MATHEMATICAL MODELING AND COMPUTING, Vol. 12, No. 2, pp. 549–557 (2025)

Numerical simulation by Deep Learning for discrete nonlinear problems involving the anisotropic $p(\cdot)$-Laplacian

Joubbi I.[1], Aqel F.[2], Sanou R.[3], Alaa N. E.[1]

[1] *Laboratory LAMAI, Faculty of Science and Technology, Cadi Ayyad University, 40000 Marrakesh, Morocco*

[2] *Laboratory for Monitoring Emerging Technologies (LAVETE), Faculty of Sciences and Technics, Hassan First University, Settat 577 Casablanca Road, Morocco*

[3] *Laboratoire de Mathématiques et Informatique (LAMI), UFR Sciences Exactes et Appliquées, Université Thomas SANKARA, 12 BP 417 Ouaga 12, Ouagadougou, Burkina-Faso*

(Received 5 April 2025; Revised 3 June 2025; Accepted 5 June 2025)

In this paper, we establish the existence of a class of discrete nonlinear systems involving the anisotropic $p(\cdot)$-Laplacian operator using an optimization based approach. We then simulate the solutions by implementing a deep learning model. The numerical results demonstrate that the proposed method is stable and robust compared to conventional approaches such as the Newton–Krylov method.

Keywords: *discrete boundary value problem; deep learning; anisotropic $p(\cdot)$-Laplacian.*

2010 MSC: 47A75, 35B38, 35P30, 34L05, 34L30 **DOI:** 10.23939/mmc2025.02.549

1. Introduction

Nonlinear difference equations have attracted considerable interest in recent years due to their wide-ranging applications in fields such as computer science, economics, neural networks, and cybernetics [1, 2]. These systems often model complex phenomena where discrete changes over time or space play a crucial role. A central challenge in the study of such systems lies in understanding the existence and behavior of weak solutions, particularly when these systems are defined in higher-dimensional Hilbert spaces [4, 5]. While substantial progress has been made in analyzing nonlinear difference equations in one or two dimensions [3, 6, 7], extending these results to 2-dimensional spaces remains a formidable task, primarily due to the increased complexity of difference operators in higher dimensions [8, 9].

In this work, the challenge is addressed by investigating the existence of nontrivial solutions for a nonlinear discrete boundary value problem in a two-dimensional space. The existence of solutions under specific conditions is established using critical point theory, a powerful framework in variational analysis [10, 11]. In this way, the aim is to generalize existing results on nonlinear difference equations to higher dimensions, and highlight the unique difficulties raised by difference operators in multidimensional contexts. Recent work by [11] has demonstrated the existence of non-trivial weak solutions for discrete nonlinear problems, providing a foundation for the present study. Moreover, Ref. [12] has explored multiple solutions for partial discrete Dirichlet problems involving the p-Laplacian, further enriching the theoretical framework.

The specific problem being considered is formulated as follows:

$$\begin{cases} -\Delta_1\big(\phi_{p_1(k-1,h)}(\Delta_1 u(k-1,h))\big) - \Delta_2\big(\phi_{p_2(k,h-1)}(\Delta_2 u(k,h-1))\big) \\ \qquad = g((k,h),u(k,h)) + f(k,h), \quad (k,h) \in \mathbb{N}[1,T_1]\times\mathbb{N}[1,T_2], \\ u(k,0) = u(k,T_2+1) = 0, \quad k \in \mathbb{N}[0,T_1+1], \\ u(0,h) = u(T_1+1,h) = 0, \quad h \in \mathbb{N}[0,T_2+1], \end{cases} \tag{1}$$

where T_1 and T_2 are positive integers, and $\mathbb{N}[1,T_s] = \{1,2,\ldots,T_s\}$ represents a discrete interval. The operators Δ_1 and Δ_2 are forward difference operators defined as:

$$\Delta_1 u(k-1,h) = u(k,h) - u(k-1,h), \quad \Delta_2 u(k,h-1) = u(k,h) - u(k,h-1).$$

The anisotropic $\boldsymbol{p}(\cdot)$-Laplacian operator is defined by

$$\Delta_{\boldsymbol{p}}(u) = \Delta_1\big(\phi_{p_1(k-1,h)}(\Delta_1 u(k-1,h))\big) + \Delta_2\big(\phi_{p_2(k,h-1)}(\Delta_2 u(k,h-1))\big),$$

where $\boldsymbol{p} = (p_1, p_2)$ and $\phi_{p_i(.)} = |t|^{p_i(.)-2}t$. Additionally, the function g is also nonlinear, adding further complexity to the problem. The nonlinearity of g may depend on the discrete variable (k,h) as well as the value of the solution $u(k,h)$ at that point. This dual nonlinearity, both in the operators ϕ_{p_i} and in the function g, makes the analysis and solution of this problem particularly interesting and potentially complex.

