QP Based Constrained Optimization for Reliable PINN Training

Alan Williams AWILLIAMS@LANL.GOV
Applied Electrodynamics Group (AOT-AE), Los Alamos National Laboratory

Christopher Leon CLEON@LANL.GOV
Applied Electrodynamics Group (AOT-AE), Los Alamos National Laboratory

Alexander Scheinker ASCHEINK@LANL.GOV
Applied Electrodynamics Group (AOT-AE), Los Alamos National Laboratory

Editors: N. Ozay, L. Balzano, D. Panagou, A. Abate

Abstract

Physics-Informed Neural Networks (PINNs) have emerged as a powerful tool for integrating physics-based constraints and data to address forward and inverse problems in machine learning. Despite their potential, the implementation of PINNs are hampered by several challenges, including issues related to convergence, stability, and the design of neural networks and loss functions. In this paper, we introduce a novel training scheme that addresses these challenges by framing the training process as a constrained optimization problem. Utilizing a quadratic program (QP)-based gradient descent law, our approach simplifies the design of loss functions and guarantees convergences to optimal neural network parameters. This methodology enables dynamic balancing, over the course of training, between data-based loss and a partial differential equation (PDE) residual loss, ensuring an acceptable level of accuracy while prioritizing the minimization of PDE-based loss. We demonstrate the formulation of the constrained PINNs approach with noisy data, in the context of solving Laplace's equation in a capacitor with complex geometry. This work not only advances the capabilities of PINNs but also provides a framework for their training.

Keywords: Optimization, Physics Informed Neural Networks, Constrained Control, Measurement Uncertainty

1. Introduction

Physics-Informed Neural Networks (PINNs) (Raissi et al. (2019)) represent a groundbreaking approach that combines physics constraints, modeled as partial differential equation (PDE) residuals, with gathered data to solve both forward and inverse problems in machine learning (ML). With complete knowledge of the physics and boundary conditions, PINNs can be used to simply solve a PDE, with a loss function given as the sum of the PDE residual and data based losses. But there are also many situations where the PINNs optimization problem may be difficult to formulate when different parts of the loss functions "fight" each other. For example, in experimental contexts, fitting a neural network (NN) to potentially noisy measurements may not exactly agree with the PDE. Or, in a setting where the PDE residual may be an incomplete description of the physics, one may not want to minimize the residual to precisely zero. Network architecture and expressiveness also dictate how small losses can be made, when training. Despite their potential, PINNs face significant challenges (Cuomo et al. (2022)). As Antonion et al. (2024) point out, "Despite significant strides in augmenting PINN capabilities through published works, numerous unresolved issues persist. These encompass a broad spectrum, spanning from theoretical considerations — such as convergence and

stability — to implementation challenges, including boundary condition management, neural network design, general PINN architecture, and optimization aspects."

We propose a training scheme aimed at simplifying the design of the loss function and ensuring convergence and stability of the training algorithm. Our approach frames the training as a constrained optimization problem, leveraging a quadratic program (QP)-based gradient descent law. Essentially, the QP-based gradient descent (QPGD) scheme dynamically balances different loss functions, shifting focus between data-based losses and PDE residual losses. The balancing is performed in a nonsmooth but continuous way, switching between the gradients of the losses. As an early case study, we consider an inverse problem to determine the voltage of a plate in a capacitor, with the potential field described by Laplace's equation, given noisy measurements. We therefore are interested in achieving an acceptable level of data based loss, determined by inherent noise level in the data, while minimizing the PDE based loss under this constraint.

A recently discovered QP based scheme gradient descent scheme, which provides the basis for this work, has been shown to have strong convergence properties (Williams et al. (2022, 2023b,a); Allibhoy and Cortés (2023)), while being able to handle equality and inequality constraints. The design of Williams et al. (2023a) presents a so called "extremum seeking" algorithm, which studies the convergence of the constrained optimization scheme when the gradients of the problem are unknown and must be estimated online — here a single inequality constraint is considered. The work of Allibhoy and Cortés (2023) studies the more general form of the nonlinear programming scheme, proving local stability under several inequality and several equality constraints.

In the machine learning literature, several approaches exist which help mitigate issues for balancing various terms in the loss function of a PINN. A learning rate annealing (Wang et al. (2021)) algorithm seeks to mitigate a particular kind of instability arising from situations in which the gradients corresponding to different elements in the loss function are of different scales, and GradNorm (Chen et al. (2018)) also provides a solution to this problem. ReLoBRaLo (Bischof and Kraus (2021)) develop heuristics which are used to adaptively tune the weights of the total loss function for more desirable training in the case of three benchmark PDEs. Our method, although distinct from the aforementioned work, addresses similar issues by a reformulation of the overall optimization problem and presents an algorithm which solves it.

