

УДК 519.63

MSC2020 65M99, 62M45, 35Q68

© Д. А. Пархоменко¹

Глубокое обучение в уравнениях математической физики

В работе представлены результаты по решению одномерного уравнения Бюргера и двумерного уравнения Пуассона с последующими инференсом модели и визуализацией. Решение было получено с помощью производительных легко-весных нейронных сетей в разработанном гибком проекте. В итоговой архитектуре модульно объединяются нейронная сеть прямого распространения (FFNN) в качестве кодировщика и нейронная сеть Колмогорова – Арнольда (KAN) как декодировщик.

Ключевые слова: *физически информированные нейронные сети, глубокое обучение, дифференциальные уравнения в частных производных.*

DOI: <https://doi.org/10.47910/FEMJ202506>

Введение

Изучение систем большого числа взаимодействующих частиц является фундаментальной и актуальной задачей современной физики в гидродинамике [1–3], механике сплошных сред [3, 4] и других [5]. Однако нахождение состояний таких систем зачастую крайне ресурсозатратно. В математической физике задача состоит в восстановлении описывающего состояния системы поля из соответствующих дифференциальных уравнений в частных производных. Например, это может быть поле температур, поле давлений и скоростей, электростатические и гравитационные поля, и так далее. Ввиду сложности решения многих дифференциальных уравнений в частных производных предлагается аппроксимирование искомого поля, причём с помощью вычислительно простых функций, что потребует значительно меньшего количества компьютерных мощностей.

Подходящими алгоритмами оптимизации являются нейронные сети. В рамках поставленных задач важно умение нейронных сетей автоматически обучать представления данных, промежуточные пространства, порождающие в конечном итоге решения уравнений математической физики. Возможность нахождения решения подкрепляется универсальной теоремой аппроксимации [6].

¹ Дальневосточный федеральный университет, Приморский край, г. Владивосток; Институт наукоёмких технологий и передовых материалов, 690922, остров Русский, п. Аякс, 10. Электронная почта: parkhomenko.dale@dvfu.ru

Особенно интересно глубокое обучение, основанное на физике (от англ. Physics-Informed Neural Networks, PINN) [7,8] — особый тип нейронных сетей, использующих знания о физических процессах, которые управляют набором данных, во время обучения. В случае информирования модели мы больше полагаемся на автоматическое обучение представлений, явно прописывая граничные, начальные и прочие условия.

В рамках данной работы целевыми для моделирования выбраны уравнение Бюргерса и уравнение Пуассона.

1. Постановка задачи

1.1. Уравнение Бюргерса

Рассмотрим частный случай уравнений Навье – Стокса — одномерное уравнение Бюргерса [9, 10]:

$$\begin{aligned} \alpha \frac{\partial u}{\partial t} + \beta u \frac{\partial u}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2} &= f(x, t), \\ u(x, 0) &= \phi(x), \quad a \leq x \leq b, \\ u(a, t) &= \mu_1(t), \quad 0 \leq t \leq \tau, \\ u(b, t) &= \mu_2(t), \quad 0 \leq t \leq \tau. \end{aligned} \tag{1}$$

Первое уравнение системы представляет собой уравнение Бюргерса, определяющее скорость течения жидкости $u(x, t)$ в одномерном пространстве. Обычно в математической физике задача трактуется как распределение скорости течения в тонком стержне во времени и пространстве.

Второе уравнение есть начальное условие для данной системы, а следующие два — постановка краевой задачи первого рода, однако несложно поставить и краевую задачу второго или третьего рода.

Во всех исследуемых системах фиксировались коэффициенты общего дифференциального уравнения, а для граничных условий всегда выбирались нулевые значения:

Таблица 1. Фиксированные параметры исследуемых уравнений Бюргерса

Гиперпараметр	Значение
α	1
β	1
ν	$\frac{0.01}{\pi} \approx 0.00318$
$\mu_1(t)$	0
$\mu_2(t)$	0

Для экспериментов использовались следующие условия и неоднородности к уравнению Бюргерса (1):

Таблица 2. Неоднородности и границы исследуемых уравнений Бюргерса (1), где l рассчитывается как среднее модулей границ $L = \frac{|a|+|b|}{2}$

ID	$f(x, t)$	$\phi(x)$	a	b	τ
burger-1	0	$-\sin(\frac{2\pi x}{L})$	-1	1	4
burger-2	0	$-\sin(\pi x)$	-3	3	10
burger-3	$\cos(\pi t)e^{-0.1t}$	$e^{\frac{x^2}{2}}$	-3	3	10
burger-4	$\cos(\pi t)e^{-0.1t}$	$-\sin(\frac{2\pi x}{L})$	-3	3	10

