



doi 10.26089/NumMet.v27r333

УДК 517.957;
519.633.2

Использование физически информированных нейронных сетей для численного решения уравнения Блэка–Шоулса

М. М. Дышаев

ООО “НПК “КОБОТОРИ”,

Челябинск, Российская Федерация

ORCID: 0000-0003-4265-1752, e-mail: Mikhail.Dyshaev@gmail.com

Аннотация: Работа посвящена применению физически информированных нейронных сетей (physics-informed neural networks, PINN) для численного решения дифференциальных уравнений в частных производных. Исследования проведены на примере уравнения Блэка–Шоулса. Представлены сравнительные показатели обучения и результатов работы двух сетей. Одна из них является базовой сетью, другая представляет собой модифицированную сеть. Для улучшения работы второй сети использовались инициализация весов Ксавьера, механизм внимания, контроль временной последовательности в обучении, эмбединг случайных функций Фурье, адаптивные веса слагаемых функции потерь. Также были использованы разработанные автором ограничения, а именно штраф за отрицательные значения решения уравнения. Результаты показывают, что модификация улучшает качественное поведение решения, особенно в локальной области около точки излома, однако это происходит за счет снижения общей точности и за счет увеличения времени обучения. Так, относительная L_2 -норма ошибки базовой PINN после модификации увеличилась с 2.14×10^{-3} до 2.63×10^{-3} . Среднее время обучения, приходящееся на одну эпоху, выросло с 1.5 до 4.2 с. Локальный показатель максимальной абсолютной ошибки в области излома снизился с 4.57×10^{-3} USD для базовой PINN до 2.55×10^{-3} USD для модифицированной, в то время как локальные значения относительной L_2 -нормы ошибки остались прежними.

Ключевые слова: физически информированные нейронные сети, PINN, уравнение Блэка–Шоулса, дифференциальные уравнения в частных производных.

Благодарности: Автор выражает искреннюю благодарность профессору Д. Ю. Голембиовскому за внимательное прочтение ранней версии статьи, за его ценные замечания и конструктивные предложения, которые способствовали улучшению содержания и качества данной работы.

Для цитирования: Дышаев М.М. Использование физически информированных нейронных сетей для численного решения уравнения Блэка–Шоулса // Вычислительные методы и программирование. 2026. 27, № 3. 523–539. doi 10.26089/NumMet.v27r333.

Application of physics-informed neural networks for the numerical solution of the Black–Scholes equation

Mikhail M. Dyshaev

COBOTORY LLC, Chelyabinsk, Russia

ORCID: 0000-0003-4265-1752, e-mail: Mikhail.Dyshaev@gmail.com

Abstract: This paper investigates the application of physics-informed neural networks (PINNs) to the numerical solution of partial differential equations. The study is conducted using the Black–Scholes equation as a representative example. A comparative analysis of the training performance and solution accuracy of two PINN architectures is presented. One network serves as the baseline model, while the other incorporates several modifications. The modified PINN employs Xavier weight initialization, an attention mechanism, a causal training algorithm, random Fourier feature embedding, and adaptive loss weighting. In addition, a penalty term proposed by the author is introduced to discourage negative values of the predicted solution. The results demonstrate that the proposed modifications improve the qualitative behavior of the solution, particularly in the local neighborhood of the solution’s non-differentiable point. However, these improvements are achieved at the cost of reduced overall accuracy and increased training time. Specifically, the relative L_2 error increased from 2.14×10^{-3} for the baseline PINN to 2.63×10^{-3} for the modified PINN, while the average training time per epoch increased from 1.5 s to 4.2 s. At the same time, the local maximum absolute error in the vicinity of the kink decreased from 4.57×10^{-3} USD for the baseline PINN to 2.63×10^{-3} USD for the modified PINN, whereas the local relative L_2 error remained essentially unchanged.

Keywords: physics-informed neural networks, PINN, Black–Scholes equation, partial differential equations.

Acknowledgements: The author gratefully acknowledges Professor D. Yu. Golembiovsky for carefully reading an earlier version of the manuscript and for his valuable comments and constructive suggestions, which contributed to improving the content and quality of this paper.

For citation: M. M. Dyshaev, “Application of physics-informed neural networks for the numerical solution of the Black–Scholes equation,” *Numerical Methods and Programming*. 27 (3), 523–539 (2026). doi 10.26089/NumMet.v27r333.

1. Введение. Быстрый рост вычислительных мощностей ведет ко все более активному использованию глубоких искусственных нейронных сетей (ИНС). ИНС активно используются как для моделирования физических процессов, так и процессов в экономике.

Одними из первых работ, посвященных использованию искусственных нейронных сетей для ценообразования опционов, были статьи [1, 2]. В статье [1] сравнивались цена опционов по модели Блэка–Шоулса [3] и цена, рассчитанная с помощью обученной ИНС. Использовалась трехслойная сеть с семью входными узлами, одним скрытым слоем из четырех нейронов и с одним выходным нейроном. На вход подавались цены страйков, время до экспирации, цена закрытия базового актива, волатильность и процентная ставка. Дополнительно подавались еще цены закрытия базового актива и опциона за предыдущий торговый день. В качестве функции активации использовалась сигмоидальная (логистическая) функция. В статье [2] авторы сравнили работу сетей радиальных базисных функций (radial basis functions, RBF) и многослойного персептрона (multilayer perceptron, MLP) и получили формулы для отношения цены опциона к цене страйка, зависящие только от срока до экспирации и денежности опциона.

В статье [4] было введено понятие физически информированного машинного обучения (physics-informed machine learning, PIML). Большинство моделей PIML являются бессеточными, что позволяет проводить расчеты для сложных геометрических форм области определения, а также для задач большой размерности. Чуть позже теми же авторами в [5] была описана базовая архитектура и предложено

несколько способов использования physics-informed neural networks (PINN). В настоящее время используется несколько переводов этого термина на русский язык: “физически информированные нейронные сети”, “физико-информированные нейронные сети”, “нейронные сети, основанные на физике” и т.д. В данной работе используется первый вариант.

Ключевым нововведением в PINN является использование автоматического дифференцирования на основе вычислительных графов [6, 7], которое позволяет получать численное значение производных, используемых в дифференциальных операторах. Как правило, такие сети на вход принимают значения координат, на выходе ИНС выдает значение решения дифференциального уравнения с учетом граничных условий. Данная область исследований развивается очень быстро. Упомянем лишь несколько тематических обзоров [8, 9] и статей [10–13], которые могут быть полезны читателям для оценки возможностей, архитектур и областей применения PIML и PINN в частности.

Отметим также, что и количество исследований моделей ценообразования опционов и решения различных видов дифференциальных уравнений Блэка–Шоулса с использованием PINN к настоящему моменту постоянно возрастает. Приведем работы, демонстрирующие возможности применения PINN для исследования широкого класса таких моделей [14–17].

Как указывается исследователями из различных областей в работах [8, 18–23] и как будет продемонстрировано ниже для рассматриваемого в данной работе уравнения, базовая модель PINN, минимизирующая функцию потерь, включающую ненулевой остаток (residual) дифференциального оператора, экспериментальных данных, граничных и начальных условий, иногда не дает адекватного решения исследуемого уравнения. Именно поэтому целью данного исследования является поиск, разработка и использование определенных приемов обучения и построения архитектуры PINN, позволяющих получать соответствующие численные решения.

В частности, впоследствии, при наличии технической возможности обучения нейросетей на графических процессорах (GPU), планируется обучить ИНС и получить решения для различных уравнений типа Блэка–Шоулса с учетом недостаточной ликвидности и транзакционных издержек, ранее исследованных методами группового анализа и численными методами [24, 25]. Разрабатываемый в данной работе подход планируется в дальнейшем применить и для поиска численного решения дробных по времени уравнений того же типа.

Вклад данной работы в развитие PINN состоит в том, что предложено новое, дополнительное слагаемое в функцию потерь, отражающее экономическую составляющую модели. Поскольку исследуемое уравнение отражает цену, являющуюся положительной величиной, вводится слагаемое, штрафующее ИНС за отрицательные значения решения.

Статья построена следующим образом. В методической части будет приведен вид аналитического решения уравнения Блэка–Шоулса. Это решение будет впоследствии использовано в качестве эталона для оценки качества работы обученных PINN. В подразделах будут описаны архитектура базовой PINN, а также модификации для получения более точного решения. После этого будут приведены результаты и предложены направления дальнейших исследований. Заключение кратко описывает итоги работы.

2. Параметры и модели для сравнения. В качестве эталона, для валидации, выбрана широко используемая на практике модель ценообразования опционов, модель Блэка–Шоулса–Мертон, которая была представлена в 1973 г. в работах [3, 26]. Тогда же была получена и известная формула Блэка–Шоулса. Модель Блэка–Шоулса для случая колл-опциона (call options) европейского типа можно записать в следующем виде:

$$\begin{aligned} u_t - \frac{1}{2}\sigma^2 x^2 u_{xx} - r(xu_x - u) &= 0, \\ u(0, t) = 0, \quad \lim_{x \rightarrow \infty} \frac{u(x, t)}{x - Ke^{-rt}} &= 1, \\ u(x, 0) &= \max(x - K, 0), \\ x &\in [0, +\infty), \quad t \in [0, T], \quad K = \text{const}, \quad K > 0, \\ \sigma &= \text{const}, \quad \sigma > 0, \quad r = \text{const}, \quad r \geq 0. \end{aligned} \tag{1}$$

Здесь x — цена базового актива, $u(x, t)$ — цена колл-опциона европейского типа, K — цена страйка, т.е. цена исполнения опциона, t — время, оставшееся до истечения (погашения) опциона, T — срок существования

опциона, σ — историческая волатильность (обычно принимается как стандартное отклонение цены за год), а r — безрисковая процентная ставка.

