

Использование методов машинного обучения для решения дифференциальных уравнений.

Аминев Дамир Айратович

Специальность: "Искусственный интеллект и машинное обучение" Научный руководитель: к.ф.-м.н. Хватов Александр Александрович

## План семинара



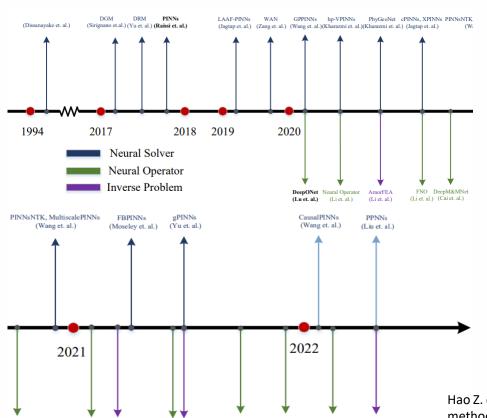
- Мотивация. Зачем нужен Physics Informed Machine Learning?
- Основная концепция physics informed neural network (PINN)
- Архитектура солвера TEDEOuS
- Актуальные направления исследований
- Предложенные нами решения
- Примеры
- Выводы

## Введение

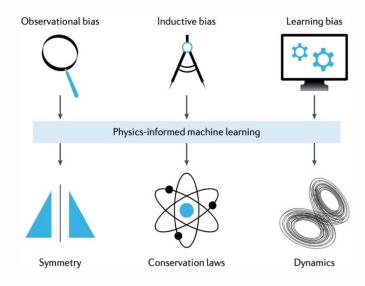
pM&MNet

ai et. al.)





DeepGreen hPINNs (PI-)DeepONet Galerkin Transformer MWT LOCA Bi-level PINN (Gin et. al.) (Lu et. al.) (Wang et. al., Hwang et. al.) (Cao et. al.) (Gupta et. al.) (Kissas et. al.) (Hao et. al.)



Karniadakis, George Em, et al. "Physics-informed machine learning." Nature Reviews Physics 3.6 (2021): 422-440.

Hao Z. et al. Physics-informed machine learning: A survey on problems, methods and applications //arXiv preprint arXiv:2211.08064. – 2022.

## Мотивация



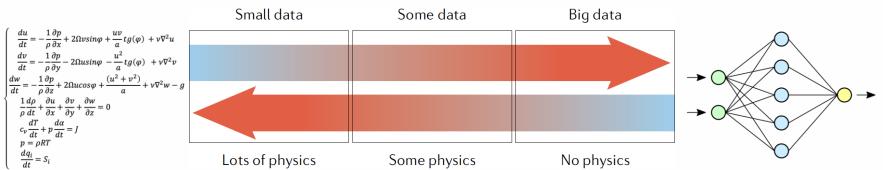
#### Дифференциальные уравнения

- **√** Точность
- ✓ Нет необходимости в данных
- ✓ Интерпретируемость
- Высокий порог вхождения
- Ресурсозатратность



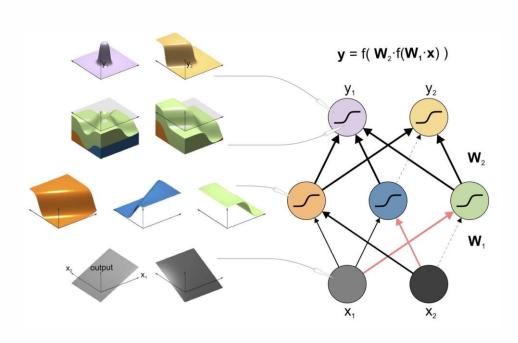
#### Нейронные сети

- ✓ Универсальный аппроксиматор
- Скорость получения решения
- ✓ Дифференцируемость
- Долго обучается
- Необходимость данных



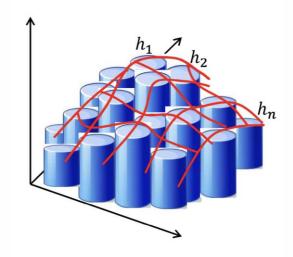
# **Универсальная теорема аппроксимации**





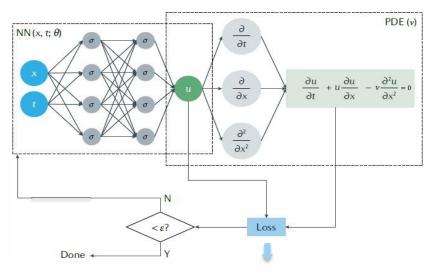
\*G. Cybenko. Approximation by superpositions of a sigmoidal function. Mathematics of Control, Signals and Systems, 2(4):303–314, 1989.

