Physics Informed Neural Networks for Heat Conduction with Phase Change

Bahae-Eddine Madir, Corentin Lothodé, Francky Luddens, Ionut Danaila

LMRS, Université de Rouen Normandie

September 24, 2024

1. Phase change materials (PCM), Stefan's problem

Navier-Stokes-Boussinesq equations

$$\nabla \cdot u = 0,$$

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)\, u + \nabla p - \frac{1}{Re}\nabla^2 u - f_B(\theta) e_y - A(\theta) u = 0,$$

$$\frac{\partial C\,\theta}{\partial t} + \nabla \cdot (C\,\theta\, u) - \nabla \cdot \left(\frac{K}{Re\,Pr} \nabla \theta \right) + \frac{\partial C\, S(\theta)}{\partial t} = 0.$$

The velocity field u in the liquid part of a PCM heated vertically from the right.

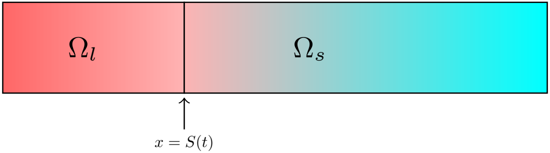
Image ref: Gong et al., 2015

A block of ice at a constant temperature[1] θ_c is heated from the left side to a temperature θ_h. The temperature θ of the system satisfies the following:

$$\partial_t \theta_l - Fo\,\partial_x^2 \theta_l = 0, \qquad \mathcal{T} \times \Omega_l,$$

$$\partial_t \theta_s - Fo\,\partial_x^2 \theta_s = 0, \qquad \mathcal{T} \times \Omega_s,$$

$$\partial_x \theta_s - \partial_x \theta_l = Fo^{-1} Ste^{-1}\, S'(t), \qquad \mathcal{T},\ x = S(t).$$



[1] In a dimensionless setting!

The enthalpy of the liquid-solid system is defined as

$$H = \theta + Ste^{-1}\,\varphi(\theta)$$

with φ representing the liquid fraction (Heaviside fct.). Substituting H into the heat equation, leads to

$$\partial_t H - Fo\,\partial_x^2\theta = 0. \tag{1}$$

For numerical and computational feasibility, φ is smoothed to

$$\varphi_\delta(\theta) = \frac{1}{2}\left(1 + \tanh\frac{\theta}{\delta}\right), \quad \delta > 0.$$

Resulting the (one insted of two) PDE

$$\partial_t\theta - Fo\,\partial_x^2\theta + Ste^{-1}\,\partial_t\varphi_\delta(\theta) = 0. \tag{2}$$

A few remarks:

- An exact analytical solution to the Stefan problem is possible, if the block of ice occupies a *semi-infinite* region.
- In an actual melting problem, the material (ice) has a finite length.
- In the enthalpy formulation, the regularized problem (with φ_δ) differs from the regular problem (with φ), since δ does not necessarily tend to zero. As a result, the solutions could also differ.

Consequences: We avoid using the exact solution, and instead generate a reference solution specific to the regularized problem.

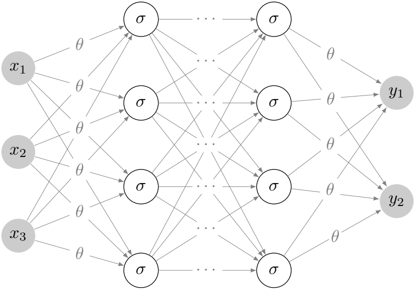
The problem is addressed numerically using the FD method: applying Crank-Nicolson's scheme for time integration and central FD for spatial discretization. To solve the resulting nonlinear problem, we employed the Newton-Raphson method.

2. Neural networks, *Physics Informed* Neural Networks (PINNs)

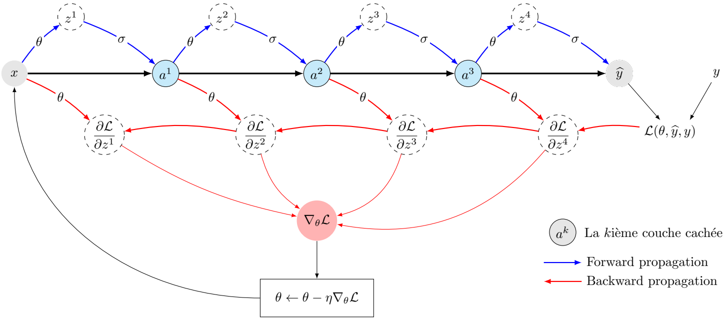
A neural network of n layers, is the composition of n functions $\ell^k : x \mapsto \ell(x, \theta_k)$, θ_k denotes the set of parameters for the kth layer

$$u_\theta(x) = \ell^n \circ \ell^{n-1} \circ \cdots \circ \ell^1(x) \tag{3}$$

The popular one: *Multi Layer Perceptron* with $\ell^k(x) = \sigma^k \left(\mathrm{W}^k x + \mathrm{b}^k \right)$



