

DLSC Project B

Thomas Baumann, Niklas Dahlmeier, Nasib Naimi, Matthias Sanicanin

July 15, 2023

1 Introduction

In this section we will briefly discuss the main methods from the paper "Respecting causality is all you need for training physics-informed neural networks" by Sifan Wang, Shyam Sankaran, and Paris Perdikaris [1]. The authors propose that the reason why historically PINNs were unable to accurately predict the solutions of more complex differential equations lies in the inability to respect causality. They give an intuitive example to showcase said causality - a linear wave spreading across a homogeneous medium.

Through some reformulations of the standard loss terms of PINNs they show that the minimization of the loss terms $\mathcal{L}(t_i, \theta)$ at time t_i is not only reliant on an accurate prediction of the solution for time t_i but also for the previous time step t_{i-1} . However, traditionally, all loss terms $\mathcal{L}(t_i, \theta)$ were minimized simultaneously, which can lead to wrong solutions. The authors provide some further insights into this problem, after which they propose a new method in order to respect causality.

1.1 How to respect causality

The authors propose a weighted residual loss $\mathcal{L}_r(\theta) = \frac{1}{N_t} \sum_{i=1}^{N_t} w_i \mathcal{L}_r(t_i, \theta)$ where the weights w_i should change over time in such a manner that $\mathcal{L}_r(t_i, \theta)$ is only really minimized if and only if all previous $\{\mathcal{L}_r(t_j, \theta)\}_{j=1}^{i-1}$ have been minimized properly. The weights w_i are chosen as follows: $w_i = \exp(-\varepsilon \sum_{k=1}^{i-1} \mathcal{L}_r(t_k, \theta))$ This choice of w_i leads to very small weights (meaning the losses will not be minimized as strongly) as long as not all $\mathcal{L}_r(t_j, \theta)$ have been decreased to a small value for all $j < i$.

The general idea of adaptive weights is not novel as can be seen in [2] for example, however this specific formulation of adaptive weights, w_i , is due to Wang et al.

After introducing the theory behind the approach, the authors then apply this to several differential equations including some that have - to their knowledge - never been solved successfully by PINNs, for example the Kuramoto–Sivashinsky equation in the chaotic regime or the Navier-Stokes equations in the turbulent regime.

2 PDEs and Solution Approach

We present a case study of the causal loss on PINNs for various PDEs. The loss for each PDE is formulated in two ways. The first is the standard PINN loss that does not enforce causality and serves as the baseline for the solution of the PDE. The second is the loss which attempts to enforce causality, as was presented in the introduction section.

For every PDE we select the boundary and initial conditions for which a reference solution can also be found. We first solve using the classical PINN formulation, as was done in the previous projects of the course and compare it to the results of the approach defined in [1], i.e. causality respecting PINNs. For all PINNs, the template code of tutorial 4 was used as a starting point and adjusted to our needs. The main change for the causal PINNs was incorporating the weights, which must be removed from the optimization process (or in their words: "prevent gradient back-propagation"), and the looping over different values for ε .

2.1 Allen-Cahn

Consider the one-dimensional Allen-Cahn equation:

$$u_t - 0.0001u_{xx} + 5u^3 - 5u = 0, \quad t \in [0, 1], x \in [-1, 1].$$

With boundary conditions $u(t, -1) = u(t, 1)$, $u_x(t, -1) = u_x(t, 1)$ and initial condition $u(x, 0) = x^2 \cos(\pi x)$. The boundary condition is strictly imposed by embedding the input coordinates into a Fourier expansion, as was done by Wang et al[1]. The loss is then reduced to only contain the loss term given by the initial condition and the loss term given by the residual of the interior points. See discussion above eq. 2.12 in Wang et al[1]. for further information.

For the baseline we used an MLP with 6 layers with 128 neurons each, which we trained with LBFGS with 20000 iterations and were able to achieve better results than Wang et al[1] in their baseline. For the causal loss PINN our MLP has 5 layers with 120 neurons, we used time marching with multiple tolerances as proposed in the paper, we further used ADAM as an optimizer with a learning rate of 0.001 and a StepLR scheduler.

Below, the plots of the solutions corresponding to the standard PINN formulation, the causal PINN and the reference solution can be seen.