The study of such systems is not only interesting from a mathematical point of view, it is also highly relevant to practical applications. For instance, in the context of neural networks, these equations can model the propagation of signals in discrete neural layers [2, 13], while in economics, they can describe dynamic systems with discrete decision-making processes [1]. The ability to solve these systems analytically or numerically is crucial for understanding and predicting the behavior of such systems. Recent advances in deep learning have provided powerful tools for approximating solutions to complex mathematical problems, including nonlinear difference systems [14–17].

In recent years, deep learning has become a groundbreaking method for simulating and approximating solutions to complex mathematical problems, including nonlinear difference systems [2, 14, 15]. Deep learning models, particularly neural networks, excel at capturing intricate patterns and relationships in high-dimensional data, making them well-suited for approximating solutions to systems like (1) [13, 18]. By training neural networks on known solutions or boundary conditions, it is possible to generate accurate approximations for systems where analytical solutions are difficult or impossible to obtain [16, 17]. This approach not only complements traditional mathematical methods but also opens new avenues for exploring the behavior of nonlinear systems in higher dimensions [19, 20].

In this article, we emphasize the importance of combining analytical techniques, such as critical point theory [10, 11], with computational tools like deep learning [2, 14, 15] to tackle the challenges posed by nonlinear difference systems. Recent developments in physics-informed neural networks (PINNs) [19–23] and adaptive activation functions [23] have shown great promise in solving differential equations, including those arising in discrete settings. Additionally, the Krylov method [24, 25] has proven to be highly effective for solving large-scale linear and nonlinear systems, particularly in sparse and high-dimensional settings. By integrating these approaches, we aim to provide a comprehensive framework for studying the existence and properties of solutions in higher-dimensional settings, ultimately advancing our understanding of these complex systems.

In this work, we address the challenges posed by nonlinear difference systems through a comprehensive approach that spans multiple key areas. First, we present some preliminary results in Section 2, which lay the theoretical foundation for our analysis. In Section 3, we establish the existence of solutions to the nonlinear discrete boundary value problem using critical point theory. Moving beyond traditional analytical methods, Section 4 introduces the application of deep learning techniques, particularly PINNs, for solving nonlinear partial differential equations (PDEs). Section 5 discusses the optimization of hyperparameters for PINNs, which is crucial for achieving accurate and efficient solutions. Finally, in Section 6, we extend our results to other nonlinear functions, demonstrating the versatility and robustness of our approach. By covering these diverse yet interconnected topics, this work aims to provide a holistic framework for understanding and solving nonlinear difference systems, bridging the gap between theoretical analysis and computational methods.

2. Some preliminary results

Throughout this section we assume that $p_i\colon \mathbb{N}[0,T_1+1]\times\mathbb{N}[0,T_2+1] \to [2,+\infty)$ for $i=1,2$, such that

$$p_i^- = \min_{(k,h)\in\mathbb{N}[1,T_1]\times\mathbb{N}[1,T_2]} p_i(k,h), \quad p_i^+ = \max_{(k,h)\in\mathbb{N}[1,T_1]\times\mathbb{N}[1,T_2]} p_i(k,h) < +\infty.$$

One defines p^-, p^+ by $p^- = \min(p_1^-, p_2^-)$, $p^+ = \max(p_1^+, p_2^+)$.

The space X is defined as follows:

$$X = \big\{u\colon \mathbb{N}[0,T_1+1]\times\mathbb{N}[0,T_2+1]\to\mathbb{R} \mid u(k,0)=u(k,T_2+1)=0,\ k\in\mathbb{N}[0,T_1+1],\\ u(0,h)=u(T_1+1,h)=0,\ h\in\mathbb{N}[0,T_2+1]\big\},$$

with the norm

$$\|u\| = \left(\sum_{h=1}^{T_2}\sum_{k=1}^{T_1+1}|\Delta_1 u(k-1,h)|^{p^-} + \sum_{k=1}^{T_1}\sum_{h=1}^{T_2+1}|\Delta_2 u(k,h-1)|^{p^-}\right)^{\frac{1}{p^-}}.$$