**Неформальная трактовка\***: искусственная нейронная сеть с одним скрытым слоем **может** аппроксимировать любую непрерывную функцию многих переменных с любой точностью.



## Основная идея PINN





$$\begin{split} \mathcal{L} &= w_{\text{data}} \mathcal{L}_{\text{data}} + w_{\text{PDE}} \mathcal{L}_{\text{PDE}}, \\ \text{where} \\ \mathcal{L}_{\text{data}} &= \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} (u(x_i, t_i) - u_i)^2 \quad \text{and} \\ \mathcal{L}_{\text{PDE}} &= \frac{1}{N_{\text{PDE}}} \sum_{j=1}^{N_{\text{PDE}}} \left( \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} - v \frac{\partial^2 u}{\partial x^2} \right)^2 |_{(x_j, t_j)}. \end{split}$$

• Система ОДУ или ДУЧП состоит из p независимых переменных  $x=(x^1,...,x^p)$  и q зависимых  $\vec{u}=(u^1,...,u^q)$ , определенная в области  $x\in\Omega\subset R^p$ :

$$S(\vec{u}) = \vec{f}$$
,  $b(\vec{u}) = \vec{g}$ 

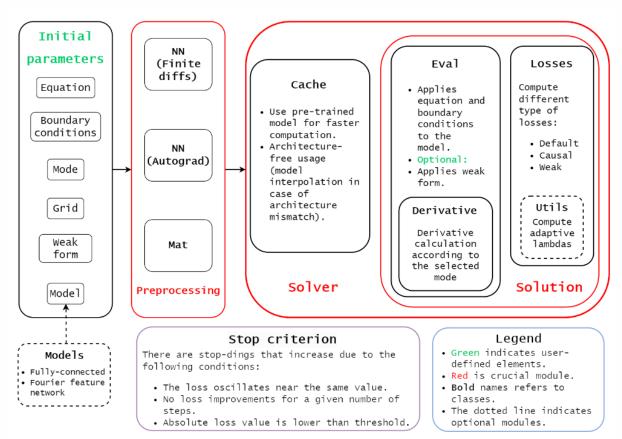
- Произвольная модель машинного обучения  $\overrightarrow{u_*}(x;\theta)$  имеет набор параметров  $\theta$  . Мы определяем ее нейронной сетью, как отображение  $\overrightarrow{u_*}(x;\theta)$ :  $R^p \to R^q$ .
- Численная задача минимизации в классической форме определяется как:

$$\underset{\theta}{argmin} \int_{\Omega} \mathbf{\lambda}_{1} \parallel S(\vec{u}_{*}(x;\theta)) - f \parallel_{i} d\Omega + \mathbf{\lambda}_{2} \parallel b(\vec{u}_{*}(x;\theta)) - g \parallel_{j}$$

нормы  $\|\cdot\|_i$  и  $\|\cdot\|_j$  могут быть произвольными (обычно  $i=l_2$  и  $j=l_1$ ).

## **Архитектура TEDEOuS**









Neural Solver	Method	Description
	Loss Reweighting	Grad Norm NTK Reweighting Variance Reweighting
	Novel Optimization Targets	Numerical Differentiation Variantional Formulation Regularization
	Novel Architectures	Adaptive Activation Feature Preprocessing Boundary Encoding Convolutional Architecture Sequential Architecture Domain Decomposition
	Other Learning Paradigms	Transfer Learning Meta-Learning

Обзор направлений исследований по теме автоматизированных солверов

## Нейронные касательные ядра

**iTMO** 

Neural Tangent Kernel (NTK)

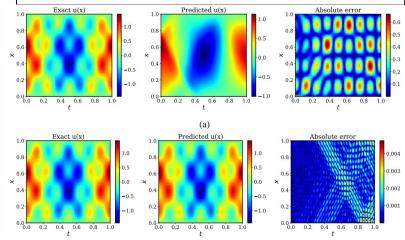
#### Волновое уравнение:

$$u_{tt}(x,t) - 4u_{xx}(x,t) = 0, \quad (x,t) \in (0,1) \times (0,1)$$

$$u(0,t) = u(1,t) = 0, \quad t \in [0,1]$$

$$u(x,0) = \sin(\pi x) + \frac{1}{2}\sin(4\pi x), \quad x \in [0,1]$$

$$u_t(x,0) = 0, \quad x \in [0,1].$$



Wang S., Yu X., Perdikaris P. When and why PINNs fail to train: A neural tangent kernel perspective //Journal of Computational Physics. – 2022.