While PINNs do not guarantee hard physics constraints, but only gently push a neural network's output towards satisfying physics via additional terms in the cost function, Physics Constrained Neural Networks (PCNNs) have been developed with hard physics constraints (Scheinker and Pokharel (2023)). For example, PCNNs have been developed utilizing three-dimensional convolutional neural network-based neural operators with hard physics constraints that enforce the spatiotemporal Maxwell's equations built into the network architectures for electrodynamics by generating vector and scalar potential fields from which the electromagnetic fields are created (Scheinker and Pokharel (2023)). One of the difficulties of the PCNN approach is that each problem is handled with a unique approach, while PINNs are much more easily applied to any physical problem by simply adding PDE-based terms to the training cost function. Our case study is motivated by the explicit presence of uncertainty. While the incorporation of uncertainty into scientific ML has been studied through a variety of techniques (Garcia-Cardona and Scheinker (2024); Rautela et al. (2024); Psaros et al. (2023); Yang and Perdikaris (2019); Zhang et al. (2019)), here we take the approach that we require the network try to satisfy the known physics while encouraging the network to fit the data only up to some error.

2. Problem Statement and Intuition

Consider the constrained optimization problem

$$\min_{\theta} f(\theta) \text{ subject to } \theta \in \mathcal{S}, \tag{1}$$

where

$$\mathcal{S} \coloneqq \{\theta \in \mathbb{R}^n : g(\theta) \leq 0\}. \tag{2}$$

with the objective $f : \mathbb{R}^n \mapsto \mathbb{R}$ and constraint $g : \mathbb{R}^n \mapsto \mathbb{R}$.

The algorithm presented, which solves (1), is inspired by the QP based safe controllers first formulated by Ames et al. (2016). To understand the algorithm, consider first the continuous time dynamics of gradient descent of the objective

$$\dot{\theta}(t) = -\nabla f(\theta) = u(\theta) \tag{3}$$

where $u(\theta)$ is a "nominal" controller solving the unconstrained problem – simply minimizing f. In this approach, one can interpret g as a control barrier function (CBF). In particular, g is a CBF-like function (of the "zeroing" type (Ames et al. (2016))) for (3) with respect to $\mathcal{S}$. (Note that in the work of Ames et al. (2016), they use h to define the constraint and consider feasible points as $h(\theta) \geq 0$.) The equivalent conditions below, if they hold,

$$\nabla g(\theta)^T u(\theta) + cg(\theta) \leq 0 \iff \dot{g} + cg(\theta) \leq 0 \tag{4}$$

imply two important properties: i) $\mathcal{S}$ is forward invariant because $g(\theta) = 0 \implies \dot{g} \leq 0$ and ii) trajectories are attracted to $\mathcal{S}$ because $g(\theta) > 0 \implies \dot{g} < 0$. Of course this condition is not automatically guaranteed by $u(\theta)$ and so one can formulate the QP below to design a control law $u_s = u(\theta) + \bar{u}$ satisfying

$$\bar{u} = \arg\min_{v \in \mathbb{R}^n} ||v||^2 \quad \text{subject to} \quad \nabla g(\theta)^T (u(\theta) + v) + cg(\theta) \leq 0, \tag{5}$$

which finds the smallest additional action to add to the nominal law, to guarantee (4) with respect to $\dot{\theta} = u_s$. If $||\nabla g(\theta)|| \neq 0$, an explicit solution for $\bar{u}$ exists:

$$\bar{u} = -\frac{\max\{-\nabla f(\theta)^T \nabla g(\theta) + cg(\theta), 0\}}{||\nabla g(\theta)||^2} \nabla g(\theta). \tag{6}$$

3. Design and Asymptotic Convergence

QPGD is based upon the discretization of the parameter update law $\dot{\theta} = u(\theta) + \bar{u}$ for $\bar{u}$ in (6), where we take a small "learning rate" as the discretized change in time. With $t \in \mathbb{N}$, learning rate $\gamma > 0$, the discrete update law is

$$\theta^{(t+1)} = \theta^{(t)} - \gamma \left(\nabla f(\theta^{(t)}) + \alpha(\theta^{(t)}) \nabla g(\theta^{(t)}) \right), \tag{7}$$

where $\alpha(\theta^{(t)}) \geq 0$ is

$$\alpha(\theta^{(t)}) = \frac{\max\{-\nabla f(\theta^{(t)})^T \nabla g(\theta^{(t)}) + cg(\theta^{(t)}), 0\}}{\max\{||\nabla g(\theta^{(t)})||^2, \epsilon_\alpha\}} \tag{8}$$

and ∇ represents the gradient, $||\cdot||$ is the l^2-norm, and $\epsilon_\alpha > 0$ is a small constant, avoiding a division by zero. The parameter $c > 0$ governs the rate of the approach to the constraint boundary.