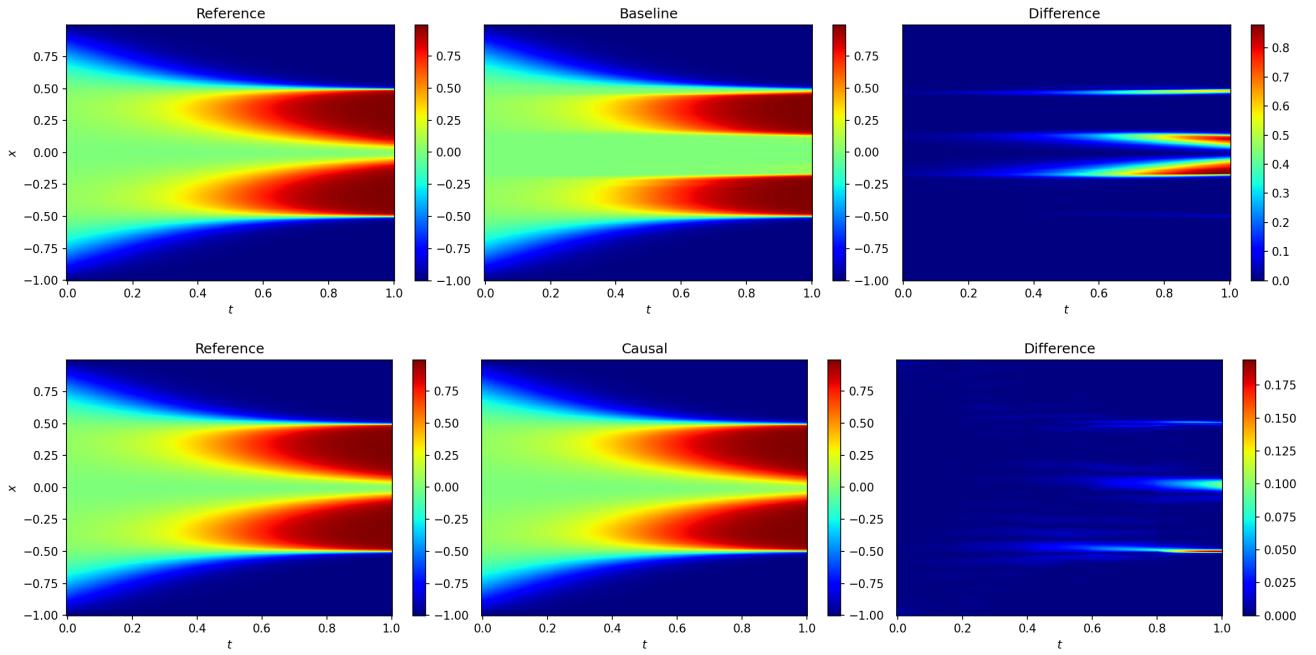


Figure 1: *Allen-Cahn equation*: Top: Reference solution versus the prediction of the baseline. Bottom: Reference solution versus the prediction of the PINN with causal loss and time marching.

2.2 Burgers

We looked at the Burger's equation

$$u_t + uu_x - \nu u_{xx} = 0, \quad t \in [0, 1], x \in [-1, 1]$$

with Dirichlet boundary conditions $u(-1, t) = u(1, t) = 0$ and the initial conditions $u(x, 0) = -\sin(\pi x)$.

Here we compared the performance of non-causal PINNs to the performance of a PINN with causal loss function. Specifically, we analysed two non-causal PINNs that were trained with LBFGS and ADAM, respectively, and a PINN with causal loss that was trained with ADAM. All models had 7 layers with 40 neurons and the ADAM training ran for 10'000 epochs, while LBFGS was limited to 1000 iterations. The causal PINN was trained with tolerance $\varepsilon = 10$.

We compared the approximate solutions from our respective PINNs to a reference solution we found here. Below, we included a plot of the reference solution as well and differences of the reference to the approximate solution of the non-causal and the causal PINNs that were trained with ADAM. The submission files contain a more complete set of plots, including comparisons between prediction and reference at fixed time points $t = 0, 0.5, 1$.

Hyperparameter tuning and more sophisticated training schemes, such as time marching, proved to be unnecessary in this case and it seemed as though the problem could be solved accurately by non-causal PINNs. (The non-causal model that was trained with LBFGS in particular matched the reference most closely, see files).

