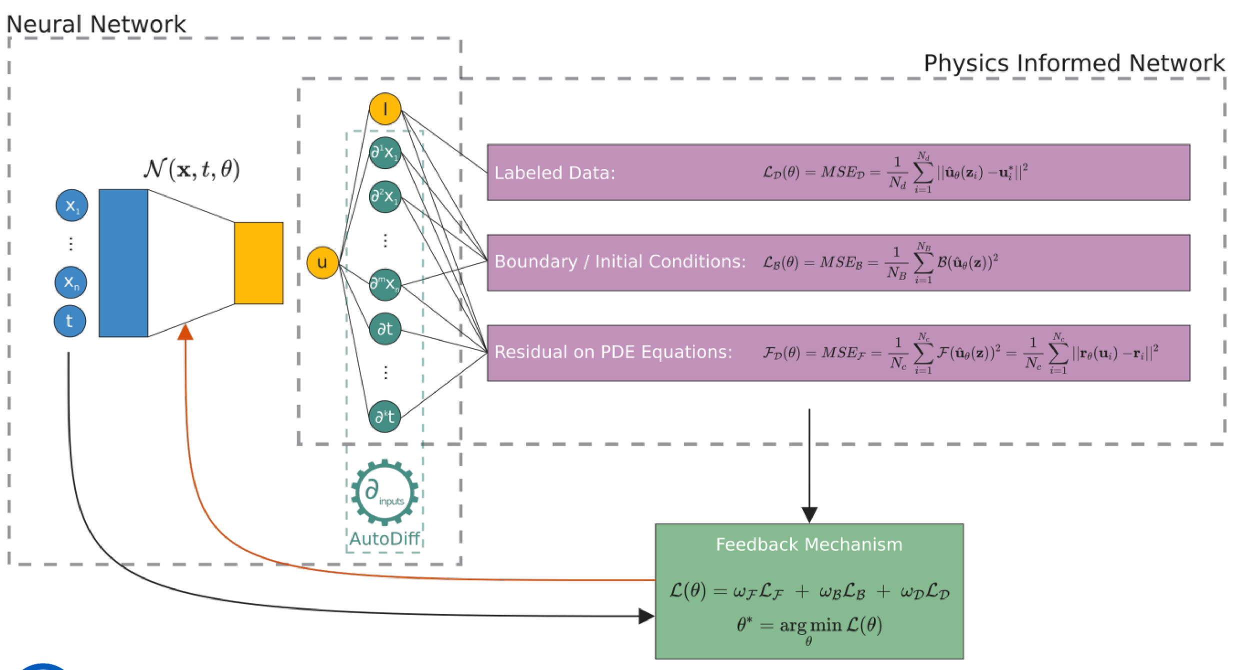
$\theta_s = 0$

$Ste = c_p(T_\infty - T_0)/L_f$

$\theta_f = (T_f - T_0)/(T_\infty - T_0)$

Replaced u_1, u_2, s with θ_s, θ_l, ε

Methodology



(A) *Schematic of the Physics-Informed Neural Network framework [2]*

For each neural network in both the problems:

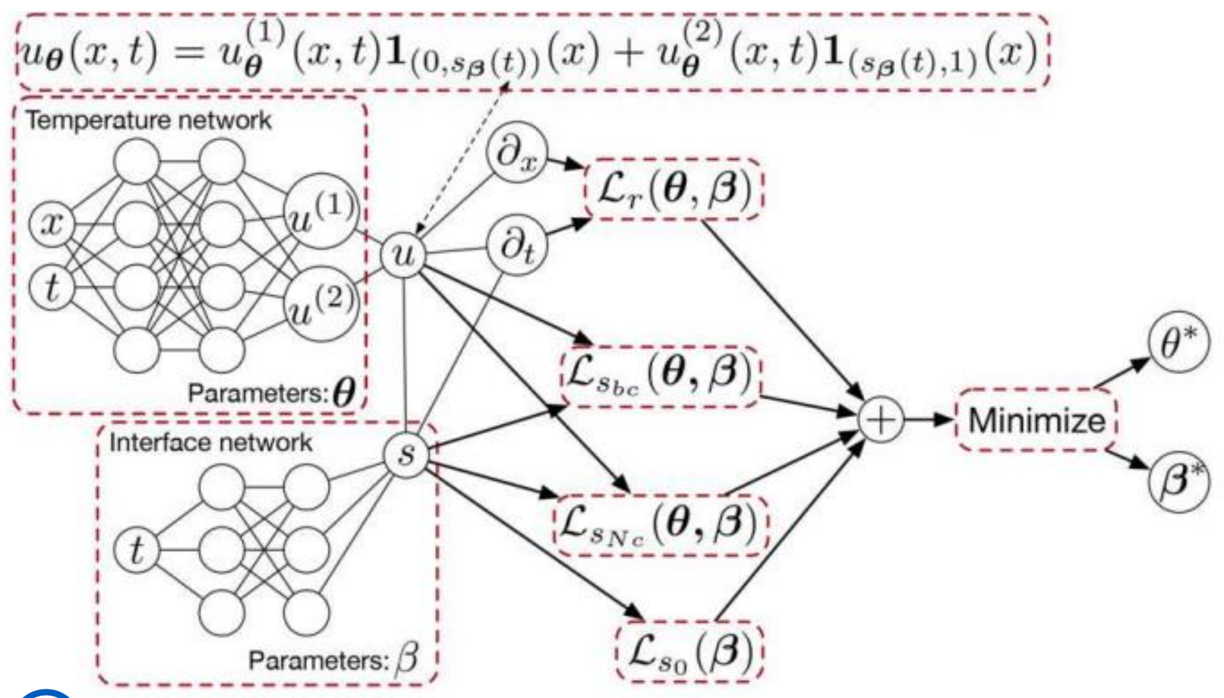
No. of Hidden Layers = 5 each with No. of Neurons = 100