Consider the following set of assumptions:

Assumption 1 *There exists a unique θ^* which solves (1) and if $\nabla f(\theta) = 0$ for $\theta \in \mathcal{S}$, then $\theta = \theta^*$.*

Assumption 2 *$f, g : \mathbb{R}^n \to \mathbb{R}$ are differentiable with Lipschitz continuous gradients, having Lipschitz constants $L_f, L_g > 0$.*

Assumption 3 *The function is g radially unbounded and there exists an $a \in \mathbb{R}^n$ such that $g(a) < 0$.*

Assumption 4 *The gradient $\nabla g(\theta) : \mathbb{R}^n \to \mathbb{R}^n$ does not vanish for values of x in or near the region defined by $g(\theta) \geq 0$. Namely, there exists a pair $\epsilon_g, l_g > 0$ such that*

$$||\nabla g(\theta)|| \geq l_g \text{ for all } \theta \in \{y : g(y) \geq -\epsilon_g\}. \tag{9}$$

For simplicity of the presentation, Assumption 1 precludes the possibility of studying a set of minimizers which solve (1), and assumes there are always nonzero gradients of f on $\mathcal{S}$ (unless $\theta = \theta^*$). With many local minimizers the scheme in (7) will converge locally and not globally. It is also possible to study an unbounded set $\mathcal{S}$, and slightly relax Assumption 3 (see Williams et al. (2024)) to assume that the penalty function associated with (1) has compact levels. Additionally, one can take these assumptions on a *particular* compact level of such a penalty function to achieve local results, which may require nonsmooth tools to perform the analysis (as the penalty function is a nonsmooth function).

The main theoretical result is asymptotic convergence to the optimum from any initial parameter, in a semiglobal sense with respect to the learning rate.

Theorem 1 *Let Assumptions 1 - 4 hold. There exists an ϵ_α^* such that for any $\epsilon_\alpha \in (0, \epsilon_\alpha^*)$ and for any initial parameter $\theta^{(0)} \in \mathbb{R}^n$ there exists a learning rate $\gamma^* > 0$ such that for all $\gamma \in (0, \gamma^*)$ the sequence $\{\theta^{(t)}\}$ asymptotically converges to θ^* as $t \to \infty$ given the the update law (7).*

See appendix for proof sketch. The key idea in proving convergence is the use of a Lyapunov function of the form

$$V_\beta(\theta) = f(\theta) - f^* + \beta \max\{g(\theta), 0\}. \tag{10}$$

This function has a minimum value of 0, at the optimum θ^*, for a sufficiently large β, under the assumptions provided. In optimization literature this function is often referred to as a "penalty function" with recent use in analysis of safe extremum seeking by Williams et al. (2023b, 2024) and gradient flow systems by Allibhoy and Cortés (2023).

On implementation: The popular optimizer Adam (Kingma (2014)) is commonly used to speed up training through the use of momentum and RMSprop. When we compute the term $\nabla f(\theta^{(t)}) + \alpha(\theta^{(t)})\nabla g(\theta^{(t)})$ in (7), we allow Adam, with momentum and RMSProp, to use this "mixed" gradient in order to update the internal exponential moving averages and perform the update to the parameter – this, in addition to a learning rate schedule, speeds up our training in practice.

In the next section, the fully connected neural network and the loss function, likely do not satisfy our assumptions and therefore there may be many solutions, or many unconnected islands of local solutions, and the level set $\{g(\theta) < \bar{g}\}$ may be a collection of disconnected sets (Assumption 4 precludes "islands"). QPGD guarantees approximate invariance of the level $\{g(\theta) < \bar{g}\}$ for $\bar{g} > 0$.

Therefore, it may be desirable to first pretrain the PINNs with a static loss $f(\theta) + \lambda g(\theta)$ to move the parameters close to a particular "island" of $\{g(\theta) < \bar{g}\}$, before implementing (7). It is also possible to clip the value of $\alpha(\theta)$ at some positive number, especially if one does not pretrain. The convergence of (7) may also be guaranteed with additional assumptions when α is clipped, but for simplicity we do not include it in our presentation in this section. If the PINN is not pretrained, and α is not clipped, a smaller learning rate may be required for stability.

4. Case Study: Capacitor with Complex Geometry

PINNs can be adapted to find solutions to real physical systems (Raissi et al. (2020); Cai et al. (2021); Jarolim et al. (2023)) and applied to solving inverse problems, something traditional PDE solvers struggle with. A straightforward formulation would include a loss function with a term corresponding to satisfying the PDE, boundary conditions (partial or full), and a term for fitting the data: $l = l_{\text{PDE}} + l_{\text{BC}} + l_{\text{DATA}}$. One issue with this approach, however, is that given measurement uncertainty, the network is likely being asked to solve an impossible task: satisfy both the known PDE and the noise corrupted data. Given the competing terms in the loss function, the network may sacrifice satisfying the known physics to fit the inaccurate data, possibly leading to overfitting. One may try to solve this by giving greater weight to the PDE term in the loss function, but this suffers from the question of how to choose the weighing hyperparameter and offers no guarantees.

In this case study we consider a 2D capacitor with perfect conducting plates of a complex shape. In the interior, between the upper and lower plates, the scalar potential $\phi : \Omega \mapsto \mathbb{R}$ satisfies the 2D Laplace's equation:

$$\nabla^2 \phi(x, y) = 0, \tag{11}$$

where $\Omega = \{(x, y) \in \mathbb{R}^2 : x \in [-1, 1],\ f_l(x) \leq y \leq f_u(x)\}$. The sides and bottom boundaries are grounded with potential $\phi = 0\ V$, while the top boundary is at some voltage $\phi = V_0$, see Fig. 2:

$$\phi(x, f_u(x)) = V_0, \quad x \in [-1, 1], \tag{12}$$
$$\phi(0, y) = 0\ V, \quad y \in (-1, 0.2), \tag{13}$$
$$\phi(1, y) = 0\ V, \quad y \in (-1, 0.2), \tag{14}$$
$$\phi(x, f_l(x)) = 0\ V, \quad x \in [-1, 1], \tag{15}$$

where $f_u(x) = -\sin(\pi x) + 0.2$ and $f_l(x) = e^{-(x+0.5)^2/0.2}(1 - x^2) - 1$.