В этом случае можно записать явное решение для задачи (1) поиска стоимости колл-опциона европейского типа:

$$u(x, t) = xN(d_+) - Ke^{-rt}N(d_-), \quad (2)$$

где

$$d_{\pm} = \frac{1}{\sigma\sqrt{t}} \left(\ln \frac{x}{K} + t \left(r \pm \frac{\sigma^2}{2} \right) \right), \quad N(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^y e^{-\frac{x^2}{2}} dx. \quad (3)$$

Здесь $N(y)$ — кумулятивная функция распределения стандартного нормального распределения. Отметим, что с наиболее полным, всесторонним и комплексным взглядом на ценообразование производных финансовых инструментов читатель может ознакомиться в монографии академика РАН А. Н. Ширяева [27].

Для оценки обучения разрабатываемой PINN в качестве основной метрики использована относительная L_2 -норма ошибки:

$$E_{L_2}^{\text{rel}} = \frac{\|(u^{\text{predict}} - u^{\text{benchmark}})\|_2}{\|u^{\text{benchmark}}\|_2} = \left(\frac{\sum_{i=1}^N (u_i^p - u_i^b)^2}{\sum_{i=1}^N (u_i^b)^2} \right)^{\frac{1}{2}}, \quad (4)$$

где в качестве u_i^b выступало аналитическое решение уравнения Блэка–Шоулса на той же области определения. Здесь $N = N_{\text{id}} + N_{\text{ic}} + N_{\text{lb}} + N_{\text{rb}}$ — совокупность точек внутренней области (id — inner domain) и точек на границах области определения (ic — initial condition, lb (rb) — left (right) boundary).

Помимо относительной L_2 -нормы, на каждой эпохе проводился расчет и других показателей, таких как средняя абсолютная ошибка (mean absolute error, MAE), среднеквадратичное отклонение (mean squared error, MSE), функция потерь Хубера [28] и квадратный корень из MSE (root mean squared error, RMSE), а также L_2 -норма решения. Сделано это было с целью поиска наилучшего показателя для остановки обучения.

В качестве валидационного набора использовались множества $\{x_i, t_i, u_i^b\}_j$, где $i = 0, \dots, N_j$, а j означает принадлежность точек соответствующей части области определения: $j \in \{\text{id}, \text{ic}, \text{lb}, \text{rb}\}$. В рамках данной работы были приняты следующие значения: $N_{\text{id}} = 10000$, $N_{\text{ic}} = 128$, $N_{\text{lb}} = N_{\text{rb}} = 256$. Величины u_i^b определялись по формулам (2), (3) для каждой пары (x_i, t_i) . Для рыночных параметров использовались значения $\sigma = 0.25$ и $r = 0.05$. Цена страйка $K = 0.5$.

3. Архитектура базовой PINN. Архитектура PINN представляет собой, как правило, многослойный персептрон. Первый слой содержит количество нейронов, соответствующее размерности вектора входных данных $\mathbf{x} \in \mathbb{R}^{n_{\text{inp}}}$. Последний слой, выходной, содержит один нейрон для выдачи предсказываемого значения функции. Математически это можно выразить следующим образом:

$$\begin{aligned} \mathbf{u}^{(0)}(\mathbf{x}) &= \phi(W^{(0)} \cdot \mathbf{x} + \mathbf{b}^{(0)}), \\ \mathbf{u}^{(m)}(\mathbf{x}) &= \phi(W^{(m)} \cdot \mathbf{u}^{(m-1)}(\mathbf{x}) + \mathbf{b}^{(m)}), \quad m = 1, \dots, l_{\text{hid}}, \\ \mathbf{u}^{(l_{\text{hid}}+1)}(\mathbf{x}) &= W^{(l_{\text{hid}}+1)} \cdot \mathbf{u}^{(l_{\text{hid}})}(\mathbf{x}) + \mathbf{b}^{(l_{\text{hid}}+1)}, \\ \mathbf{u}^p(\mathbf{x}) &= \mathbf{u}^{(l_{\text{hid}}+1)}(\mathbf{x}). \end{aligned} \quad (5)$$

Здесь $\mathbf{b}^{(0)}, \mathbf{b}^{(m)}, \mathbf{u}^{(0)}, \mathbf{u}^{(m)} \in \mathbb{R}^{n_{\text{hid}}}$, $W^{(0)} \in \mathbb{R}^{n_{\text{hid}} \times n_{\text{inp}}}$, $W^{(m)} \in \mathbb{R}^{n_{\text{hid}} \times n_{\text{hid}}}$, $W^{(l_{\text{hid}}+1)} \in \mathbb{R}^{n_{\text{out}} \times n_{\text{hid}}}$, $\mathbf{u}^{(l_{\text{hid}}+1)}, \mathbf{b}^{(l_{\text{hid}}+1)} \in \mathbb{R}^{n_{\text{out}}}$.

В исследованиях различных авторов варьируется глубина (количество скрытых слоев l_{hid} , обычно от 2 до 8 слоев) и ширина (количество нейронов в скрытых слоях n_{hid} , встречается от 20 до 500 нейронов) сети. В данной работе используется сеть, содержащая $l_{\text{hid}} = 4$ скрытых слоя по $n_{\text{hid}} = 256$ нейронов в каждом. На слое входных данных используется $n_{\text{inp}} = 2$ нейрона, на выходном $n_{\text{out}} = 1$ нейрон.

Обычно в качестве функций активации $\phi(\cdot)$ для PINN используют гиперболический тангенс или сигмоиду. Такие популярные функции активации, как ReLu, содержащие линейные участки, не подходят, поскольку при автоматическом дифференцировании производные (и градиенты) могут обращаться в нуль.



В данной работе в качестве функции активации также используется гиперболический тангенс

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}.$$

Для исследования базовой модели PINN первоначальные веса инициализируются случайной величиной с нормальным стандартным распределением $\xi \sim \mathcal{N}(0, 1)$. Обучение происходит методом градиентного спуска с применением оптимизатора Adam¹. Скорость обучения снижается с коэффициентом 0.95 каждые 1000 эпох.

В отличие от других методов PIML, когда дополнительные знания о модели инкорпорируются непосредственно в архитектуру, в случае PINN дополнительная информация (начальное и граничные условия, вид дифференциального уравнения) включается непосредственно в функцию потерь в виде меры отклонения, например MSE. Общим случаем является включение следующих слагаемых:

отклонения в начальном условии

$$L_{ic}(w) = \frac{1}{N_{ic}} \sum_{i=1}^{N_{ic}} (u_{ic}^p(x_i, t_i, w) - u_{ic}^0(x_i, t_i))^2,$$

отклонения на левой границе

$$L_{lb}(w) = \frac{1}{N_{lb}} \sum_{i=1}^{N_{lb}} (u_{lb}^p(x_i, t_i, w) - u_{lb}^0(x_i, t_i))^2,$$

отклонения на правой границе

$$L_{rb}(w) = \frac{1}{N_{rb}} \sum_{i=1}^{N_{rb}} (u_{rb}^p(x_i, t_i, w) - u_{rb}^0(x_i, t_i))^2,$$

отклонения во внутренней области

$$L_{id}(w) = \frac{1}{N_{id}} \sum_{i=1}^{N_{id}} (\mathcal{F}_{id}(x_i, t_i, w))^2.$$

Здесь w — значения всех обучаемых параметров сети (весов W и смещений $\mathbf{b}$), $\mathbf{u}_{ic}^0$, $\mathbf{u}_{lb}^0$, $\mathbf{u}_{rb}^0$ задаются соответствующими условиями на границах из (1), а $\mathcal{F}_{id}(x_i, t_i, w)$ представляет собой отклонение численного значения дифференциального оператора от нуля, в нашем случае это задача (1):

$$\mathcal{F}_{id}(x_i, t_i, w) = u_t(x_i, t_i, w) - \frac{1}{2}\sigma^2 x_i^2 u_{xx}(x_i, t_i, w) - r(x_i u_x(x_i, t_i, w) - u^p(x_i, t_i, w)).$$

Расчет производных u_t , u_x и u_{xx} проводится с помощью автоматического дифференцирования [6, 7], реализованного в основных фреймворках для машинного обучения. В частности, для данного исследования используется метод `torch.autograd.grad` из PyTorch.

Таким образом, функция потерь базовой PINN может быть представлена в виде

$$\mathcal{L}(w) = L_{ic}(w) + L_{lb}(w) + L_{rb}(w) + L_{id}(w), \quad (6)$$

а соответствующая задача обучения базовой PINN в виде

$$\min_w \mathcal{L}(w) = \min_w [L_{ic}(w) + L_{lb}(w) + L_{rb}(w) + L_{id}(w)]. \quad (7)$$

¹См., например, препринт *Kingma D.P., Ba J.* Adam: A method for stochastic optimization // arXiv:1412.6980v9. doi 10.48550/arXiv.1412.6980.

4. Результаты работы базовой PINN. Приведем результаты работы PINN базовой архитектуры. Показатель относительной L_2 -нормы (4) составил в среднем 0.0021 на последних 100 эпохах обучения PINN. Сравнивая полученную величину с аналогичными показателями в работах других авторов, можно отметить вполне удовлетворительный уровень точности полученных значений.

Однако, если мы визуализируем решение, будет заметно принципиальное отличие полученного с помощью базовой PINN решения от точного аналитического решения уравнения Блэка–Шоулса (2), (3). На рис. 1, 2 представлены графики решения задачи (1). В выделенной эллипсом области на рис. 1 в сравнении с рис. 2 обращают на себя внимание две особенности решения базовой PINN:

- во-первых, отсутствует сходимость к точке излома (0.5, 0), отражающей цену опциона близко к моменту погашения, если при этом базовый актив торгуется около цены страйка (довольно частая ситуация в практике);
- во-вторых, слева от указанной точки решение базовой PINN $u(x, t)$ находится в отрицательной области, что противоречит и эталонному аналитическому решению, и экономическому смыслу задачи.

Отметим также, что используемые в работах некоторых других авторов по тематике PINN изображения решений и ошибок в сравнении с эталонным решением в виде “тепловых карт” (рис. 3) могут иногда скрывать принципиальные отличия в поведении решений.