Эталонной выбрана первая система burger-1, точность решения которой сравнивается с известными результатами [10, 11]:

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} - \frac{0.01}{\pi} \frac{\partial^2 u}{\partial x^2} &= 0, \\ u(x, 0) &= -\sin\left(\frac{2\pi x}{L}\right), \quad -1 \leq x \leq 1, \\ u(-1, t) &= u(1, t) = 0, \quad 0 \leq t \leq 4. \end{aligned} \quad (2)$$

Дополнительно приведём пример системы с неоднородностью и нормальным распределением в начальных условиях:

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} - \frac{0.01}{\pi} \frac{\partial^2 u}{\partial x^2} &= \cos(\pi t)e^{-0.1t}, \\ u(x, 0) &= e^{\frac{x^2}{2}}, \quad -3 \leq x \leq 3, \\ u(a, t) &= u(b, t) = 0, \quad 0 \leq t \leq 10. \end{aligned}$$

1.2. Уравнение Пуассона

Вторым целевым уравнением выбрано уравнение Пуассона [11]

$$\begin{aligned} \Delta \phi(x, y) &= -2\pi^2 \cos(\pi x) \cos(\pi y), \\ x, y &\in \Omega, \\ (x, y)|_{\partial\Omega} &= \mu(t), \end{aligned} \quad (3)$$

для которого граничные условия аналогично выбираются нулевыми $\mu(t)=0$, но форма границы $\partial\Omega$ может изменяться:

Таблица 3. Границы исследуемых уравнений Пуассона

ID	Ω	$\partial\Omega$
poisson-1	квадрат	-
poisson-2	круг	$x^2 + y^2 = 2^2$
poisson-3	эллипс	$\frac{x^2}{3^2} + \frac{y^2}{1^2} = 1$

В работе сопоставляются решения для первой системы poisson-1 без пространственных границ с известными результатами [11]:

$$\begin{aligned}\Delta\phi(x, y) &= -2\pi^2 \cos(\pi x) \cos(\pi y), \\ x, y &\in \Omega.\end{aligned}\tag{4}$$

2. Модель и методы решения

2.1. Физически информированная нейронная сеть

Как и большинство остальных алгоритмов глубокого обучения, PINN являются нелинейным отображением из пространства признаков в пространство целевых переменных; с другой стороны, они могут относиться к задачам обучения без учителя, полностью определяемым дифференциальным уравнением и соответствующим ему условиям. Моделью называем универсальную аппроксимирующую функцию

$$\begin{aligned}\hat{u}_\theta(\mathbf{x}, t) &= f(X) \equiv NN(X), \\ X &= (\mathbf{x}, t),\end{aligned}$$

где в качестве признаков выступает облако точек — неструктурированный набор векторов вида $X = (x, t)$.

Технически архитектуру PINN в общем виде разделяют на две части [7]:

- 1) отображение $f : X \rightarrow \hat{u}$ посредством сложной композиции функций, соответствующих разным слоям;
- 2) использование полученного массива чисел $\hat{u}$ в целевых уравнениях и условиях с последующим расчётом ошибки. Зачастую в равенствах все объекты по типу $u_{xx}, u_x, f(u)$ переносятся в одну сторону, а условия, которые мы задаём на целевое уравнение, в другую сторону. Это позволяет оптимизировать среднеквадратичную ошибку в понятном виде.

В такой постановке задачи функция потерь для обратного распространения представляет собой взвешенную сумму всех ошибок:

$$\begin{aligned}L(X, X_B; \theta) &= \omega_{pde} L_{pde}(X; \theta) + \omega_B L_B(X_B; \theta), \\ \theta^* &= \underset{\theta}{\operatorname{argmin}} L(X, X_B; \theta).\end{aligned}$$

В качестве потерь $L_{pde}(X; \theta)$ выступает математическое ожидание квадратов отклонений между аппроксимацией решения $\hat{u}_\theta(X)$ и внешним условием $g_{pde}(X)$ на множестве точек в заданных интервалах. Для потерь $L_B(X_B; \theta)$ аналогично выполняется оптимизация на квадратах разностей с соответствующими уравнению условиями и множеством точек:

$$\begin{aligned}L_{pde}(X; \theta) &= \mathbb{E}_{f \in pde} [f(X)^2], \\ L_B(X_B; \theta) &= \mathbb{E}_{B \in B} [B(X_B)^2], \\ f : X &\rightarrow \hat{u}_\theta(X) - g_{pde}(X), \\ B : X_B &\rightarrow \hat{u}_\theta(X_B) - g_B(X_B),\end{aligned}$$

где $\mathbb{E}_{B \in \mathcal{B}}$ есть условное математическое ожидание при выборе условия B из множества граничных и начальных условий $\mathcal{B}$, X_B — множество точек на границе или в начальный момент времени, $\hat{u}_\theta(X)$ — прогноз модели с параметрами θ .