#### Идея NTK:

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_b(\boldsymbol{\theta}) + \mathcal{L}_r(\boldsymbol{\theta})$$
  $\mathcal{L}_b(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{N_b} |u(\boldsymbol{x}_b^i, \boldsymbol{\theta}) - g(\boldsymbol{x}_b^i)|^2$   $\frac{d\boldsymbol{\theta}}{dt} = -\nabla \mathcal{L}(\boldsymbol{\theta})$   $\mathcal{L}_r(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=1}^{N_r} |r(\boldsymbol{x}_r^i, \boldsymbol{\theta})|^2.$ 

$$\begin{bmatrix} \frac{du(\boldsymbol{x}_b,\boldsymbol{\theta}(t))}{dt} \\ \frac{d\mathcal{L}u(\boldsymbol{x}_r,\boldsymbol{\theta}(t))}{dt} \end{bmatrix} = -\begin{bmatrix} \boldsymbol{K}_{uu}(t) & \boldsymbol{K}_{ur}(t) \\ \boldsymbol{K}_{ru}(t) & \boldsymbol{K}_{rr}(t) \end{bmatrix} \cdot \begin{bmatrix} u(\boldsymbol{x}_b,\boldsymbol{\theta}(t)) - g(\boldsymbol{x}_b) \\ \mathcal{L}u(\boldsymbol{x}_r,\boldsymbol{\theta}(t)) - f(\boldsymbol{x}_r) \end{bmatrix}$$

$$\begin{aligned} & (\boldsymbol{K}_{uu})_{ij}(t) = \left\langle \frac{du(\boldsymbol{x}_b^i, \boldsymbol{\theta}(t))}{d\boldsymbol{\theta}}, \frac{du(\boldsymbol{x}_b^j, \boldsymbol{\theta}(t))}{d\boldsymbol{\theta}} \right\rangle \\ & (\boldsymbol{K}_{ur})_{ij}(t) = \left\langle \frac{du(\boldsymbol{x}_b^i, \boldsymbol{\theta}(t))}{d\boldsymbol{\theta}}, \frac{d\mathcal{L}u(\boldsymbol{x}_r^j, \boldsymbol{\theta}(t))}{d\boldsymbol{\theta}} \right\rangle \\ & (\boldsymbol{K}_{rr})_{ij}(t) = \left\langle \frac{d\mathcal{L}(\boldsymbol{x}_r^i, \boldsymbol{\theta}(t))}{d\boldsymbol{\theta}}, \frac{d\mathcal{L}(\boldsymbol{x}_r^j, \boldsymbol{\theta}(t))}{d\boldsymbol{\theta}} \right\rangle \end{aligned}$$

$$\lambda_b = \frac{\sum_{i=1}^{N_r + N_b} \lambda_i(n)}{\sum_{i=1}^{N_b} \lambda_i^{uu}(n)} = \frac{Tr(\boldsymbol{K}(n))}{Tr(\boldsymbol{K}_{uu}(n))}$$
$$\lambda_r = \frac{\sum_{i=1}^{N_r + N_b} \lambda_i(n)}{\sum_{i=1}^{N_r} \lambda_i^{rr}(n)} = \frac{Tr(\boldsymbol{K}(n))}{Tr(\boldsymbol{K}_{rr}(n))}$$

## Анализ чувствительности

# **iTMO**

#### на основе индексов Соболя

$$Y = f_0 + \sum_{i=1}^d f_i(X_i) + \sum_{i < j}^d f_{ij}(X_i, X_j) + \dots + f_{1, 2, \dots, d}(X_1, X_2, \dots, X_d)$$

$$\int_0^1 f_{i_1 i_2 \ldots i_s}(X_{i_1}, X_{i_2}, \ldots, X_{i_s}) dX_k = 0, ext{ for } k = i_1, \ldots, i_s$$

$$egin{aligned} f_0 &= E(Y) \ f_i(X_i) &= E(Y|X_i) - f_0 \ f_{ij}(X_i,X_j) &= E(Y|X_i,X_j) - f_0 - f_i - f_j \end{aligned}$$