A neural network with 4 layers, each containing one neuron

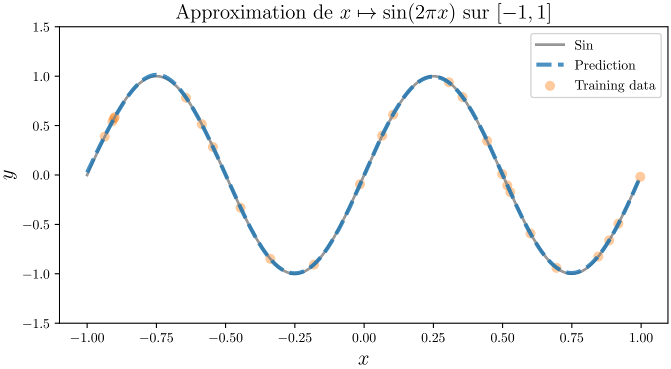


In other words

$$\theta^\star = \arg\min_\theta \mathcal{L}(\theta, \widehat{y}, y). \tag{4}$$

$z^k = \mathrm{W}^k a^{k-1} + \mathrm{b}^k$
$a^k = \sigma(z^k)$, $a^0 = x$

A shallow neural network (with one hidden layer) can approximate any continuous function with any given accuracy, provided it has a sufficient number of neurons (Hornik. 1991)



Consider PDE

$$\begin{aligned} \partial_t u + \mathcal{N}[u] &= 0, & \mathcal{T} \times \Omega, \\ u(t,x) &= g(t,x), & \mathcal{T} \times \partial\Omega, \\ u(0,x) &= h(x), & \Omega. \end{aligned} \tag{5}$$

The solution u to the problem can be approximated with a neural network $\widehat{u} := u_\theta$, by minimizing with respect to parameters θ, the loss functions:

$$\begin{aligned} \mathcal{L}_r &= \|\partial_t \widehat{u} + \mathcal{N}[\widehat{u}]\|_{\mathcal{T}\times\Omega,\, N_r}, \\ \mathcal{L}_b &= \|\widehat{u} - g\|_{\mathcal{T}\times\partial\Omega,\, N_b}, \\ \mathcal{L}_0 &= \|\widehat{u} - h\|_{\Omega,\, N_0}. \end{aligned}$$

$\|f\|_{A,\,N} = \frac{1}{N}\sum_{k=1}^{N} |f(x_k)|^2$, x_k are randomly chosen uniformly on A.

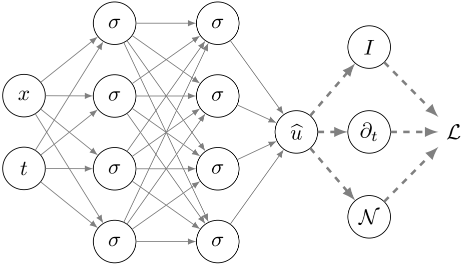


Figure: Physics Informed Neural Network (PINN): a neural network is employed to predict the solution *u* for the problem. Then, using automatic differentiation, the loss functions are computed and minimized with respect to the network parameters.

Consider Poisson's problem in $[0,1]$

$$-u''(x) = 4\pi^2 \sin(2\pi x), \qquad u(0) = u(1) = 0.$$

Let $\widehat{u}$ be a shallow neural network of 10 neurons, with $\sigma = \tanh$. We set:

$$\mathcal{L}_r = \frac{1}{15} \sum_{k=1}^{15} \left|\widehat{u}''(x_r^k) + 4\pi^2 \sin(2\pi x_r^k)\right|^2,$$

$$\mathcal{L}_u = \widehat{u}(0)^2 + \widehat{u}(1)^2, \quad \mathcal{L} = \mathcal{L}_r + \mathcal{L}_u.$$

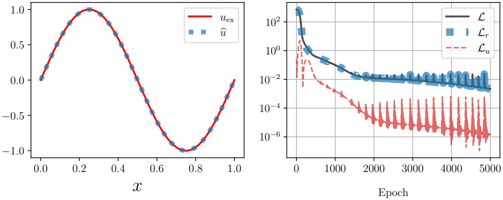


Figure: Approximation of $u_{\text{ex}}(x) = \sin(2\pi x)$, relative L_2 error of $O(10^{-4})$ at the end.

3. *PINNs* for Stefan's problem

We will test the behavior of the method (PINNs) for two cases:

- When the enthalpy jump is moderate e.g. $Ste = 0.5$.
- When the enthalpy jump is large e.g. $Ste = 0.005$.

$Fo = 10^{-2}$, $\delta = 0.05$ remain fixed. For both cases we use a neural network of two inputs (t, x), six hidden layers of 20 neurons, with activation function $\sigma = \tanh$.

In the following, $\mathcal{L}_r$, $\mathcal{L}_0$, and $\mathcal{L}_b$ represent the physics-informed loss (residual) terms, corresponding to the PDE, the initial condition, and the boundary condition, respectively.