Предварительные эксперименты с длительностью обучения и изменением гиперпараметров не улучшили ситуацию с приведенными отклонениями. О наличии похожей ситуации также сообщается в работах [8, 18, 20–23].

Вследствие обнаруженных расхождений для получения более точного решения с помощью PINN потребовались дополнения базовой архитектуры, а также модификации процесса обучения и функции потерь. В данной работе не проводится анализ влияния таких отдельных дополнений и модификаций на скорость обучения и точность работы PINN (методом последовательного добавления и исключения).

Учитывая, что новые области знания иногда изначально образуются эмпирическим путем без должного теоретического обоснования, для улучшения ситуации с расхождением в решениях была предпринята попытка реализовать несколько разработанных другими авторами модификаций и дополнений для PINN, а также использовать собственные разработки, совместно представленные в следующем разделе 5.

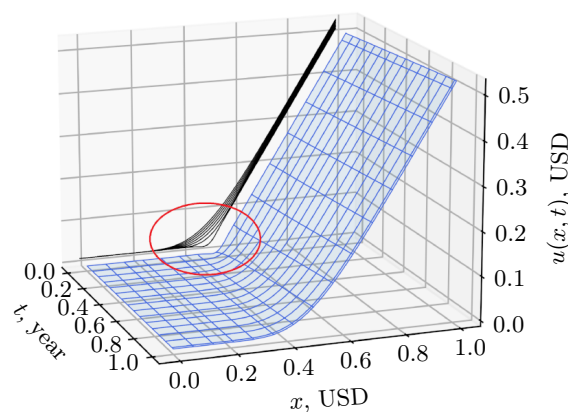


Рис. 1. График решения задачи (1) с помощью PINN базовой архитектуры

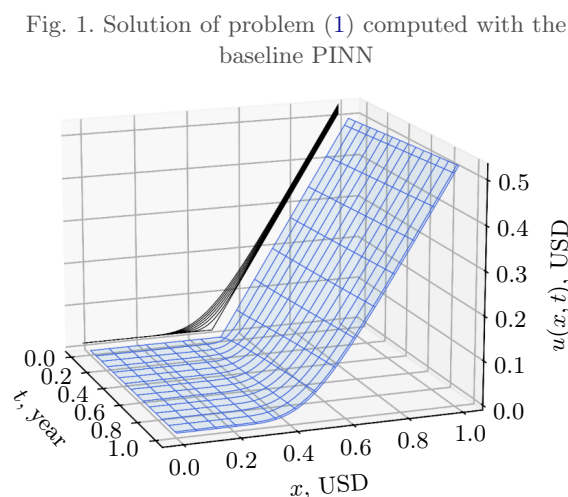


Рис. 2. График аналитического решения задачи (1)

Fig. 2. Close-form solution of problem (1)

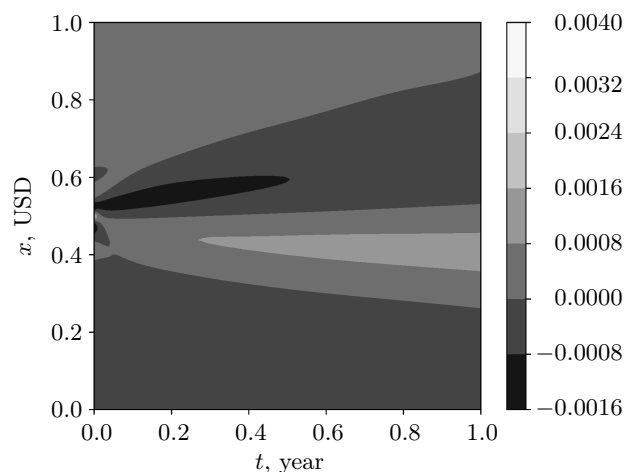


Рис. 3. Разность между решением базовой PINN для задачи (1) и аналитическим решением

Fig. 3. Signed difference between the baseline PINN solution of problem (1) and the analytical solution

5. Модификация PINN. По отношению к вышеописанной базовой архитектуре PINN в данной работе были сделаны следующие изменения:

- введено ограничение на знак значений решения $u(x, t)$,
- добавлен механизм адаптации весов перед слагаемыми функции потерь $\mathcal{L}(w)$,
- изменен порядок инициализации всех тензоров,
- добавлен механизм последовательного обучения по времени,
- добавлен механизм внимания с помощью двух энкодеров,
- реализован эмбединг случайных функций Фурье.

Рассмотрим каждое из них подробнее.

5.1. Ограничение на знак значений решения $u(x, t)$. Искомое решение $u(x, t)$ представляет собой цену опциона и, следовательно, может принимать только неотрицательные значения. Поскольку решение базовой PINN имеет устойчивые отрицательные значения в областях близких к цене страйка $K = 0.5$, было введено дополнительное слагаемое в функцию потерь, штрафующее за отрицательные значения:

$$L_{\text{sgn}}(w) = \begin{cases} \frac{1}{N_{\text{sgn}}} \sum_{i=1}^{N_{\text{id}}} \left(u_{\text{id}}^p(x_k, t_k, w) \cdot \mathbf{1}_{\{u_{\text{id}}^p(x_i, t_i, w) < 0\}} \right)^2, & N_{\text{sgn}} > 0, \\ 0, & N_{\text{sgn}} = 0, \end{cases}$$

$$N_{\text{sgn}} = \sum_{i=1}^{N_{\text{id}}} \mathbf{1}_{\{u_{\text{id}}^p(x_i, t_i, w) < 0\}}, \quad \mathbf{1}_{\mathcal{A}(x)} = \begin{cases} 1, & x \in \mathcal{A}, \\ 0, & x \notin \mathcal{A}. \end{cases}$$

Дополнительное слагаемое L_{sgn} , несомненно, оказывает влияние на ландшафт функции потерь и скорость обучения PINN. Теперь общая функция потерь (6) принимает вид

$$\mathcal{L}(w) = L_{\text{ic}}(w) + L_{\text{lb}}(w) + L_{\text{rb}}(w) + L_{\text{id}}(w) + L_{\text{sgn}}(w). \quad (8)$$

Соответствующим образом изменится и задача обучения (7).

5.2. Адаптивные веса слагаемых функции потерь. Предварительный анализ во время обучения показал, что слагаемые функции потерь (6) могут в разное время отличаться друг от друга на 1–2 порядка. Следовательно, в такие моменты вклад большего слагаемого в общую функцию потерь будет преобладать и оказывать тем самым преобладающее влияние на качество обучения всех обучаемых параметров w . Проблема балансировки вклада слагаемых, как указывается в исследованиях, лежит в недостатках метода градиентного спуска, применяемого к многоцелевой или состоящей из нескольких частей функции потерь [22, 23, 29]. Стандартным подходом исправить дисбаланс является введение весов в (8):

$$\mathcal{L}(w) = \lambda_{\text{ic}} L_{\text{ic}}(w) + \lambda_{\text{lb}} L_{\text{lb}}(w) + \lambda_{\text{rb}} L_{\text{rb}}(w) + \lambda_{\text{id}} L_{\text{id}}(w) + \lambda_{\text{sgn}} L_{\text{sgn}}(w).$$

В работе [21] приведены некоторые современные методы определения коэффициентов λ_j . Еще один вариант предложен в работе [30].

В данной работе используется алгоритм, как в работах [31] и [32], когда веса λ_j приводят слагаемые функции потерь к одному порядку. Задается минимальный порог отклонений, при превышении которого требуется переопределение весового коэффициента. Веса пересматриваются каждую эпоху, а порог установлен как $\lambda_{\text{th}} = 0.1$. Веса определяются следующим образом:

$$\lambda_j = \begin{cases} \max \left(\lambda_{\text{th}} \cdot \left(\frac{\max_L}{L_j} \right), 1 \right), & L_j > 0, \\ 1, & L_j = 0, \end{cases}$$

где $\max_L = \max(L_j)$, $j \in \{\text{id}, \text{ic}, \text{lb}, \text{rb}, \text{sgn}\}$.

5.3. Изменение способа инициализации всех тензоров. Вместо инициализации начальных значений весов с помощью нормально стандартно распределенной случайной величины, как в случае с базовой PINN, в качестве модификации предусмотрена инициализация Ксавьера [33]. В этом случае

веса инициализируются нормально распределенной случайной величиной $\xi \sim \mathcal{N}(0, s^2)$, где

$$s = C \cdot \sqrt{\frac{2}{n_{\text{prev}} + n_{\text{next}}}}, \quad C = \text{const.}$$

Здесь n_{prev} и n_{next} — количество нейронов на предшествующем и последующем слоях.

Как указывают авторы [33], для сетей с функцией активации в виде гиперболического тангенса предлагаемая инициализация может быть весьма полезной, поскольку сохраняет величину значений функции активаций при прямом проходе и градиентов при обратном. Используемая в настоящей работе функция активации $\tanh(y)$ сжимает входные данные в диапазон от -1 до 1 , поэтому значения производных стремятся к нулю для относительно больших значений y , и градиенты могут исчезать во время обучения. Применение инициализации Ксавьера с рекомендуемым коэффициентом $C = 1.666$ может помочь этого избежать.

5.4. Механизм последовательного обучения по времени. Поиск причин, почему базовая PINN не точно “улавливает” начальное условие, привел к работам, в которых авторы при обучении PINN обращают внимание на существование внутренней причинно-следственной структуры в физических системах [20, 34, 35]. Подобная структура определяет, как локальные изменения в начальных и граничных условиях системы отражаются в ее состояниях в более поздние моменты времени. Наиболее явно она проявляется в классических численных методах, где дискретизация дифференциального уравнения обычно проводится так, чтобы гарантировать наличие решения в момент времени t_n прежде чем аппроксимировать решение в момент времени t_{n+1} .

