

УДК 519.62

МРНТИ 27.35.21; 27.35.47; 27.41.19

<https://doi.org/10.55452/1998-6688-2026-23-3-33-46>

¹*Исахов А.А.,

профессор, ORCID ID: 0000-0002-1937-8615,

*e-mail: alibek.issakhov@gmail.com

¹Бейсембеков М.Е.,

бакалавр, ORCID ID: 0009-0004-5395-4569,

e-mail: m_beisembekov @kbtu.kz

¹Казахстанско-Британский технический университет, г. Алматы, Казахстан

ПРИМЕНЕНИЕ ФИЗИКО-ОРИЕНТИРОВАННЫХ НЕЙРОННЫХ СЕТЕЙ ДЛЯ РЕШЕНИЯ ЗАДАЧ ЛИНЕЙНЫХ УРАВНЕНИЙ В ЧАСТНЫХ ПРОИЗВОДНЫХ С НЕОДНОРОДНЫМИ СМЕШАННЫМИ ГРАНИЧНЫМИ УСЛОВИЯМИ

Аннотация

В работе рассматривается применение Physics-Informed Neural Networks (PINN) для решения двумерного уравнения Пуассона с неоднородными смешанными граничными условиями. Метод PINN объединяет нейросетевые аппроксимации и физические законы, включая невязку уравнения и ошибки на границе в единую функцию потерь. Модель реализована в среде PyTorch и обучалась на выборке, сгенерированной методом латинской гиперкубической выборки. Для проверки точности проведено сравнение с аналитическим решением, а также с традиционными численными методами. Систематически исследовано влияние ключевых гиперпараметров: числа обучающих точек, весов в функции потерь, архитектуры сети и шага обучения. Установлено, что увеличение веса на граничные условия улучшает сходимость и точность, а оптимальные архитектуры с 4–5 скрытыми слоями и 64 нейронами обеспечивают относительную ошибку менее 0.1–0.2%. В отличие от численных схем, требующих сеточной дискретизации, PINN продемонстрировал универсальность и высокую точность при работе с разреженными данными и сложными условиями, что делает его перспективным инструментом для задач вычислительной математики и моделирования физических процессов в условиях сложной геометрии и неполной информации.

Ключевые слова: методы машинного обучения, модель PINN, уравнение Пуассона, функция одиночных потерь, гиперпараметры.

Введение

Развитие методов машинного обучения и нейронных сетей в последние годы оказало значительное влияние на широкий спектр научных и инженерных дисциплин. Современные модели, опирающиеся на глубокое обучение, продемонстрировали высокую эффективность в таких областях, как компьютерное зрение, обработка естественного языка, генерация изображений и прогнозирование временных рядов. Постепенно эти методы начали внедряться в сферу научных вычислений, включая моделирование физических процессов.

Особое внимание привлекли подходы, направленные на решение дифференциальных уравнений, лежащих в основе большинства моделей в физике, механике, теплообмене и других отраслях. Благодаря свойству универсальной аппроксимации нейросети оказались способными приближать решения таких уравнений даже в условиях разреженных данных, сложной геометрии и ограниченной информации о граничных условиях. Эти особенности сделали нейросетевые методы привлекательной альтернативой традиционным численным схемам, таким как метод конечных разностей и метод конечных элементов, особенно в случаях, когда доступ к данным ограничен или постановка задачи требует гибкости.

Идея интеграции физических законов непосредственно в структуру нейросетевой модели была систематизирована и формализована в работах Raissi, Perdikaris и Karniadakis, заложивших основу концепции Physics-Informed Neural Networks (PINNs) [1]. В отличие от традиционных подходов, PINN минимизирует функцию потерь, которая включает не только отклонение от известных данных, но и невязку уравнения в частных производных, тем самым обеспечивая физическую согласованность модели.

Несмотря на широкое распространение Physics-Informed Neural Networks (PINNs) лишь в последние годы, сама идея использования нейросетей для приближенного решения дифференциальных уравнений уходит корнями в 1980–90-е годы. Одними из первых эту тему систематически изучали Lagaris, Likas и Fotiadis [2], а также Dissanayake и Phan-Thien [3]. Они рассматривали возможность аппроксимации решения уравнений с помощью нейронных сетей, при этом особое внимание уделялось точному соблюдению граничных условий – так называемому подходу с жесткими ограничениями, при котором граничные условия встроены в функциональную форму приближения и выполняются автоматически.

Теоретическую основу этим работам придала статья Hornik, Stinchcombe и White [4], в которой было доказано, что многослойные полносвязные нейронные сети являются универсальными аппроксиматорами. Это означало, что при достаточном числе параметров и подходящей архитектуре нейросети способны приближать решения самых разных уравнений в частных производных. Другие исследователи, такие как Lee и Kang [5], Psychogios и Ungar [6], изучали возможность объединения знаний из первых принципов с данными, создавая гибридные подходы к моделированию физических процессов.

Возвращение к нейросетевым методам решения дифференциальных уравнений стало возможным благодаря стремительному развитию технологий в области искусственного интеллекта и глубокого обучения. Одним из ключевых факторов стало широкое распространение графических процессоров (GPUs), что сделало обучение нейросетей более масштабируемым и быстрым. Важным шагом также стало появление библиотек автоматического дифференцирования, таких как TensorFlow и PyTorch [7, 8], что позволило напрямую включать производные по входным переменным в вычислительный граф и использовать их в функции потерь.

На этом фоне начала активно развиваться концепция physics-informed подходов. Raissi, Yazdani и Karniadakis [9] предложили метод скрытой гидродинамики, позволяющий по визуализации течения восстанавливать распределения давления и скорости. Haghghat et al. [10] расширили PINN на задачи механики твердого тела, включая инверсные задачи и моделирование переноса. Wu et al. [11] провели всесторонний анализ методов адаптивного семплирования и их влияния на точность PINN. Mishra и Molinaro [12] предложили строгие оценки обобщающей ошибки при применении PINN к обратным задачам.

Современные PINN-модели находят применение в самых различных областях науки и техники, демонстрируя гибкость подхода и его способность работать с частичными, шумными и разреженными данными. В области гидродинамики PINN используется как для ламинарных течений [13], так и для турбулентных режимов [14], включая моделирование ближнестеночных эффектов в крупномасштабных вихревых моделях (LES) [15]. В биомеханике и гемодинамике PINN позволяет восстанавливать параметры кровотока по данным магнитно-резонансной томографии (4D flow MRI) [16], реконструировать поля давления в артериях [17], а также моделировать процессы электрической активации миокарда [18].

