

# ПЛАНУВАННЯ ЕКСПЕРИМЕНТІВ ТА МЕТОД РОЮ ЧАСТОК ОПТИМІЗАЦІЇ НЕЙРОМЕРЕЖЕВИХ МЕТОДІВ РОЗВ'ЯЗАННЯ КРАЙОВИХ ЗАДАЧ

**Ярош А.О.**

аспірантка

*Запорізький національний університет, Україна*

**Кудін О.В.**

канд. фіз.-мат. наук, доцент, професор кафедри програмної інженерії

*Запорізький національний університет, Україна*

**Вступ.** Крайові задачі є фундаментальними в інженерії через їх ключову роль у моделюванні і аналізі фізичних систем. Вони дозволяють інженерам точно прогнозувати поведінку матеріалів і конструкцій, оцінювати стійкість і надійність інженерних рішень, а також оптимізувати проектування систем і процесів. Завдяки крайовим задачам інженери можуть аналізувати і прогнозувати поведінку систем при різних умовах навантаження, температури, тиску тощо, що допомагає виявляти потенційні проблеми і здійснювати заходи для їх усунення до впровадження проєкту.

Актуальність розвитку існуючих методів розв'язання крайових задач та розробки нових підходів зумовлена зростаючою складністю інженерних і наукових проблем, що потребують високої точності та ефективності в моделюванні. Сучасні технології, такі як матеріалознавство, біоінженерія, енергетика та кліматологія, вимагають розв'язання диференціальних рівнянь зі складною геометрією області визначення та з нелінійними граничними умовами, що часто виходять за рамки можливостей класичних методів. Традиційні підходи, такі як методи Гальоркіна, Рітца та кінцевих різниць, хоча і є потужними, можуть бути обмежені в умовах великої кількості ступенів свободи або високої розмірності задач.

Метод нейронних мереж з фізичною інформацією [1] відрізняється від класичних методів, таких як методи Гальоркіна, Рітца, колокації та кінцевих різниць, тим, що він використовує нейронні мережі для розв'язання диференціальних рівнянь, вбудовуючи фізичні закони безпосередньо у функцію втрат мережі. Класичні методи зазвичай використовують дискретизацію області та апроксимацію розв'язку шляхом побудови базисних функцій, що задовольняють граничні умови. Метод PINN, з іншого боку, не вимагає дискретизації і може працювати з нерегулярними даними та складними геометріями. Перевагами метода PINN є його здатність обробляти складні задачі з нелінійними граничними умовами, природна інтеграція фізичних знань і здатність працювати з обмеженою кількістю даних. Однак, до недоліків відноситься висока обчислювальна вартість навчання нейронної мережі та можливість недостатньої точності в разі неправильного налаштування моделі або вибору гіперпараметрів.

**Огляд бібліотек.** Бібліотека DeepXDE [2] є відкритою, програмний код знаходиться за посиланням <https://github.com/lululxvi/deepxde>. Саме ця бібліотека часто використовується дослідниками для розв'язання математичних моделей різних типів, офіційна документація містить посилання на дослідницькі публікації, які

використовують DeepXDE <https://deepxde.readthedocs.io/en/latest/user/research.html>. Бібліотека містить набір засобів для розв'язання прямих та обернених задач диференціальних та інтегро-диференціальних рівнянь, дробових диференціальних рівнянь [2]. Також реалізовано глибинні нейромережеві оператори (DeepONet) та нейромережеві оператори Фур'є. Використовуються методи покращення збіжності, зокрема, адаптивний вибір пробних точок даних (точок колокації), градієнтний метод, ознаки Фур'є. Бібліотека підтримує можливість моделювати складні форми областей визначення функцій засобами конструктивної твердотільної геометрії [2]. Слід зазначити, що використання бібліотеки потребує знання мов програмування та, що найголовніше, дотримання певного внутрішнього формату запису задачі. Наприклад, для розв'язання звичайного диференціального рівняння другого порядку, необхідно виконати такий програмний код (відповідно до офіційної документації [https://deepxde.readthedocs.io/en/latest/demos/pinn\\_forward/ode.2nd.html](https://deepxde.readthedocs.io/en/latest/demos/pinn_forward/ode.2nd.html)).