Важно, что набор параметров для каждой функции потерь, участвующей в комбинации итоговых потерь, одинаков, потому что основное тело модели не зависит от расчёта производных и условий. Как раз это и необходимо для обучения представления: выучить решение целевого уравнения с учётом всех указанных ограничений и внешних воздействий.

Условие с точки зрения алгоритма является функцией от координат или времени, вычисляющей разность между прогнозом $\hat{u}$ и заданным поведением g согласно положению в пространстве-времени. Приведём для начального условия:

$$B_{ic}(\hat{u}_\theta(\mathbf{x}, 0))^2 = MSE(\hat{u}_\theta(\mathbf{x}, 0), g_B(\mathbf{x}))$$

Архитектуру можно представить схематично (см. рис. 1):

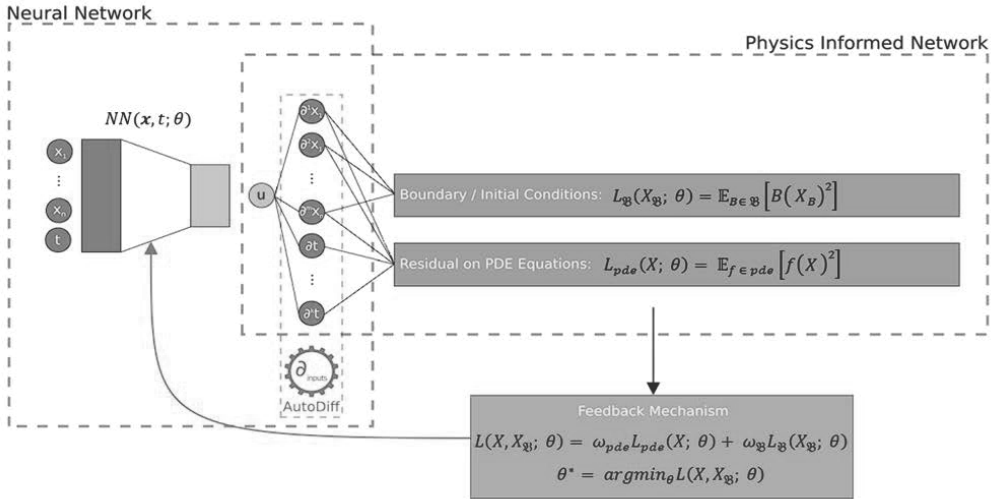


Рис. 1. Шаблон физически информированной нейронной сети. Входные наблюдения отображаются в пространство решения, которое впоследствии используется для расчёта частных производных и построения целевого дифференциального уравнения с данными условиями [7]

Предлагаемые архитектуры PINN должны быть достаточны с точки зрения объёма и универсальности ради получения решения в принципе — без функции потерь, рассчитываемой на аналитическое решение. То есть в условиях генерации случайных векторов, отсутствия реальных данных и отсутствия целевой переменной необходимо получить хотя бы какое-то приближение, описывающее динамику системы большого числа частиц.

2.2. Определение набора данных

Задачи PINN уникальны. Если обычно приходится предварительно обрабатывать данные перед подачей в модели машинного обучения, то для решения дифференциальных уравнений в частных производных важен фундаментальный вопрос генерации самих данных. Эксперименты с уравнением Пуассона (3) проводились в двух различных доменах:

- квадратной области: здесь разреженное пространство формируется из большого числа N точек, заполненного с помощью дискретного равномерного распределения. Причём координаты и время генерируются как независимые и одинаково распределённые случайные величины в некоторых заранее определённых интервалах;
- эллиптической области: то же разреженное пространство формируется из N точек, но сначала в полярных координатах с учётом сохранения равномерности после перехода в декартову систему координат. Координаты теперь зависимы и одинаково распределены, потому что для генерации эллипсоидальной области необходим общий порождающий угол.

С уравнением Бюргерса (1) более просто: точки размещаются с повторением из равномерного распределения от левой до правой границы, то есть укладываются на прямую, координатной или временной.

Дополнительно на каждой эпохе обучения случайно выбирается $k: k \ll N$ точек. Такая выборка с возвращением предостерегает модель от переобучения и повышает её обобщаемость:

$$\begin{aligned} coords &\in \mathbb{R}^{k \times d}, \\ time &\in \mathbb{R}^{k \times 1}, \end{aligned}$$

где d — число координат. Выбор гиперпараметров для построения доменов описан ниже, в пункте 3.1 «Итоговый алгоритм».