Метод также успешно применяется в геофизике, например, для моделирования волнового распространения и решения задач полной волновой инверсии [19]; в нефтегазовой инженерии – для моделирования фильтрации в пористых средах [20]; в задачах теплопереноса [21, 22], в области нанопотоники и метаматериалов [23], а также в многомасштабном моделировании сложных материалов. Такой широкий спектр применения демонстрирует, что PINN может служить универсальным инструментом научного моделирования и предсказания в условиях ограниченной или неполной информации.

Материалы и методы

Постановка задачи

Рассматривается двумерное эллиптическое уравнение с реакцией на прямоугольной области $\Omega \in \mathbb{R}^2$:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + 3u = -2 \sin(x + 2y) + 16e^{2x+3y}, \quad (x, y) \in \Omega$$

Область определения выбирается как $\Omega = [0, 1/4] \times [0, 1/2]$.

На границе $\partial\Omega$ задаются неоднородные граничные условия смешанного типа:

На нижней границе ($y = 0$) устанавливается условие Неймана:

$$\frac{\partial u}{\partial y} = 2 \cos(x + 2y) + 3e^{2x+3y}$$

На остальных трех границах ($x = 0, x = \frac{1}{4}, y = 1/2$) устанавливается условие Дирихле:

$$u = \sin(x + 2y) + e^{2x+3y}$$

Аналитическое решение данной задачи известно и имеет вид:

$$u(x, y) = \sin(x + 2y) + e^{2x+3y}$$

Приближенное решение аппроксимируется нейросетевой моделью $u_\theta(x, y)$, параметризованной вектором θ . Модель обучается на основе невязки (residual) в уравнении и граничных условиях.

Реализация PINN

Physics-Informed Neural Networks (PINNs) представляют собой класс нейросетевых моделей, ориентированных на решение дифференциальных уравнений путем включения физической информации непосредственно в функцию потерь. Такой подход позволяет обойти необходимость в традиционной дискретизации и использовать как данные, так и структуру уравнения для обучения модели.

Типичная архитектура PINN представляет собой многослойную полносвязную нейросеть с активацией $\tanh$, которая широко используется из-за ее гладкости и способности точно приближать решения гладких уравнений [24]. На вход подаются пространственно-временные координаты (x, y, t) , а на выходе – предсказание физической величины u (давления, температуры, скорости и т.п.). В дальнейшем будем обозначать нейросетную аппроксимацию точного решения $u(x, y)$ как $u_\theta(x, y)$, где θ – совокупность всех обучаемых параметров сети и имеющей архитектуру глубины D : входной слой, $D - 2$ скрытых слоя и выходной слой

$$u_\theta(x, y) = L^{(D)} \circ \sigma \circ L^{(D-1)} \circ \sigma \circ \dots \circ \sigma \circ L^{(1)}(x, y)$$

где $L^{(d)}(z) = W^{(d)}z + b^{(d)}$ – линейное преобразование на слое d , σ – функция активации, а $\theta = \{W^{(d)}, b^{(d)}\}_{d=1}^D$ – совокупность всех обучаемых параметров модели.

Функция потерь имеет вид:

$$\mathcal{L} = \omega_f \mathcal{L}_f + \omega_b \mathcal{L}_b + \omega_o \mathcal{L}_o$$

где $\mathcal{L}_f$ – невязка уравнения (PDE residual loss), $\mathcal{L}_b$ – ошибка на границе (boundary condition loss), $\mathcal{L}_0$ – ошибка начального условия (initial condition loss), $\omega_f, \omega_b, \omega_0$ – гиперпараметры весов.

Модель обучается на трех типах точек:

- 1) граничные точки
- 2) начальные точки, если задача нестационарная
- 3) внутренние точки

Производные для $\mathcal{L}_f$ вычисляются автоматически с помощью механизма автоматического дифференцирования (automatic differentiation), реализованного в современных библиотеках, таких как PyTorch и TensorFlow [7, 8].

Инициализация весов сети играет ключевую роль в стабильности и сходимости обучения. Помимо стандартных подходов, таких как Xavier initialization, в литературе предлагаются и более адаптированные стратегии [25].

Хотя в данной работе используется базовая архитектура, в более сложных постановках PINN может быть расширена. Например, в модели PhyGeoNet [26] реализована геометрически адаптивная сверточная сеть (CNN), а в gradient-enhanced PINNs [27] добавляются производные высших порядков в лосс-функцию для повышения точности в обратных задачах.

Наиболее распространенным выбором функция потерь в PINN строится как среднеквадратичная ошибка (MSE) [1] и имеет вид:

$$\mathcal{L} = \omega_f MSE_f + \omega_b MSE_b + \omega_0 MSE_0$$

И включает несколько слагаемых: одно отвечает за выполнение самого дифференциального уравнения во внутренних точках области, другое – за выполнение граничных условий, и отдельное – за начальные условия. Каждое из этих слагаемых можно умножить на свой вес, чтобы регулировать их вклад в общую функцию потерь. Таким образом, итоговый loss представляет собой взвешенную сумму ошибок, что позволяет сбалансировать точность аппроксимации решения в уравнении и на границах.

$$\begin{aligned} MSE_f &= \frac{1}{N_f} \sum_{i=1}^{N_f} (u_\theta(x_i, y_i) - u(x_i, y_i))^2 \\ &= \frac{1}{N_f} \sum_{i=1}^{N_f} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + 3u - (-2 \sin(x + 2y) + 16e^{2x+3y}) \right)^2 \end{aligned}$$

Граничные условия

$$MSE_b = \frac{1}{N_b} \sum_{i=1}^{N_b} (u_\theta(x_i, y_i) - u(x_i, y_i))^2$$

Для поставленной задачи среднеквадратичная ошибка на граничных условиях раскладывается как:

$$\begin{aligned} MSE_b &= MSE_{b(neumann)} + MSE_{b(dirichlet)} \\ &= \frac{1}{N_d} \sum_{i=1}^{N_d} (u_\theta(x_i, y_i) - u(x_i, y_i))^2 + \frac{1}{N_n} \sum_{i=1}^{N_n} \left(\frac{\partial u_\theta}{\partial y}(x_i, 0) - (2 \cos(x_i) + 3e^{2x_i}) \right)^2 \end{aligned}$$

Это позволяет отдельно учитывать вклад условий Неймана и Дирихле в общую функцию потерь.

$$N_b = N_d + N_n$$

где N_f – количество внутренних точек, N_d – количество точек Дирихле, N_n – количество точек Неймана, N_b – количество граничных точек.