Отже, як можна побачити з наведеного прикладу, коректне застосування бібліотеки DeepXDE потребує ґрунтовних знань з програмування та розуміння загальної архітектури.

Бібліотека NeuralPDE [3] (<https://github.com/SciML/NeuralPDE.jl>) реалізована на мові програмування Julia. Відмінність від інших бібліотек PINN, наприклад, DeepXDE, полягає в тому, що вони, зазвичай, вимагають від користувача написання низькорівневого формулювання рівняння в частинних похідних, NeuralPDE абстрагує реалізацію від структури за допомогою символічного інтерфейсу (визначеного через Symbolics.jl). Результатом є те, що запис задачі дуже нагадує математичну постановку. Наприклад, розв'язок двовимірного рівняння Пуассона потребує такого програмного коду: <https://github.com/SciML/NeuralPDE.jl?tab=readme-ov-file#example-solving-2d-poisson-equation-via-physics-informed-neural-networks>.

Бібліотека SimNet (Nvidia Modulus) [4] є потужним засобом моделювання мультифізичних процесів. Підтримує роботу з конструктивною твердотільною геометрією, масивами точок, STL файлами для визначення геометрії області визначення задач. Є можливість безпосередньої роботи з графічними процесорами засобами бібліотеки. З точки зору користувача, процес програмування задачі та її розв'язку може бути складним, у порівнянні з . Наприклад, програмний код для розв'язання одновимірного рівняння поширення хвилі наведено [https://docs.nvidia.com/deeplearning/modulus/modulus-sym/user\\_guide/foundational/1d\\_wave\\_equation.html](https://docs.nvidia.com/deeplearning/modulus/modulus-sym/user_guide/foundational/1d_wave_equation.html).

Бібліотека SciANN [5] (<https://github.com/sciann/sciann>) використовує широко використовувані пакети глибокого навчання TensorFlow і Keras для побудови глибоких нейронних мереж і оптимізаційних моделей, таким чином успадковуючи багато функціональних можливостей Keras, наприклад пакетну оптимізацію та повторне використання моделі для переносу навчання. SciANN розроблено для абстрактної побудови нейронної мережі для наукових обчислень і розв'язання та виявлення диференціальних рівнянь із частинними похідними за допомогою архітектури PINN мереж, що забезпечує гнучкість налаштування складних функціональних форм.

Бібліотека PINNs-Torch [6] (<https://github.com/rezaakb/pinns-torch>) представляє нейронні мережі з фізичними даними, реалізовані за допомогою PyTorch. Відмінною функцією є використання графів CUDA та компіляторів JIT (TorchScript) для компіляції моделей, що призводить до збільшення продуктивності у порівнянні з використанням TensorFlow.

Для дослідників, таких як інженери та математики, важливо мати простий інтерфейс роботи з бібліотекою розв'язання крайових задач засобами PINN.

З порівняння форми запису тестових задач бібліотек DeepXDE, NeuralPDE, Nvidia Modulus, SciANN та PINNs-Torch можна зробити висновок, що формат запису задачі у символічному вигляді засобами NeuralPDE більш наближений до математичної постановки, однак, все ще потребує знання внутрішнього формату запису.

**Методи покращення збіжності.** Розглянемо далі деякі методи покращення збіжності PINN мереж, так звані improved PINN моделі.

Налаштування гіперпараметрів нейромереж, зокрема PINN, є актуальною задачею з кількох причин. Гіперпараметри, такі як кількість шарів, кількість нейронів у кожному шарі, швидкість навчання, тип активаційних функцій та параметри регуляризації, значно впливають на здатність моделі навчатися та узагальнювати дані. Для PINN правильне налаштування гіперпараметрів може значно покращити точність та стабільність моделювання фізичних процесів, забезпечуючи краще задоволення граничних умов і диференціальних рівнянь.

Бібліотеки оптимізації гіперпараметрів [7, 8] дозволяють використовувати поширені методи, зокрема, еволюційні. Одним з підходів є метод рою часток [9].