If $V_0 = 1\ V$ is known, the above can be solved with the traditional PINN approach using $l(\theta) = l_{\text{PDE}}(\theta) + l_{\text{BC}}(\theta)$, for some parameter weights θ – we use this method to obtain a ground truth solution $\phi_{\text{TRUE}}(x, y)$.

We consider V_0 unknown, with some measurement data of the interior subject to some uncertainty. The goal is to estimate the $\phi_{\text{TRUE}}(x, y)$ field using a network $\phi(x, y; \theta)$ and determine the unknown voltage V_0, known as the "inverse" problem. The measurement data is corrupted by the addition of Gaussian noise:

$$\phi_{\text{LABEL}}(x, y) = \phi_{\text{TRUE}}(x, y) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \delta^2), \tag{16}$$

where ϵ is identically distributed and independent for each measurement, while the uncertainty scale is known: $\delta = 0.1\ V$. In real world problems, the uncertainty scale can often be reasonably estimated.

We construct the losses with the data loss with a p-norm:

$$l_{\text{DATA}}(\theta) = \left(\frac{1}{N_M - 1} \sum_{i=1}^{N_M} |\phi(x_i, y_i; \theta) - \phi_{\text{LABEL},i}(x_i, y_i)|^p \right)^{1/p} \quad \text{for } (x_i, y_i) \in \mathcal{M}, \tag{17}$$

as well as:

$$l_{\text{PDE}}(\theta) = \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} |\nabla^2 \phi(x_i, y_i; \theta)|^2 \text{ for } (x_i, y_i) \in \text{int}(\Omega), \tag{18}$$

$$l_{\text{BC},0}(\theta) = \frac{1}{N_{\text{BC},0}} \sum_{i=1}^{N_{\text{BC},0}} |\phi(x_i, y_i; \theta) - 0|^2 \text{ for } (x_i, y_i) \in \partial\Omega, \tag{19}$$

$$l_{\text{BC, V}}(\theta) = \frac{1}{N_{\text{BC, V}}} \sum_{i=1}^{N_{\text{BC, V}}} |\phi(x_i, y_i; \theta) - \hat{V}_0|^2 \text{ for } (x_i, y_i) \in \partial\Omega, \tag{20}$$

where $N_M = 4$ measurement points from the measurement set $\mathcal{M} \subset \Omega$, $N_{\text{PDE}} = 20,000$ PDE points, and $N_{\text{BC},0} = 300$ boundary condition points for the grounded boundaries (corresponding to (13), (14), (15)), and $N_{\text{BC, V}} = 100$ for upper boundary (corresponding to (12)). The scalar $\hat{V}_0$ is a trainable/learnable parameter, considered as an element of the vector θ which does not contribute to the output of the network $\phi(x, y; \theta)$. For $p = 2$, (17) is the root mean squared error – appropriate for Gaussian errors. For $p = \infty$, (17) is maximum absolute error, which is most appropriate for uniformly distributed errors. We formulate the constrained optimization problem:

$$l(\theta) = l_{\text{PDE}}(\theta) + l_{\text{BC},0}(\theta) + l_{\text{BC,V}}(\theta), \tag{21}$$

$$\text{subject to: } l_{\text{DATA}}(\theta) \leq z\delta, \tag{22}$$

where z is some hyperparameter that quantifies the tolerance for error. For the case of heteroscedastic uncertainties with variable δ_i, (22) can be easily modified: $\left(\frac{1}{N_M} \sum_i^{N_M} |\frac{\phi_i - \phi_{\text{LABEL},i}}{\delta_i}|^p \right)^{1/p} \leq z$. To implement (7) we take $f(\theta) = l(\theta)$ and $g(\theta) = l^2_{\text{DATA}}(\theta) - z^2\delta^2$, using squared version of (22) since $l_{\text{DATA}}(\theta)^2$ is a scaled mean squared error, similar in form to the mean squared errors in (18) - (20).

To test our constraint optimization, we trained a regular PINN (Naive-PINN) with loss $l_{\text{PDE}}(\theta) + l_{\text{BC},0}(\theta) + l_{\text{BC,V}}(\theta) + l_{\text{DATA}}(\theta)^2$ against two networks trained under (21) and (22) for $c = 1$ and $c = 10$, with the latter networks labeled as PINN-QPGD. Since we are given the variance of the noise, we should expect the fit to follow:

$$\frac{1}{N_M - 1} \sum_{i=1}^{N_M} |\phi(x_i, y_i; \theta) - \phi_{\text{LABEL},i}(x_i, y_i)|^2 \lessapprox \delta^2. \tag{23}$$