В данной работе точки выбираются с использованием метода латинской гиперкубической выборки (LHS), что обеспечивает равномерное покрытие области без регулярной сетки. Такой подход является стандартным и хорошо зарекомендовал себя при решении задач с PINN. Более сложные стратегии, включая остаточно-адаптивное распределение, могут существенно повышать точность, особенно в задачах с локализованными особенностями решения. Их сравнительный анализ представлен в работе Wu et al. [11].

Значения весов в функции потерь выбираются эмпирически: для усиления выполнения граничных условий используется $\omega_b = 100$, при этом вес на уравнение равен $\omega_f = 1$. Для минимизации функции потерь используется оптимизатор Adam с фиксированным шагом обучения $\eta = 10^{-3}$. Общая продолжительность обучения составляет от 5000 до 10 000 эпох в зависимости от задачи. Дополнительные оптимизаторы, такие как L-BFGS, в рамках данной работы не использовались. Обучающая выборка состоит из $N_f = 10000$ внутренних точек и $N_b = 1000$ граничных точек.

Точки генерируются с использованием метода латинской гиперкубической выборки (LHS), обеспечивающего равномерное покрытие области и границы. Все вычисления проводятся на графическом процессоре NVIDIA GeForce GTX 1650 Ti (GPU) с использованием фреймворка PyTorch. Инициализация весов нейросети осуществляется по схеме Xavier normal initialization.

Базовая настройка модели

Для начального этапа обучения модели PINN была выбрана следующая конфигурация гиперпараметров: количество внутренних точек $N_f = 1000$, количество граничных точек $N_b = 100$, количество эпох $Epochs = 10000$, вес функции потерь на граничных условиях $\omega_b = 1$, вес функции потерь в уравнении $\omega_f = 1$, архитектура сети: 4 скрытых слоя по 32 нейрона в каждой, функция активации tanh, оптимизатор Adam, скорость обучения (learning rate) 10^{-3} .

Результаты и обсуждение

Влияние плотности обучающих точек на точность PINN

Одним из ключевых факторов, влияющих на точность решений, получаемых с помощью Physics-Informed Neural Networks (PINNs), является количество точек, на которых рассчитываются невязки: внутренних и граничных. В этом исследовании фиксировались все параметры модели, кроме N_f и N_b , чтобы изолировать их влияние (таблица 1). Использовалась следующая конфигурация: $Epochs = 10000$, Loss weights: $\omega_f = 1$, $\omega_b = 1$, Architecture: 4 hidden layers, 32 neurons each layer, Activation function: tanh, Optimizer: Adam.

На рисунке 1 представлено сопоставление аналитического решения, численного решения, предсказанного моделью, а также распределение абсолютной ошибки для трех уровней плотности обучающих точек. Анализ показывает, что во всех рассматриваемых случаях модель корректно воспроизводит форму аналитического решения, а локализация областей повышенной ошибки сохраняется независимо от плотности выборки. Это свидетельствует о стабильности аппроксимационных свойств модели при варьировании

количества обучающих данных. На рисунке 2 показано распределение обучающих точек, сгенерированных с использованием метода латинского гиперкуба (LHS). Данный метод обеспечивает квазирегулярное покрытие расчетной области, минимизируя кластеры точек и снижая вероятность возникновения плохо аппроксимируемых зон. Равномерность распределения точек позволяет интерпретировать различия в точности модели преимущественно как следствие архитектурных и оптимизационных особенностей, а не как эффект неравномерного семплирования пространства. Как видно из рисунка 3, после примерно 2000 эпох обучения наблюдается появление выраженных высокочастотных пиков функции потерь. Данное поведение указывает на неустойчивость процесса оптимизации и может быть интерпретировано как «перепрыгивание» оптимального решения вследствие слишком большого шага обучения или накопления градиентных колебаний. Несмотря на это, итоговые значения ошибок остаются ограниченными, что свидетельствует о способности модели сохранять приемлемое качество аппроксимации даже при наличии локальной неустойчивости на поздних этапах обучения. На рисунке 3 также представлены результаты для трех расчетных кейсов (А, В и С), отличающихся количеством внутренних и граничных точек. Сравнительный анализ показывает, что увеличение плотности как внутренних, так и граничных точек не приводит к заметному улучшению точности решения. Это подтверждается тем, что все метрики ошибки – максимальная ошибка, средняя абсолютная ошибка (MAE), среднеквадратичная ошибка (MSE) и относительная ошибка в норме L2 – находятся в узком диапазоне значений (таблица 2). Количественно установлено, что максимальная ошибка варьируется в пределах 0.023–0.025, различие значений MAE не превышает 0.001, а расчетное время изменяется незначительно и составляет 126–130 секунд. Эти результаты указывают на насыщение точности модели: при фиксированной архитектуре нейронной сети и неизменной стратегии обучения дальнейшее увеличение плотности обучающих точек не приводит к существенному снижению ошибки, но сопровождается ростом вычислительных затрат. В целом полученные данные позволяют заключить, что для рассматриваемой задачи плотность обучающих точек не является определяющим фактором точности решения при условии адекватного покрытия расчетной области и корректно выбранной архитектуры модели. Основные ограничения точности, по-видимому, обусловлены выразительной способностью нейронной сети и особенностями процедуры оптимизации, что указывает на целесообразность дальнейших исследований, направленных на адаптацию архитектуры, стратегии регуляризации и схемы обучения, а не на простое увеличение объема обучающей выборки.

Таблица 1 – Параметры для трех уровней плотности

Case	N_f	N	Description
A	400	40	Sparse
B	1000	100	Medium
C	10000	1000	High dense

Таблица 2 – Сравнение различных параметров ошибок для трех уровней плотности

Case	Max error	MAE	MSE	L2 relative error %	Calculation time (s)
A	0.024198	0.007315	6.5040e-05	0.2186 %	128.47
B	0.024743	0.007557	7.7533e-05	0.2237 %	126.30
C	0.023278	0.006587	6.1723e-05	0.1996 %	130.21