Метод рою часток (Particle Swarm Optimization, PSO) [9] є метаевристичним алгоритмом оптимізації. PSO використовує популяцію часток, де кожна частка представляє потенційне рішення задачі оптимізації і переміщується в пошуковому просторі, коригуючи свої положення на основі свого власного досвіду та досвіду сусідніх часток. Це дозволяє автоматично налаштовувати гіперпараметри PINN, такі як кількість шарів, кількість нейронів, швидкість навчання та параметри регуляризації, а також безпосередньо оптимізувати ваги нейронної мережі. PSO добре підходить для нелінійних та високорозмірних задач, що часто зустрічаються при розв'язанні диференціальних рівнянь за допомогою PINN, оскільки частки рою можуть ефективно досліджувати великий пошуковий простір і знаходити глобальні оптимуми.

Метод планування експериментів Generalized Subset Designs (GSD) [10] є сучасним підходом до оптимізації та аналізу експериментів, який дозволяє ефективно зменшувати кількість необхідних експериментів, зберігаючи при цьому інформативність та якість отриманих результатів. Цей метод базується на використанні підмножин експериментальних точок, які обираються таким чином, щоб максимально покрити простір можливих факторів і взаємодій між ними.

В роботі запропонована бібліотека, що дозволить розширити практику використання нейронних мереж як обчислювальних методів розв'язання крайових задач. Репозиторій прототипу бібліотеки знаходиться за посиланням <https://github.com/avk256/AutoPINN>. Проведено обчислювальні експерименти з нейромережами PINN та методами покращення збіжності, а саме: методом рою часток та методом планування експериментів GSD.

### Список використаних джерел:

1. Raissi M., Perdikaris P., Karniadakis G.E. Physics-informed neural net-works: A deep learning framework for solving forward and inverse prob-lems involving nonlinear partial differential equations. *Journal of Computa-tional Physics* 378, 2019, 686–707.
2. Lu L., Meng X., Mao Z. and Karniadakis G. Em. DeepXDE: A deep learning library for solving differential equations. *SIAM Review*, volume 63, 1,pp. 208-228, 2021, DOI: 10.1137/19M1274067
3. Zubov K. , McCarthy Z. , Ma Y. , Calisto F., Pagliarino V., Azeglio S., Bottero L., Luján E., Sulzer V., Bharambe A., Vinchhi N., Balakrishnan K., Upadhyay D., Rackauckas C. NeuralPDE: Automating

- Physics-Informed Neural Networks (PINNs) with Error Approximations, 2021. <https://doi.org/10.48550/arXiv.2107.09443>.
4. Hennigh O., Narasimhan S., Nabian M.A., Subramaniam A., Tangsali K., Rietmann M., Aguila Ferrandis J., Byeon W., Fang Z., Choudhry S. NVIDIA SimNet™: an AI-accelerated multi-physics simulation framework, 2020. <https://doi.org/10.48550/arXiv.2012.07938>
  5. Ehsan Haghighat, Ruben Juanes. SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks, *Computer Methods in Applied Mechanics and Engineering*, Volume 373, 2021, 113552, ISSN 0045-7825, <https://doi.org/10.1016/j.cma.2020.113552>.
  6. Reza Akbarian Bafghi and Maziar Raissi. PINNs-Torch: Enhancing Speed and Usability of Physics-Informed Neural Networks with PyTorch, 2023. URL: <https://openreview.net/forum?id=nl1ZzdHpab>
  7. Liaw R., Liang E., Nishihara R., Moritz P., Gonzalez J.E., Stoica I. Tune: A Research Platform for Distributed Model Selection and Training, 2018. arXiv preprint arXiv:1807.05118
  8. Akiba T., Sano S., Yanase T., Ohta T., Koyama M. Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2019. <https://doi.org/10.1145/3292500.3330701>
  9. Simon D. *Evolutionary Optimization Algorithms*. Wiley, 2013. 784 p.
  10. Surowiec I., Vikström L., Hector G., Johansson E., Vikström C., Trygg J. Generalized Subset Designs in Analytical Chemistry. *Anal. Chem.* 2017, 89, 12, 6491–6497. <https://doi.org/10.1021/acs.analchem.7b00506>