В работе [34] представлен анализ, подтверждающий гипотезу о том, что PINN при обучении нарушают причинно-следственную связь и неявно смещают решение в область начальных решений более “позднего” времени. Доказательство приведено в [34, section 2]. Чтобы устранить этот фундаментальный недостаток, авторы предлагают изменить методику одновременного обучения на всех данных и ввести дополнительные временные веса, которые явно учитывают причинно-следственную структуру, характеризующую решение нелинейных PDE.

Идея состоит в том, чтобы разбить область определения на равные по времени сегменты. Затем вычислять функцию потерь для каждого сегмента, однако в общую функцию потерь для внутренней области включать их с обучаемыми весами, причем вес “позднего” сегмента будет мал до того момента, пока сеть не обучится и не уменьшит потери во всех предшествующих, “ранних”, сегментах.

Другими словами, временная область разбивается на N сегментов, добавляются веса и тогда

$$L_{\text{id}}(w) = \frac{1}{N} \sum_{i=1}^N h_i L_{\text{id}}^i(w), \quad h_i = \exp \left(-\varepsilon \sum_{k=1}^{i-1} L_{\text{id}}^k(w) \right).$$

В результате $L_{\text{id}}^i(w)$ не будет влиять на общую функцию потерь до тех пор, пока все предыдущие остатки $\{L_{\text{id}}^k(w)\}_{k=1}^{i-1}$ не уменьшатся до достаточно малых значений. Тем самым реализуется причинно-следственная структура решения уравнения. Авторы рекомендуют использовать гиперпараметр *причинно-следственной толерантности* (causal tolerance) $\varepsilon \in [1, 10]$, а количество сегментов $N \in [10, 100]$. В данной работе использованы значения $\varepsilon = 1$ и $N = 16$.

Для большей наглядности процесса обучения PINN с соблюдением причинно-следственной связи на рис. 4 представлена динамика изменения решения PINN в первые 10000 эпох обучения (использованы также все представленные в данной статье методы).

Видно, что сначала обучаются области решения, лежащие ближе к начальному условию. Необходимо отметить, что введение принципа причинно-следственной связи в PINN в нашем случае увеличило временные затраты на обучение на одну эпоху примерно в 1.5 раза. Анализ и результаты будут представлены ниже в соответствующем разделе.

5.5. Механизм внимания. Основываясь на достижениях в области компьютерного зрения и обработки естественного языка [36], базирующихся на разработке и использовании механизмов внимания [37], авторы работы [22] представили новую архитектуру для PINN. Эта архитектура почти такая же, как у стандартного многослойного персептрона, но с добавлением двух энкодеров и изменением в порядке прямого прохода.

В частности, входные данные $\mathbf{x} \in \mathbb{R}^{n_{\text{inp}}}$ проецируются в пространство признаков $\mathbb{R}^{n_{\text{hid}}}$ с помощью двух энкодеров, $\mathbf{U}$ и $\mathbf{V}$, и затем с помощью поэлементного умножения обновляют веса в каждом скрытом

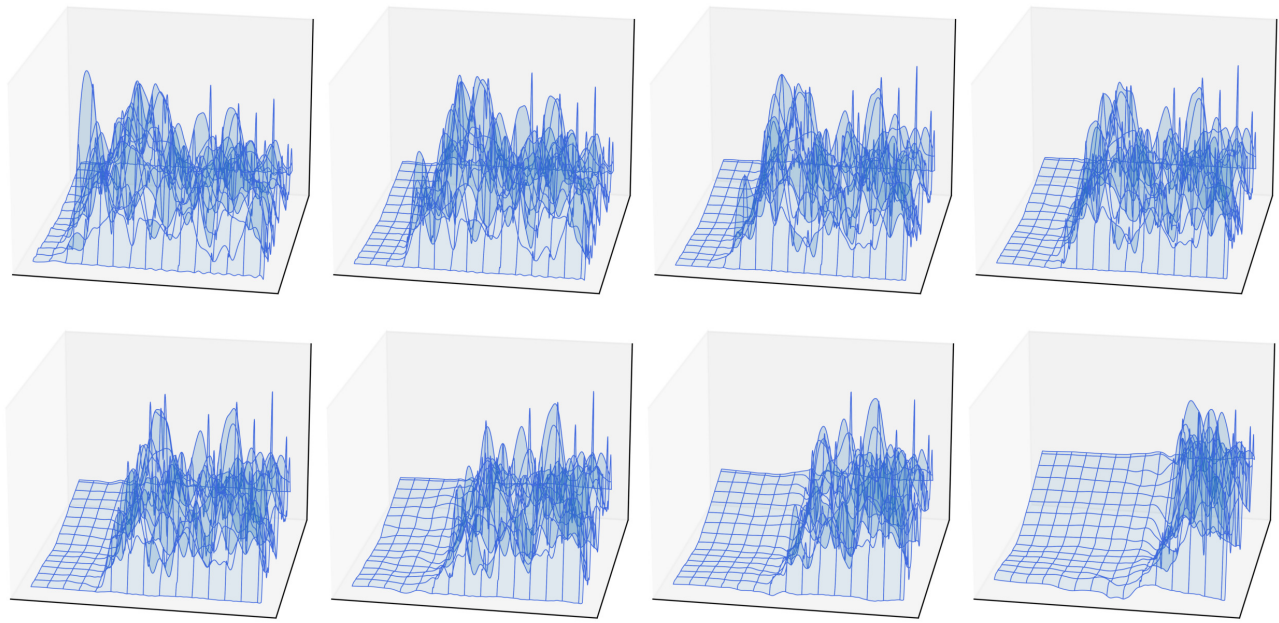


Рис. 4. Динамика изменения решения задачи (1) с помощью PINN при использовании последовательного обучения. Для решения применяются все представленные в статье методы

Fig. 4. Evolution of the problem (1) solution via PINN with sequential training strategy. The solution is computed using all methods described in this work

слое PINN. Как отмечено в [22], это позволяет учитывать взаимодействия между различными входными измерениями и усиливать скрытые состояния PINN с помощью остаточных связей. В этом случае базовая PINN (5) преобразуется к следующему виду (см. также [22, рис. 9]):

$$\begin{aligned} \mathbf{u}^{(0)}(\mathbf{x}) &= \phi(W^{(0)} \cdot \mathbf{x} + \mathbf{b}^{(0)}), \\ \mathbf{U} &= \phi(W_U \cdot \mathbf{x} + \mathbf{b}_U), \quad \mathbf{V} = \phi(W_V \cdot \mathbf{x} + \mathbf{b}_V), \\ \mathbf{u}_0^{(m)}(\mathbf{x}) &:= \phi(W^{(m)} \cdot \mathbf{u}^{(m-1)}(\mathbf{x}) + \mathbf{b}^{(m)}), \\ \mathbf{u}^{(m)}(\mathbf{x}) &= \mathbf{u}_0^{(m)}(\mathbf{x}) \odot \mathbf{U} + (\mathbf{I} - \mathbf{u}_0^{(m)}(\mathbf{x})) \odot \mathbf{V}, \quad m = 1, \dots, l_{\text{hid}}, \\ \mathbf{u}^{(l_{\text{hid}}+1)}(\mathbf{x}) &= W^{(l_{\text{hid}}+1)} \cdot \mathbf{u}^{(l_{\text{hid}})}(\mathbf{x}) + \mathbf{b}^{(l_{\text{hid}}+1)}, \\ \mathbf{u}^p(\mathbf{x}) &= \mathbf{u}^{(l_{\text{hid}}+1)}(\mathbf{x}), \end{aligned}$$

при этом, в дополнение к (5),

$$\begin{aligned} W_U, W_V &\in \mathbb{R}^{n_{\text{hid}} \times n_{\text{inp}}}, \\ \mathbf{I}, \mathbf{b}_U, \mathbf{b}_V &\in \mathbb{R}^{n_{\text{hid}}}. \end{aligned}$$

Здесь символом $\odot$ обозначено поэлементное умножение. В обучаемые параметры сети w добавляются веса и смещения энкодеров $\mathbf{U}$ и $\mathbf{V}$. Авторы архитектуры отмечают возрастающие при ее использовании вычислительные затраты, однако отмечают также и существенное повышение точности результатов. Исходя из этого, архитектура рассматриваемой в данной работе базовой PINN была трансформирована соответствующим образом.

5.6. Эмбе́ддинг случайных функций Фурье. Еще одной возможной причиной получения с помощью PINN численных решений с недостаточной точностью является спектральное смещение или F-принцип. Эффект был одновременно обнаружен двумя группами исследователей ([38] и [39], см. также [40]) и заключается в том, что глубокие нейронные сети быстрее обучаются низкочастотным компонентам решения, чем высокочастотным. Это подразумевает неявное смещение, согласно которому сети

имеют тенденцию сначала аппроксимировать обучающие данные с помощью низкочастотной функции, что сказывается на точности. Возможно, что именно этот эффект в большей степени не позволяет решению базовой PINN корректно аппроксимировать область, выделенную эллипсом на рис. 1.

В работе [41], опираясь на методы использования нейронного касательного ядра (neural tangent kernel, NTK) [42], авторы показали, что стандартные многослойные перцептроны не способны к обучению высоким частотам как в теории, так и на практике. Для исключения такого спектрального смещения авторы предложили подход, с помощью которого можно выбрать и использовать для конкретной задачи в качестве эмбедингов случайные функции Фурье [43]. Такой подход значительно повышает производительность многослойных перцептронов для задач регрессии малой размерности, что соответствует рассматриваемому в данной работе случаю.

До того, как вектор входных данных $\mathbf{x} \in \mathbb{R}^{n_{\text{inp}}}$ проходит через PINN, он преобразуется в высокочастотные сигналы как $\gamma: \mathbb{R}^{n_{\text{inp}}} \rightarrow \mathbb{R}^{n_{\text{hid}}}$, а именно,

$$\gamma(\mathbf{x}) = \begin{bmatrix} \cos(B\mathbf{x}) \\ \sin(B\mathbf{x}) \end{bmatrix},$$

где каждый элемент в $B \in \mathbb{R}^{\frac{n_{\text{hid}}}{2} \times n_{\text{inp}}}$ выбирается из нормального распределения $\mathcal{N}(0, b^2)$, где $b > 0$ является гиперпараметром. На практике авторы рекомендуют использовать умеренно большое $b \in [1, 10]$. В данной работе использовано значение $b = 1$.