For the data g we impose the following conditions: for each $(k,h)\in\mathbb{N}[1,T_1]\times\mathbb{N}[1,T_2]$, the function $g((k,h),\cdot)\colon\mathbb{R}\to\mathbb{R}$ is continuous and there exists a constant $\beta>0$ and $\alpha(k,h)$ such that:

$$|g((k,h),\xi)| \leqslant \beta\,|\xi|^{\alpha(k,h)-1} \tag{2}$$

with

$$0<\alpha^- = \min_{(k,h)\in\mathbb{N}[1,T_1]\times\mathbb{N}[1,T_2]}\alpha(k,h), \qquad \alpha^+ = \max_{(k,h)\in\mathbb{N}[1,T_1]\times\mathbb{N}[1,T_2]}\alpha(k,h) < +\infty.$$

We denote

$$G((k,h),r) = \int_0^r g((k,h),s)\,\mathrm{d}s \quad \text{for} \quad (k,h,r)\in\mathbb{N}[1,T_1]\times\mathbb{N}[1,T_2]\times\mathbb{R}.$$

We deduce that

$$|G((k,h),\xi)| \leqslant \tau\,|\xi|^{\alpha(k,h)}, \tag{3}$$

where τ is a nonnegative constant.

We make the following assumption about f:

$$f\colon \mathbb{N}[0,T_1+1]\times\mathbb{N}[0,T_2+1]\longrightarrow\mathbb{R} \quad \text{such that} \quad C_f = \sum_{h=0}^{T_2+1}\sum_{k=0}^{T_1+1}|f(k,h)|<\infty. \tag{4}$$

Lemma 1. *a)* $\max_{(k,h)\in\mathbb{N}[0,T_1+1]\times\mathbb{N}[0,T_2+1]}|u(k,h)| \leqslant \frac{T_1+T_2+2}{4}\|u\|$, $\forall u\in H$.

b) There exist positive constants C_1 such that $\forall u\in H$ with $\|u\|>1$, we have

$$\sum_{h=1}^{T_2}\sum_{k=1}^{T_1+1}|\Delta_1 u(k-1,h)|^{p_1(h,k)} + \sum_{k=1}^{T_1}\sum_{h=1}^{T_2+1}|\Delta_2 u(k,h-1)|^{p_2(h,k)} \geqslant C_1\|u\|^{p^-} - C_2. \tag{5}$$

Proof. See (Ref. [26], Lemma 61 pp. 251) for similar arguments. ■

3. Existence of solutions

In this section, we show the existence of at least a weak solution to problem (1).

We define the energy functional $J\colon X\to\mathbb{R}$ by

$$J(u) = \sum_{h=1}^{T_2}\sum_{k=1}^{T_1+1}\frac{1}{p_1(k,h)}|\Delta_1 u(k-1,h)|^{p_1(k,h)} + \sum_{k=1}^{T_1}\sum_{h=1}^{T_2+1}\frac{1}{p_2(k,h)}|\Delta_2 u(k,h-1)|^{p_2(k,h)}\\ - \sum_{k=1}^{T_1}\sum_{h=1}^{T_2}G\big((k,h),u(k,h)\big) - \sum_{k=1}^{T_1}\sum_{h=1}^{T_2}f(k,h)\,u(k,h).$$

Then J is well-defined and $J\in C^1(X,\mathbb{R})$. Its derivative is

$$\begin{aligned}(J'(u),v) &= \lim_{t\to0}\frac{J(u+tv)-J(u)}{t}\\ &= \sum_{h=1}^{T_2}\sum_{k=1}^{T_1+1}\phi_{p_1}\big(\Delta_1 u(k-1,h)\big)\,\Delta_1 v(k-1,h) + \sum_{k=1}^{T_1}\sum_{h=1}^{T_2+1}\phi_{p_2}\big(\Delta_2 u(k,h-1)\big)\,\Delta_2 v(k,h-1)\\ &\quad - \sum_{k=1}^{T_1}\sum_{h=1}^{T_2}g\big((k,h),u(k,h)\big)\,v(k,h) - \sum_{k=1}^{T_1}\sum_{h=1}^{T_2}f(k,h)\,v(k,h).\end{aligned}$$

Definition 1. *A weak solution of problem (1) is $u \in X$ such that*

$$\sum_{h=1}^{T_2}\sum_{k=1}^{T_1+1} \phi_{p_1(k,h)}\big(\Delta_1 u(k-1,h)\big)\,\Delta_1 v(k-1,h) + \sum_{k=1}^{T_1}\sum_{h=1}^{T_2+1} \phi_{p_2(k,h)}\big(\Delta_2 u(k,h-1)\big)\,\Delta_2 v(k,h-1) - \sum_{k=1}^{T_1}\sum_{h=1}^{T_2} g\big((k,h),u(k,h)\big)\,v(k,h) - \sum_{k=1}^{T_1}\sum_{h=1}^{T_2} f(k,h)\,v(k,h) = 0$$

for every $v \in X$.

