Physics Informed Neural Networks for coupled radiation transport equations

Laetitia Laguzet[1], Gabriel Turinici[2]

[1]CEA-DAM-DIF, F-91297 Arpajon, France

[2]CEREMADE, Université Paris Dauphine - PSL, 75016 Paris, France

Computational Methods for Multi-scale, Multi-uncertainty and Multi-physics Problems, 2-4 July 2025, Porto, Portugal

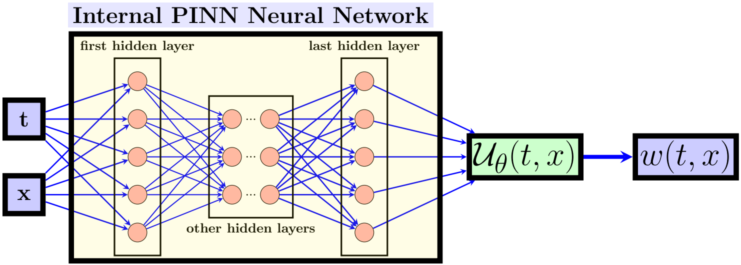
Outline

1. **Introduction of Physics Informed Neural Networks (PINNs)**

2. **PINNs for coupled radiative equation**

PINNs: general framework

⋆ Find u with: $\partial_t u = \mathcal{G}(u)$, $u(0,x) = u_0(x)$, $u(t,x) = u_{bc}$, $t \in [0,T]$, $x \in \Omega \subset \mathbb{R}^d$ $(d \geq 1)$ where $u_0(x)$ (initial conditions), u_{bc} (boundary conditions) are known functions and $\mathcal{G}$ some operator.



⋆ PINNS gives the approximation $\mathcal{U}_\theta(t,x)$ of $u(t,x)$ with θ the NN parameters.

⋆ w is the equation error: $w = \partial_t \mathcal{U}_\theta - \mathcal{G}(\mathcal{U}_\theta)$

PINNs: loss contributors

⋆ Initial and boundary conditions:

- Imposed it using:

$$t \cdot \mathcal{U}_\theta(t, x) + u_0(x)$$

- Shift the result of network

$$\mathcal{U}_\theta(t, x) - \mathcal{U}_\theta(0, x) + u_0(x)$$

⋆ Structure of the loss:

- Original proposition :

$$\mathcal{L}(\theta) = \int_0^T \int_\Omega |w(t, x)|^2 dxdt$$

- Weighting schemes (see example later):

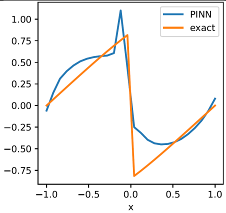
$$\mathcal{L}(\theta) = \int_0^T \int_\Omega \rho(t) |w(t, x)|^2 dxdt$$

Specification when using PINNs : examples

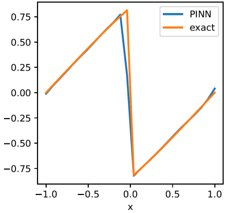
⋆ Burgers' equation: one dimensional flow of viscous fluid:

$$\partial_t u + u\partial_x u = v\partial_{xx} u \text{ with } t \in]0, T], x \in]-1, 1[\text{ and } v = 0.01/\pi$$

$$u(0, x) = -\sin(\pi x) \text{ and } u(t, \pm 1) = 0 \ \forall t \in [0, T]$$



$$\rho(t) = \exp\{-\epsilon \int_0^t ||w(s,x)||^2_{L^2_x} ds\}$$



$$\rho(t) = \exp\{-\int_0^t \lambda_n(s) ds\}$$

with $\lambda_n(t) \sim \langle \mathcal{G}_t(\mathcal{U}_{\theta_n}(t,\cdot)) - \mathcal{G}_t(0), \mathcal{U}_{\theta_n}(t,\cdot)\rangle / \|\mathcal{U}_{\theta_n}(t,\cdot)\|^2$ and $\mathcal{G} = \partial_{xx}$

Outline

1. Introduction of Physics Informed Neural Networks (PINNs)

2. **PINNs for coupled radiative equation**

Transport equation coupled with matter evolution

⋆ Radiative transfer equation in gray general form:

- transfer equation:

$$\underbrace{\frac{1}{c}\partial_t I + \Omega \cdot \nabla I}_{\text{spatial process evolution}} + \underbrace{\sigma_a(t,x) I}_{\text{absorption term}} = \underbrace{\sigma_s(t,x) \left(\int_{S^2} I d\Omega - I \right)}_{\text{isotropic scattering term}} + \underbrace{f(t,x,\Omega)}_{\text{source term}}$$

- matter evolution:

$$\partial_t(\rho\epsilon) = \int_{S^2} (\sigma_a I - f) d\Omega$$

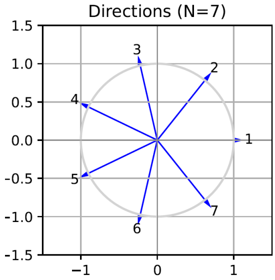
with $I(t,x,\Omega)$ the radiative intensity, and $\rho(t,x)\epsilon(t,x)$ the internal energy density. S^2 is the unit sphere and $\Omega \in S^2$.

⋆ General methods use to solve this class of non linear coupled equation: deterministic method (like P_N or S_N) and Monte Carlo (explicit or implicit also called symbolic) with a linearization.

Particular case

⋆ PINNs in the simplified case:

- $c_n^t = 1$; c_F and θ are constants;
- $F(a) = c_F / a^3$;
- constant c_F and θ are chosen to have a Marshak wave equation;
- approach like S_N method: $N = 7$ directions $c_n^x = \cos(2\pi n/N)$.



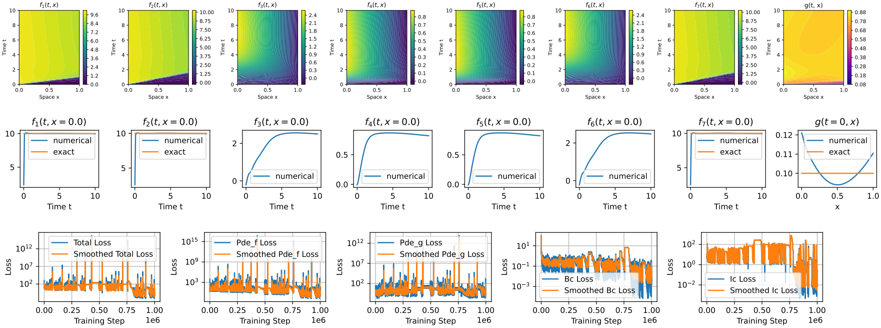
$$c_n^t \partial_t f_n(t,x) + c_n^x \partial_x f_n(t,x) + F(g(t,x)) f_n = \theta \cdot g(t,x) \ \forall n \in \{1, ..., 7\}$$

$$c_g^t \partial_t g(t,x) = F(g(t,x)) \sum_{n=1}^{N} f_n(t,x)/N - \theta \cdot g(t,x)$$

$f_n(t, x_{min}) = 10$ for $n = 1, 2, 7$ and $g(0, x) = 0.10$.

Activation: first test

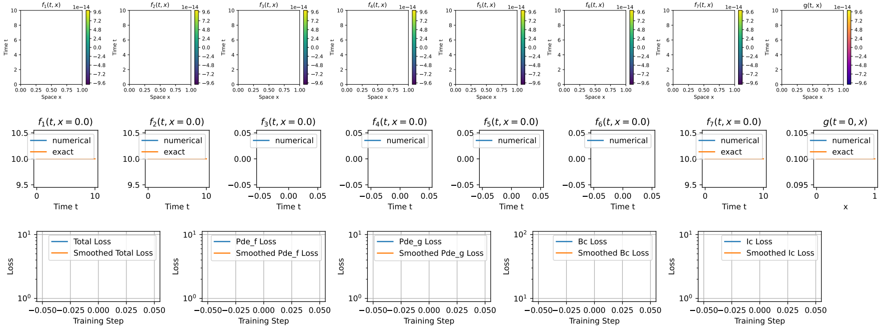
⋆ Hidden layer activation function: tanh and linear final activation function.