We therefore choose $p = 2$ and $z = 1$. When the left hand side of 23 is less than δ^2, then the network focuses on minimizing $l(\theta)$ and avoids overfitting to the data. For each network, we used a dense neural network with 4 hidden layers and 64 neurons per layer, along with the learnable parameter $\hat{V}_0$. For the activation function, we used GELU (Gaussian Error Linear Unit) (Hendrycks and Gimpel (2016)): $\text{GELU}(x) = x\Phi(x)$ where $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt$. We found this trained

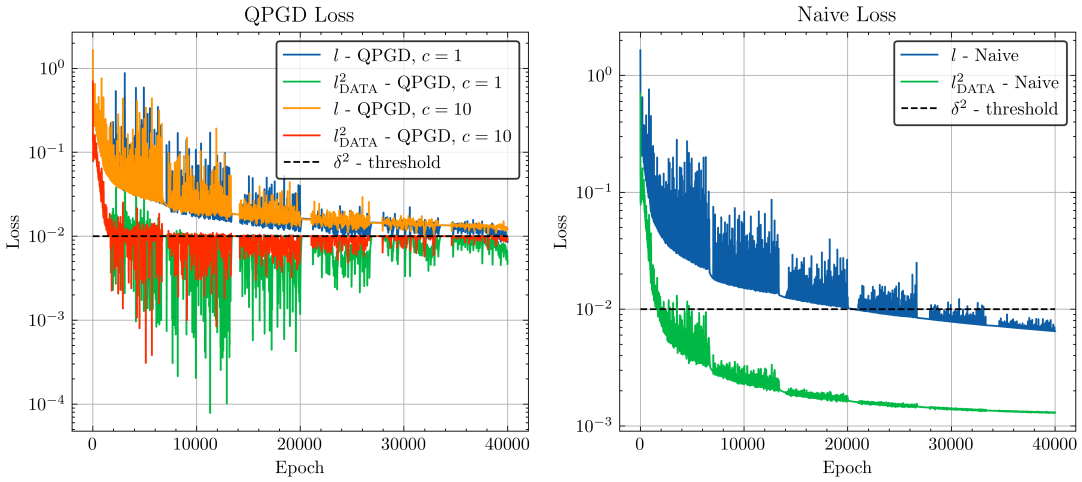

Figure 1: Losses over training for QPGD-PINN and Naive-PINN.

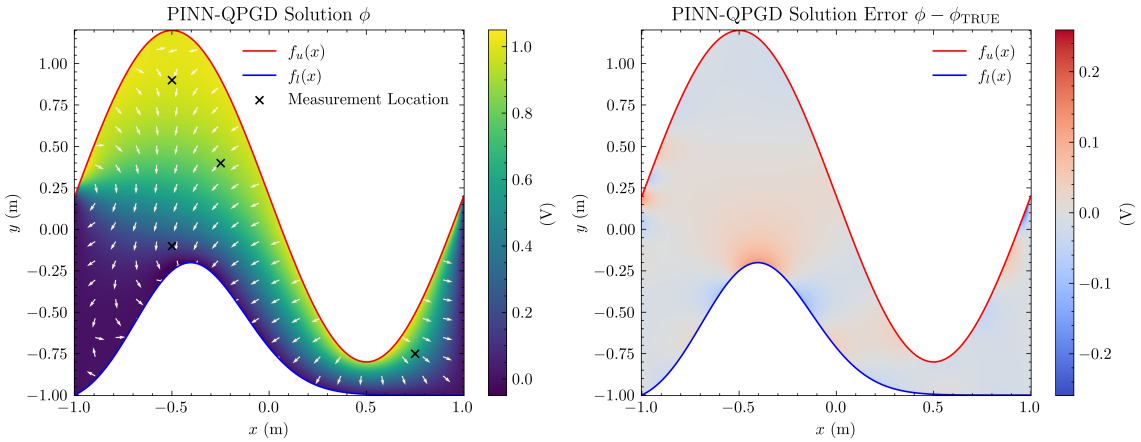

Figure 2: *Left*: QPGD solution with $c = 1$. Arrows illustrate the normalized direction of the predicted electric field. *Right*: QPGD error with $c = 1$.

better than the usual tanh function used in PINNs, and performed better than a sigmoid as well. This is likely due to the fact GELU, unlike the other two activation functions, does not suffer from the vanishing gradient problem. The GELU curve resembles the commonly used RELU in machine learning, but has the advantage that it is smooth.

	PINN-Naive	PINN-QPGD $c = 1$	PINN-QPGD $c = 10$
$\hat{V}_0$	1.035 V	**0.996** V	0.966 V
Average Absolute Error (Interior)	0.029 V	**0.017** V	0.019 V
Average Absolute Laplacian	0.0120 V/m^2	0.0114 V/m^2	**0.0106** V/m^2

Table 1: Results of training. Bold indicates best performance.