6. Результаты и обсуждение. Напомним, что результаты решения базовой PINN уравнения Блэка–Шоулса представлены на рис. 1. Рис. 5 отражает решение уравнения Блэка–Шоулса с помощью PINN, в которой присутствуют ранее описанные и реализованные модификации.

Видно, что данная PINN уже качественно правильно отражает область около страйка в момент погашения и все решение $u(x, t)$ находится в положительной области. Поэтому вопросы, которые возникли при рассмотрении результатов работы базовой PINN, можно считать условно решенными. Однако окончательно решить вопрос с конфигурацией станет возможным после исследований методом последовательного исключения или добавления всех модификаций, а также подбора гиперпараметров, чтобы оценить вклад каждого из них в производительность PINN.

Поскольку исследование потребует многих циклов обучения сети, необходимы работа по оптимизации исполняемого кода программы и наличие соответствующего аппаратного обеспечения. Как уже указывалось ранее, среднее время на одну эпоху обучения увеличилось примерно в полтора раза при добавлении контроля причинно-следственной связи. Это вызвано тем, что производится расчет градиентов многих величин потерь внутри каждого сегмента. Вероятно, требуется оптимизация некоторых алгоритмов для снижения количества таких расчетов. Что касается аппаратной части, то расчеты в данной работе проводились на ноутбуке на CPU (Intel(R) Core(TM) i7-6700HQ, 16 GB RAM, CPU 2.60 GHz). Однако существенного ускорения расчетов можно добиться при использовании ресурсов графических процессоров GPU, отсутствующих в имеющемся аппаратном обеспечении.

Приведем сравнительные показатели работы двух PINN в табл. 1. Строка “baseline PINN” означает данные для базовой PINN, а “modified PINN”, соответственно, для модифицированной. В столбце, обозначенном Epochs, отражено количество эпох обучения сети. Скорость обучения на момент остановки приведена в столбце LR. Столбец t_{avg} отражает среднее время работы в секундах, приходящееся на одну эпоху. Текущий минимум функции потерь $\min_w \mathcal{L}(w)$ фиксировался в ходе обучения только в случае снижения $\mathcal{L}(w)$ на 5% от предыдущего минимума. На рис. 6 представлена динамика десятичного логарифма данного показателя для базовой и модифицированной PINN.

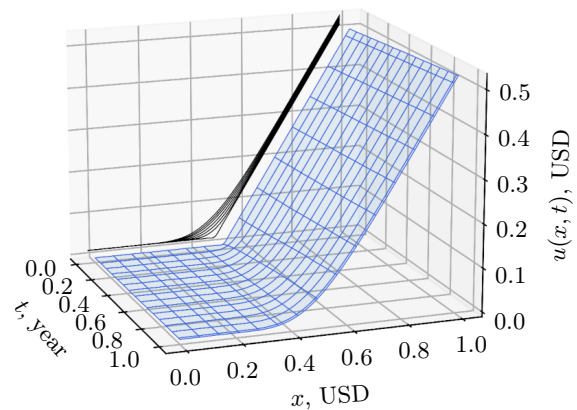


Рис. 5. График решения задачи (1) с помощью модифицированной PINN

Fig. 5. Solution of problem (1) computed with the modified PINN



Таблица 1. Показатели работы базовой и модифицированной PINN

Table 1. Comparative performance of the baseline and modified PINN

	Epochs	LR	$\min_w \mathcal{L}(w)$	MAE	$E_{L_2}^{\text{rel}}$	t_{avr}
baseline PINN	51640	7.31×10^{-5}	1.28×10^{-7}	3.35×10^{-4}	2.14×10^{-3}	1.5
modified PINN	51640	7.31×10^{-5}	1.47×10^{-6}	4.40×10^{-4}	2.63×10^{-3}	4.2

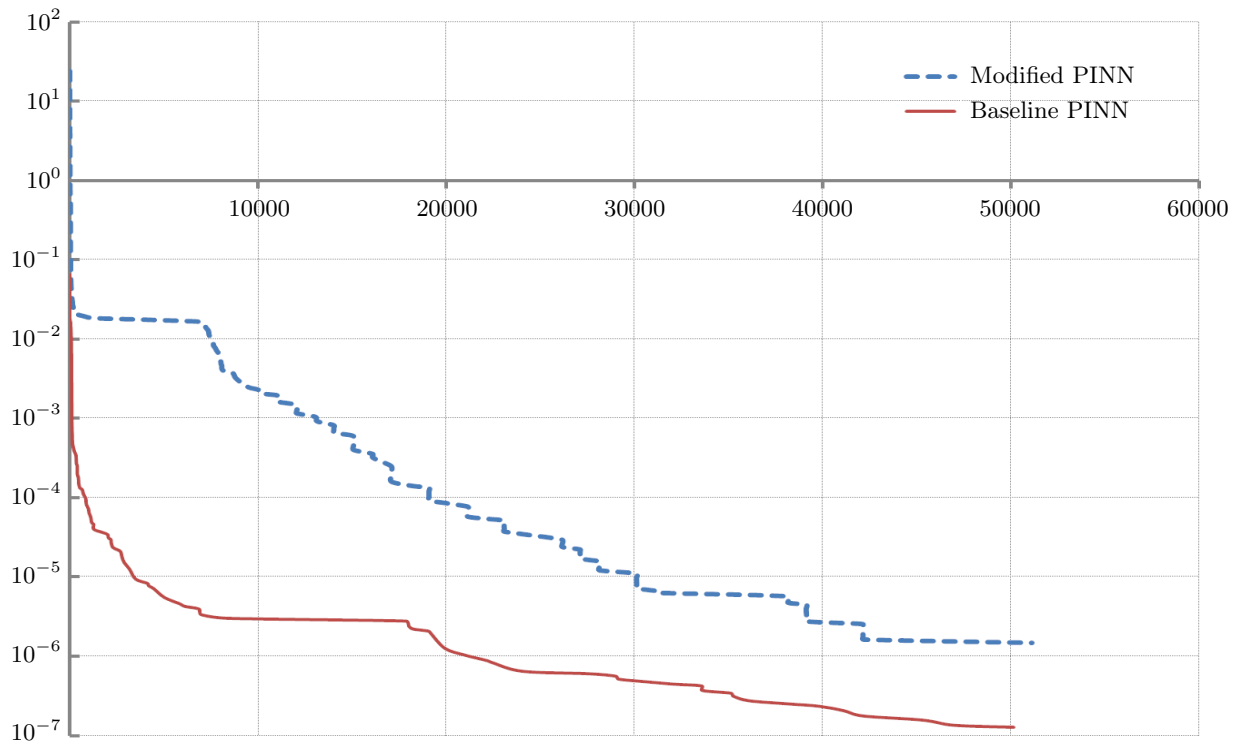


Рис. 6. Динамика $\lg \left(\min_w \mathcal{L}(w) \right)$ для базовой и модифицированной PINN

Fig. 6. Evolution of $\lg \left(\min_w \mathcal{L}(w) \right)$ for the baseline and modified PINN

Вопрос о прекращении обучения в данной работе решался по изменению выбранного показателя качества относительной L_2 -нормы (4). Экспериментально было определено, что $E_{L_2}^{\text{rel}}$ для базовой PINN перестает существенно меняться после примерно 4.5×10^4 эпох, поэтому обучение как базовой, так и модифицированной PINN было остановлено после 51640-й эпохи для сравнения показателей.

Из табл. 1 видно, что базовая PINN показывает меньшие значения $\min_w \mathcal{L}(w)$, MAE и $E_{L_2}^{\text{rel}}$, хотя одна из целей модификации была как раз превзойти по точности базовую PINN. Вероятно, реализовался именно тот случай, когда внесение модификаций усложнило ландшафт функции потерь так, что сеть стала обучаться заметно медленнее.

Несмотря на то, что модифицированная PINN качественно лучше охватывает проблемные области (наличие в решении базовой PINN отрицательных значений и отсутствие излома около страйка на момент экспирации), произошло снижение общей точности и увеличение времени обучения сети. Относительная L_2 -норма (4) для базовой PINN увеличилась с 2.14×10^{-3} до 2.63×10^{-3} после проведения модификации сети. При этом среднее время обучения, составлявшее 1.5 с на одну эпоху для базовой PINN, для модифицированной PINN выросло до 4.2 с.

Чтобы охарактеризовать качество решений базовой и модифицированной PINN вблизи точки излома $(0.5, 0)$, была введена локальная область $\Omega = [0.475, 0.525] \times [0, 0.05]$, соответствующая диапазону $\pm 5\%$ относительно значения страйка $x = 0.5$ и временному интервалу $t \in [0, 0.05]$. По решениям, полученным

с помощью нескольких последних сохраненных состояний сети, были вычислены локальные показатели ошибок в этой области. Рассчитывались средняя и наихудшая относительные L_2 -нормы, а также средняя и наихудшая максимальные абсолютные ошибки:

$$E_{\max} = \max_{k \in K} E_{\max}^{(k)}, \quad \bar{E}_{\max} = \frac{1}{|K|} \sum_{k \in K} E_{\max}^{(k)},$$

$$E_{\max}^{(k)} = \max_{(x,t) \in \Omega} \left| u_{(k)}^{\text{predict}}(x,t) - u^{\text{benchmark}}(x,t) \right|, \quad k \in K,$$

$$\Omega = [0.475, 0.525] \times [0, 0.05], \quad K = \{46000, 47000, 48000, 49000, 50000, 51000\}.$$

Результаты представлены в табл. 2. Видно, что при близких значениях локальной относительной L_2 -нормы модифицированная PINN демонстрирует существенно меньшую локальную максимальную абсолютную ошибку. Это может служить количественным подтверждением более точного воспроизведения решения в окрестности точки недифференцируемости именно модифицированной PINN, что ранее наблюдалось при визуальном анализе графиков решений.