$$\int f^2({f X}) d{f X} - f_0^2 = \sum_{s=1}^d \sum_{i_1 < \cdots < i_s}^d \int f_{i_1 \ldots i_s}^2 dX_{i_1} \ldots dX_{i_s}$$

$$\mathrm{Var}(Y) = \sum_{i=1}^d V_i + \sum_{i < j}^d V_{ij} + \cdots + V_{12\ldots d}$$

$$\left|S_i = rac{V_i}{\mathrm{Var}(Y)}\;
ight|\, S_{T_i} = ext{1} - rac{\mathrm{Var}_{\mathbf{X}_{\sim i}}(E_{X_i}(Y \mid \mathbf{X}_{\sim i}))}{\mathrm{Var}(Y)}
ight|$$

$$\begin{aligned} \mathbf{L} &= \mathbf{L}_{\text{pde1}} + \mathbf{L}_{\text{pde2}} + \ldots + \mathbf{L}_{\text{bc1}} + \mathbf{L}_{\text{bc2}} + \ldots \\ \mathbf{S}_{\text{t}} &= \mathbf{S}_{\text{t\_pde1}} + \mathbf{S}_{\text{t\_pde2}} + \ldots + \mathbf{S}_{\text{t\_bc1}} + \mathbf{S}_{\text{t\_bc2}} + \ldots \\ \lambda_{pde1} &= \frac{S_t}{S_{t\_pde1}}, \lambda_{bc1} = \frac{S_t}{S_{t\_bc1}}, \ldots \\ \mathbf{Loss} &= \sum_{i} \lambda_i \mathbf{L}_i \end{aligned}$$

## Анализ чувствительности

#### на основе индексов Соболя



#### Волновое уравнение:

$$u_{tt}(x,t) - 4u_{xx}(x,t) = 0, \quad (x,t) \in (0,1) \times (0,1)$$

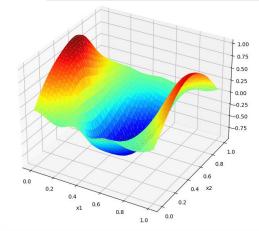
$$u(0,t) = u(1,t) = 0, \quad t \in [0,1]$$

$$u(x,0) = \sin(\pi x) + \frac{1}{2}\sin(4\pi x), \quad x \in [0,1]$$

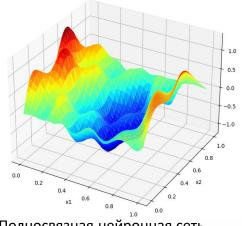
$$u_t(x,0) = 0, \quad x \in [0,1].$$

#### Уравнение Пуассона:

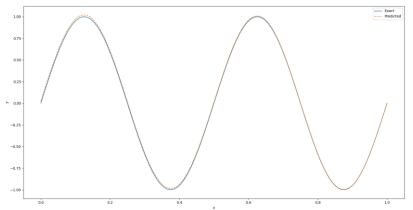
$$u_{xx} = g(x), \quad x \in [0, 1]$$
  
 $u(0) = u(1) = 0.$   
 $g(x) = -(4\pi)^2 \sin(4\pi x)$ 



Полносвязная нейронная сеть



Полносвязная нейронная сеть + адаптивные весовые коэф-ты



Решение уравнения Пуассона

## Модифицированный loss (causal)



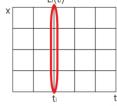


 $10^{-1}$ 

 $10^{-5}$ 

0.00

0.25



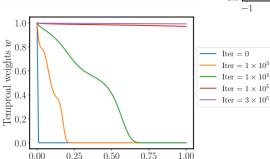
$$w_i = \exp\left(-\epsilon \sum_{k=1}^{i-1} \mathcal{L}_r(t_k, \boldsymbol{\theta})\right), \text{ for } i = 2, 3, \dots, N_t$$

$$\mathcal{L}_r(oldsymbol{ heta}) = \sum_{i=1}^{N_t} \exp\left(-\epsilon \sum_{k=1}^{i-1} \mathcal{L}_r(t_k, oldsymbol{ heta})
ight) \mathcal{L}_r(t_i, oldsymbol{ heta})$$

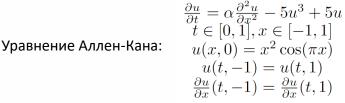
0.75

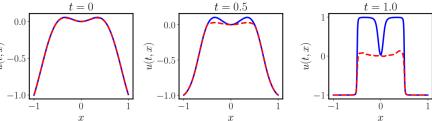
0.50

1.00

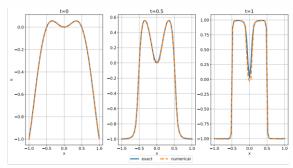


Wang S., Sankaran S., Perdikaris P. Respecting causality is all you need for training physics-informed neural networks //arXiv preprint arXiv:2203.07404. – 2022.