Activation Function = Swish = x*sigmoid(x)

Weight Initialiser = Glorot Normal and Optimiser = Adam

Approach for Problem 1:

- Domain data = 200 labelled (fixed) + 10,000 (each epoch)
- PINN architecture of Fig B was trained for 5000 epochs
- Minimisation of addition of loss terms with decaying LR



(B) *The parallel network architecture for solving Stefan's Problem [3]*

Approach for Problem 2:

- Add a network for C_l(x,t) and a trainable parameter for C_s
- Loss terms were scaled as per data collected (Fig C)
- Employ causal training for 1000 epochs with LR = 0.005
- Stopping Condition on C_s = Error tolerance of 10^{-4}
- Fix network for s(t), apply adaptive weighting (1000 epochs)
- Apply another round of causal training + adaptive weighting

Causal Training → Temporal Weights

L(ti,θ) = Residual loss at a discrete timestep

Total weighted average loss function:

$$\mathcal{L}(\theta) = \frac{1}{N_t}\sum_{i=0}^{N_t} w_i \mathcal{L}(t_i, \theta),$$

Labelled Data:

900 domain + 50 interface + 50 (t=0)

Collocation Data:

N_t=100 x N_x=256 grid (each epoch)

For ϵ in [0.01, 0.1, 1, 10, 100]:

Repeat for S iterations:

Compute temporal weights w_i $\quad w_i = \exp\left(-\epsilon \sum_{k=1}^{i-1} \mathcal{L}(t_k, \theta)\right)$

Gradient descent parameter update $\quad \theta_{n+1} = \theta_n - \eta \nabla_\theta \mathcal{L}(\theta_n)$

(C) *The causal training algorithm [4]*

Adaptively Weighted Loss

Inspired by Adam's Optimisation

Let the loss function be given by $\mathcal{L}(\theta) := \mathcal{L}_r(\theta) + \sum_{i=1}^{M} \lambda_i \mathcal{L}_i(\theta),$

Repeat for each epoch:

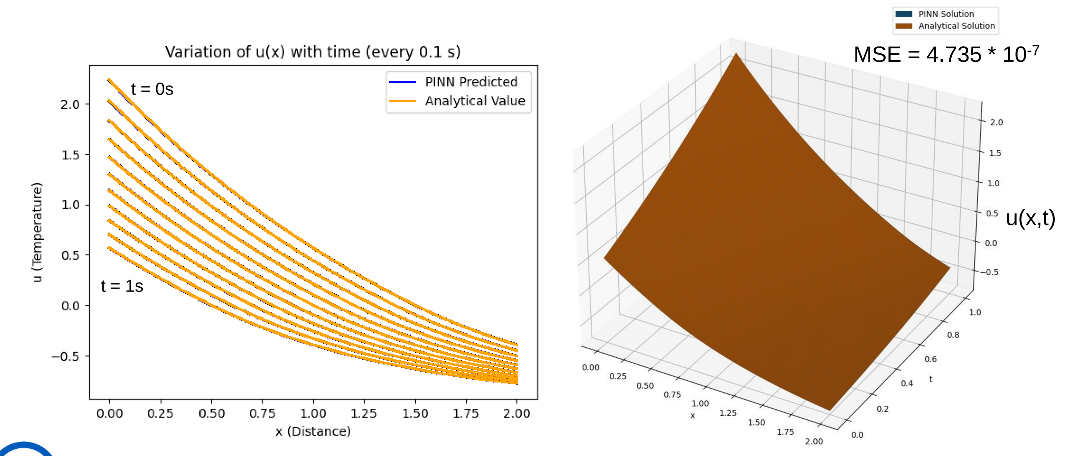
Calculate instantaneous weight as $\hat{\lambda}_i = \frac{\max_\theta\{|\nabla_\theta \mathcal{L}_r(\theta_n)|\}}{\overline{|\nabla_\theta \mathcal{L}_i(\theta_n)|}}, \quad i = 1, \ldots, M,$