Theorem 1. *Assume that (2)–(4) holds and $2 \leqslant \alpha^+ < p^-$ then problem (1) admits at least one weak solution. If in addition, $r \to g(\cdot, r)$ is decreasing, then the solution is unique.*

Proof. Since $J \in C^1(X,R)$ and is weakly lower semicontinuous. The existence of a solution is equivalent to showing that J is coercive. Let us consider $u \in X$ with $\|u\| \geqslant 1$, we deduce from inequalities (5) that

$$J(u) \geqslant \frac{2C_2}{p^+}\|u\|^{p^-} - \frac{2C_3}{p^+} - C_\alpha\big(\|u\|^{\alpha^+} + C_f^2 + \|u\|^2\big). \tag{6}$$

Hence $\alpha^+ < p^-$, J is coercive. Then there exists $u^* \in X$ such that $\inf_{x\in X} J(x) = J(u^*)$ and u^* is also a critical point of J, i.e. $J'(u^*) = 0$. Therefore u^* is a weak solution of (1). Now if we also assume that $r \to g(\cdot, r)$ is decreasing, then J is sctrically convex, so the solution is unique. ∎

4. Introduction to deep learning for solving non-linear PDEs

Partial Differential Equations (PDEs) are simply, equations used to explain diverse physics phenomena, such as heat transfer, fluid dynamics, and electromagnetic fields. The solution process for non-linear PDEs can become more difficult when domain complexity increases and coefficient variability occurs. PINNs have proven to be a powerful tool for solving PDEs. Unlike traditional numerical methods such as finite difference or finite element methods, PINNs use neural networks to approximate solutions. They embed physical laws (PDEs, boundary conditions, and labeled data, if available) directly into the training process as constraints. A key advantage of PINNs is that they do not necessarily require labeled data to approximate PDE solutions, as the governing physics can guide the optimization of the loss function. Below is our approach for implementing PINNs.

Algorithm 1 Algorithm for PINNs.

Statement 1. ***Define*** *the PDE, boundary/initial conditions.*

Statement 2. ***Initialize*** *a neural network $u_\theta(\mathbf{x}, t)$ with parameters θ.*

Statement 3. ***(Optional)*** *Include data points (observations) if available.*

Statement 4. ***Compute*** *PDE residuals and boundary/data errors via automatic differentiation.*

Statement 5. ***Construct*** *the total loss $\mathcal{L} = \mathcal{L}_{data} + \lambda_{phys}\mathcal{L}_{physics} + \mathcal{L}_{BC/IC}$.*

Statement 6. ***Use*** *the Adam optimizer to minimize $\mathcal{L}$ for a specified number of epochs:*
1. *Perform a forward pass, compute $\mathcal{L}$, and backpropagate to update θ.*
2. *Save model checkpoints every M epochs.*
3. *Repeat until the prescribed number of Adam epochs is reached.*

Statement 7. ***Refine*** *the model using LBFGS:*
1. *Initialize LBFGS with the final parameters θ from Adam.*
2. *Minimize $\mathcal{L}$ until convergence or a stopping criterion is met.*

Statement 8. ***Obtain*** *the trained network u_θ^* as the approximate PDE solution.*

5. Recommended hyperparameter settings

We conducted experiments to optimize PINNs for solving the discrete nonlinear p-Laplacian problem by tuning hyperparameters such as neurons per layer, hidden layers, training epochs, and grid size. Our objective was to achieve a balance between computational accuracy and efficiency. The simulations were executed on a machine equipped with an NVIDIA RTX 3050 GPU, 16 GB RAM, and an Intel i5-11400H CPU.

In all experiments, we used a spatially varying exponent

$$p(k,h) = 2 + \frac{1}{1+k^2+h^2},$$

and a quadratic-type nonlinearity

$$g((k,h),u(k,h)) = -0.1\,|u|^{1+\frac{1}{\sqrt{1+k^2+h^2}}}, \qquad f(k,h) = 5.$$

We selected these specific functions to highlight how effectively PINNs can handle spatially varying exponents and nonlinearities in discrete p-Laplacian problems.

5.1. Baseline configuration

The baseline setup was:

— **Architecture**: 4 hidden layers, 128 neurons each.

— **Training**: 30 000 Adam epochs, then 2000 LBFGS epochs.