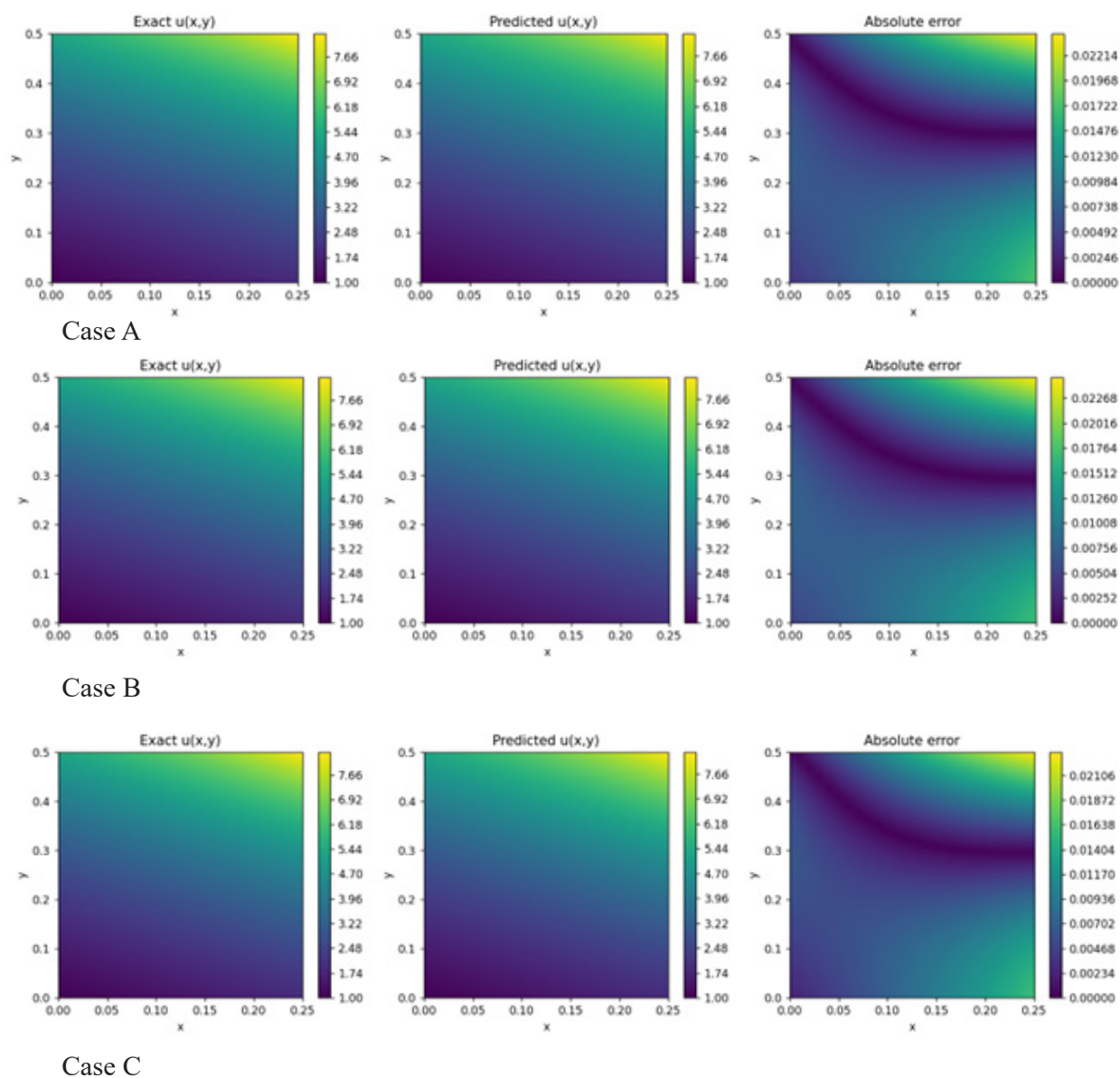


Рисунок 1 – Сравнение численного решения, прогнозированного решения и аналитического решения для трех уровней плотности

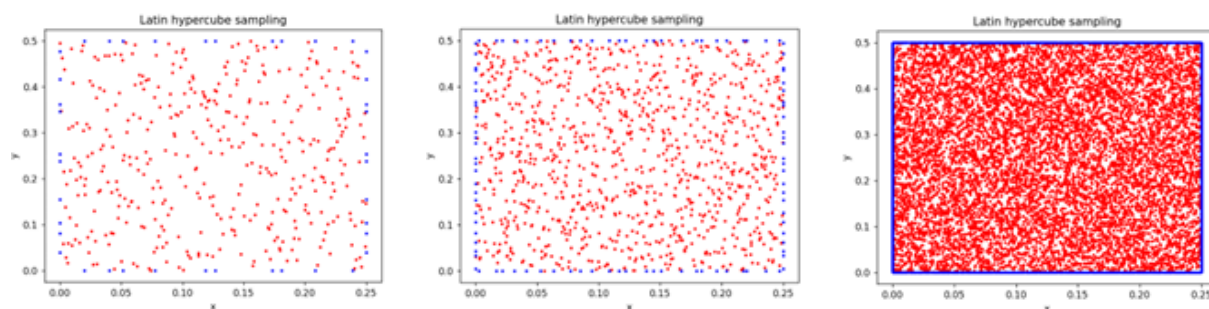
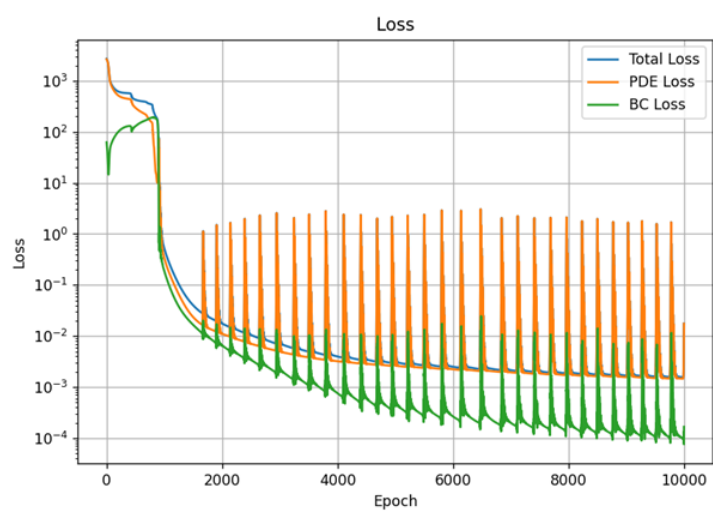
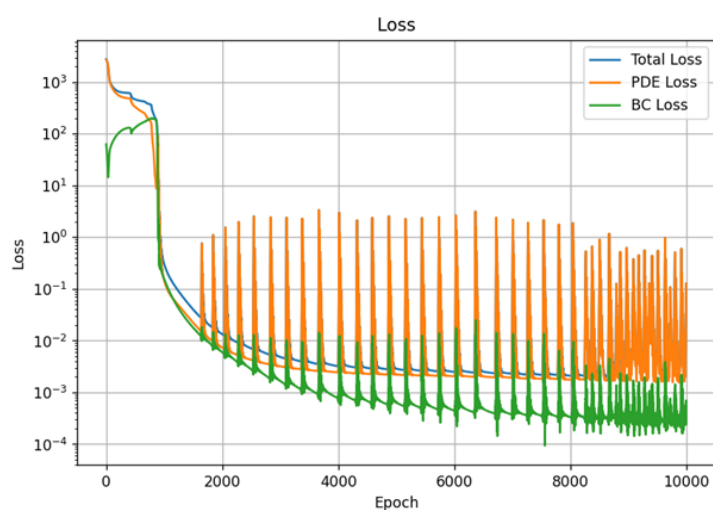