Update weight using moving average $\lambda_i = (1-\alpha)\lambda_i + \alpha\hat{\lambda}_i, \quad i = 1, \ldots, M.$

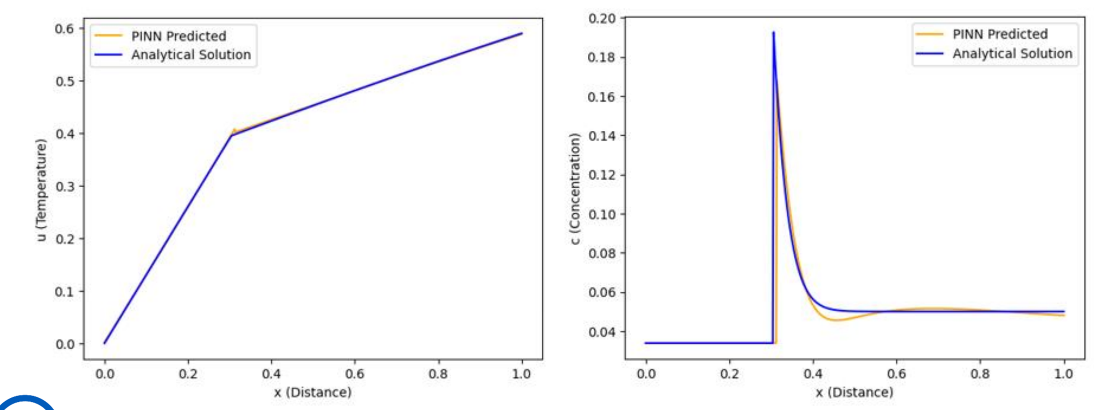
Update parameters using gradient descent $\theta_{n+1} = \theta_n - \eta\nabla_\theta \mathcal{L}_r(\theta_n) - \eta \sum_{i=1}^{M} \lambda_i \nabla_\theta \mathcal{L}_i(\theta_n)$

