Task 4 - Robustness of PINNs and Transferability

Guido Putignano

April 12, 2024

1 Problem Description

The main subject is a linear elliptic PDE given by the formula

$$\nabla \cdot (A(x)\nabla u(x)) = \sum_{i,j} \frac{\partial}{\partial x_i}\left(a_{ij}(x)\frac{\partial u(x)}{\partial x_j}\right) = f(x) \text{ in } B_1$$

within a domain $\Omega = B_1$, which is the n-dimensional unit ball. Here,

$$A(x) = (a_{ij}(x))_{ij}$$

is a matrix function that satisfies an ellipticity condition. This condition ensures that the matrix $A(x)$ is bounded between two multiples of the identity matrix, controlled by constants λ and Λ, which are strictly positive and finite. The coefficients $a_{ij}(x)$ of $A(x)$ are continuous functions, and the function $f(x)$ is essentially bounded within B_1.
When defining the first problem, it's present the ellipticity condition given by $0 < \lambda\mathrm{Id} \leq A \leq \Lambda\mathrm{Id}$. It guarantees that the PDE is well-posed and that the solutions exhibit certain regularity properties. This condition is crucial for the analysis and solution of elliptic PDEs, as it ensures the operator is neither degenerate nor too irregular.
The Schauder estimates is also described in the text given it applies to any solution $u \in C^2(B_1)$ of the elliptic PDE. The reason why it has been describes is because it provides bounds on the norms of the solutions in terms of the norm of the function f and the solution u itself, within the specified domain. Specifically, the estimate given by

$$\|u\|_{C^1(B_{1/2})} \leq C_\epsilon(\|u\|_{L^\infty(B_1)} + \|f\|_{L^\infty(B_1)})$$

relates the C^1 norm of u on a smaller ball $B_{1/2}$ to the L^∞ norms of u and f on the larger ball B_1, with C_ϵ being a constant dependent on several parameters including n, ϵ, λ, Λ, and the coefficients a_{ij}.
Given a bound of the norms of the solutions, it's possible to understand what would be the effects when there is an introduction of u and u_δ that are solutions

to the original and a perturbed PDE. In this case, there is also the perturbation in the forcing term f given by an addition of $\delta \cdot \phi$, where ϕ is a smooth function supported in B_1 and bounded by 1. This setup is common in analysis to understand the sensitivity of solutions to changes in the input data or parameters.

2 Establish some bound

To establish a bound on $\|u_\theta - u_\delta\|_{C^1(B_{1/2})}$, where u_θ is a solution obtained from a trained Physics-Informed Neural Network (PINN) and u_δ is a solution to a perturbed version of the original PDE we can follow steps:

1. **Differential Operator Linearity:** The first step is to consider the linearity of the differential operator that allows us to consider the difference between solutions directly. If u and u_δ are solutions to their respective PDEs, then $u - u_\delta$ satisfies the PDE associated with the difference of the forcing terms, $f - f_\delta$, due to the linearity. Indeed, PINNs aim to minimize the residual $R = D(u_\theta) - f$.

2. **PDE for the Difference:** The PDE satisfied by $u - u_\delta$ is

$$\nabla \cdot (A(x)\nabla(u - u_\delta)) = f - f_\delta$$

with $f_\delta = f + \delta \cdot \phi$, so the PDE becomes

$$\nabla \cdot (A(x)\nabla(u - u_\delta)) = f - f - \delta \cdot \phi = -\delta \cdot \phi$$

3. **Applying Schauder Estimates:** By applying the Schauder estimate to $u - u_\delta$, we obtain:

$$\|u - u_\delta\|_{C^1(B_{1/2})} \leq C_\epsilon \|u - u_\delta\|_{L^\infty(B_1)} + C_\epsilon \delta$$

In this way is possible to ensuring small total errors through careful management of the PDE residual.

4. **Bounding $u_\theta - u_\delta$:** To relate u_θ to u and u_δ, we use the triangle inequality:

$$\|u_\theta - u_\delta\|_{C^1(B_{1/2})} \leq \|u_\theta - u\|_{C^1(B_{1/2})} + \|u - u_\delta\|_{C^1(B_{1/2})}$$

Given $\|u - u_\theta\|_{C^1(B_1)} \leq \epsilon$, we can bound the first term. The second term can be bounded by the result from step 3. In this case, using the coercivity of PDEs and the bounds on total error in terms of residuals, we provide a robust theoretical backdrop to our bounding strategy.

5. **Combining Bounds:** By substituting our bounds into the inequality, we get:

$$\|u_\theta - u_\delta\|_{C^1(B_{1/2})} \leq \epsilon + C_\epsilon \|u - u_\delta\|_{L^\infty(B_1)} + C_\epsilon \delta$$

Numerical Results: (SM, Molinaro, Tanios, 2021)

- Heat Equation:

Dimension	Training Error	Total error
20	0.006	0.79%
50	0.006	1.5%
100	0.004	2.6%

- Black-Scholes type PDE with Uncorrelated Noise:

Dimension	Training Error	Total error
20	0.0016	1.0%
50	0.0031	1.5%
100	0.0031	1.8%

- Heston option-pricing PDE

Dimension	Training Error	Total error
20	0.0064	1.0%
50	0.0037	1.3%
100	0.0032	1.4%

Figure 1: Description of Total Error depending on the dimension, to describe the importance of dimensions apart from data.