2.3 Korteweg–De Vries

Korteweg–De Vries (KdV) equation was also solved using the standard and causal approach. The PDE is given by

$$u_t + u_{xxx} - 6uu_x = 0, \quad t \in [0, 3], x \in [0, 3],$$

In order to compare the solutions learned by the PINN and the Causal PINN, we chose to select the initial condition and boundary condition that yield the exact single soliton solution. The exact solution to the single soliton is

$$\frac{-c}{2\cosh^2(\frac{1}{2}\sqrt{c}(x - ct + C))}$$

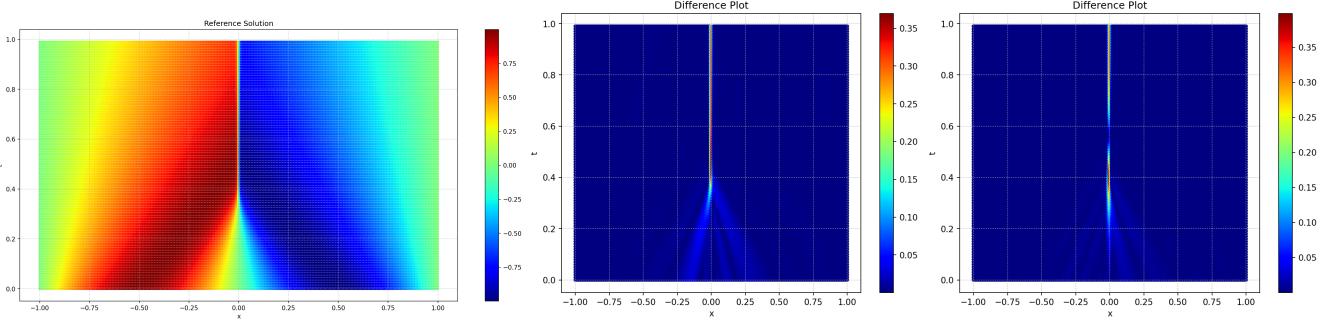


Figure 2: Left: Reference Solution for Burgers Equation, Middle: Difference plot from a classical PINN after 10'000 epochs, Right: Difference plot from a causality respecting PINN after 10'000 epochs with $\varepsilon = 10$

Selecting $c = 1$ and $C = 0$, initial and boundary conditions become $u(0, t) = \frac{-1}{2\cosh^2(-\frac{1}{2}t)}$ and $u(x, 0) = \frac{-1}{2\cosh^2(\frac{1}{2}x)}$. For the kdv equation, we were able to achieve very good results using the LBFGS optimizer that converged very quickly compared to training with ADAM. The LBFGS optimizer we limited the number of iterations to 1000 and were able to achieve close results for both the standard PINN formulation and the causal PINN with a tolerance of $\varepsilon = 1$. In both cases a standard MLP with width 128 and 4 hidden layers was used. The learned solution to the problem was compared to the exact solution of the PDE for the soliton case given above. In Figure 3, the exact solution is given together with the difference between the exact solution and the PINN solutions (standard and causal).

This problem may be considered sufficiently simple for a standard PINN to solve it, hence why we saw little difference in the final predictions when varying the tolerance factor for the causal loss. Using ADAM with more tolerance steps for the causal loss may improve, however we were constrained by compute and were unable to run ADAM for long enough to get a competitive result.

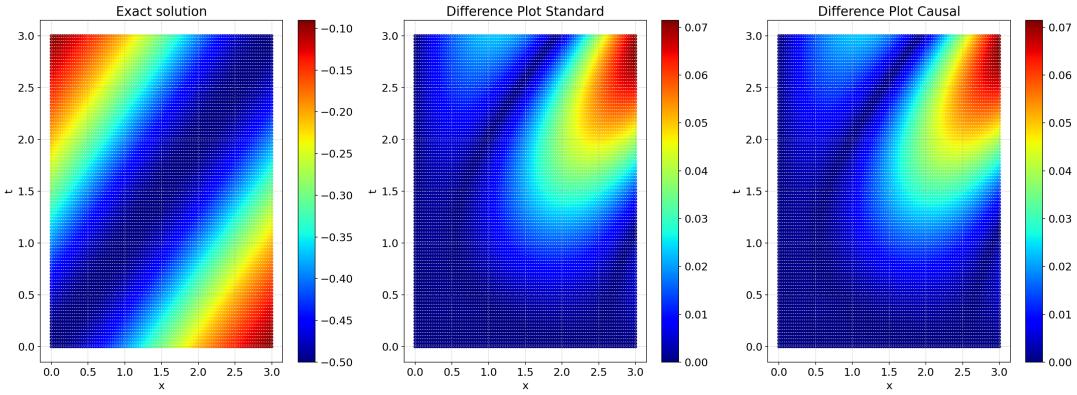


Figure 3: Left: Exact solution for the single soliton case of the KdV equation, Middle: Difference plot between the classical PINN and the exact solution, Right: Difference plot between causality respecting PINN trained with $\varepsilon = 1$ and exact solution

