

УДК 519.63

Реализация PINN для вихря Тейлора–Грина

Подгорнов А.А.¹, Каледина Е.А.¹

Национальный исследовательский Мордовский государственный университет¹

Аннотация: Реализована физико-информированная нейронная сеть для численного решения уравнений Навье–Стокса несжимаемой жидкости в двумерной постановке. На примере классического вихря Тейлора–Грина проведена оценка точности и устойчивости метода при различных уровнях зашумления входных данных, а также выполнено сравнение с методом конечных разностей.

Ключевые слова: физико-информированная нейронная сеть, уравнения Навье–Стокса, вихрь Тейлора–Грина.

1. Постановка задачи

Физико-информированные нейронные сети (Physics-Informed Neural Networks, PINNs) представляют собой бессеточный метод аппроксимации решений дифференциальных уравнений в частных производных. Искомая функция моделируется нейронной сетью, а функция потерь помимо невязки начальных и граничных условий, включает невязку самого уравнения [1].

Исследуется реализация PINN для задачи двумерного распада вихря Тейлора–Грина [2]. Рассматривается течение несжимаемой жидкости в квадратной области с периодическими граничными условиями. Начальное поле скорости задаётся аналитическими выражениями $u(x, y, 0) = \sin x \cos y$, $v(x, y, 0) = -\cos x \sin y$. Благодаря линейности уравнений в данном случае существует точное решение. Его наличие делает задачу удобным эталоном для оценки точности и сходимости вычислительных алгоритмов.

Входными переменными PINN служат пространственные координаты $x = (x, y, z)$ и время t ; выходом – предсказанные поля компонент вектора скорости $\hat{\vec{u}} = (\hat{u}, \hat{v}, \hat{w})$ и давления $\hat{p}$. Сеть реализует отображение $[\hat{\vec{u}}(\vec{x}, t; \vec{\theta}), \hat{p}(\vec{x}, t; \vec{\theta})] = \mathcal{N}_{\vec{\theta}}(\vec{x}, t)$, где $\vec{\theta}$ – полный набор обучаемых параметров. Архитектура сети $\mathcal{N}_{\vec{\theta}}$ – многослойный персептрон, содержащий 4 скрытых слоя по 256 нейронов в каждом с функцией активации $\sigma(z) = \tanh(z)$.

В оптимизируемый функционал явно включаются невязки уравнений Навье–Стокса, а также ошибки выполнения начальных и граничных условий. Для этого на множестве коллокационных точек, случайно распределённых по области $\Omega \times [0, T]$, с помощью автоматического дифференцирования вычисляются все частные производные сетевого предсказания по входам x, y, z, t [3]. Подстановка нейросетевых аппроксимаций $\hat{\vec{u}}, \hat{p}$ даёт векторную невязку уравнений моментов

$$\vec{R}_{NS}(\vec{x}, t; \vec{\theta}) = \frac{\partial \hat{\vec{u}}}{\partial t} + (\hat{\vec{u}} \cdot \nabla) \hat{\vec{u}} + \nabla \hat{p} - \nu \Delta \hat{\vec{u}}$$

и скалярную невязку уравнения неразрывности $R_{div}(\vec{x}, t; \vec{\theta}) = \nabla \cdot \hat{\vec{u}}$. Функция потерь $L(\vec{\theta})$, минимизируемая в процессе обучения, представляет собой взвешенную сумму

нескольких слагаемых [4]

$$L(\vec{\theta}) = \lambda_{PDE} L_{PDE}(\vec{\theta}) + \lambda_{IC} L_{IC}(\vec{\theta}) + \lambda_{BC} L_{BC}(\vec{\theta}) + \lambda_{data} L_{data}(\vec{\theta}).$$