2.3. Полносвязная нейронная сеть прямого распространения

Полносвязная нейронная сеть прямого распространения (от англ. Feed-Forward Neural Network, FFNN) представляет собой последовательное применение линейных операций с нелинейными функциями активации к исходным точкам, то есть композицию нелинейных блоков z_i в количестве n штук, отображающую исходное пространство в целевое решение PDE [12]:

$$\begin{aligned} z_i(x; W_i, b_i) &= \sigma(W_i^T x + b_i), \\ FFNN(x; W) &= (z_n(\cdot; W_n, b_n) \circ z_{n-1}(\cdot; W_{n-1}, b_{n-1}) \circ \dots \circ z_1(\cdot; W_1, b_1))(x). \end{aligned}$$

В качестве FFNN выбрана архитектура «кодировщик–декодировщик». В кодировщике каждая отдельная координата и время отображаются в пространства большей размерности $\mathbb{R}^{emb_dim}$, после чего конкатенируются и как единая структура приводятся к этой же размерности emb_dim , являющейся основным гиперпараметром модели. В качестве функции активации выбран гиперболический тангенс. Упро-

щённо кодировщик для одного пакета размером B выглядит так:

$$\begin{aligned} z_x &: \mathbb{R}^{B \times d} \rightarrow \mathbb{R}^{B \times emb_dim} \\ z_t &: \mathbb{R}^{B \times 1} \rightarrow \mathbb{R}^{B \times emb_dim} \\ z_{xt} &: [z_x z_t] \in \mathbb{R}^{B \times (emb_dim \times (d+1))} \rightarrow \mathbb{R}^{B \times emb_dim} \end{aligned}$$

Следом используется декодировщик, состоящий из линейных слоёв, для понижения размерности до единицы, что в итоге соответствует отображению $d+1$ чисел, координат и времени в одно – аппроксимацию решения $\hat{u}$:

$$z_u : \mathbb{R}^{B \times emb_dim} \rightarrow \mathbb{R}^{B \times 1}. \quad (5)$$

Основной гиперпараметр emb_dim , который управляет объёмом модели, для полносвязной сети выбирается от 16 до 128, причём чем сложнее задача, тем больший объём может понадобиться. Для решаемых в работе систем достаточным оказалось выбрать $emb_dim = 32$.

Далее исследовались различные модификации кодировщика модели для уравнений с временной составляющей: наложение простых функций по типу натурального логарифма, синуса, квадратного корня и прочего на время перед подачей в линейный слой; добавление обходящих связей как есть, со взвешиванием, с обучаемыми параметрами, суммой Эйнштейна и так далее. Заключительная версия FFNN содержит обходящую связь как сумму Эйнштейна по второй размерности, пропорциональной emb_dim , между координатами и временем. В случае одномерных задач результат операции соответствует поточечной сумме. Причём общая схема кодировщика, описанная выше, сохраняется и используется как первый слой итоговой модели.

Построенная FFNN модель двумерные задачи решает с довольно большой ошибкой, и особенно заметно несоответствие масштабу, который очевидно должен быть от -1 до 1 :

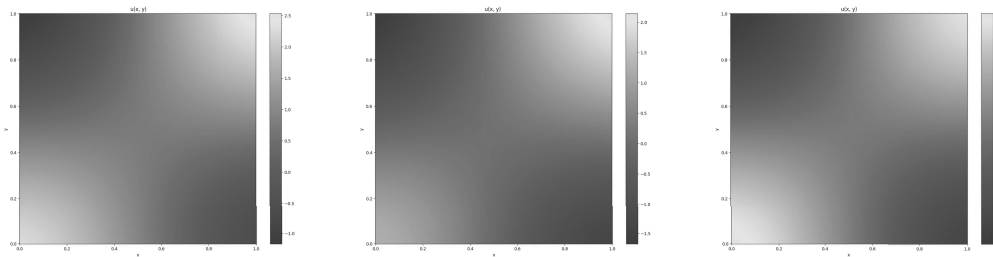


Рис. 2. Решение уравнения Пуассона (4) разноразмерными FFNN с emb_dim 32, 64 и 128

Однако в одномерной задаче, эталонном уравнении Бюргерса (2), нейронная сеть прямого распространения показывает себя с лучшей стороны — качество соответствует результатам при обучении с таргетом [10]:

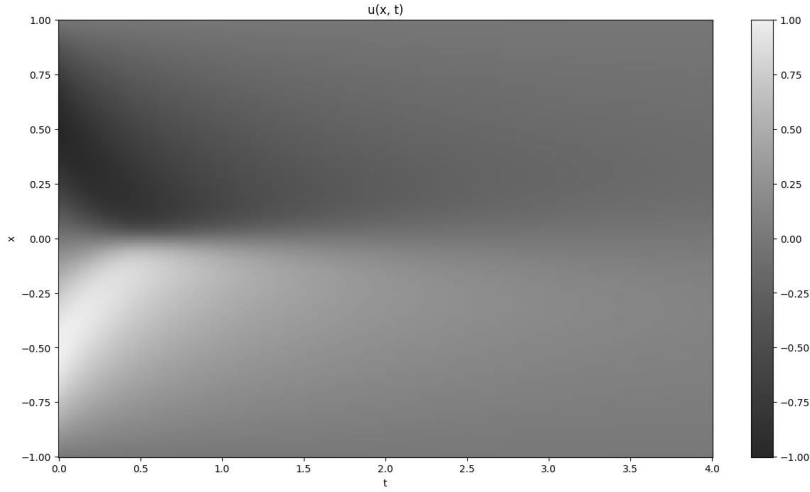


Рис. 3. Решение уравнения Бюргерса (2) на основе FFNN

2.4. Нейронная сеть Колмогорова – Арнольда

Теорема о представлении Колмогорова – Арнольда состоит в том, что непрерывную функцию в некотором интервале можно разложить в конечную композицию одномерных функций с операцией сложения [13]:

$$f(x) = f(x_1, \dots, x_n) = \sum_{q=1}^{2n+1} \Phi_q \left(\sum_{p=1}^n \phi_{q,p}(x_p) \right)$$

По аналогии с полносвязными сетями прямого распространения, KAN (Kolmogorov – Arnold Network) можно трактовать как композицию линейных слоёв с одномерными обучаемыми функциями на рёбрах, заменяющими функции активации.

Одна из главных проблем — проблема неверного масштаба прогноза. Сложно научить FFNN воспринимать пространство изотропным и однородным вне зависимости от расстояния относительно начала системы координат. С другой стороны, в KAN вместо привычных линейных параметров модель содержит параметризованные обучаемые одномерные функции, называемые сплайнами, составляющими аппроксимирующую функцию. Таким образом, оптимизируясь в множество одномерных пространств, мы можем частично обойти проклятие размерности.

Из самой идеи построения нейронной сети Колмогорова – Арнольда и экспериментальных результатов ясно, что модели, содержащей KAN-слой, не нужен большой объём в принципе. Можно сжать модель в 4–5 раз и получить лучшее качество, однако сложность вычислений всё равно намного выше, чем у привычных линейных слоёв.

Модель KAN использовалась не сама по себе, а добавлялась в общую архитектуру как декодировщик с аналогичным понижением размерности (5). Кодировщиком остаётся лучший вариант из пункта 3.3, однако *emb_dim* понижается до 4 или 8.

Пример работы алгоритмы для эталонного двумерного уравнения Пуассона (4):

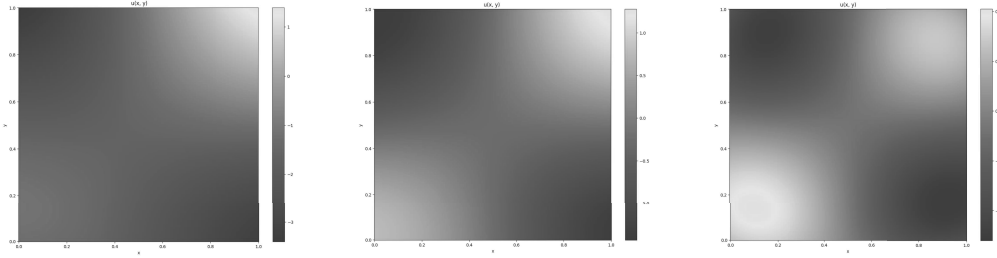


Рис. 4. Решение уравнения Пуассона (4) разноразмерной KAN emb_dim 4, 8 и 16

Видно, что теперь масштаб более корректный, но чрезмерное увеличение объёма модели на базе KAN, наоборот, ухудшает аппроксимацию относительно простых систем.

3. Результаты численных экспериментов

Далее представлены схема алгоритма, архитектура нейронной сети и обученные под целевые уравнения модели. Положительным результатом считается получение качественной интерпретируемой и обоснованной с точки зрения задаваемой системы уравнений двумерной поверхности, поля, отложенного либо в пространстве ХУ для уравнения Пуассона, либо в пространстве ХТ для уравнения Бюргерса.

Суммарные потери складываются из ошибок на всех условиях, а также зависят от размера пакета B и масштаба. Дополнительно потери различаются в зависимости от системы, в том числе из-за отсутствия аналитического решения, играющего важную роль в оптимизации. Поэтому суммарные потери при обучении или инференсе не приводятся из-за несодержательности.

