1 Radiative Transfer with PINNs

1.1 Motivation

We chose to treat the radiative transfer equation which is a PDE in high dimension, hence very expensive to solve numerically. To counter this, [Mishra and Molinaro, 2021] have explored the use of physics-informed neural networks (PINNs).

Here we describe our approach to solve the radiative transfer equation in the stationary, monochromatic (both) 1D (baseline) and 3D (extension) case. For the 3D case, we followed the approach in [Grella, 2013] but this time with PINNs instead of other numerical methods.

1.2 Methods

1.2.1 Monochromaticity and stationarity

The radiative transfer model is defined as

$$\frac{1}{c}u_t + \omega \cdot \nabla_x u + ku + \sigma\left(u - \frac{1}{s_d}\int_\Lambda \int_S \Phi\left(\omega, \omega', \nu, \nu'\right) u\left(t, x, \omega', \nu'\right) d\omega' d\nu'\right) = f$$

and the corresponding variables as

$$\text{domain: } t \in [0,T], x \in D \subseteq \mathbb{R}^d, D_T = [0,T] \times D, \quad \text{angle: } \omega \in \mathbb{S}^{d-1} = S, \quad \text{group energy: } \nu \in \Lambda \subset \mathbb{R},$$

$$\text{surface area of the d-dimensional unit sphere: } s_d, \quad \text{speed of light: } c, \quad \text{radiative intensity: } u : D_T \times S \times \Lambda \to \mathbb{R},$$

$$\text{absorption coefficient: } k : D \times \Lambda \to \mathbb{R}, \quad \text{scattering coefficient: } \sigma : D \times \Lambda \to \mathbb{R}, \quad \text{scattering kernel: } \Phi : S \times S \times \Lambda \times \Lambda$$

$$\text{with normalization } \int_{S\times\Lambda} \Phi(\cdot, \omega', \cdot, \nu') d\omega' d\nu' = 1 \quad \text{and source term } f : D \times \Lambda \to \mathbb{R}$$

We can subsume the equation in operator form $Au = f, \quad A = T + Q, \quad T = (\omega \cdot \nabla_x + k)u,$

$$Q = \sigma(\mathbb{1} - \Sigma)u = \sigma(x)u(x,\omega) - \sigma(x)\int_S \Phi(\omega, \omega' u(x, \omega') d\omega' \text{ as scattering operator.}$$

We are interested in the monochromatic and stationary states. Hence, all dependencies on t are dropped in our model.

Additionally, the group energy ν is constant. Thus the integral simplifies to

$$\omega \cdot \nabla_x u + ku + \sigma\left(u - \frac{1}{s_d}\int_S \Phi\left(\omega, \omega'\right) u\left(x, \omega'\right) d\omega'\right) = f$$

1.2.2 1D case: baseline

Firstly, there is the one-dimensional case. The x-domain is an interval $[0,1]$. There we have inflow boundary conditions. Thus the previous equation becomes

$$\mu\frac{\partial}{\partial x}u(x,\mu) + (\sigma(x) + k(x))u(x,\mu) = \frac{\sigma(x)}{2}\int_{-1}^{1} \Phi\left(\mu, \mu'\right) u(x,\mu) d\mu', \quad \mu = \cos(\theta), \quad (x,\mu) \in [0,1] \times [-1,1].$$

with inflow boundary conditions

$$\begin{aligned} u(0,\mu) &= 1, \quad \mu \in (0,1], \\ u(1,\mu) &= 0, \quad \mu \in [-1,0). \end{aligned}$$

$$\sigma(x) = x, \quad k(x) = 0, \quad \Phi\left(\mu', \mu\right) = \sum_{\ell=0}^{L} d_\ell P_\ell(\mu) P_\ell\left(\mu'\right), \quad d_0 = 1$$

which is equation (3.1) from [Mishra and Molinaro, 2021] (note the variable name correction in our model). P_l denotes the Legendre-Polynomials of order l. Furthermore, the variables ω and μ are identically defined as was done in the paper. For the next step, we will look at the PyTorch implementation. The code can be found in part_b.py. We used the PINN class from the exercise sessions as inspiration.