Здесь $L_{PDE}(\vec{\theta})$ – невязка уравнений Навье–Стокса, $L_{IC}(\vec{\theta})$ – невязка начальных условий, $L_{BC}(\vec{\theta})$ – невязка граничных условий, $L_{data}(\vec{\theta})$ – невязка аппроксимации аналитического решения, вычисляемые соответственно по формулам

$$\begin{aligned} L_{PDE} &= \frac{1}{N_f} \sum_{i=1}^{N_f} (\|\vec{R}_{NS}(\vec{x}_f^i, t_f^i; \vec{\theta})\|^2 + |R_{div}(\vec{x}_f^i, t_f^i; \vec{\theta})|^2), \\ L_{IC}(\vec{\theta}) &= \frac{1}{N_{IC}} \sum_{i=1}^{N_{IC}} \|\hat{u}(\vec{x}_{IC}^i, 0; \vec{\theta}) - \vec{u}_0(\vec{x}_{IC}^i)\|^2, \\ L_{BC}(\vec{\theta}) &= \frac{1}{N_{BC}} \sum_{i=1}^{N_{BC}} \|\hat{u}(\vec{x}_{bc,1}^i, t_{bc}^i; \vec{\theta}) - \hat{u}(\vec{x}_{bc,2}^i, t_{bc}^i; \vec{\theta})\|^2, \\ L_{data}(\vec{\theta}) &= \frac{1}{N_d} \sum_{i=1}^{N_d} \|\hat{u}(\vec{x}_d^i, t_d^i; \vec{\theta}) - \vec{u}_{ref}^i\|^2. \end{aligned}$$

Для минимизации функционала $L(\vec{\theta})$ на начальном этапе применяется оптимизатор Adam, а затем для тонкой настройки используется квазиньютоновский метод L-BFGS.

2. Результаты вычислительных экспериментов

Для проведения вычислительных экспериментов использовалась библиотека PyTorch с поддержкой CUDA. Использованы следующие параметры сети: количество точек коллокации $N_f = 1000$, количество точек начальных условий $N_{ic} = 500$, количество пар точек для периодических ГУ $N_{bc} = 200$.

На рис. 1–2 представлены графики аналитического решения и прогноза PINN, поля скорости и давления, графики ошибок.

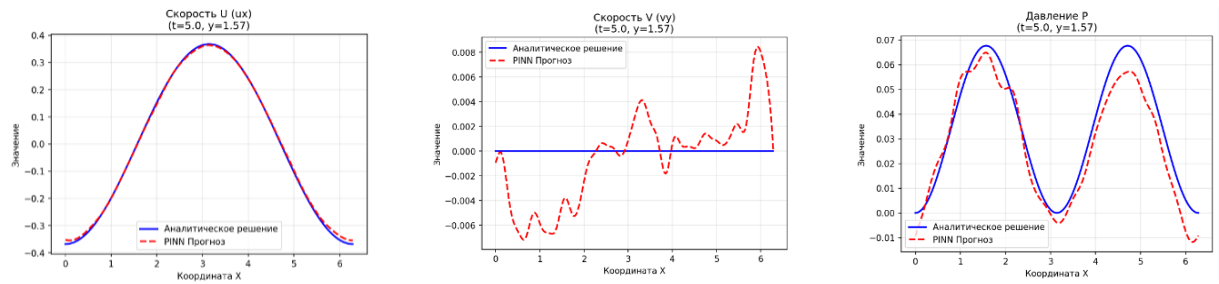


Рис. 1. Сравнение значений скорости параметров u, v, p аналитического решения и прогноза PINN

3. Устойчивость к зашумлённым данным

Чтобы оценить устойчивость PINN к ошибкам во входных данных, в начальные поля скорости добавлялся гауссовский шум с нулевым средним и возрастающим стандартным отклонением. Для сравнения использовался классический метод конечных разностей (МКР) на сетке 50×50 . В качестве метрики точности вычислялась среднеквадратичная ошибка (RMSE) между полученным решением и аналитическим. В