#### relative RMSE=4.98e-1



relative RMSE=1.43e-3

## Модифицированный loss+ Фурье эмбеддинг



Уравнение Лотки-Вольтерра:

$$\frac{\partial x}{\partial t} = x\alpha - yx\beta$$

$$\frac{\partial y}{\partial t} = -y\delta + yx\gamma$$

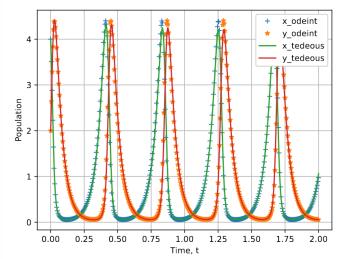
$$x(0) = x_0 = 4$$

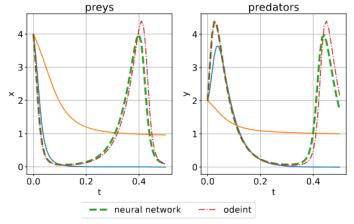
$$y(0) = y_0 = 2$$

$$t \in [0, 2]$$

$$\mathbf{v}(x) = (1, \cos(\omega x), \sin(\omega x), \cos(2\omega x), \sin(2\omega x), \cdots, \cos(m\omega x), \sin(m\omega x))$$

$$\omega = \frac{2\pi}{L}$$





## Рой частиц (PSO)

#### particle swarm optimization



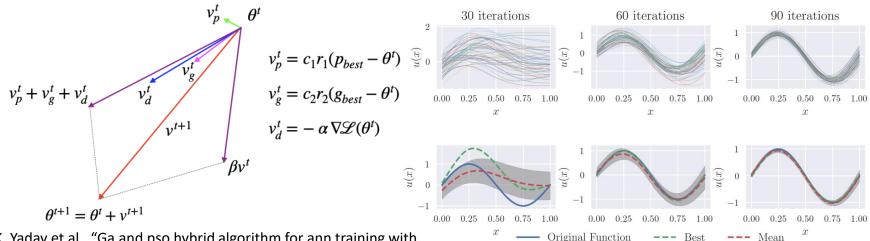
$$\theta^{t+1} = \theta^{t} + v^{t+1}$$

$$v^{t+1} = \beta v^{t} + c_{1}r_{1}(p_{best} - \theta^{t}) + c_{2}r_{2}(g_{best} - \theta^{t})$$

$$v^{t+1} = \beta v^{t} + c_{1}r_{1}(p_{best} - \theta^{t}) + c_{2}r_{2}(g_{best} - \theta^{t}) - \alpha \nabla \mathcal{L}(\theta^{t})$$

#### Уравнение Пуассона:

$$u_{xx} = g(x), \quad x \in [0, 1]$$
  
 $u(0) = u(1) = 0.$   
 $g(x) = -(2\pi)^2 \sin(2\pi x)$ 



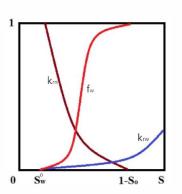
R. K. Yadav et al., "Ga and pso hybrid algorithm for ann training with application in medical diagnosis," in 2019

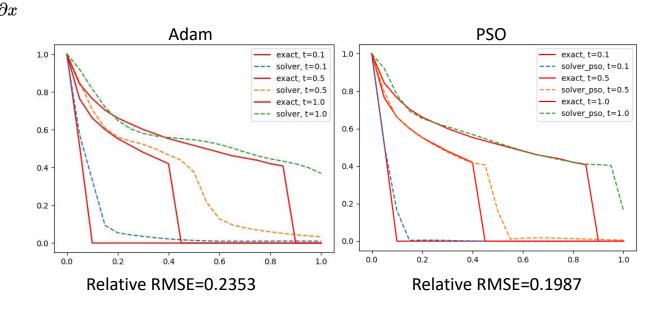
## Рой частиц (PSO)