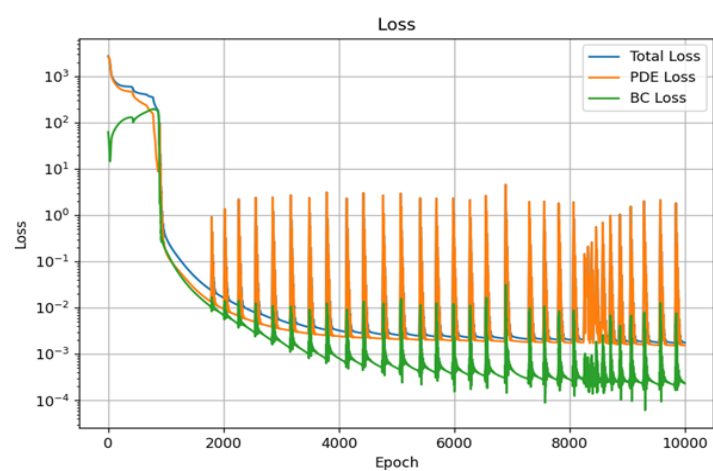
Рисунок 2 – Распределение точек по методу LHS



A



B



C

Рисунок 3 – Графики падения функции потерь для трех уровней плотности.

Влияние весовых коэффициентов функции потерь на точность модели

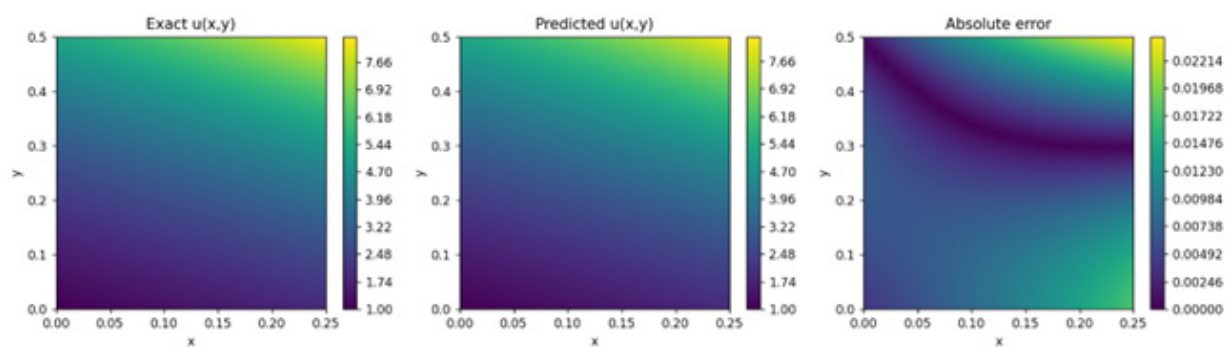
В данном разделе исследуется влияние весового коэффициента при расчете функции потерь на граничных условиях на общую точность модели. В то время как вес уравнения сохраняется постоянным и равным 1, параметр варьируется в пределах [1, 10, 100] (таблица 3). Такое масштабирование позволяет понять, насколько важным является точное соблюдение граничных условий при обучении PINN и как оно влияет на аппроксимацию решения в области.

Для каждого случая фиксируются все остальные параметры: архитектура сети (4 скрытых слоя по 32 нейрона), количество обучающих точек, количество эпох и скорость обучения. Основное внимание уделяется сравнительному анализу ошибок и времени расчета. Исходя из этих наблюдений, в дальнейшем фиксируется значение во всех следующих экспериментах.

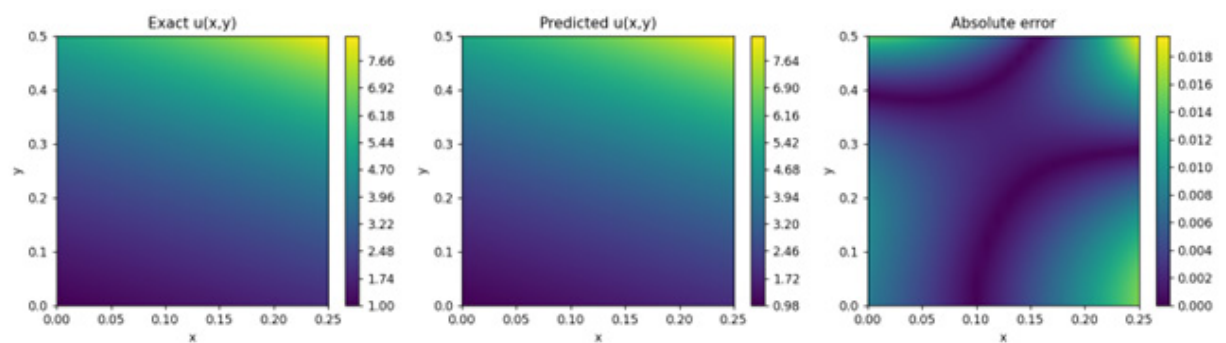
На рисунке 4 представлено сравнение трех расчетных кейсов, отличающихся степенью сгущения обучающих точек, с учетом влияния весовых коэффициентов функции потерь. Анализ полученных распределений показывает, что варьирование плотности точек в сочетании с изменением весового коэффициента граничных условий приводит к более выраженным изменениям точности решения, чем изолированное увеличение числа обучающих точек. Это указывает на ключевую роль балансировки компонент функции потерь в задачах физически информированных нейронных сетей (PINN). На рисунке 5 приведены графики эволюции функции потерь в процессе обучения для трех уровней плотности обучающих точек при различных значениях весовых коэффициентов. Видно, что при увеличении наблюдается более крутое начальное падение функции потерь и более раннее достижение плато, что свидетельствует об ускорении сходимости оптимизационного процесса. При этом различия между кейсами с разной плотностью точек остаются вторичными по сравнению с влиянием весовых коэффициентов. Количественные данные, представленные в таблице 3, показывают, что, несмотря на увеличение значения, общее время обучения не возрастает пропорционально и в ряде случаев даже уменьшается. Данный эффект объясняется тем, что усиление вклада граничных условий в целевую функцию заставляет модель на ранних этапах обучения быстрее удовлетворять физическим и геометрическим ограничениям на границе расчетной области. Это, в свою очередь, снижает степень конфликта между остатками дифференциального уравнения и граничными условиями, облегчая последующее глобальное минимизирование функции потерь. С точки зрения оптимизации увеличение веса приводит к более направленному градиентному потоку, уменьшая вероятность попадания модели в локальные минимумы, обусловленные несогласованностью различных компонент функции потерь. В результате достигается не только повышение точности аппроксимации вблизи границы, но и более стабильное и быстрое обучение в целом. Таким образом, полученные результаты демонстрируют, что корректный выбор весовых коэффициентов функции потерь является более эффективным инструментом повышения точности и ускорения сходимости PINN по сравнению с простым увеличением плотности обучающих точек. Это подчеркивает необходимость адаптивных или проблемно-ориентированных стратегий балансировки весов компонент функции потерь, особенно для задач с жесткими граничными условиями и сложной физической структурой решения.