⋆ Results admissible but after a lot of iteration. Note that after 0.5M iteration, g is still negative

Activation function: second test

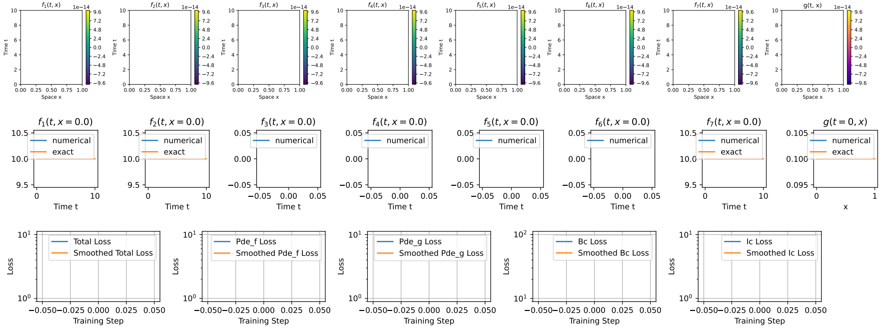
⋆ Hidden layer activation function: tanh and ReLu final activation function.



⋆ NaN results

Activation function: third test

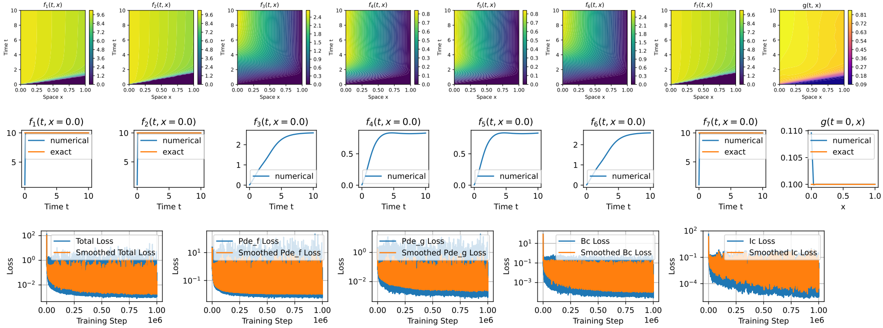
⋆ Hidden layer activation function: tanh and $(ReLU)^2 = (\cdot_+)^2$ final activation function.



⋆ NaN results

Activation function: final test

⋆ Hidden layer activation function: tanh, safeSoftPlus ($\log(1 + e^x) + 10^{-3}$) final activation function.



⋆ Good results.

Encountered difficulty

⋆ Need to use two different networks (one for all f_n and one for g);

⋆ NN sensitive to the random initialization: can start with an error of 10^{18} or 10^3

⋆ Difficulty to set the optimal learning rate: value 0.01 is instable very quickly and value 0.001 is converging slowly;

⋆ Tests were done with different (final) activation functions (to ensure positive value of g); the last activation was either ReLU, ReLU2, no activation (aka Linear) or SafeSoftPlus;

Conclusion

⋆ With no activation option:

• negative values for the solutions at the start occurs, which is not acceptable even if after some iterations it becomes positive;

• the error is however still large and convergence slow. For instance after 0.5M iterations g is still negative; at about 1M iterations solution starts to be reasonable;

⋆ ReLU, ReLU2 gives NaN-s immediately;

⋆ safeSoftPlus ($\log(1 + e^x) + 10^{-3}$) works better and faster.

Perspectives

⋆ Time distribution / weights of points : as a remark there are two causes that require adaptation of the distribution of weights of points:

• the solution has non-smooth behavior: this requires many points thus adapting the **distribution**;

• the solution is chaotic, causal or converging to equilibrium: this requires adaptation of **weights**: $\rho(\cdot)$ in the loss formula $\mathcal{L}(\theta) = \int_0^T \int_\Omega \rho(t)|\omega(t,x)|^2 dxdt$.

• Tests for time sampling: uniform vs. non-uniform, with current setup (GPU, many points); test also the effect of the number of hidden layers (use 8 for instance).

⋆ Try to find the optimal number of layers (results presented with 8) and neuron by layers (here 32).

References

1. Maziar Raissi, Paris Perdikaris, and George E Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational physics* 378 (2019), pp. 686–707

2. Gabriel Turinici. "Optimal time sampling in physics-informed neural networks". In: *International Conference on Pattern Recognition*. Springer. 2025, pp. 218–233

3. Laetitia Laguzet and Gabriel Turinici. "Model free collision aggregation for the computation of escape distributions". In: *ESAIM: Mathematical Modelling and Numerical Analysis* 58.5 (2024), pp. 2061–2077

4. Laetitia Laguzet and Gabriel Turinici. "The Quantization Monte Carlo method for solving radiative transport equations". In: *Journal of Quantitative Spectroscopy and Radiative Transfer* 329 (2024), p. 109178