3 Robustness to Perturbations

Robustness is another important parameter when considering PINNs, as PINNs tend to be excellent at generalising solutions with the lack of the data. Indeed, especially as it has been formulated, PINNs work in an unsupervised way so that there is no training data that are needed. The training is mainly based on the loss function. Regarding the training of the network, and the number of points that are needed, much may depend on the level of depth we want to delve into. Indeed, given the use of Hoeffding's inequality and Lipschitz bounds on u_θ, we consider the approximation of the generalization error $EG(\theta)$ in terms of the training error $ET(\theta)$ and network parameters. The relationship can be expressed as:

$$EG(\theta) \approx O\left(ET(\theta) + \frac{C\left(M, \log(\|W\|)\right)\log(\sqrt{N})}{\sqrt{N}}\right) \tag{1}$$

where C is a constant dependent on M (a parameter reflecting model complexity or depth) and the logarithm of the norm of the weight matrix W, and N represents the number of training samples or network width.
The impact of perturbations on the solution accuracy depends on the specific problem and the nature of the perturbations. For linear problems or small perturbations, the error might propagate linearly given an approximation of linearity under small perturbations. However, for nonlinear problems or large perturbations, the error can grow non-linearly, potentially leading to significant inaccuracies.

After looking online, several techniques have been found to improve the robustness of PINNs to perturbations and enhance their adaptability:

- **Pre-training on a Range of Scenarios:** By pre-training the network on a diverse set of scenarios that include a wide range of source terms and initial conditions, the PINN can learn a broader representation of the solution space. This approach helps the network to generalize better to new perturbations. One side effect of it is the need of understanding what type of scenarios we may be looking at. Indeed, the main problem is that we don't fully know the possible scenarios we may consider

- **Fine-tuning with Perturbed Data:** Once pre-trained, the network can be fine-tuned on a smaller dataset that includes specific perturbations of interest. This step helps the network adjust its parameters to accommodate the changes introduced by the perturbations, improving accuracy for these specific conditions. A similar approach has also been used in the class when describing Quasi-Monte Carlo methods or other techniques that would follow the same approach of "Local Elevations".

- **Domain Adaptation:** Another techniques is to start with the incorporation of domain-invariant features into the learning process to be sure on what may be constant. When the perturbations lead to solutions in a slightly different domain (e.g., a change in the regime of the physical process), domain adaptation techniques can be employed. An example can be Conditioned PINNs where multiple parameters can also be taken into account.

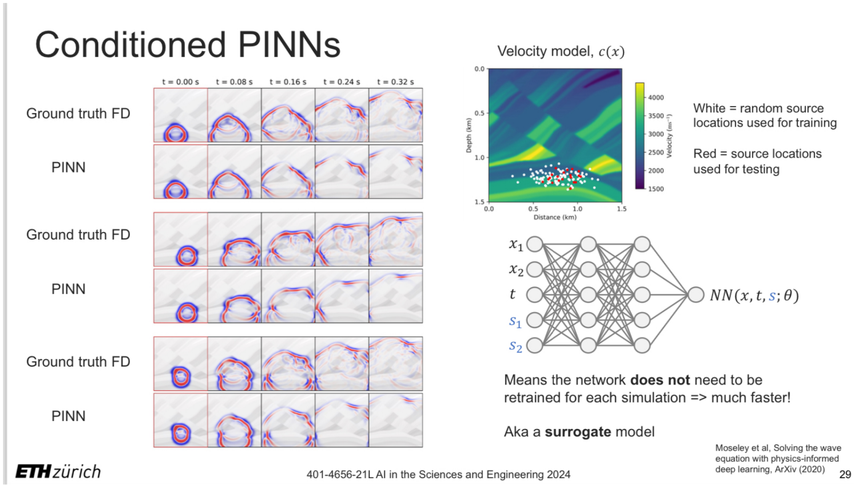

Figure 2: Conditioned PINNs and their influence for generalisation. Reference Included

- **Meta-learning:** What can be called as meta-learning defines an approach involves training the network on a variety of tasks, including solving PDEs with different source terms and initial conditions. The goal is to learn a model that can quickly adapt to a new task (e.g., a new set of perturbations) with minimal additional training. In general this element isn't always possible as there may be the risk of having a PINN to train again depending on the considered conditions. Moreover, as seen during the class having new parameters to train on can't be ideal for the precision of the algorithm

- **Incorporation of Uncertainty Quantification:** Integrating methods for uncertainty quantification within the PINN framework can help in assessing the confidence level of the solutions under perturbations. This can guide the network in focusing its learning on regions of the solution space where the uncertainty is high due to perturbations. Some examples may be the use of Gaussian Processes and all that is related to Probabilistic Artificial Intelligence

4 Impact of the Differential Operator on Robustness

The differential operator is fundamental to directly influences the sensitivity of the problem to such perturbations and the strategies required to ensure robust and accurate solutions. For this reason, its impact depends on multiple component:

Linear vs. Nonlinear Operators: Linear differential operators typically lead to problems where the response to perturbations in source terms or initial conditions is proportional and predictable. Nonlinear operators, on the other hand, can cause solutions to respond to perturbations in a more complex and sometimes unpredictable manner, potentially leading to highly sensitive dependence on initial conditions or source terms.