Таблица 3 – Сравнение относительных ошибок для трех уровней плотности при влиянии весовых коэффициентов функции потерь

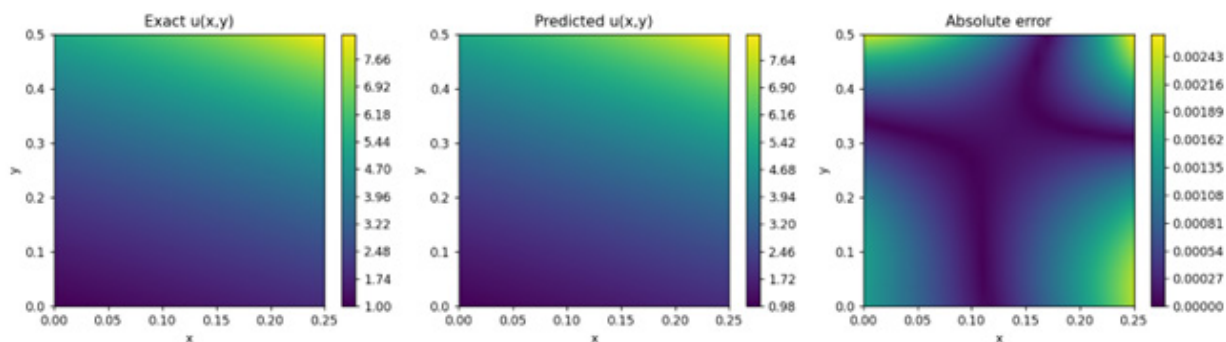
Case	ω_b	Max error	MAE	MSE	L2 relative error	Calculation time (s)
A	1	0.024198	0.007315	6.5040e-05	0.2186 %	128.47
B	10	0.022492	0.005535	4.9847e-05	0.1794 %	131.13
C	100	0.002640	0.000713	8.4998e-07	0.0234 %	137.77



Case A

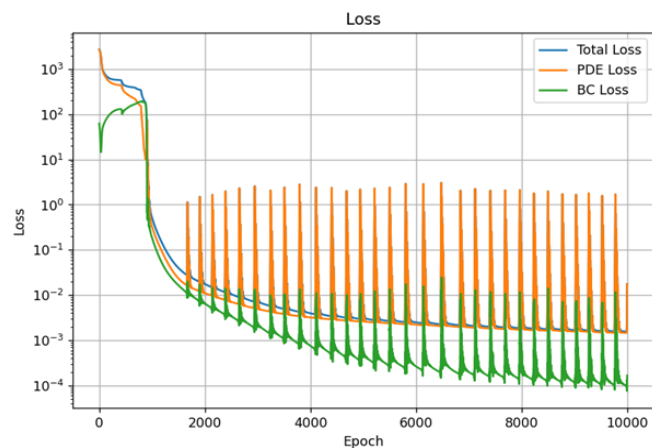


Case B

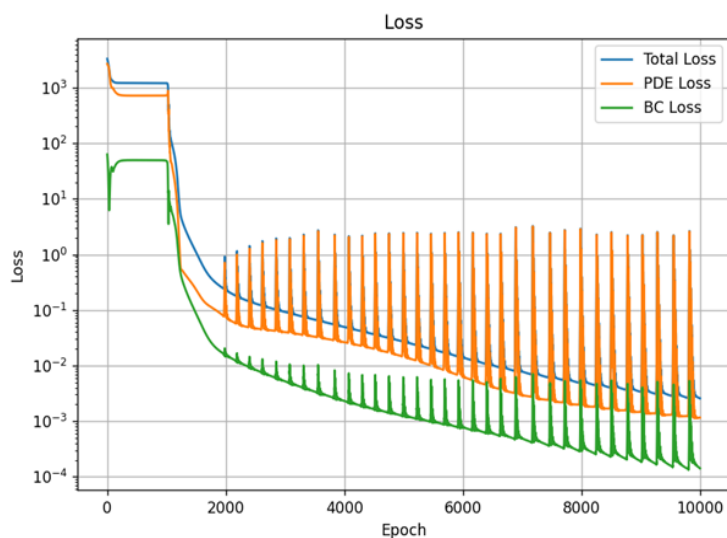


Case C

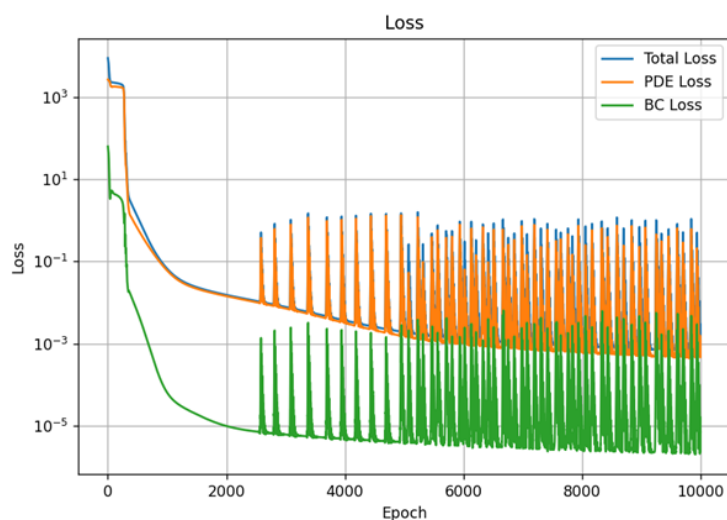
Рисунок 4 – Сравнение численного решения, прогнозированного решения и аналитического решения при влиянии весовых коэффициентов функции потерь



Case A



Case B



Case C

Рисунок 5 – Графики падения функции потерь для трех уровней плотности при влиянии весовых коэффициентов функции потерь

Заключение

В данной работе была рассмотрена эффективность применения Physics-Informed Neural Networks (PINN) для решения двумерного уравнения Пуассона. Модель была реализована с использованием библиотеки PyTorch и обучалась на сгенерированных с помощью латинского гиперкубического метода точках. Для валидации решений проводилось сравнение с аналитическим и численным решениями.