Таблица 2. Локальные показатели ошибки решений PINN в окрестности точки недифференцируемости (0.5, 0)
 Table 2. Local error metrics of PINN solutions near the non-differentiable point (0.5, 0)

Method	$E_{\max}$	$\bar{E}_{\max}$	$(E_{L_2}^{\text{rel}})_{\max}$	$\bar{E}_{L_2}^{\text{rel}}$
baseline PINN	4.57×10^{-3}	$(4.23 \pm 0.20) \times 10^{-3}$	6.49×10^{-2}	$(5.95 \pm 0.41) \times 10^{-2}$
modified PINN	2.55×10^{-3}	$(2.09 \pm 0.31) \times 10^{-3}$	6.27×10^{-2}	$(5.70 \pm 0.47) \times 10^{-2}$

Анализ полученных значений в табл. 1 поднимает вопрос о необходимости исследования влияния и целесообразности каждого дополнения и модификации с точки зрения производительности сети и ее практического использования с учетом допустимых при этом погрешностей. Поэтому одним из направлений будущих исследований является разработка плана и проведение абляции (ablation study) — последовательного исключения/включения рассмотренных в статье модифицирующих компонентов (и/или их комбинаций) с последующей оценкой, как наличие каждого такого компонента отражается на показателях. Реализация такого исследования потребует наличия и использования графических процессоров GPU и соответствующей доработки программного обеспечения.

Еще одно возможное направление исследований лежит в применении полученных результатов для обучения ИНС и получения численных решений для различных уравнений типа Блэка–Шоулса с учетом недостаточной ликвидности и транзакционных издержек, а также для поиска численного решения дробных по времени уравнений того же типа.

7. Заключение. В работе описаны архитектуры и приведены значения гиперпараметров для двух физически информированных нейронных сетей (PINN), с помощью которых решается классическое уравнение Блэка–Шоулса. Решение базовой PINN по сравнению с аналитическим решением качественно недостаточно точно отражает область, соответствующую времени погашения опциона при цене базового актива в районе страйка (в нашем случае $x \approx 0.5$, $t \rightarrow 0$). Кроме того, решение базовой PINN содержит отрицательные значения. Проведенная модификация PINN позволила получить решение, которое в указанной области является экономически обоснованным (отсутствуют отрицательные значения цены). Также качественно улучшилось локальное поведение решения около страйка. Необходимо отметить, что внесенная модификация снизила при этом общую точность и увеличила время обучения сети. Для практического использования построенных PINN необходимо искать баланс между точностью получаемых решений и допустимых при ценообразовании опционов погрешностей.

Список литературы

1. Malliaris M., Salchenberger L. A neural network model for estimating option prices // Applied Intelligence. 1993. **3**, N 3. 193–206. doi [10.1007/BF00871937](https://doi.org/10.1007/BF00871937).
2. Hutchinson J.M., Lo A.W., Poggio T. A nonparametric approach to pricing and hedging derivative securities via learning networks // The Journal of Finance. 1994. **49**, N 3. 851–889. doi [10.1111/j.1540-6261.1994.tb00081.x](https://doi.org/10.1111/j.1540-6261.1994.tb00081.x).
3. Black F., Scholes M. The pricing of options and corporate liabilities // Journal of Political Economy. 1973. **81**, N 3. 637–654. doi [10.1086/260062](https://doi.org/10.1086/260062).



4. Raissi M., Perdikaris P., Karniadakis G. E. Machine learning of linear differential equations using Gaussian processes // *Journal of Computational Physics*. 2017. **348**, N 11. 683–693. doi [10.1016/j.jcp.2017.07.050](https://doi.org/10.1016/j.jcp.2017.07.050).
5. 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*. 2019. **378**. 686–707. doi [10.1016/j.jcp.2018.10.045](https://doi.org/10.1016/j.jcp.2018.10.045).
6. Baydin A. G., Pearlmutter B. A., Radul A. A., et al. Automatic differentiation in machine learning: a survey // *Journal of Machine Learning Research*. 2018. **18**, N 153. 1–43. doi [10.48550/arXiv.1502.05767](https://doi.org/10.48550/arXiv.1502.05767).
7. Rall L. B. Automatic differentiation: techniques and applications. *Lecture Notes in Computer Science*. Vol. 120. 1 ed. Berlin: Springer, 1981. doi [10.1007/3-540-10861-0](https://doi.org/10.1007/3-540-10861-0).
8. Cuomo S., Di Cola V. S., Giampaolo F., et al. Scientific machine learning through physics-informed neural networks: Where we are and what's next // *Journal of Scientific Computing*. 2022. **92**, N 3. Article Number 88. doi [10.1007/s10915-022-01939-z](https://doi.org/10.1007/s10915-022-01939-z).
9. Karniadakis G. E., Kevrekidis I. G., Lu L., et al. Physics-informed machine learning // *Nature Reviews Physics*. 2021. **3**, N 6. 422–440. doi [10.1038/s42254-021-00314-5](https://doi.org/10.1038/s42254-021-00314-5).
10. Вершинин В. Е., Пономарев Р. Ю. Применение методов нейросетевого моделирования при решении начально-краевых задач для дифференциальных уравнений в частных производных // *Вестник Тюменского государственного университета. Серия: Физико-математическое моделирование. Нефть, газ, энергетика*. 2023. **9**, № 3(35). 132–147. doi [10.21684/2411-7978-2023-9-3-132-147](https://doi.org/10.21684/2411-7978-2023-9-3-132-147).
11. Tarkhov D., Lazovskaya T., Antonov V. Adapting PINN models of physical entities to dynamical data // *Computation*. 2023. **11**, N 9. Article Number 168. doi [10.3390/computation11090168](https://doi.org/10.3390/computation11090168).
12. Зрелова Д. П., Ульянов С. В. Модели физически информированных / осведомленных классических Лагранжевых / Гамильтоновых нейронных сетей в глубоком обучении // *Современные информационные технологии и ИТ-образование*. 2022. **18**, № 2. 310–325. doi [10.25559/SITITO.18.202202.310-325](https://doi.org/10.25559/SITITO.18.202202.310-325).
13. Лазовская Т. В., Тархов Д. А., Шемякина Т. А., Чориев М. Д. Эволюционный подход к формированию архитектуры PINN для приближенного решения уравнения Лапласа в двух постановках: с разрывным краевым условием и данными измерений внутри квадратной области // *Современные информационные технологии и ИТ-образование*. 2023. **19**, № 2. 438–446. doi [10.25559/sitito.019.202302.438-446](https://doi.org/10.25559/sitito.019.202302.438-446).
14. Gatta F., Di Cola V. S., Giampaolo F., et al. Meshless methods for American option pricing through physics-informed neural networks // *Engineering Analysis with Boundary Elements*. 2023. **151**, N 3. 68–82. doi [10.1016/j.engana-bound.2023.02.040](https://doi.org/10.1016/j.engana-bound.2023.02.040).
15. Dey R., Srivastava S. N., Domoshnitsky A., et al. A data driven study on fractional Black–Scholes–merton equation using physics informed neural network // Available at SSRN 4995226. doi [10.2139/ssrn.4995226](https://doi.org/10.2139/ssrn.4995226).
16. Song L., Tan Y., Yu F., et al. Optimal approximations for the free boundary problems of the space-time fractional Black–Scholes equations using a combined physics-informed neural network // *Scientific Reports*. 2024. **14**, N 1. Article Number 25289. doi [10.1038/s41598-024-77073-7](https://doi.org/10.1038/s41598-024-77073-7).
17. Hainaut D., Casas A. Option pricing in the Heston model with physics inspired neural networks // *Annals of Finance*. 2024. **20**, N 3. 353–376. doi [10.1007/s10436-024-00452-7](https://doi.org/10.1007/s10436-024-00452-7).
18. Daw A., Bu J., Wang S., Perdikaris P., et al. Mitigating propagation failures in physics-informed neural networks using retain-resample-release (R3) sampling // *Proceedings of the 40th International Conference on Machine Learning Honolulu, Hawaii, USA, July 23–29, 2023*. PMLR, Vol. 202, 2023, pp. 7264–7302. <https://proceedings.mlr.press/v202/daw23a.html>. Cited August 23, 2026.
19. Jagtap A. D., Kawaguchi K., Karniadakis G. E. Adaptive activation functions accelerate convergence in deep and physics-informed neural networks // *Journal of Computational Physics*. 2020. **404**, N C. Article Number 109136. doi [10.1016/j.jcp.2019.109136](https://doi.org/10.1016/j.jcp.2019.109136).
20. Krishnapriyan A., Gholami A., Zhe S., et al. Characterizing possible failure modes in physics-informed neural networks // *35th Conference on Neural Information Processing Systems (NeurIPS 2021)*, Online Conference, December 6–14, 2021. *Advances in Neural Information Processing Systems*, Vol. 32, 2021. pp. 26548–26560. <https://proceedings.neurips.cc/>. Cited August 23, 2026.
21. McClenny L. D., Braga-Neto U. M. Self-adaptive physics-informed neural networks // *Journal of Computational Physics*. 2023. **474**, N C. Article Number 111722. doi [10.1016/j.jcp.2022.111722](https://doi.org/10.1016/j.jcp.2022.111722).
22. Wang S., Teng Y., Perdikaris P. Understanding and mitigating gradient flow pathologies in physics-informed neural networks // *SIAM Journal on Scientific Computing*. 2021. **43**, N 5. A3055–A3081. doi [10.1137/20M1318043](https://doi.org/10.1137/20M1318043).
23. Wang S., Yu X., Perdikaris P. When and why PINNs fail to train: A neural tangent kernel perspective // *Journal of Computational Physics*. 2022. **449**, N C. Article Number 110768. doi [10.1016/j.jcp.2021.110768](https://doi.org/10.1016/j.jcp.2021.110768).