— **Grid Size**: 12×12 points on $[0,11] \times [0,11]$.

5.2. Varying neurons per layer

To evaluate how the model capacity affects performance, we varied the number of neurons per layer.

Table 1. Results for varying neurons per layer.

Neurons	Adam loss	LBFGS loss	Physics loss	Boundary loss	Time
64	6e-5	6e-5	2e-5	4e-5	228.36 seconds
128	6e-5	1e-5	7e-6	6e-6	295.89 seconds
256	8e-5	4e-6	1e-6	3e-6	294.36 seconds

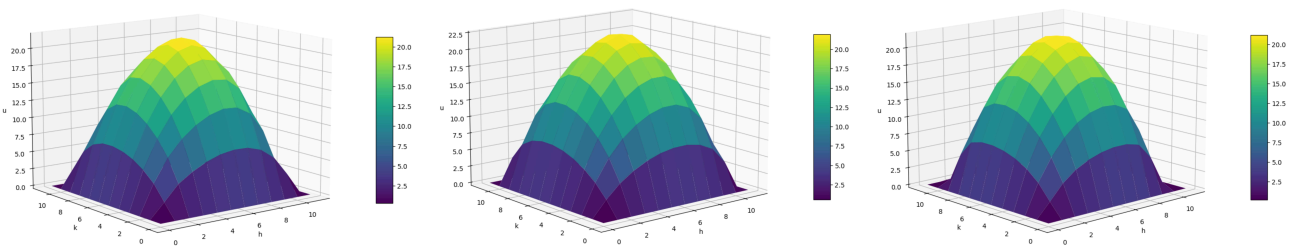

Fig. 1. 3D surface plot for different numbers of neurons per layer(64,128,256).

5.3. Varying grid size (number of training points)

We assessed the effect of training data resolution by varying the grid size used for discretizations.

Table 2. Results for varying Grid Size.

Grid size	Adam loss	LBFGS loss	Physics loss	Boundary loss	Time
7×7	0	0	0	0	510.75 seconds
22×22	2e-3	7e-5	5e-5	2e-5	572.23 seconds
50×50	4e-1	4e-4	3e-4	6e-5	1618.41 seconds

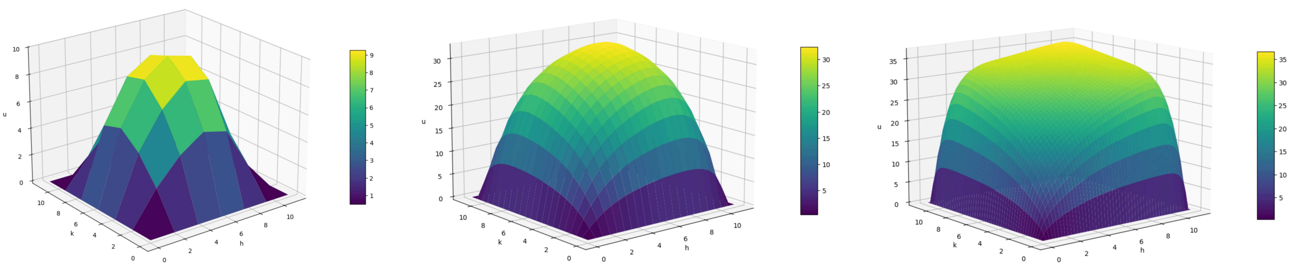

Fig. 2. 3D surface plot for different grid sizes (7×7, 22×22, 50×50).

5.4. Varying hidden layers

Here, we explored the impact of network depth on solution accuracy by varying the number of hidden layers.

Table 3. Results for varying hidden layers.

Layers	Adam loss	LBFGS loss	Physics loss	Boundary loss	Time
2	7e-2	9e-6	7e-6	2e-6	464.49 seconds
3	9e-3	1e-5	2e-6	1e-5	390.94 seconds
4	6e-5	1e-5	7e-6	6e-6	295.89 seconds

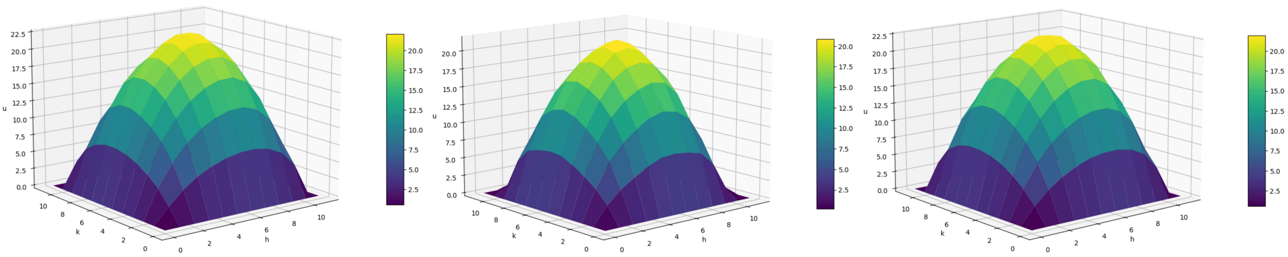

Fig. 3. 3D surface plot for different numbers of hidden layers(2,3,4).