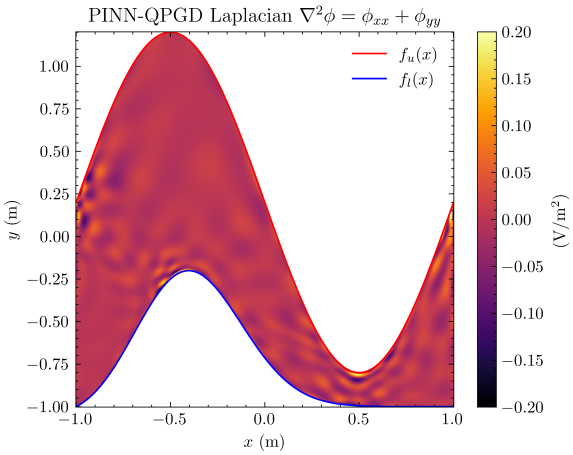

Figure 3: Laplacian $\nabla^2\phi(x, y; \theta)$ plotted over Ω of the QPGD solution with $c = 1$.

All models were trained on identical learning rate schedules, and initialized with the same weights. The optimizer was one commonly used in machine learning, Adam (Kingma (2014)), which uses gradient descent along with momentum and RMSProp. Both models were trained for 40,000 epochs, with the learning rate starting at $\gamma = 4 \times 10^{-3}$ and decreased by half approximately every 6,6667 epochs. The QPGD-PINN models were pretrained with the Naive-PINN loss until l_{DATA^2} reached the value δ^2, around epoch 100.

It is possible to train a QPGD-PINN using the update law (7) without pretraining. We found during our tests that training overall took longer without pretraining. This is likely due to the fact that the set $\{\theta : g(\theta) \leq 0\}$ likely is not connected, and there may be many "islands" that the parameter θ may be driven to due during training – this certainly violates at least Assumptions 1 and 4 in practice. Therefore we recommend pretraining with a naively designed PINNs loss until the threshold value of the constraint is reached.

The potential field predicted by PINN-QPGD and its error can be seen in Fig. 2, while how well it satisfies Laplace's equation can be seen in Fig. 3. As seen in Table 1, the PINN-QPGD outperformed PINN-Naive according to several metrics. The PINN-Naive model overfit the data at the price of sacrificing the known physics of the PDE and boundary conditions. With the QPGD scheme, the network can refine the the physics based loss indefinitely after satisfying the data based loss.

The same methodology was also tested using 4 different measurement points, with similar results. When the number of measurement points was increased to 43, naive PINN and PINN-QPGD were about equally effective – we hypothesize that this is due to the fact that the relative small NN hindered the overfit to such a larger amount of erroneous data. This suggests that PINN-QPGD is most useful when the amount of measurement data is low, or the network is large, where there is a higher risk of overfitting.

5. Conclusion

This study presents a methodology for training PINNs for solving forward and inverse problems, addressing critical challenges that have hindered their application. By developing a training scheme

based on constrained optimization and a QP-based gradient descent law, QPGD, we provide a solution that simplifies the loss function design process and ensures optimal parameter convergence. This approach balances the competing demands of data-based and PDE residual losses, thereby enhancing the performance and applicability of PINNs in solving physics-based problems. We present a case study of solving Laplace's equation and determining the unknown voltage on a capacitor surface, using noisy measurements of the field in between the plates – a situation in which overfitting to noise may lead to a poor estimate of the unknown voltage and estimation of the solution of the potential field. Further work can close the gap between theory and practice, in particular how the inclusion of momentum and RMSProp to gradient descent or the existence of stationary points affects the analysis. Future work will also expand this method by including multiple constraints in the formulation.

Acknowledgments

This work was supported by the Los Alamos National Laboratory (LANL) LDRD Program Directed Research (DR) project 20220074DR.

Appendix

Proofs omitted due to space.

Proposition 2 (Lyapunov Function) *Let Assumptions 1, 2, and 3 hold. Consider a set* $\mathcal{S}_{\bar{g}} = \{g(\theta) \leq \bar{g}\}$ *for some* $\bar{g} > 0$*. There exists a* $\beta^* > 0$ *such that for all* $\beta > \beta^*$ *the function* V_β *in* (10) *has a unique minimizer* θ^* *on* $\mathcal{S}_{\bar{g}}$*.*

Proposition 3 (Existence of Small ϵ_α**)** *Let Assumptions 2, 3, and 4 hold. There exists an* ϵ_α^* *such that for all* $\epsilon_\alpha \in (0, \epsilon_\alpha^*)$*,* $\|\nabla g(\theta)\|^2 \leq \epsilon_\alpha \implies \alpha(\theta) = 0$ *in* (8)*.*

Proposition 4 (Forward Invariance of Positive Levels of g**)** *Let Assumptions 2, 3, and 4 hold and let* $\epsilon_\alpha \in (0, \epsilon_\alpha^*)$ *for* ϵ_α^* *given in Proposition 3. For any* $\bar{g} > 0$ *there exists a* γ^* *such that the set* $\{g(\theta) \leq \bar{g}\}$ *is forward invariant and compact for all* $\gamma \in (0, \gamma^*)$*, under the update law* (7)*.*

Proof Sketch of Main Result: We first state two preliminaries: 1) V_β is locally Lipschitz everywhere (it is composed of Lipschitz functions) with a unique minimum of 0 at θ^* because of Proposition 2, and V_β is radially unbounded by Assumption 3. 2) The right hand side of the update law (7) is locally Lipschitz with respect to $\theta^{(t)}$ given Assumptions 1 - 4 .