In order to enforce the discontinuity in the boundary conditions at $\mu = 0$ we introduce the constant $\epsilon = 0.01$ such that any point with μ close enough to zero gets discarded from the loss computation.
The internal nodes do not have this constraint. However, inspired by fig. 3 in [Pontaza and Reddy, 2005], we have one set of points uniformly distributed over the whole μ,x domain, an equal-sized one only with points with x around zero or one, and a last one with values μ at around 0. The objective of the second set is to increase the accuracy of our model on the boundaries as for those values we have exact solutions as stated in [Pontaza and Reddy, 2005]. We dropped the last set as it did not have a visible impact on the results.
After profiling our code we observed that the scattering kernel was the most expensive part. We noted that $\Phi(\mu, \mu')$ could be computed upon initialization of the PINN object. We made this high-performing by computing Φ as a matrix (see function *build_phi*) and using the optimized matrix-matrix multiplication Φu in the function *kernel_mat*. This resulted in a formidable speedup. As in the paper we use 10 points of a Gauss-Legendre quadrature rule.
The model is trained with an ADAM optimizer with a learning rate of 9e-4 as well as a scheduler. The scheduler successfully stops the loss function from oscillating with values between 10e-1 and 10e-3 and brings it to converge to around 10e-3.

1.2.3 3D case: extension

Next, we examine the three-dimensional case. For this special case on the domain $[0, 1]^3$, we chose to implement section (3.3) from [Mishra and Molinaro, 2021] which builds on (8.2, experiment 3) from [Grella, 2013].

The 2-sphere has mass $s_d = 4\pi$. We define the source term

$$f(x) = k(x)I_b(x), \quad I_b(x) = \begin{cases} 0.5 - r, & r \leqslant 0.5 \\ 0, & \text{otherwise} \end{cases}, \Phi(x) = 1, \sigma(x) = 1$$

Our model thus reads

$$(I_b(x) + 1)u + \omega \nabla_x \cdot u - \frac{1}{4\pi}\frac{1}{2}\int_S u(x, \omega')d\omega' = I_b^2(x)$$

Reading out the solution amounts to computing k-th moments; we're interested in the so-called irradiant radiation, the 0-th moment, given by $G(x)$.

$$G(x) = \int_S u(x, s)ds$$

We tried to keep our parameters as close as possible as suggested in [Mishra and Molinaro, 2021]. Notable differences are that we used only a 6-layer neural net and 8 times fewer nodes, due to performance reasons. Analogously to the 1D case, the scattering integral is an expensive method. We had to optimize this hot path to get acceptable performance for training. We prepared the function evaluations in a convenient way in a 3-tensor and let the einsum function efficiently do contractions in the right dimensions. Here we used a Gauss-Legendre quadrature as was done in the reference paper. It's of order 20 in one dimension amounting to overall 400 model evaluations per training point.

The model is trained with an ADAM optimizer with a learning rate of 5e-4.

1.3 Results

1.3.1 1D case

The model was trained with 4096 internal nodes, 4096 nodes for our additional set of points as well as 2048 boundary points. As in the paper, we applied a depth of 8. The following results were obtained after one hour of training on Google Colab (CPUs only as GPUs did not significantly accelerate execution).
As you can see in plot 1 at the right boundary the values are similar to those presented in figure 3 in
[Mishra and Molinaro, 2021]. One discrepancy in the left plot is that the values of u rise to one already for μ smaller than 0. We observed that this issue was not present during the first 100 epochs but could not stop training this early as the overall loss value was still too high.

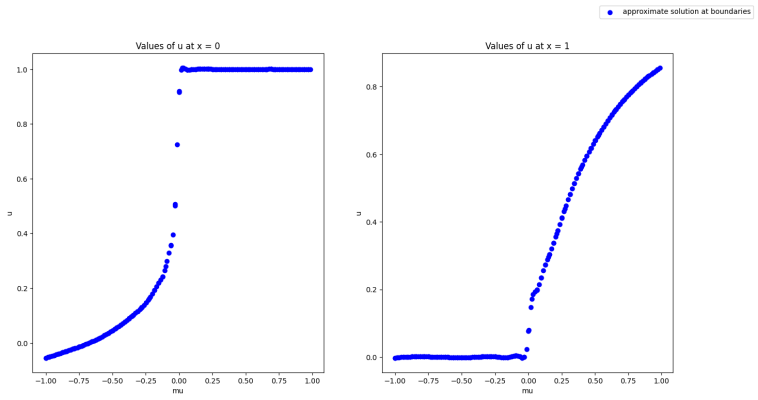

Figure 1: PINN radiative intensity at the physical domain boundaries for the stationary monochromatic radiative transfer in one space dimension.