5.5. Varying training epochs

To examine the influence of training duration, we varied the number of Adam epochs before switching to L-BFGS.

Table 4. Results for varying training epochs.

Epochs	Adam loss	LBFGS loss	Physics loss	Boundary loss	Time
20 000	7e-4	1e-5	7e-6	8e-6	263.23 seconds
30 000	6e-5	1e-5	7e-6	6e-6	295.89 seconds
40 000	2e-5	2e-5	2e-5	4e-6	325.58 seconds

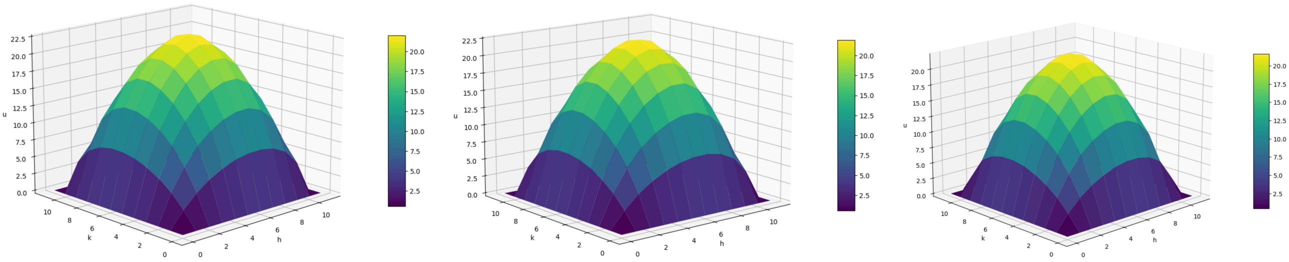

Fig. 4. 3D surface plot for different numbers of training epochs(20k,30k,40k).

5.6. Conclusion of hyperparameter studies

Based on these results, we recommend using:

— **Neurons per layer**: 256.
— **Number of layers**: 4.
— **Grid size**: 22×22.
— **Training epochs**: 40 000 for Adam, followed by 1500 LBFGS epochs.

5.7. Final results with recommended hyperparameters

The figures below show the final solution and training success.

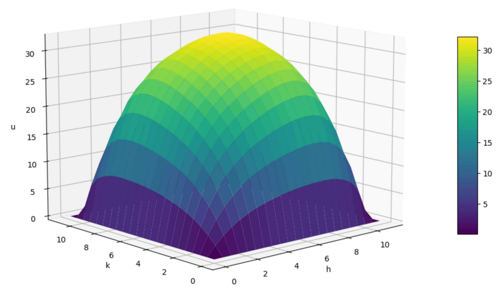

Fig. 5. 3D surface plot of the final PINNs solution.

Fig. 6. Training Loss by Adam and LBFGS.

6. Results for other nonlinear functions

In this section, we showcase two different examples of combined nonlinearity terms. Throughout these examples, we retain:

— The same PINN framework as in the previous sections (four hidden layers, 256 neurons each, and 40 000/1 500 training epochs).
— The same discrete domain and Dirichlet boundary conditions ($u = 0$ on the boundary).
— The same spatially varying exponent

$$p(k, h) = 2 + \frac{1}{1 + k^2 + h^2}.$$

For each example, we display three plots: a **3D surface** of the solution, a **heatmap**, and the **loss** function curve over training.

Example 1. $g((k, h), u(k, h)) = -0.1u^2$, $f(k, h) = 1 + \frac{1}{\sqrt{1+k^2+h^2}}$.
Results:

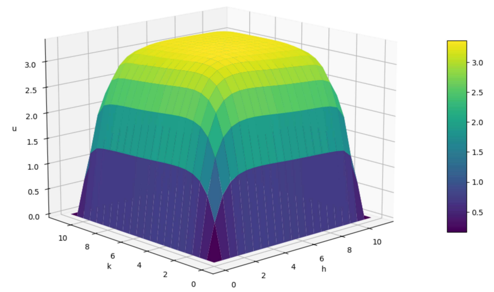

Fig. 7. 3D surface plot of the PINNs solution.