3.1. Итоговый алгоритм

Сначала была выстроена базовая архитектура сети в виде последовательного применения полносвязных линейных слоёв с различными функциями активации на основе наблюдений вида $inputs = (x, t)$, где наблюдения генерируются для одномерных задач равномерно в заданном интервале, а для двумерных — в квадратной либо эллиптической области.

Соответствующие граничным и начальным условиям векторы подставляются на лету с повторным проходом по модели. Максимум за один шаг выполняется прогноз трёх составляющих систему частей: целевого уравнения, начальных и граничных условий. По умолчанию на одну эпоху обучения выделяется $k=50000$ точек, которые случайно выбираются из $N=1000000$ точек, равномерно распределённых в заданной области. При обратном распространении градиенты сепарируются по числу условий, и здесь важно правильно их взвешивать.

Algorithm 1: Обзор ключевых этапов алгоритма**Input:**

Исследуемая область пространства-времени: форма, масштаб;

Случайные векторы вида $(\mathbf{x}, t)$ в заданной области.

Require:

Построение основного, параметрического тела модели $\hat{u}_\theta(\mathbf{x}, t)$ с параметрами θ ;

Выбор целевых граничных и/или начальных условий;

Выбор неоднородности и функций потерь.

Ensure: $epoch \leq 15$

Формирование пакета (batch) точек с привязкой к графу;

Прогноз $f(\mathbf{x}, t) \equiv \hat{u}(\mathbf{x}, t)$ и расчёт ошибки $L_{pde}(\mathbf{x}, t)$;

Прогноз $B_{ic}(\hat{u}(\mathbf{x}, 0))^2$ и расчёт ошибки $L_{ic}(\mathbf{x}, 0)$;

Прогноз $B_{bc}(\hat{u}(\mathbf{x}_{bc}, t))^2$ и расчёт ошибки $L_{bc}(\mathbf{x}_{bc}, t)$;

Агрегация потерь и шаг обратного распространения;

Логирование потерь на каждое условие, сохранение параметров.

Output:

Прогноз решения PDE $\hat{u}(\mathbf{x}, t)$.

Для временной составляющей выделен гиперпараметр *extreme_time*, принимающий булевы значения. Если указать его в *True*, сгенерированные точки по времени разделяются в соотношении 9:1, где меньшая часть растягивается от правой временной границы до десятикратного её значения. Например, для интервала $0 \leq t \leq 10$ время будет распределено на 90% в нём, а на 10% в интервале $10 < t \leq 100$. Такое расширение по времени повышает обобщаемость модели в будущем.

После экспериментов в главе 2 «Модель и методы решения» итоговой архитектурой выбрана нейронная сеть, состоящая из FFNN-кодировщика и KAN-декодировщика с основным гиперпараметром $emb_dim = 8$:

Таблица 4. Основные гиперпараметры итоговой модели

Гиперпараметр	Значение	Описание
<i>emb_dim</i>	8	Управление объёмом
<i>base_func</i>	tanh	Базовая нелинейная функция

Далее приведены результаты работы итогового алгоритма на нестандартных системах.

3.2. Решение уравнения Бюргерса

Полносвязная сеть прямого распространения отлично справлялась с эталонной задачей burger-1 в интервале от -1 до 1 , но намного хуже в кратно раз большем интервале, например, от -3 до 3 , тяжело усваивая периодические начальные условия.

Итоговая модель на основе KAN лишена этого недостатка:

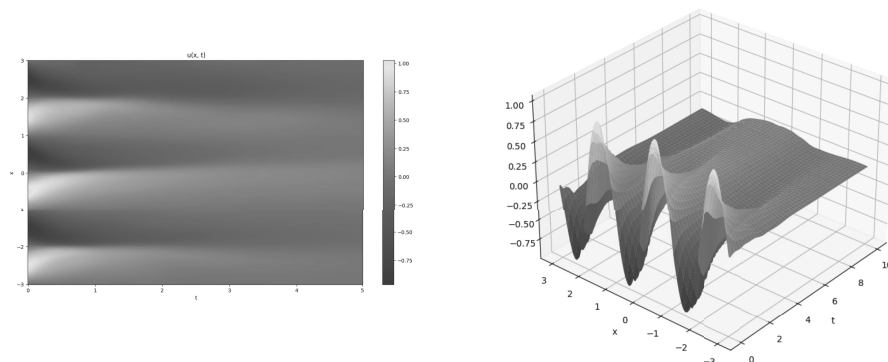


Рис. 5. Решение однородного вязкого burger-2 в интервале от -3 до 3 и временем до 10 экстремальными значениями до 100