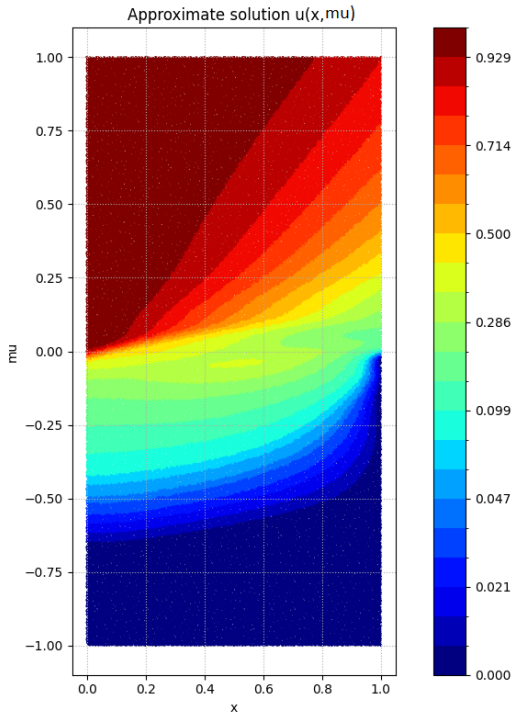

Figure 2: Contour plot of the radiative intensity $u(x, \mu)$ for the 1D monochromatic case.

The contour plot on the full domain is depicted in figure 2. Comparing it to figure 2 in [Mishra and Molinaro, 2021] we observe again that the values of u at the left physical boundary for μ between 0 and -0.25 are too high.

Another observation is that at the top left and bottom right corners there are clearly more values respectively equal to one and zero than in the paper. The contour lines, however, predominantly do behave as in the paper. Except for those two observations, the contour plot looks close to the one in [Mishra and Molinaro, 2021].

1.3.2 3D case

The model was trained with 2048 internal and 1536 external nodes for 20000 epochs on a GPU. This made it more feasible to run the model with more nodes. The model output was the irradiant radiation, as was done in [Mishra and Molinaro, 2021]. We observe that the results are in the same order of magnitude and somewhat close. The differences are that the surface of our sphere is not exactly zero everywhere. This is most likely because we used fewer nodes. The most perceivable difference is that the center of our model has higher values than theirs which may be just an artifact of having fewer nodes.

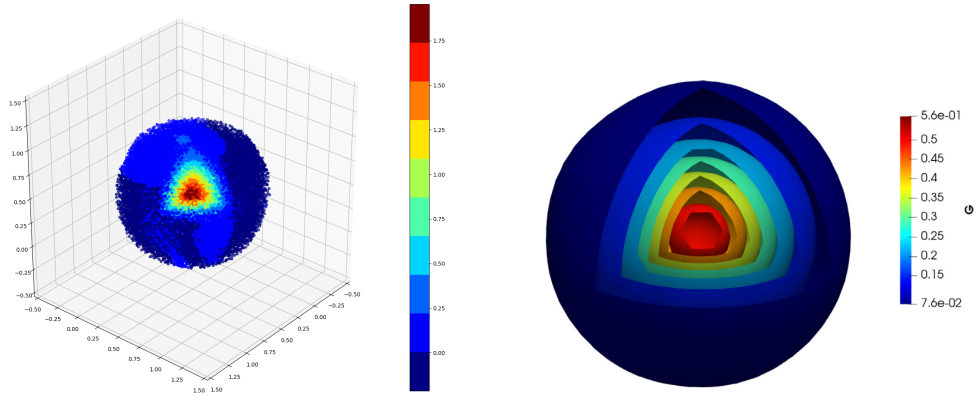

Figure 3: $G(x)$. left: ours; right: [Mishra and Molinaro, 2021]

1.4 Individual team member's contribution

All-in-all this was a successful team project with many interesting discussions. We distributed the workload between us but were still able to discuss everyone's questions together. Here is an overall summary of each team member's contribution. Konrad: proof of concept for both 1D and 3D; analytical development for both; profiling for 1D; kernel optimization for both 1D and 3D cases; visualizations.

Claudio mainly worked on the 3D case, GPU implementation, kernel optimization and profiling, verification of the analytical work, and summarizing it for the presentation in the report.

Florian mainly worked on the 1D case starting from Konrad's first version. This includes adapting the functions already present in the PINN class, implementing new ones (like the build_mat() and optimized kernel for example), fine-tuning the network, and extending the plotting function to the three in the final code.

The report was created collaboratively.

References

[Grella, 2013] Grella, K. (2013). *Sparse tensor approximation for radiative transport*. Doctoral Thesis, ETH Zurich. Accepted: 2017-11-03T14:09:51Z.

[Mishra and Molinaro, 2021] Mishra, S. and Molinaro, R. (2021). Physics Informed Neural Networks for Simulating Radiative Transfer. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 270:107705. arXiv:2009.13291 [cs, stat].

[Pontaza and Reddy, 2005] Pontaza, J. and Reddy, J. (2005). Least-squares finite element formulations for one-dimensional radiative transfer. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 95(3):387–406.