Reminder, the enthalpy is defined as

$$H_\delta = \theta + Ste^{-1}\,\varphi_\delta(\theta).$$

The PDE

$$\partial_t\theta - Fo\,\partial_x^2\theta + Ste^{-1}\,\partial_t\varphi_\delta(\theta) = 0.$$

Case of $Ste = 0.5$

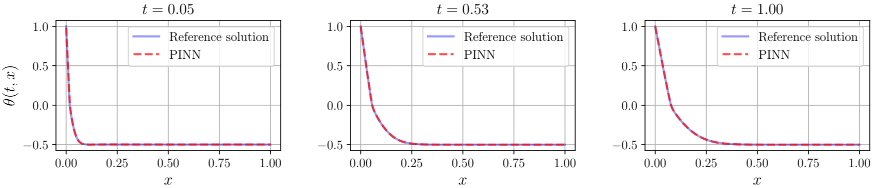


Figure: Solution of (2) by minimizing $\mathcal{L} = \mathcal{L}_0 + \mathcal{L}_b + \mathcal{L}_r$ over 10^5 iterations of gradient descent (Adam type). The relative L_2 error at the end is of $O(10^{-3})$.

Case of $Ste = 0.005$

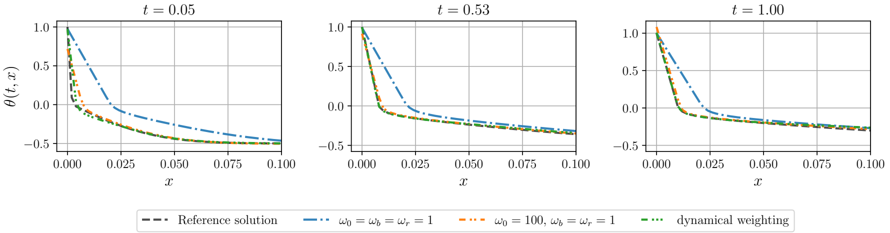


Figure: Solution of (2) by minimizing $\mathcal{L} = \omega_0 \mathcal{L}_0 + \omega_b \mathcal{L}_b + \omega_r \mathcal{L}_r$ over 10^5 iterations of gradient descent (Adam type).

	$\omega_0 = \omega_b = \omega_r = 1$	$\omega_0 = 100,\ \omega_b = \omega_r = 1$	dynamical weighting[2]
Rel. L_2 error	1.3×10^{-1}	3.5×10^{-2}	2.5×10^{-2}

[2] see Sifan Wang et al. 2021.

Pointwise weighting (still in the case $Ste = 0.005$)

Instead of using one weight for one loss function, we attribute to each training point (t, x) a weight $\omega(t, x)$ e.g.

$$\text{(before)} \quad \omega_r \mathcal{L}_r = \frac{\omega_r}{N_r} \sum_{k=1}^{N_r} \left| \partial_t \widehat{\theta}(t_k, x_k) - Fo\, \partial_x^2 \widehat{\theta}(t_k, x_k) + Ste^{-1}\, \partial_t \varphi_\delta(\widehat{\theta})(t_k, x_k) \right|^2,$$

$$\text{(after)} \quad \omega_r \mathcal{L}_r' = \frac{1}{N_r} \sum_{k=1}^{N_r} m(\omega_r(t_k, x_k))^2 \left| \partial_t \widehat{\theta}(t_k, x_k) - Fo\, \partial_x^2 \widehat{\theta}(t_k, x_k) + Ste^{-1}\, \partial_t \varphi_\delta(\widehat{\theta})(t_k, x_k) \right|^2,$$

with m is a strictly $\nearrow$ non linear function. Same thing for $\omega_0 \mathcal{L}_0$ and $\omega_b \mathcal{L}_b$. The problem is formulated as follows:

$$(\Theta^\star, \omega_i^\star, \omega_b^\star, \omega_r^\star) = \min_{\Theta} \max_{\omega_i, \omega_b, \omega_r} \mathcal{L}(\Theta, \omega_i, \omega_b, \omega_r) \tag{6}$$

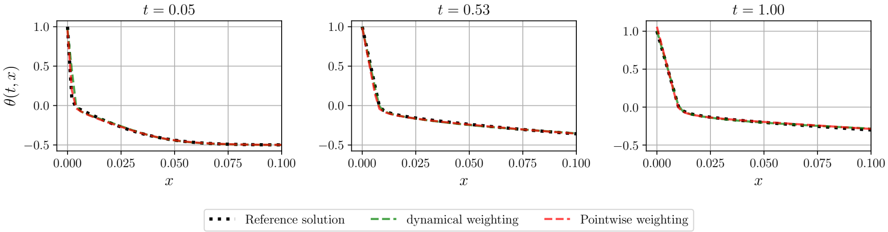


Figure: Solution of (2) by minimizing $\mathcal{L} = \omega_0 \mathcal{L}'_0 + \omega_b \mathcal{L}'_b + \omega_r \mathcal{L}'_r$ over 10^5 iterations of gradient descent (Adam type). The relative L_2 error at the end is 2.4×10^{-2}.

4. Conclusion

Conclusion, Perspectives

- For $Ste = 0.5$ we can directly approximate the solution of the problem.
- For $Ste = 0.005$ "sharp solution" $\Rightarrow$ difficulties in the learning process, it is necessary to balance the components of the loss function, globally or locally.

- Next step: coupling Stefan's problem with the Navier-Stokes equations (Navier-Stokes-Boussinesq).

Thank you for your attention!