Order of the Differential Operator: Higher-order differential operators often encapsulate more complex physical phenomena, including those with rapid changes in the solution space. Perturbations in problems governed by higher-order operators may lead to more significant changes in the solution, requiring careful control of input data to ensure stability and accuracy.

Ellipticity and Well-Posedness: Well-posedness is a concept in mathematics, particularly in the theory of partial differential equations (PDEs) and other mathematical problems, that describes a set of criteria a problem must satisfy to be considered properly formulated and solvable in a practical sense. The properties of differential operators, such as ellipticity, affect the well-posedness of the problem. Problems governed by elliptic operators, for instance, often exhibit robustness to small perturbations, as solutions tend to be smooth and unique.

Ellipticity is typically defined in terms of the symbol of the operator. Consider a linear differential operator L of order m acting on a function $u(x)$ as follows:

$$L(u) = \sum_{|\alpha| \leq m} a_\alpha(x) D^\alpha u(x),$$

where D^α denotes partial derivatives, α is a multi-index, and $a_\alpha(x)$ are coefficient functions. The symbol of L, denoted $L(\xi)$, is a polynomial in ξ obtained by replacing each derivative D^α with ξ^α and evaluating at a point x.

The operator L is said to be elliptic if its symbol $L(\xi)$ does not vanish for any nonzero ξ in the dual space, excluding the origin. Mathematically, this condition can be expressed as:

$$L(\xi) \neq 0 \quad \text{for all } \xi \neq 0.$$

5 Control in Higher-Order Sobolev Spaces

Controlling the source terms, initial data, and solution in higher-order Sobolev spaces is beneficial for several reasons:

- **Improved Regularity and Solution Smoothness:** Higher-order Sobolev spaces ($W^{k,p}$ spaces, for instance) involve control over both the function and its derivatives up to order k. Ensuring that the source terms and initial data belong to such spaces often leads to solutions that are smoother and possess higher regularity. This is particularly important for differential equations where the behavior of derivatives is crucial to understanding the physical phenomena being modeled. For example, in fluid dynamics, the Navier-Stokes equations benefit significantly from initial data in higher-order Sobolev spaces, leading to smoother velocity and pressure fields that are more realistic

- **Enhanced Stability and Convergence:** For numerical and computational methods, including PINNs, controlling the input data in higher-order Sobolev spaces can lead to enhanced stability and convergence of the solution. It ensures that the computational method is dealing with functions that have well-defined properties over the domain of interest, reducing the potential for numerical instabilities.

- **Robustness to Perturbations:** When initial data and source terms are controlled in higher-order Sobolev spaces, the resulting solutions are often more robust to perturbations. This robustness is due to the boundedness of derivatives, which limits the rate at which small changes can propagate through the system. This is especially critical in nonlinear systems where responses to perturbations can be amplified. In the context of weather modeling, for example, improved robustness can lead to more reliable forecasts in the face of small observational errors or model uncertainties.

- **Accuracy in Approximation:** In the context of PINNs, ensuring that the solution and its derivatives are accurately represented in higher-order Sobolev spaces allows for a more precise approximation of the underlying physical laws. This is crucial for capturing complex behaviors and ensuring that the PINN can generalize well across different scenarios.

In those compontents, it's also important to consider that the curse of dimensionality is a significant challenge that arises when dealing with higher-order Sobolev spaces, especially as the dimensionality of the problem increases. It states that as the dimensionality of the space increases, the number of degrees of freedom associated with a function in a higher-order Sobolev space grows exponentially. This growth demands significantly more computational resources for both storage and processing. In the context of differential equations, this means that discretization methods (e.g., finite element methods) require a drastically larger number of elements or points to achieve a similar level of accuracy compared to lower-dimensional problems. Moreover, in higher dimensions, data becomes increasingly sparse. This sparsity makes it difficult to ensure that the initial and boundary conditions, as well as the source terms, are well-represented in higher-order Sobolev spaces. For models that rely on empirical data, such as Physics-Informed Neural Networks (PINNs), this can lead to challenges in training the model to accurately reflect the underlying physics, as the model may not have enough data points to learn from across the high-dimensional space. Finally, higher-order Sobolev spaces can lead to models that are overly complex, capturing noise in the data rather than the underlying signal. This complexity makes the model less generalizable and more prone to overfitting, especially when the amount of available data is limited compared to the dimensionality of the space.