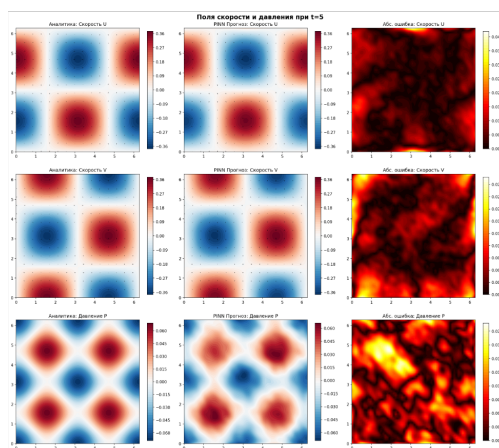


Рис. 2. Сравнение полей скорости, давления и ошибок аналитического решения и PINN

таблице 1 представлено сравнение двух подходов. Относительный рост ошибок для PINN и метода конечных разностей рассчитан следующим образом

$$\text{Относительный рост ошибок} = (RMSE_{\text{с шумом}}) / (RMSE_{\text{без шума}}).$$

Таблица 1. Сравнение RMSE для PINN и метода конечных разностей

Уровень шума	Абс. значение RMSE PINN			Отн. рост RMSE PINN			Абс. значение RMSE МКР			Отн. рост RMSE МКР		
	u	v	p	u	v	p	u	v	p	u	v	p
0.00	0.0074	0.0078	0.0078	—	—	—	0.0002	0.0002	0.0003	—	—	—
0.05	0.0189	0.0210	0.0117	2.6	2.7	1.5	0.0026	0.0024	0.0008	13.0	12.0	2.7
0.10	0.0313	0.0331	0.0159	4.2	4.2	2.0	0.0051	0.0049	0.0014	25.5	24.5	4.7
0.20	0.0468	0.0493	0.0205	6.3	6.3	2.6	0.0102	0.0097	0.0028	51.0	48.5	9.3

В отсутствие шума PINN и метод конечных разностей показывают сопоставимую точность. Однако из таблицы 1 видно, что при увеличении шума ошибка метода конечных разностей растёт гораздо быстрее, чем ошибка PINN. Таким образом, физические законы (уравнения Навье–Стокса), заложенные в обучение сети, помогают сглаживать случайные помехи, не давая им катастрофически нарастать, в отличие от классического численного метода.

Литература

1. Raissi M., Perdikaris P., Karniadakis G.E. Physics informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations // Journal of Computational Physics. 2019. Vol. 378. P. 686-707. DOI: 10.1016/j.jcp.2018.10.045.
2. Taylor G.I., Green A.E. Mechanism of the production of small eddies from large ones // Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences. 1937. Vol. 158, No. 895. P. 499-521. DOI: 10.1098/rspa.1937.0036.

3. Бетелин В.Б., Галкин В.А. О построении искусственной нейронной сети для решения системы уравнений Навье-Стокса в случае несжимаемой жидкости // Доклады Российской академии наук. Математика, информатика, процессы управления. 2024. Т. 517. С. 115-119. DOI: 10.31857/S2686954324700412.
4. Xiaowei Jin, Shengze Cai, Hui Li, Karniadakis G.E. NSFnets (Navier-Stokes flow nets): Physics informed neural networks for the incompressible Navier-Stokes equations // Journal of Computational Physics. 2021. Vol. 426. P. 109951. DOI: 10.1016/j.jcp.2020.109951.

MSC 65M99

PINN implementation for the Taylor–Green vortex

A.A. Podgornov¹, E.A. Kaledina¹

National Research Mordovia State University¹

Abstract: A physically-informed neural network for the numerical solution of the Navier–Stokes equations for an incompressible fluid in a two-dimensional setting has been implemented. Using the classical Taylor–Green vortex as an example, the accuracy and stability of the method are assessed for various levels of input data noise, and a comparison with the finite difference method is made.

Keywords: Physics-informed neural network, Navier–Stokes equations, Taylor–Green vortex.