уравнение Бакли-Леверетта



$$egin{align} rac{\partial S_w}{\partial t} + rac{\partial}{\partial x} \left(rac{Q}{\phi A}f_w(S_w)
ight) &= \phi rac{\partial S_w}{\partial x} \ f_w(S_w) &= rac{\lambda_w}{\lambda_w + \lambda_n} &= rac{rac{k_{rw}}{\mu_w}}{rac{k_{rw}}{\mu_w} + rac{k_{rn}}{\mu_n}} \ S_w(x,0) &= 0, \qquad S_w(0,t) &= 1 \ (x,t) &\in (0,1) imes (0,1) \end{aligned}$$





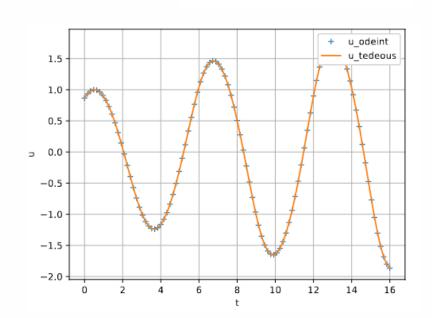
## Примеры

$$u'' + \epsilon(u^2 - 1)u' + u = 0$$
  

$$u'(t = 0) = 0.5$$
  

$$u(t = 0) = \sqrt{3}/2$$
  

$$t \in [0, 16]$$



# **iTMO**

Уравнение Шредингера

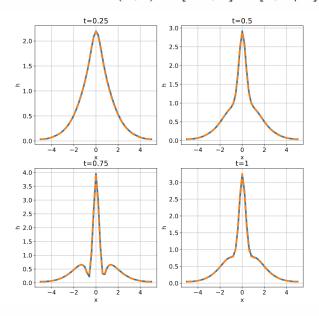
$$i\frac{\partial h}{\partial t} + \frac{1}{2}\frac{\partial^2 h}{\partial x^2} + |h^2|h = 0$$

$$h(x,0) = 2\mathrm{sech}(x)$$

$$h(-5,t) = h(5,t)$$

$$\frac{\partial h}{\partial x}(-5,t) = \frac{\partial h}{\partial x}(5,t)$$

$$(x,t) \in [-5,5] \times [0,\pi/2] = \Omega$$



## Примеры

# **ITMO**

## Задача о распаде произвольного разрыва

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} = 0$$

$$B(x,0) = \begin{cases} B_L & x \le 0.5 \\ B_R & x > 0.5 \end{cases}$$

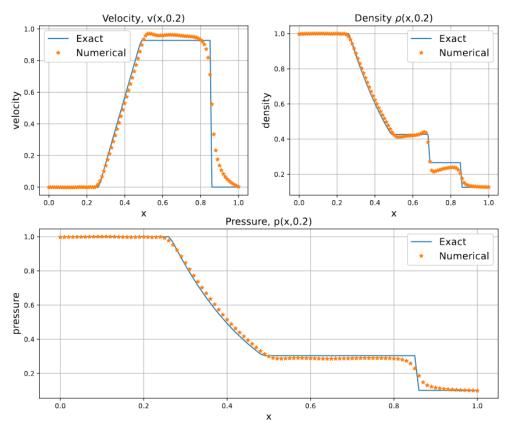
$$(x,t) \in [0,1] \times [0,0.2] = \Omega$$

$$U = \begin{pmatrix} \rho \\ \rho v \\ \rho E \end{pmatrix}, F(U) = \begin{pmatrix} \rho v \\ \rho v^2 + p \\ v(\rho E + p) \end{pmatrix};$$

$$E = \frac{v^2}{2} + e$$

$$B_L = (\rho_L, v_L, p_L)^T = (1, 0, 1)^T$$

$$B_R = (\rho_R, v_R, p_R)^T = (0.125, 0, 0.1)^T$$



#### **ИТОГИ**



- Реализован алгоритм адаптивного подбора весовых коэффициентов в функционале ошибок на основе индексов Соболя.
- Реализован и обобщен для систем дифференциальных уравнений (ДУ) алгоритм построения causal loss.
- Реализован алгоритм оптимизации на основе роя частиц для поиска решений ДУ
- Невысокая степень доверия к решению уравнений, полученных на основе данных
- Плохая сходимость для начальных уравнений
- Большое количество параметров, влияющих на сходимость нейронной сети

# Спасибо за внимание!

ITSIMOre than a UNIVERSITY