Далее исследовались системы с неоднородностью $\cos(\pi t)e^{-0.1t}$ и различными начальными условиями:

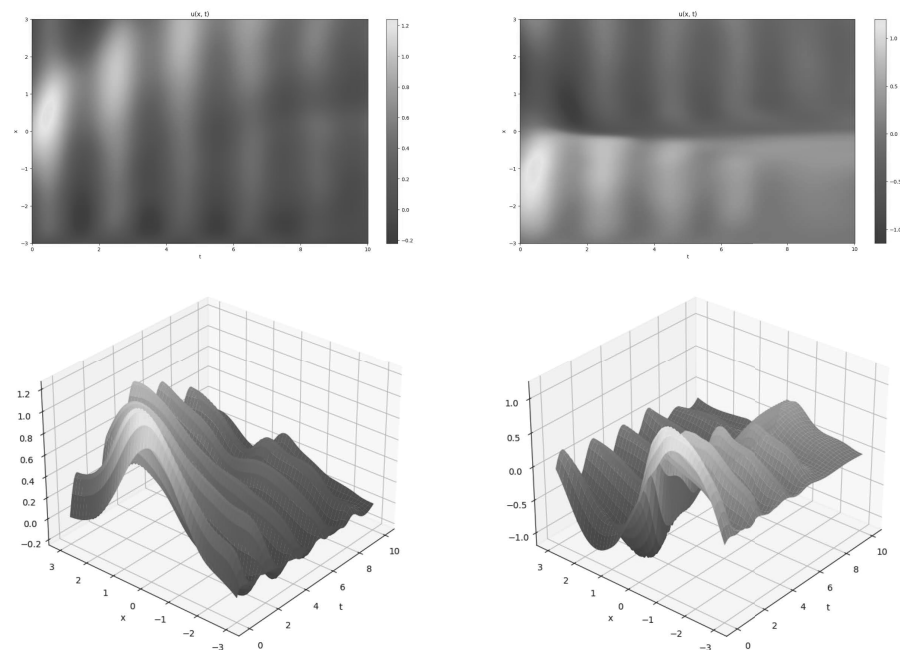


Рис. 6. Решение неоднородного вязкого уравнения Бюргерса с нормальным (слева, burger-3) и синусоидальным (справа, burger-4) распределением в начальном условии

3.3. Решение уравнения Пуассона

Аппроксимировано решение для неоднородного двумерного уравнения Пуассона с нулевым граничным условием на эллипсе, где точки генерируются внутри данной эллиптической области:

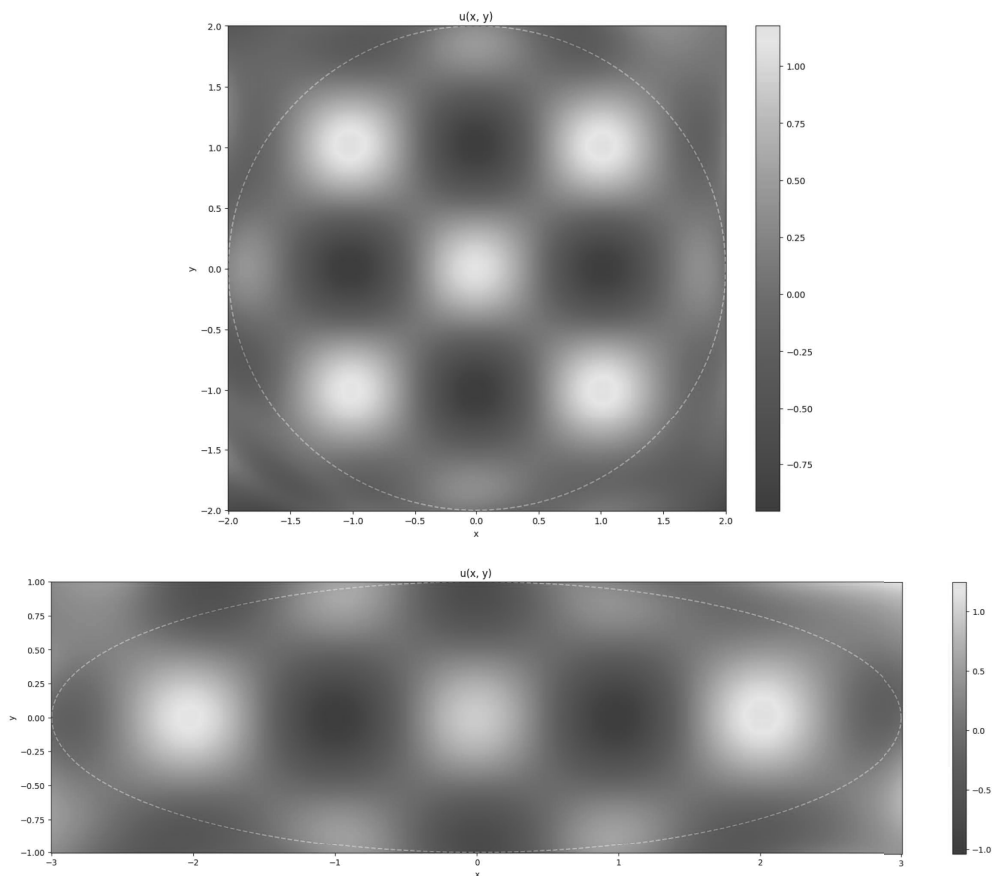


Рис. 7. Решение неоднородного уравнения Пуассона (3) в системах poisson-2 и poisson-3

Заключение

В работе представлены результаты аппроксимации решений дифференциальных уравнений в частных производных с помощью нейронной сети на основе KAN. В отличие от стандартных многослойных перцептронов (MLP), нейронные сети Колмогорова–Арнольда интерпретируемы, всегда находятся в активном состоянии, не страдая от забывания в результате переобучения или дообучения, хорошо выявляют физические закономерности и масштабируются практически на произвольный

размер пространства-времени.

Не менее важно как отсутствие аналитического решения таргета для обучения сети, так и отсутствие реальных данных. Разработанный проект позволяет получать достаточно качественные решения дифференциальных уравнений в частных производных за разумное время с помощью только явно запрограммированных условий и уравнений.

Список литературы

- [1] Ngan et V. N. al., “A reduced-order model for segregated fluid-structure interaction solvers based on an ALE approach”, 2023, arXiv: 2305.13613.
- [2] Lu J., Peters E., Kuipers J. A. M., “Direct numerical simulation of coupled heat and mass transfer in fluid-solid systems”, *Proceedings of the 12th International Conference on Computational Fluid Dynamics in the Oil & Gas, Metallurgical and Process Industries*, SINTEF Academic Press, 2017.
- [3] Faroughi et S. A. al., “Physics-guided, physics-informed, and physics-encoded neural networks in scientific computing”, 2022, arXiv: 2211.07377.
- [4] Jin H., Zhang E., Espinosa H. D., “Recent advances and applications of machine learning in experimental solid mechanics: A review”, *Applied Mechanics Reviews*, 2023.
- [5] Hu et X. al., “Physics-guided deep neural networks for power flow analysis”, *IEEE Transactions on Power Systems*, 2020.
- [6] Hornik K., Stinchcombe M., White H., “Multilayer feedforward networks are universal approximators”, *Neural networks*, 1989.
- [7] Cuomo et S. al., “Scientific machine learning through physics-informed neural networks: Where we are and what’s next”, *Journal of Scientific Computing*, 2022.
- [8] Guasti Junior W., Santos I. P., “Solving differential equations using feedforward neural networks”, *Computational Science and Its Applications – ICCSA 2021: 21st International Conference, Cagliari, Italy, September 13–16, 2021, Proceedings, Part IV 21*, Springer, 2021.
- [9] Bonkile et M. P. al., “A systematic literature review of Burgers’ equation with recent advances”, *Pramana*, 2018.
- [10] Li Y., Mei F., “Deep learning-based method coupled with small sample learning for solving partial differential equations”, *Multimedia Tools and Applications*, 2021.
- [11] Wang Y., Zhong L., “NAS-PINN: neural architecture search-guided physics-informed neural network for solving PDEs”, *Journal of Computational Physics*, 2024.
- [12] Goodfellow I., *Deep learning*, MIT press, 2016.
- [13] Liu et Z. al., “Kan: Kolmogorov-arnold networks”, 2024, arXiv: 2404.19756.

Поступила в редакцию
11 октября 2024 г.

*Parkhomenko D. A.*¹ Inequalities for modulus of rational functions. *Far Eastern Mathematical Journal*. 2025. V. 25. No 1. P. 67–80.

¹Institute of High Technologies and Advanced Materials, Far Eastern Federal University, Vladivostok, Russia

ABSTRACT

The paper presents results on the solution of the one-dimensional Burgers equation and the two-dimensional Poisson equation followed by model inference and visualization. The solution has been obtained using performance lightweight neural networks in the developed agile project. The final architecture modularly combines a feed-forward neural network (FFNN) as an encoder and a Kolmogorov–Arnold neural network (KAN) as a decoder.

Key words: *Physics-Informed Neural Networks, Deep Learning, Partial Differential Equations.*