24. Дышаев М.М., Фёдоров В.Е. Симметрии и точные решения одного нелинейного уравнения ценообразования опционов // Уфимск. матем. журн. 2017. **9**, № 1. 29–40. doi [10.13108/2017-9-1-29](https://doi.org/10.13108/2017-9-1-29).
25. Dyshaev M.M., Fedorov V.E. Comparing of some sensitivities (Greeks) for nonlinear models of option pricing with market illiquidity // Математические заметки СВФУ. 2019. **26**, N 2. 94–108. doi [10.25587/SVFU.2019.102.31514](https://doi.org/10.25587/SVFU.2019.102.31514).
26. Merton R.C. Theory of rational option pricing // The Bell Journal of Economics and Management Science. 1973. **4**, N 1. 141–183. doi [10.2307/3003143](https://doi.org/10.2307/3003143).
27. Ширяев А.Н. Основы стохастической финансовой математики: 2 тома. М.: МЦНМО, 2016.
28. Huber P.J. Robust Estimation of a Location Parameter // The Annals of Mathematical Statistics. 1964. **35**, N 1. 73–101. doi [10.1214/aoms/1177703732](https://doi.org/10.1214/aoms/1177703732).
29. Shin Y., Darbon J., Karniadakis G. E. On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs // Communications in Computational Physics. 2020. **28**, N 5. 2042–2074. doi [10.4208/cicp.0A-2020-0193](https://doi.org/10.4208/cicp.0A-2020-0193).
30. Liu D., Wang Y. A Dual-Dimer method for training physics-constrained neural networks with minimax architecture // Neural Networks. 2021. **136**. 112–125. doi [10.1016/j.neunet.2020.12.028](https://doi.org/10.1016/j.neunet.2020.12.028).
31. Zobeiry N., Humfeld K.D. A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications // Engineering Applications of Artificial Intelligence. 2021. **101**. Article Number 104232. doi [10.1016/j.engappai.2021.104232](https://doi.org/10.1016/j.engappai.2021.104232).
32. Чупров И.А., Гао Ц., Ефременко Д.С. и др. Оптимизация физико-информированных нейронных сетей для решения нелинейного уравнения Шредингера // Доклады Российской академии наук. Математика, информатика, процессы управления. 2023. **514**, № 2. 28–38. doi [10.31857/S2686954323601586](https://doi.org/10.31857/S2686954323601586).
33. Glorot X., Bengio Y. Understanding the difficulty of training deep feedforward neural networks // Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS), Chia Laguna Resort, Sardinia, Italy, May 13–15, 2010. Proceedings of Machine Learning Research, Vol. 9, 2010. pp. 249–256.
34. Wang S.-F., Sankaran S., Perdikaris P. Respecting causality for training physics-informed neural networks // Computer Methods in Applied Mechanics and Engineering. 2024. **421**. Article Number 116813. doi [10.1016/j.cma.2024.116813](https://doi.org/10.1016/j.cma.2024.116813).
35. Matthey R., Ghosh S. A novel sequential method to train physics informed neural networks for Allen Cahn and Cahn Hilliard equations // Computer Methods in Applied Mechanics and Engineering. 2022. **390**, N 12. Article Number 114474. doi [10.1016/j.cma.2021.114474](https://doi.org/10.1016/j.cma.2021.114474).
36. Vaswani A., Shazeer N., Parmar N., et al. Attention is all you need // Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, December 4–9, 2017. Advances in Neural Information Processing Systems, Vol. 30. Red Hook, NY, USA: Curran Associates, Inc., 2017. pp. 6000–6010.
37. Niu Z., Zhong G., Yu H. A review on the attention mechanism of deep learning // Neurocomputing. 2021. **452**. 48–62. doi [10.1016/j.neucom.2021.03.091](https://doi.org/10.1016/j.neucom.2021.03.091).
38. Rahaman N., Baratin A., Arpit D., et al. On the spectral bias of neural networks // Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, June 9–15, 2019. Proceedings of Machine Learning Research (PMLR), Vol. 97. Red Hook, NY, USA: Curran Associates, Inc., 2019. pp. 5301–5310. doi [10.48550/arXiv.1806.08734](https://doi.org/10.48550/arXiv.1806.08734).
39. Xu Z.Q.J., Zhang Y., Xiao Y. Training behavior of deep neural network in frequency domain // Neural Information Processing. 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019. Lecture Notes in Computer Science, Vol. 11953. Cham: Springer, 2019. pp. 264–274. doi [10.1007/978-3-030-36708-4_22](https://doi.org/10.1007/978-3-030-36708-4_22).
40. Xu Z.-Q. J., Zhang Y., Luo T., et al. Frequency principle: Fourier analysis sheds light on deep neural networks // Communications in Computational Physics. 2020. **28**, N 5. 1746–1767. doi [10.4208/cicp.0A-2020-0085](https://doi.org/10.4208/cicp.0A-2020-0085).
41. Tancik M., Srinivasan P., Mildenhall B., et al. Fourier features let networks learn high frequency functions in low dimensional domains // Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada, December 6–12, 2020. Advances in Neural Information Processing Systems, Vol. 33. Red Hook, NY, USA: Curran Associates, Inc., 2020. pp. 7537–7547.
42. Jacot A., Gabriel F., Hongler C. Neural tangent kernel: convergence and generalization in neural networks // Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada, December 2–8, 2018. Advances in Neural Information Processing Systems, Vol. 31. Red Hook, NY, USA: Curran Associates, Inc., 2018. pp. 8580–8589.



43. Rahimi A., Recht B. Random features for large-scale kernel machines // Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3–6, 2007. Advances in Neural Information Processing Systems (NIPS), Vol. 20. Red Hook, NY, USA: Curran Associates, Inc., 2007. pp. 1177–1184.

Получена
4 июля 2026 г.

Принята
29 июля 2026 г.

Опубликована
11 сентября 2026 г.

Информация об авторе

Михаил Михайлович Дышаев — к.ф.-м.н., науч. сотр.; ООО “НПК “КОБОТОРИ”, ул. Аношкина, д. 4, помещ. 1, 454100, Челябинск, Российская Федерация.

References

1. M. Malliaris and L. Salchenberger, “A neural network model for estimating option prices,” *Applied Intelligence* **3** (3), 193–206 (1993). doi [10.1007/BF00871937](https://doi.org/10.1007/BF00871937).
2. J. M. Hutchinson, A. W. Lo, and T. Poggio, “A Nonparametric Approach to Pricing and Hedging Derivative Securities via Learning Networks,” *The Journal of Finance* **49** (3), 851–889 (1994). doi [10.1111/j.1540-6261.1994.tb00081.x](https://doi.org/10.1111/j.1540-6261.1994.tb00081.x).
3. F. Black and M. Scholes, “The Pricing of Options and Corporate Liabilities,” *Journal of Political Economy* **81** (3), 637–654 (1973). doi [10.1086/260062](https://doi.org/10.1086/260062).
4. M. Raissi, P. Perdikaris, and G. E. Karniadakis, “Machine learning of linear differential equations using Gaussian processes,” *Journal of Computational Physics* **348** (11), 683–693 (2017). doi [10.1016/j.jcp.2017.07.050](https://doi.org/10.1016/j.jcp.2017.07.050).
5. M. Raissi, P. Perdikaris, and G. 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). doi [10.1016/j.jcp.2018.10.045](https://doi.org/10.1016/j.jcp.2018.10.045).
6. A. G. Baydin, B. A. Pearlmutter, A. A. Radul, et al., “Automatic differentiation in machine learning: a survey,” *Journal of Machine Learning Research* **18** (153), 1–43 (2018). doi [10.48550/arXiv.1502.05767](https://doi.org/10.48550/arXiv.1502.05767). Cited August 23, 2026.
7. L. B. Rall, *Automatic differentiation: techniques and applications* (Lecture Notes in Computer Science Vol. 120, 1 ed.) (Springer, Berlin, 1981). doi [10.1007/3-540-10861-0](https://doi.org/10.1007/3-540-10861-0).
8. S. Cuomo, V. S. Di Cola, F. Giampaolo, et al., “Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What’s Next,” *Journal of Scientific Computing* **92** (3), Article Number 88 (2022). doi [10.1007/s10915-022-01939-z](https://doi.org/10.1007/s10915-022-01939-z).
9. G. E. Karniadakis, I. G. Kevrekidis, L. Lu, et al., “Physics-informed machine learning,” *Nature Reviews Physics* **3** (6), 422–440 (2021). doi [10.1038/s42254-021-00314-5](https://doi.org/10.1038/s42254-021-00314-5).
10. V. E. Vershinin and R. Ju. Ponomarev, “Application of neural network modeling methods in solving initial boundary value problems for partial differential equations,” *Tyumen State University Herald. Physical and Mathematical Modeling. Oil, Gas, Energy* **9** (3), 132–147 (2023). doi [10.21684/2411-7978-2023-9-3-132-147](https://doi.org/10.21684/2411-7978-2023-9-3-132-147). [in Russian].
11. D. Tarkhov, T. Lazovskaya, and V. Antonov, “Adapting PINN Models of Physical Entities to Dynamical Data,” *Computation* **11** (9), Article Number 168 (2023). doi [10.3390/computation11090168](https://doi.org/10.3390/computation11090168).
12. D. P. Zrelova and S. V. Ul’janov, “Physics-Informed Classical Lagrange / Hamilton Neural Networks in Deep Learning,” *Modern Information Technologies and IT-Education* **18** (2), 310–325 (2022). doi [10.25559/SITITO.18.202202.310-325](https://doi.org/10.25559/SITITO.18.202202.310-325). [in Russian].
13. T. V. Lazovskaya, D. A. Tarhov, T. A. Shemjakina, and M. D. Choriev, “Evolutionary Approach to PINN Architecture Design for Approximate Solving the Laplace Equation in two Statements: With Discontinuous Initial Condition or with Measurement Data Inside a Square Domain,” *Modern Information Technologies and IT-Education* **19** (2), 438–446 (2023). doi [10.25559/sitito.019.202302.438-446](https://doi.org/10.25559/sitito.019.202302.438-446). [in Russian].
14. F. Gatta, V. S. Di Cola, F. Giampaolo, et al., “Meshless methods for American option pricing through physics-informed neural networks,” *Engineering Analysis with Boundary Elements* **151** (3), 68–82 (2023). doi [10.1016/j.enganabound.2023.02.040](https://doi.org/10.1016/j.enganabound.2023.02.040).
15. R. Dey, S. N. Srivastava, A. Domoshnitsky, et al., “A data driven study on fractional Black–Scholes–merton equation using physics informed neural network,” Available at SSRN 4995226. doi [10.2139/ssrn.4995226](https://doi.org/10.2139/ssrn.4995226).