Next we choose $\epsilon_\alpha, \bar{g}, \beta$. Let ϵ_α be sufficiently small such that $\|\nabla g(\theta^{(t)})\|^2 \leq \epsilon_\alpha \implies \alpha(\theta^{(t)}) = 0$, which exists by Proposition 3. This allows cases of $\|\nabla g(\theta^{(t)})\|^2 \leq \epsilon_\alpha$, in the arguments below, to be a subset of the case of $\alpha(\theta^{(t)}) = 0$, simplifying the proof and presentation. Intuitively, this is true because that we assume ∇g is not "close" to zero near $g \geq 0$ (Assumption 4), and so, it can be shown that if ∇g is small, then it must be on the interior of the constraint set implying α is zero in this scenario (due to the construction of update law in (8)).

Now choose $\bar{g} > 0$ such that the initial condition $\theta^{(0)} \in \{g(\theta) \leq \bar{g}\}$, an invariant set by Proposition 4. Then choose β sufficiently large such that i) V_β has a unique minimum of 0 at θ^*, by Proposition 2, ii) $\beta \geq \frac{F^*}{l_h} + \frac{c\bar{g}}{2l_h^2} + \frac{1}{2}$ where F^* is the supremum of $\|\nabla f(\theta)\|$ on the compact set $\{\bar{g} \geq g(\theta) \geq 0\}$.

We present the inequalities below, following from expansions and Lipschitzness of the gradients. The points θ^+ and θ are shorthand for $\theta^{(t+1)}$ and $\theta^{(t)}$ respectively. Substituting the update law $\theta^+ - \theta = -\gamma(\nabla f(\theta) + \alpha(\theta)\nabla g(\theta))$:

$$\begin{aligned} f(\theta^+) - f(\theta) &\leq \nabla f(\theta)^T(\theta^+ - \theta) + \frac{1}{2}L_f||\theta^+ - \theta||^2 \\ &\leq -\gamma\nabla f(\theta)^T(\nabla f(\theta) + \alpha(\theta)\nabla g(\theta)) + \frac{1}{2}L_f\gamma^2||\nabla f(\theta) + \alpha(\theta)\nabla g(\theta)||^2 \qquad (24) \\ g(\theta^+) - g(\theta) &\leq \nabla g(\theta)^T(\theta^+ - \theta) + \frac{1}{2}L_g||\theta^+ - \theta||^2 \\ &\leq -\gamma c g(\theta) + \frac{1}{2}L_g\gamma^2||\nabla f(\theta) + \alpha(\theta)\nabla g(\theta)||^2 \qquad (25) \end{aligned}$$

where we use the fact $z - \max\{z, 0\} \leq 0$ and $\|g(\theta)\|^2 \leq \epsilon_\alpha \implies \alpha(\theta) = 0$.

Next we show that V_β is strictly decreasing on the invariant set $\{g(\theta) \leq \bar{g}\}$. Consider the change in V_β under the update law:

$$\Delta V_\beta = V_\beta(\theta^+) - V_\beta(\theta) = f(\theta^+) - f(\theta) + \beta\max\{g(\theta^+), 0\} - \beta\max\{g(\theta), 0\}. \qquad (26)$$

We take two main cases, $\alpha(\theta) = 0$ for **case 1**, and $\alpha(\theta) > 0$ for **case 2**, with each of the two cases consisting of two sub-cases involving the signs of $g(\theta^+)$. Each case makes use of the expansions (24) - (25). We summarize the results in each case.

Case 1i: $g(\theta^+) \leq 0$. Then with $\gamma \leq 1/L_f$:

$$\Delta V_\beta \leq -\frac{1}{2}\gamma\|\nabla f(\theta)\|^2 - \beta\max\{g(\theta), 0\}. \qquad (27)$$

Case 1ii: $g(\theta^+) > 0$ then $\Delta V_\beta = f(\theta^+) - f(\theta) + \beta g(\theta^+) - \beta\max\{g(\theta), 0\}$ and

$$\Delta V_\beta \leq -\frac{1}{2}\gamma\|\nabla f(\theta)\|^2 - \beta c\gamma|g(\theta)| \qquad (28)$$

with $\gamma \leq 1/(L_f + \beta L_g)$ and $\gamma c \leq 1/2$.

Case 2i: $g(\theta^+) \leq 0$. If $g(\theta) < 0$, then with the choice $\gamma \leq 1/L_f$,

$$\Delta V_\beta \leq -\frac{1}{2}\gamma u(\theta) - \frac{1}{2}\gamma\frac{(cg(\theta))^2}{\|\nabla g(\theta)\|^2}. \qquad (29)$$

with $u(\theta) = \nabla f(\theta)^T M(\theta)\nabla f(\theta) \geq 0$ due to the matrix $M(\theta) = I - \frac{\nabla g(\theta)\nabla g(\theta)^T}{||\nabla g(\theta)||^2} \succeq 0$.