Fig. 8. Training Loss by Adam and LBFGS.

Example 2. $g((k, h), u(k, h)) = -\sin\big(u\big)^2 - 0.6u^3$, $f(k, h) = 10$.
Results:

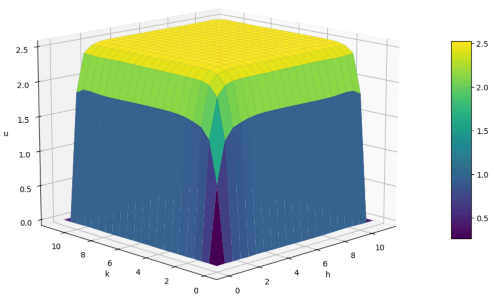

Fig. 9. 3D surface plot of the PINNs solution.

Fig. 10. Training Loss by Adam and LBFGS.

Throughout all the provided examples, the PINN framework consistently exhibited robustness, stable convergence, and accurately captured complex nonlinear behaviors. The provided 3D plots, heatmaps, and training-loss curves highlight the accuracy and stability of this approach.

7. Conclusion and observations

When comparing the traditional *Newton–Krylov solver* approach and our *PINNs* approach to solve the discrete $p(\cdot)$-Laplacian problem, a crucial difference emerges as we refine the grid and increase its size (e.g., from a 12×12 mesh to a 25×25 or beyond). In our numerical experiments:

— **Krylov solver scalability.** For moderate grid sizes, the Krylov-based method converges reasonably fast and provides accurate solutions. However, Increasing the grid density rapidly expands the nonlinear system, since every additional interior point introduces another unknown to solve for.Our numerical experiments indicate that for grid sizes of approximately 25×25 or greater, the Krylov solver either fails or faces significant convergence issues, primarily due to increased system size and nonlinear complexity. Iterative solvers that rely on building or approximating large Jacobians can become prohibitively expensive in terms of both memory and compute time, especially if no sophisticated preconditioning is used.

— **PINns with larger grids.** By contrast, *PINNs* benefit from having more training points when the grid size increases. With additional data points, the neural network has more explicit constraints guiding it toward the true solution. Although training a neural network also requires significant computation, the PINN approach does not require forming and solving a large coupled system of equations in one go. Instead, it minimizes a loss function via gradient-based methods (Adam, LBFGS, etc.). Our experiments suggest that increasing the number of collocation points (grid points) often *improves* PINNs final accuracy. The network effectively leverages the additional data points to improve its approximation of the solution.

In summary, increasing grid resolution can reduce the practicality of traditional Krylov-based methods due to escalating memory demands and system complexity, leading to convergence difficulties. Meanwhile, the *PINNs approach* appears more robust and scales better with additional points, consistently yielding good approximations. This robustness highlights the growing appeal of machine-learning-based methods, especially for high-dimensional or highly nonlinear problems where classical iterative methods may slow down and become costly.

Acknowledgement

The authors wish to thank you, our esteemed referees, for your effort and time spent evaluating our article.

[1] Agarwal R. P. Difference equations and inequalities: Theory, methods and applications. Boca Raton (2000).

[2] Aggarwal C. C. Neural Networks and Deep Learning. A Textbook. Springer Cham (2018).

[3] Cai X., Yu J. Existence theorems for second-order discrete boundary value problems. Journal of Mathematical Analysis and Applications. **320** (2), 649–661 (2006).

[4] Sanou R., Kone B. Weak nontrivial solutions of Dirichlet discrete nonlinear problems. Asia Pacific Journal of Mathematics. **7**, 33 (2020).

[5] Ibrango I., Koné B., Guiro A., Ouaro S. Weak solutions for anisotropic nonlinear discrete Dirichlet boundary value problems in a two-dimensional Hilbert space. Nonlinear Dynamics and Systems Theory. **21** (1), 90–99 (2021).

[6] Tian Y., Ge W. Existence of multiple positive solutions for discrete problems with p-Laplacian via variational methods. Electronic Journal of Differential Equations. **2011** (45), 1–8 (2011).

[7] Zhang G., Liu S. On a class of semipositone discrete boundary value problem. Journal of Mathematical Analysis and Applications. **325** (1), 175–182 (2007).

[8] Galewski M., Wieteska R. Existence and multiplicity of positive solutions for discrete anisotropic equations. Turkish Journal of Mathematics. **38** (2), 297–310 (2014).