2.4 Kuramoto-Sivashinsky

Finally, we consider the one-dimensional Kuramoto-Sivashinsky equation

$$u_t + \alpha uu_x + \beta u_{xx} + \gamma u_{xxxx} = 0, \quad t \in [0, 1], \quad x \in [-1, 1].$$

Where $\alpha = 5, \beta = 0.5, \gamma = 0.005$ and with periodic boundary conditions and initial condition $u(0, x) = -\sin(\pi x)$. Similarly to the previous equations we tried both causal and non causal PINNs. The computational complexity of this problem was very high, which is at least partly due to us computing fourth order derivatives in the loss function, so we were unable to predict the solution on the entire domain, however, we were able to approximate the solution on the first 3 time frames, as we used time marching. Specifically, we used a set of 10 neural nets, one for each time frame $[t_{i-1}, t_i]$ for $t_i = \frac{i}{10}$, $i \in [10]$, with 5 layers and 120 neurons to predict the solution and were able to get somewhat sensible results for the first 3 intervals by training with Adam for 18'000 (2000 per ε ,

3 times). Similarly to before we enforced the boundary conditions exactly with a Fourier feature embedding. The method was introduced by Dong et al. [3]. The Figure 4 shows our approximate solution, trained for the first 3 time intervals. The model described above was our only successful attempt, non-causal and even non-marching

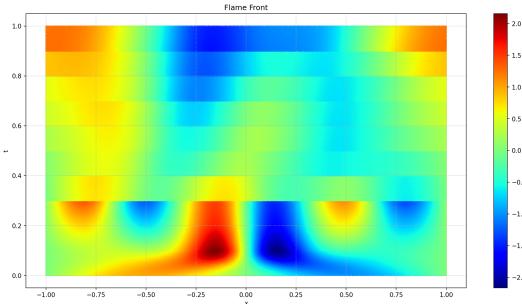


Figure 4: Approximate Solution for KS after 18'000 training epochs, with a list of tolerances $\varepsilon = 0.01, 0.1, 10$.

models were unsuccessful. For the sake of completeness, we decided to enclose our non-marching and non-causal approaches in the submission folder (trial and error).

3 Conclusion and Discussion

With our somewhat limited computational power, the causality respecting PINNs were able to match and slightly exceed the performance of the standard PINNs. However, due to computational restraints, we were unable to improve much on the standards PINNs with the causality respecting PINNS. With more compute resources, we may have been able to achieve even better results with stronger tolerances for the causal loss. Furthermore, we noted that the baseline Allen-Cahn solution we produced far exceeded the baseline that was reported in Wang et al. This may be due to us using LBFGS and fourier encodings of the coordinates.

PDE	Baseline	Causality Loss
Allen-Cahn	0.16680	0.01386
Burgers Equation	0.0004	0.0003
KdV	0.05790	0.05483
Kuramoto	Not applicable	Not applicable

Table 1: Relative L2 error of baseline and the PINN with the causality loss with respect to the reference solution.

4 Labor Division

For each equation, we assigned a group member who was primarily responsible. However, we all contributed to each others PDEs if help was needed or somebody was stuck. Matthias & Niklas were responsible for Burgers equations and Kuramoto-Sivashinsky, Nasib was responsible for kdv equation, and Thomas for the Allen-Cahn equation. The report was written by all group members. We all believe that the work of the entire project was split fairly and equally.

References

- [1] Sifan Wang, Shyam Sankaran, and Paris Perdikaris. Respecting causality is all you need for training physics-informed neural networks, 2022.
- [2] Levi McClenny and Ulisses Braga-Neto. Self-adaptive physics-informed neural networks using a soft attention mechanism, 2022.
- [3] Suchuan Dong and Naxian Ni. A method for representing periodic functions and enforcing exactly periodic boundary conditions with deep neural networks. *Journal of Computational Physics*, 435:110242, jun 2021.