Было проведено систематическое исследование влияния различных гиперпараметров модели, которые включают плотность выборки точек, веса функций потерь, архитектуру сети (число слоев и нейронов), шаг обучения.

Основные выводы:

- ♦ Модель PINN продемонстрировала высокую точность решения, с относительной ошибкой по норме L2, не превышающей 1% во всех конфигурациях и достигающей менее 0.1% при оптимальных архитектурах.
- ♦ Изменение числа обучающих точек (от 400 до 10000) оказало незначительное влияние на точность, особенно при достаточном покрытии области.
- ♦ Увеличение веса граничных условий до 100 существенно улучшило сходимость и точность.

В дополнение к этому были проведены сравнения с традиционными численными методами. Последние показали более низкое время расчета и высокую точность (относительная L2 ошибка 0.01%), но требуют явной дискретизации и зависят от качества сетки. В отличие от них, PINN не требует сеточной структуры и может быть применен к произвольным геометриям и условиям.

Таким образом, PINN-подход подтвердил свою универсальность и точность при решении уравнений математической физики, особенно в случаях сложных геометрий и неполных данных. Работа не только демонстрирует применимость PINN на классическом уравнении Пуассона, но и служит отправной точкой для более глубокого изучения PINN в области вычислительной математики и моделирования физических процессов.

Информация о финансировании

Работа поддержана грантами Министерства науки и высшего образования Республики Казахстан (AP32320067).

ЛИТЕРАТУРА

- 1 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. *J. Comput. Phys.* 378, 686–707 (2019). <https://doi.org/10.1016/j.jcp.2018.10.045>
- 2 Lagaris, I., Likas, A., Fotiadis, D.: Artificial neural networks for solving ordinary and partial differential equations. *IEEE Trans. Neural Networks* 9(5), 987–1000 (1998). <https://doi.org/10.1109/72.712178>
- 3 Dissanayake, M.W.M.G., Phan-Thien, N.: Neural-network-based approximations for solving partial differential equations. *Commun. Numer. Methods Eng.* 10(3), 195–201 (1994). <https://doi.org/10.1002/cnm.1640100303>
- 4 Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. *Neural Networks* 2(5), 359–366 (1989). [https://doi.org/10.1016/0893-6080\(89\)90020-8](https://doi.org/10.1016/0893-6080(89)90020-8)
- 5 Lee, H., Kang, I.S.: Neural algorithm for solving differential equations. *J. Comput. Phys.* 91(1), 110–131 (1990). [https://doi.org/10.1016/0021-9991\(90\)90007-N](https://doi.org/10.1016/0021-9991(90)90007-N)
- 6 Psychogios, D.C., Ungar, L.H.: A hybrid neural network-first principles approach to process modeling. *AIChE J.* 38(10), 1499–1511 (1992). <https://doi.org/10.1002/aic.690381003>
- 7 Haghighat, E., Juanes, R.: SciANN: A Keras/Tensorflow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Comput. Methods Appl. Mech. Engrg.* 373, 113,552 (2021). <https://doi.org/10.1016/j.cma.2020.113552>
- 8 Paszke, A., Gross, S., Chintala, S., et al.: Automatic differentiation in PyTorch. Tech. rep., (2017) <https://openreview.net/forum?id=BJJsrnfCZ>
- 9 Raissi, M., Yazdani, A., Karniadakis, G.E.: Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Sci.* 367(6481), 1026–1030 (2020). <https://doi.org/10.1126/science.aaw4741>
- 10 Haghighat, E., Raissi, M., Moure, A., et al.: A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Comput. Methods Appl. Mech. Engrg.* 379, 113,741 (2021b). <https://doi.org/10.1016/j.cma.2021.113741>

- 11 Wu, C., Zhu, M., Tan, Q., Kartha, Y. and Lu, L., A Comprehensive Study of Non-Adaptive and Residual-Based Adaptive Sampling for Physics-Informed Neural Networks, *Computer Methods in Applied Mechanics and Engineering*, vol. 403, p. 115671, October 22, 2022. <https://doi.org/10.1016/j.cma.2022.115671>
- 12 Mishra, S., Molinaro, R.: Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs. *IMA J. Numer. Anal.* (2021). <https://doi.org/10.1093/imanum/drab032>
- 13 Rao, C., Sun, H., & Liu, Y. (2020). Physics-informed deep learning for incompressible laminar flows. *Theoretical and Applied Mechanics Letters*, 10(3), 207–212. <https://doi.org/10.1016/j.taml.2020.01.039>
- 14 Wang, R., Kashinath, K., Mustafa, M., Albert, A., & Yu, R. (2019). Towards Physics-informed Deep Learning for Turbulent Flow Prediction. *ArXiv.org*. <https://arxiv.org/abs/1911.08655>
- 15 Yang, X. I. A., Zafar, S., Wang, J.-X. ., & Xiao, H. (2019). Predictive large-eddy-simulation wall modeling via physics-informed neural networks. *Physical Review Fluids*, 4(3). <https://doi.org/10.1103/physrevfluids.4.034602>
- 16 Kissas, G., Yang, Y., Hwuang, E., Witschey, W. R., Detre, J. A., & Perdikaris, P. (2020). Machine learning in cardiovascular flows modeling: Predicting arterial blood pressure from non-invasive 4D flow MRI data using physics-informed neural networks. *Computer Methods in Applied Mechanics and Engineering*, 358, 112623. <https://doi.org/10.1016/j.cma.2019.112623>
- 17 Arzani, A., Wang, J.-X., & D'Souza, R. M. (2021). Uncovering near-wall blood flow from sparse data with physics-informed neural networks. *Physics of Fluids*, 33(7), 071905. <https://doi.org/10.1063/5.0055600>
- 18 Sahli Costabal, F., Yang, Y., Perdikaris, P., Hurtado, D. E., & Kuhl, E. (2020). Physics-Informed Neural Networks for Cardiac Activation Mapping. *Frontiers in Physics*, 8. <https://doi.org/10.3389/fphy.2020.00042>
- 19 Rasht-Behesht, M., Huber, C., Shukla, K., & Karniadakis, G. E. (2022). Physics-Informed Neural Networks (PINNs) for Wave Propagation and Full Waveform Inversions. *Journal of Geophysical Research: Solid Earth*, 127(5). <https://doi.org/10.1029/2021jb023120>
- 20 Almajid, M. M., & Abu-Al-Saud, M. O. (2022). Prediction of porous media fluid flow using physics informed neural networks. *Journal of Petroleum Science and Engineering*, 208, 109205. <https://doi.org/10.1016/j.petrol.2021.109205>
- 21 Cai, S., Wang, Z., Wang, S., Perdikaris, P., & Karniadakis, G. (2021). Physics-Informed Neural Networks (PINNs) for Heat Transfer Problems. *Journal of Heat Transfer*. <https://doi.org/10.1115/1.4050542>
- 22 Zobeiry, N., & Humfeld, K. D. (2021). A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications. *Engineering Applications of Artificial Intelligence*, 101, 104232. <https://doi.org/10.1016/j.engappai.2021.104232>
- 23 Chen, Y., Lu, L., Karniadakis, G. E., & Dal Negro, L. (2020). Physics-informed neural networks for inverse problems in nano-optics and metamaterials. *Optics Express*, 28(8), 11618. <https://doi.org/10.1364/oe.384875>
- 24 De Ryck, T., Lanthaler, S., & Mishra, S. (2021). On the approximation of functions by tanh neural networks. *Neural Networks*, 143, 732–750. <https://doi.org/10.1016/j.neunet.2021.08.015>
- 25 Liu, X., Zhang, X., Peng, W., Zhou, W., & Yao, W. (2022). A novel meta-learning initialization method for physics-informed neural networks. *Neural Computing and Applications*, 34(17), 14511–14534. <https://doi.org/10.1007/s00521-022-07294-2>
- 26 Gao, H., Sun, L., & Wang, J.-X. (2021). PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parameterized steady-state PDEs on irregular domain. *Journal of Computational Physics*, 428, 110079. <https://doi.org/10.1016/j.jcp.2020.110079>
- 27 Yu, J., Lu, L., Meng, X., & Karniadakis, G. E. (2022). Gradient-enhanced physics-informed neural networks for forward and inverse PDE problems. *Computer Methods in Applied Mechanics and Engineering*, 393, 114823. <https://doi.org/10.1016/j.cma.2022.114823>