(D) *The adaptive weighting algorithm [5]*

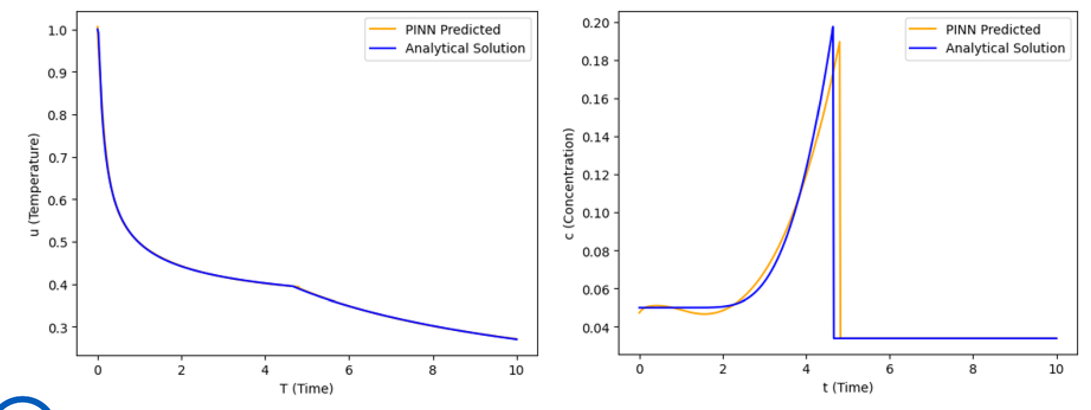
Results



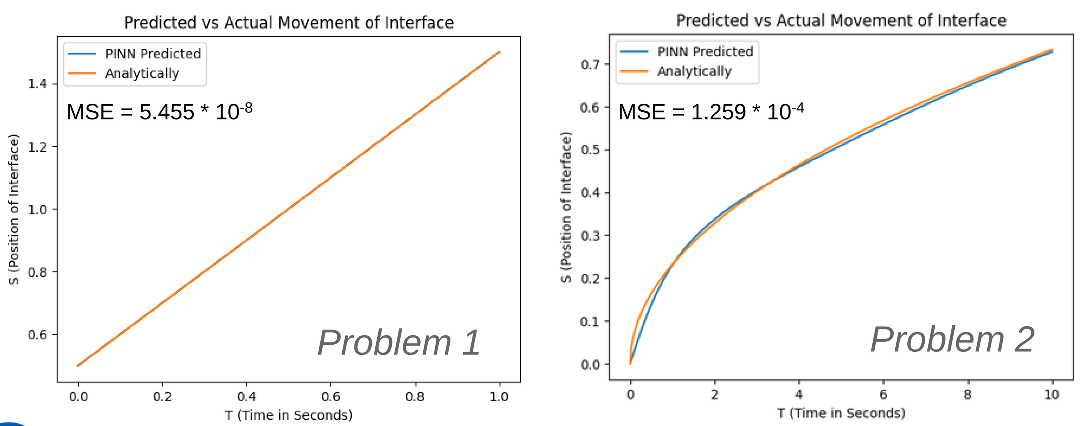
(E) *Problem 1: u vs x every 0.1s (L) and temperature surface plot (R)*



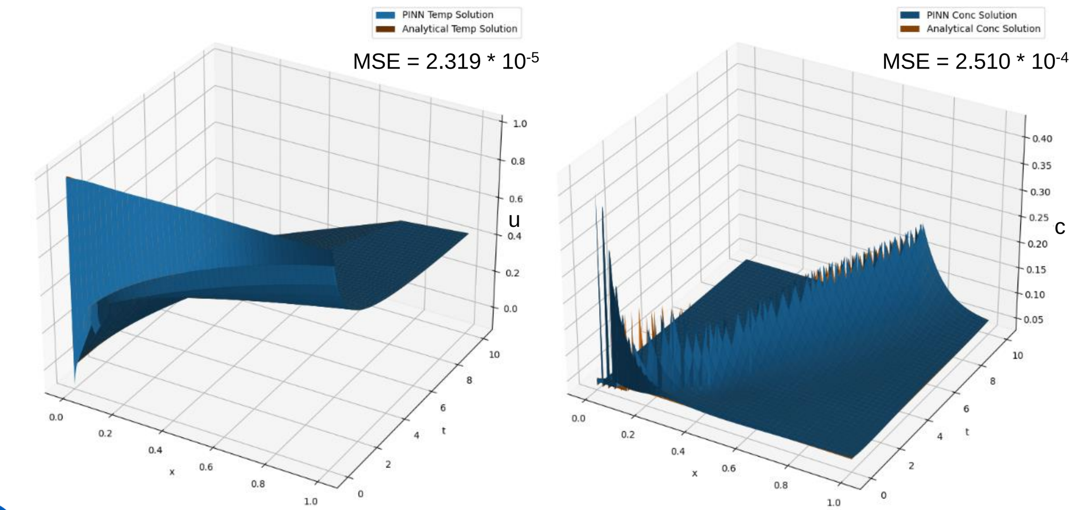
(F) *Problem 2: Temperature (L) and Composition (R) vs x at t = 1.725s*



(G) *Problem 2: Temperature (L) and Composition (R) vs t at x = 0.5*



(H) *Evolution of interface position with time for the two problems*



(I) *Problem 2: Temperature (L) and Composition (R) surface plots*

Conclusion

- PINN is a data-driven + physics-informed framework useful in solving real-world, complex, multiphysics problems
- A careful approach is needed with strategies like separate networks, alternate application of causal training + adaptive weighting, still, loss convergence requires attention (Problem 2)

Acknowledgements

I thank my faculty advisor for guidance, C-MInDS for scholarship and MEMS & IITB for coursework

References

1. M. Raissi et al., Journal of Computational Physics, 2018
2. S. Cuomo et al., Journal of Scientific Computing, 2022
3. S Cai et al., Journal of Heat Transfer, 2021
4. S Wang et al., arXiv preprint 2203.07404, 2022
5. S Wang et al., SIAM Journal on Scientific Computing, 2021
6. J A Dantzig and M Rapazz, "Solidification" Book, 2009

INDIAN INSTITUTE OF TECHNOLOGY BOMBAY
Centre for Machine Intelligence and Data Science [CMInDS]
www.minds.iitb.ac.in

Inter-Disciplinary Dual Degree Program (IDDDP) in AI and Data Science (2022-2023)

Metallurgical Engineering and Materials Science (2018-2022)