If $g(\theta) \geq 0$, with $\gamma c \leq 1/2$ and $\gamma L_f \leq 1$,

$$\Delta V_\beta \leq -\gamma\frac{1}{2}u(\theta) - \frac{1}{2}\gamma c|g(\theta)|. \qquad (30)$$

Case 2ii: $g(\theta^+) > 0$. For $g(\theta) < 0$ with $t \leq 1/L_b$, $L_b = L_f + \beta L_g$,

$$\Delta V_\beta \leq -\gamma\frac{1}{2}u(\theta) - \frac{1}{2}\gamma\frac{(cg(\theta))^2}{\|\nabla g(\theta)\|^2} - \beta\gamma c|g(\theta)|. \qquad (31)$$

For $\bar{g} \geq g(\theta) \geq 0$, and with $\gamma \leq 1/L_b$,

$$\Delta V_\beta \leq -\gamma\frac{1}{2}u(\theta) - \frac{1}{2}\gamma c|g(\theta)|. \qquad (32)$$

Assumption 1 and KKT theory (Proposition 3.3.1, 3.3.4 Bertsekas (1997)) guarantee $\Delta V_\beta < 0$ everywhere except $\theta = \theta^*$. Therefore the update law asymptotically converges to θ^* (Bof et al. (2018)).

References

Ahmed Allibhoy and Jorge Cortés. Control barrier function-based design of gradient flows for constrained nonlinear programming. *IEEE Transactions on Automatic Control*, 2023.

Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8): 3861–3876, 2016.

Klapa Antonion, Xiao Wang, Maziar Raissi, and Laurn Joshie. Machine learning through physics–informed neural networks: Progress and challenges. *Academic Journal of Science and Technology*, 9(1):46–49, 2024.

Dimitri Bertsekas. Nonlinear programming. *Journal of the Operational Research Society*, 48(3): 334–334, 1997.

Rafael Bischof and Michael Kraus. Multi-objective loss balancing for physics-informed deep learning. *arXiv preprint arXiv:2110.09813*, 2021.

Nicoletta Bof, Ruggero Carli, and Luca Schenato. Lyapunov theory for discrete time systems. *arXiv preprint arXiv:1809.05289*, 2018.

Shengze Cai, Zhiping Mao, Zhicheng Wang, Minglang Yin, and George Em Karniadakis. Physics-informed neural networks (pinns) for fluid mechanics: A review. *Acta Mechanica Sinica*, 37(12): 1727–1738, 2021.

Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. In *International conference on machine learning*, pages 794–803. PMLR, 2018.

Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific machine learning through physics–informed neural networks: Where we are and what's next. *Journal of Scientific Computing*, 92(3):88, 2022.

Cristina Garcia-Cardona and Alexander Scheinker. Machine learning surrogate for charged particle beam dynamics with space charge based on a recurrent neural network with aleatoric uncertainty. *Physical Review Accelerators and Beams*, 27(2):024601, 2024.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Robert Jarolim, JK Thalmann, AM Veronig, and Tatiana Podladchikova. Probing the solar coronal magnetic field with physics-informed neural networks. *Nature Astronomy*, 7(10):1171–1179, 2023.

Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Apostolos F Psaros, Xuhui Meng, Zongren Zou, Ling Guo, and George Em Karniadakis. Uncertainty quantification in scientific machine learning: Methods, metrics, and comparisons. *Journal of Computational Physics*, 477:111902, 2023.

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. *Journal of Computational physics*, 378:686–707, 2019.

Maziar Raissi, Alireza Yazdani, and George Em Karniadakis. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science*, 367(6481):1026–1030, 2020.

Mahindra Rautela, Alan Williams, and Alexander Scheinker. Time-inversion of spatiotemporal beam dynamics using uncertainty-aware latent evolution reversal. *arXiv preprint arXiv:2408.07847*, 2024.

Alexander Scheinker and Reeju Pokharel. Physics-constrained 3d convolutional neural networks for electrodynamics. *APL Machine Learning*, 1(2), 2023.

Sifan Wang, Yujun Teng, and Paris Perdikaris. Understanding and mitigating gradient flow pathologies in physics-informed neural networks. *SIAM Journal on Scientific Computing*, 43(5):A3055–A3081, 2021.

Alan Williams, Miroslav Krstić, and Alexander Scheinker. Practically safe extremum seeking. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 1993–1998. IEEE, 2022.

Alan Williams, Miroslav Krstić, and Alexander Scheinker. Semi-global practical extremum seeking with practical safety. In *2023 IEEE 62st Conference on Decision and Control (CDC)*. IEEE, 2023a.

Alan Williams, Alexander Scheinker, En-Chuan Huang, Charles Taylor, and Miroslav Krstic. Experimental safe extremum seeking for accelerators. *arXiv preprint arXiv:2308.15584*, 2023b.

Alan Williams, Miroslav Krstic, and Alexander Scheinker. Semiglobal safety-filtered extremum seeking with unknown cbfs. *IEEE Transactions on Automatic Control*, 2024.

Yibo Yang and Paris Perdikaris. Adversarial uncertainty quantification in physics-informed neural networks. *Journal of Computational Physics*, 394:136–152, 2019.

Dongkun Zhang, Lu Lu, Ling Guo, and George Em Karniadakis. Quantifying total uncertainty in physics-informed neural networks for solving forward and inverse stochastic problems. *Journal of Computational Physics*, 397:108850, 2019.