[9] Galewski M., Głąb S., Wieteska R. Positive solutions for anisotropic discrete boundary-value problems. Electronic Journal of Differential Equations. **32**, 1–9 (2013).

[10] Zhikov V. Averaging of functionals in the calculus of variations and elasticity. Mathematics of the USSR-Izvestiya. **29**, 33–66 (1987).

[11] Ibrango I., Ouedraogo D., Guiro A. Existence of non trivial weak solutions for discrete non linear problems in n-dimensional HILBERT space. Electronic Journal of Mathematical Analysis and Applications. **11** (1), 190–197 (2023).

[12] Du S., Zhou Z. Multiple Solutions for Partial Discrete Dirichlet Problems Involving the p-Laplacian. Mathematics. **8** (11), 2030 (2020).

[13] Li S., Song W., Fang L., Chen Y., Ghamisi P., Benediktsson J. A. Deep learning for hyperspectral image classification: An overview. IEEE Transactions on Geoscience and Remote Sensing. **57** (9), 6690–6709 (2019).

[14] Nascimento R. G., Fricke K., Viana F. A. C. A tutorial on solving ordinary differential equations using Python and hybrid physics-informed neural network. Engineering Applications of Artificial Intelligence. **96**, 103996 (2020).

[15] Ranade R., Hill C., He H., Maleki A., Chang N., Pathak J. A composable autoencoder-based iterative algorithm for accelerating numerical simulations. Preprint arXiv:2110.03780 (2021).

[16] Pham B., Nguyen T., Nguyen T. T., Nguyen B. T. Solve systems of ordinary differential equations using deep neural networks. 2020 7th NAFOSTED Conference on Information and Computer Science (NICS). 42–47 (2020).

[17] Nam H., Baek K. R., Bu S. Error estimation using neural network technique for solving ordinary differential equations. Advances in Continuous and Discrete Models. **2022**, 45 (2022).

[18] Hornik K., Stinchcombe M., White H. Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks. Neural Networks. **3** (5), 551–560 (1990).

[19] Raissi M., Perdikaris P., Karniadakis G. E. 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).

[20] Raissi M., Perdikaris P., Karniadakis G. E. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. Preprint arXiv:1711.10561 (2017).

[21] Hariri I., Radid A., Rhofir K. Physics-informed neural networks for the reaction-diffusion Brusselator model. Mathematical Modeling and Computing. **11** (2), 448–454 (2024).

[22] Hariri I., Radid A., Rhofir K. Embedding physical laws into Deep Neural Networks for solving generalized Burgers–Huxley equation. Mathematical Modeling and Computing. **11** (2), 505–511 (2024).

[23] Jagtap A. D., Kawaguchi K., Karniadakis G. E. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks. Journal of Computational Physics. **404**, 109136 (2020).

[24] Saad Y. Iterative Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, 2nd edition (2003).

[25] Golub G. H., Van Loan C. F. Matrix Computations. Johns Hopkins University Press, Baltimore, 4th edition (2013).

[26] Radulescu V. D., Repovs D. D. Partial differential equations with variable exponents: variational methods and qualitative analysis. Chapman and Hall/CRC, New York (2015).

Чисельне моделювання дискретних нелінійних задач із анізотропним $\boldsymbol{p}(\cdot)$-лапласіаном за допомогою глибинного навчання

Джуббі І.[1], Акель Ф.[2], Сану Р.[3], Алаа Н. Е.[1]

[1] *Лабораторія LAMAI, Факультет наук і технологій, Університет Каді Айяда, Марракеш, Марокко*
[2] *Лабораторія моніторингу новітніх технологій (LAVETE), Факультет наук та техніки, Університет Хассана Першого, Сеттат 577, Касабланка Роуд, Марокко*
[3] *Лабораторія математики та інформатики (LAMI), Факультет точних та прикладних наук, Університет Томаса САНКАРИ, 12 BP 417 Оуага 12, Уагадугу, Буркіна-Фасо*

У статті доведено існування класу дискретних нелінійних систем із анізотропним $\boldsymbol{p}(\cdot)$-лапласіаном шляхом застосування підходу, заснованого на оптимізації. Здійснено чисельне моделювання розв'язків шляхом реалізації моделі глибинного навчання. Чисельні результати показують, що запропонований метод є стабільним та надійним порівняно з традиційними підходами, такими як метод Ньютона–Крилова.

Ключові слова: *дискретна крайова задача; глибинне навчання; анізотропний* $\boldsymbol{p}(\cdot)$*-лапласіан.*