^{1*}Исахов А.А.,

профессор, ORCID ID: 0000-0002-1937-8615, *e-mail: alibek.issakhov@gmail.com

¹Бейсембеков М.Е.

бакалавр, ORCID ID: 0009-0004-5395-4569, e-mail: m_beisembekov@kbtu.kz

¹Қазақстан-Британ техникалық университет, Алматы қ., Қазақстан**БІРТЕКТІ ЕМЕС АРАЛАС ШЕКАРАЛЫҚ ШАРТТАРМЕН БЕРІЛГЕН
СЫЗЫҚТЫҚ ДЕРБЕС ТУЫНДЫ ТЕНДЕУЛЕРДІ ШЕШУГЕ АРНАЛҒАН
ФИЗИКАҒА НЕГІЗДЕЛГЕН НЕЙРОНДЫҚ ЖЕЛІЛЕРДІ ҚОЛДАНУ****Аңдатпа**

Мақалада біртекті емес аралас шекаралық шарттары бар екі өлшемді Пуассон теңдеуін шешу үшін физикалық ақпараттандырылған нейрондық желілерді (PINN) қолдану қарастырылады. PINN тәсілі нейрондық желінің аппроксимациясын физикалық заңдармен үйлестіріп, теңдеу қалдығын және шекаралық шарттар қателіктерін бірыңғай шығын функциясына біріктіреді. Модель PyTorch ортасында жүзеге асырылады және латын гиперкубтық сынама алу әдісімен жасалған үлгіде оқытылады. Дәлдігін тексеру үшін аналитикалық шешіммен, сондай-ақ дәстүрлі сандық әдістермен салыстыру жүргізіледі. Негізгі гиперпараметрлердің әсері жүйелі түрде зерттеледі: жаттығу нүктелерінің саны, жоғалту функциясындағы салмақтар, желі архитектурасы және жаттығу қадамы. Шекаралық шарттардағы салмақты арттыру конвергенция мен дәлдікті жақсартатыны анықталды, ал 4–5 жасырын қабаты және 64 нейрондары бар оңтайлы архитектуралар 0,1–0,2%-дан аз салыстырмалы қателікті қамтамасыз етеді. Тордың дискретизациясын талап ететін сандық схемалардан айырмашылығы, PINN сирек деректермен және күрделі шарттармен жұмыс істеу кезінде әмбебаптығы мен жоғары дәлдігін көрсетті, бұл оны күрделі геометрия және толық емес ақпарат жағдайында есептеу математикасы мен физикалық процестерді модельдеу үшін перспективалы құралға айналдырды.

Түйін сөздер: машиналық оқыту әдістері, PINN моделі, Пуассон теңдеуі, жалғыз жоғалту функциясы, гиперпараметрлер.

^{1*}Issakhov A.A.,

Professor, ORCID ID: 0000-0002-1937-8615, *e-mail: alibek.issakhov@gmail.com

¹Beisembekov M.E.

Bachelor, ORCID ID: 0009-0004-5395-4569, e-mail: m_beisembekov@kbtu.kz

¹Kazakh-British Technical University, Almaty, Kazakhsan**APPLICATION OF PHYSICALLY-ORIENTED NEURAL NETWORKS
TO SOLVING PROBLEMS OF LINEAR PARTIAL DERIVATIVE EQUATIONS
WITH INHOMOGENEOUS MIXED BOUNDARY CONDITIONS****Abstract**

The paper considers the application of Physics-Informed Neural Networks (PINN) to solving a two-dimensional Poisson equation with inhomogeneous mixed boundary conditions. The PINN method combines neural network approximations and physical laws, including the residual of the equation and errors at the boundary, into a single loss function. The model is implemented in the PyTorch environment and trained on a sample generated by the Latin hypercubic sampling method. To verify the accuracy, a comparison is made with an analytical solution, as well as with traditional numerical methods. The influence of key hyperparameters is systematically studied: the number of training points, weights in the loss function, network architecture, and training step. It is found that increasing the weight on the boundary conditions improves convergence and accuracy, and optimal architectures with 4–5 hidden layers and 64 neurons provide a relative error of less than 0.1–0.2%. Unlike numerical schemes requiring grid discretization, PINN has demonstrated versatility and high accuracy when dealing with sparse data and complex conditions, making it a promising tool for computational mathematics and modeling of physical processes in conditions of complex geometry and incomplete information.

Keywords: machine learning methods, PINN model, Poisson equation, single loss function, hyperparameters.

Received September 2, 2025; revised May 29, 2026; accepted June 11, 2026.