16. L. Song, Y. Tan, F. Yu, et al., “Optimal approximations for the free boundary problems of the space-time fractional Black–Scholes equations using a combined physics-informed neural network,” *Scientific Reports* **14** (1), Article Number 25289 (2024). doi [10.1038/s41598-024-77073-7](https://doi.org/10.1038/s41598-024-77073-7).
17. D. Hainaut and A. Casas, “Option pricing in the Heston model with physics inspired neural networks,” *Annals of Finance* **20** (3), 353–376 (2024). doi [10.1007/s10436-024-00452-7](https://doi.org/10.1007/s10436-024-00452-7).
18. A. Daw, J. Bu, S. Wang, et al., “Mitigating propagation failures in physics-informed neural networks using retain-resample-release (R3) sampling,” in *Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA, July 23–29, 2023* (PMLR, Vol. 202, 2023), pp. 7264–7302. <https://proceedings.mlr.press/v202/daw23a.html>. Cited August 23, 2026.
19. A. D. Jagtap, K. Kawaguchi, and G. E. Karniadakis, “Adaptive activation functions accelerate convergence in deep and physics-informed neural networks,” *Journal of Computational Physics* **404** (C), Article Number 109136 (2020). doi [10.1016/j.jcp.2019.109136](https://doi.org/10.1016/j.jcp.2019.109136).
20. A. Krishnapriyan, A. Gholami, S. Zhe, et al., “Characterizing possible failure modes in physics-informed neural networks,” in *35th Conference on Neural Information Processing Systems (NeurIPS 2021), Online Conference, December 6–14, 2021* (Advances in Neural Information Processing Systems, Vol. 32, 2021), pp. 26548–26560. <https://proceedings.neurips.cc/> Cited August 23, 2026.
21. L. D. McClenny and U. M. Braga-Neto, “Self-adaptive physics-informed neural networks,” *Journal of Computational Physics* **474** (C), Article Number 111722 (2023). doi [10.1016/j.jcp.2022.111722](https://doi.org/10.1016/j.jcp.2022.111722).
22. S. Wang, Y. Teng, and P. Perdikaris, “Understanding and Mitigating Gradient Flow Pathologies in Physics-Informed Neural Networks,” *SIAM Journal on Scientific Computing* **43** (5), A3055–A3081 (2021). doi [10.1137/20M1318043](https://doi.org/10.1137/20M1318043).
23. S. Wang, X. Yu, and P. Perdikaris, “When and why PINNs fail to train: A neural tangent kernel perspective,” *Journal of Computational Physics* **449** (C), Article Number 110768 (2022). doi [10.1016/j.jcp.2021.110768](https://doi.org/10.1016/j.jcp.2021.110768).
24. M. M. Dyshaev and V. E. Fedorov, “Symmetries and exact solutions of a nonlinear pricing options equation,” *Ufa Mathematical Journal* **9** (1), 29–40 (2017). doi [10.13108/2017-9-1-29](https://doi.org/10.13108/2017-9-1-29).
25. M. M. Dyshaev and V. E. Fedorov, “Comparing of some sensitivities (Greeks) for nonlinear models of option pricing with market illiquidity,” *Mathematical Notes of NEFU* **26** (2), 94–108 (2019). doi [10.25587/SVFU.2019.102.31514](https://doi.org/10.25587/SVFU.2019.102.31514).
26. R. C. Merton, “Theory of rational option pricing,” *The Bell Journal of Economics and Management Science* **4** (1), 141–183 (1973). doi [10.2307/3003143](https://doi.org/10.2307/3003143).
27. A. N. Shirjaev, *Fundamentals of Stochastic financial Mathematics*: Vol. 1 and 2. (MCNMO, M., 2016). [in Russian].
28. P. J. Huber, “Robust Estimation of a Location Parameter,” *The Annals of Mathematical Statistics* **35** (1), 73–101 (1964). doi [10.1214/aoms/1177703732](https://doi.org/10.1214/aoms/1177703732).
29. Y. Shin, J. Darbon, and G. E. Karniadakis, “On the convergence of physics informed neural networks for linear second-order elliptic and parabolic type PDEs,” *Communications in Computational Physics* **28** (5), 2042–2074 (2020). doi [10.4208/cicp.0A-2020-0193](https://doi.org/10.4208/cicp.0A-2020-0193).
30. D. Liu and Y. Wang, “A Dual-Dimer method for training physics-constrained neural networks with minimax architecture,” *Neural Networks* **136**, 112–125 (2021). doi [10.1016/j.neunet.2020.12.028](https://doi.org/10.1016/j.neunet.2020.12.028).
31. N. Zobeiry and K. D. Humfeld, “A physics-informed machine learning approach for solving heat transfer equation in advanced manufacturing and engineering applications,” *Engineering Applications of Artificial Intelligence* **101**, Article Number 104232 (2021). doi [10.1016/j.engappai.2021.104232](https://doi.org/10.1016/j.engappai.2021.104232).
32. I. A. Chuprov, C. Gao, D. S. Efremenko, et al., “Optimization of Physics-Informed Neural Networks for Non-linear Schrödinger Equation,” *Doklady RAN. Mat. Inf. Pr. Upr.* **514** (2), 28–38 (2023). doi [10.31857/S2686954323601586](https://doi.org/10.31857/S2686954323601586). [in Russian].
33. X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics (AISTATS), Chia Laguna Resort, Sardinia, Italy, May 13–15, 2010* (Proceedings of Machine Learning Research, Vol. 9, 2010), pp. 249–256.
34. S.-F. Wang, S. Sankaran, and P. Perdikaris, “Respecting causality for training physics-informed neural networks,” *Computer Methods in Applied Mechanics and Engineering* **421**, Article Number 116813 (2024). doi [10.1016/j.cma.2024.116813](https://doi.org/10.1016/j.cma.2024.116813).
35. R. Matthey and S. Ghosh, “A novel sequential method to train physics informed neural networks for Allen Cahn and Cahn Hilliard equations,” *Computer Methods in Applied Mechanics and Engineering* **390** (12), Article Number 114474 (2022). doi [10.1016/j.cma.2021.114474](https://doi.org/10.1016/j.cma.2021.114474).
36. A. Vaswani, N. Shazeer, N. Parmar, et al., “Attention is all you need,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, December 4–9, 2017* (Advances in Neural Information Processing Systems. (Curran Associates, Inc., Red Hook, NY, USA, 2017), Vol. 30, pp. 6000–6010.



37. Z. Niu, G. Zhong, and H. Yu, “A review on the attention mechanism of deep learning,” *Neurocomputing* **452**, 48–62 (2021). doi [10.1016/j.neucom.2021.03.091](https://doi.org/10.1016/j.neucom.2021.03.091).
38. N. Rahaman, A. Baratin, D. Arpit, et al., “On the spectral bias of neural networks,” *Proceedings of the 36th International Conference on Machine Learning, Long Beach, California, June 9–15, 2019* Proceedings of Machine Learning Research (PMLR). (Curran Associates, Inc., Red Hook, NY, USA, 2019), Vol. 97, pp. 5301–5310. doi [10.48550/arXiv.1806.08734](https://doi.org/10.48550/arXiv.1806.08734).
39. Z.-Q. J. Xu, Y. Zhang, and Y. Xiao, “Training behavior of deep neural network in frequency domain,” in *Neural Information Processing. 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019* Lecture Notes in Computer Science, Vol. 11953. (Springer, Cham., 2019), pp. 264–274. doi [10.1007/978-3-030-36708-4_22](https://doi.org/10.1007/978-3-030-36708-4_22).
40. Z.-Q. J. Xu, Y. Zhang, T. Luo, et al., “Frequency principle: Fourier analysis sheds light on deep neural networks,” *Communications in Computational Physics* **28** (5), 1746–1767 (2020). doi [10.4208/cicp.0A-2020-0085](https://doi.org/10.4208/cicp.0A-2020-0085).
41. M. Tancik, P. Srinivasan, B. Mildenhall, et al., “Fourier features let networks learn high frequency functions in low dimensional domains,” in *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada, December 6–12, 2020* Advances in Neural Information Processing Systems. (Curran Associates, Inc., Red Hook, NY, USA, 2020), Vol. 33, pp. 7537–7547.
42. A. Jacot, F. Gabriel, and C. Hongler, “Neural tangent kernel: convergence and generalization in neural networks,” in *Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada, December 2–8, 2018* Advances in Neural Information Processing Systems. (Curran Associates, Inc., Red Hook, NY, USA, 2018), Vol. 31, pp. 8580–8589.
43. A. Rahimi and B. Recht, “Random features for large-scale kernel machines,” in *Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3–6, 2007* Advances in Neural Information Processing Systems (NIPS). (Curran Associates, Inc., Red Hook, NY, USA, 2007), Vol. 20, pp. 1177–1184.

Received
July 4, 2026

Accepted
July 29, 2026

Published
September 11, 2026

Information about the author

Mikhail M. Dyshaev — Ph.D., Scientist; COBOTORY LLC, Anoshkina ulitsa, 4, Room 1, 454100, Chelyabinsk, Russia.