

ӨОЖ 05.13.10
ФТАХР 21.21.29

<https://doi.org/10.55452/1998-6688-2025-22-4-143-154>

¹Нұртас М.,

PhD, қауымдастырылған профессор, ORCID ID: 0000-0003-4351-0185,
e-mail: m.nurtas@iitu.edu.kz

^{1*}Абдикаликова З.Т.,

PhD, қауымдастырылған профессор, ORCID ID: 0000-0003-0131-4469,
*e-mail: z.abdikalikova@iitu.edu.kz

¹Адиат Л.,

магистр, лектор, ORCID ID: 0009-0006-2969-1695,
e-mail: l.adiat@iitu.edu.kz

¹Халықаралық ақпараттық технологиялар университеті, Алматы қ., Қазақстан

ӨЗІНЕ-ӨЗІ ҰҚСАС ШЕШІМДЕРГЕ АРНАЛҒАН ФИЗИКАҒА НЕГІЗДЕЛГЕН НЕЙРОНДЫҚ ЖЕЛІ (PINN) ӘДІСІ

Аңдатпа

Жылу өткізгіштік пен газ динамикасы сияқты күрделі физикалық процестерді сипаттайтын дербес туындысы бар дифференциалдық теңдеулерді сандық әдістер арқылы шешу барысында көбіне үлкен есептеу қуаты қажет етіледі. Осы қиындықтарды шешу үшін соңғы жылдары ғылым мен техниканың назарында физикаға негізделген нейрондық желілер (Physics-Informed Neural Networks, PINN) ерекше орын алған. Бұл мақалада PINN әдісі арқылы жылу және газ динамикасы теңдеулерінің шешімдерін табу мәселесі қарастырылады. Дәстүрлі сандық әдістерден ерекшелігі, физикаға негізделген нейрондық желі әдісі физикалық заңдарды нейрон желісінің құрылымына енгізу арқылы есепті шешуге мүмкіндік береді. Яғни, шешім тек қана мәліметтерге емес, сонымен қатар теңдеудің өзіне бағынады. Мақалада PINN әдісінің архитектурасы, шығын функцияларының құрылымы және олардың жылу өткізгіштік және Эйлер теңдеулерімен байланысы нақты мысалдармен сипатталады. Сонымен қатар, есептің бастапқы және шекаралық шарттарын енгізу механизмдері және шешімдердің орнықтылығы мен дәлдігіне әсер ететін факторлар талданған. Нәтижелер арқылы PINN-нің тиімділігі мен болашақта күрделі көп фазалы, көпөлшемді есептерге қолдану мүмкіндігі көрсетілді. Сонымен қатар, PINN-нің есептеу үдерісін жылдамдату және тұрақтылығын арттыру бағытындағы зерттеулер де ұсынылған.

Тірек сөздер: PINN, өзіне-өзі ұқсас, жылу теңдеуі, Эйлер теңдеуі, соққы түтігі (shock tube).

Кіріспе

Қазіргі таңда дифференциалдық теңдеулерді шешу әдістері ғылыми және инженерлік есептерде кеңінен қолданылады. Атап айтқанда, жылуөткізгіштік және газ динамикасы сияқты күрделі физикалық құбылыстарды сипаттайтын дербес туындылы дифференциалдық теңдеулер (ДТДТ) – табиғи және техникалық процестерді модельдеудің негізгі құралдарының бірі. Осындай есептерді сандық әдістермен шешу көптеген жағдайларда үлкен есептеу ресурстарын талап етеді, әсіресе көпөлшемді, шекаралық және бастапқы шарттары күрделі жүйелерде. Бұл шектеулерді жеңу мақсатында соңғы жылдары ғылым мен техникада физикаға негізделген нейрондық желілер (Physics-Informed Neural Networks, PINN) ерекше назарға иілінді.

PINN – машиналық оқыту мен классикалық физиканың синтезі болатын жаңа парадигма. Бұл әдіс алғаш рет (2019–2024 жж.) Raissi және т.б. [1–2] еңбектерде ұсынылып, кейіннен көптеген күрделі ДТДТ есептерін шешуде тиімділігін көрсетті. PINN әдісінің басты ерекшелігі – ол деректерге ғана емес, сонымен қатар физикалық заңдылықтарға (мысалы: энергия

сақталуы, масса сақталуы, импульс заңдары және т.б.) негізделеді. Мұндай тәсіл модельдің физикалық шындыққа сәйкестігін қамтамасыз етіп, мәліметтердің жеткіліксіздігі жағдайында да жақсы нәтиже алуға мүмкіндік береді.

PINN әдісімен салыстырғанда классикалық әдістер (мысалы: ақырлы элементтер әдісі (FEM), ақырлы айырмалар әдісі (FDM)) тор генерациясына, дәлдікке және есептеу тұрақтылығына тәуелді. Бұл әдістер әсіресе көпөлшемді, бейсызық немесе стохастикалық ДТДТ-де қиындықтарға тап болады. Ал PINN әдісі кеңістіктік торларды қажет етпейді, оның орнына кездейсоқ коллокациялық нүктелер арқылы шешімді үйретуге мүмкіндік береді. Сонымен қатар, PINN әдісі параллельді есептеуге ыңғайлы, бұл оны үлкен деректермен жұмыс істеу мен нақты уақыттағы есептерге бейімделуге мүмкіндік береді.

Бүгінгі таңда PINN әдісі жылуөткізгіштік, Био теңдеулері [3], Навье–Стокс [4], Коши есебі [5], реакция диффузиялық жүйелер, акустикалық толқындар [6], фазалық өріс модельдері сынды көптеген салаларда сәтті қолданылуда. Атап айтқанда, жылу мен газ динамикасы теңдеулері физикалық процестерді сипаттаудың маңызды мысалдары ретінде PINN әдісін қолдануға өте қолайлы. Бұл есептерде шекаралық және бастапқы шарттар, сондай-ақ күрделі шешім құрылымы жиі кездесетіндіктен, PINN әдісінің артықшылықтары айқын байқалады.

PINNs әдісінің даму барысымен, қазіргі жағдайы және болашақ зерттеу бағыттарымен жүйелі түрде [7] жұмыста танысуға болады. [8] әдебиетте PINN әдісін ғылыми есептеулерде қолданудың әртүрлі қырларын қамтитын кең көлемді шолу, оның ішінде PINN-есептегіштерін баптау, оқыту теориясының аспектілері, қолжетімді құралдар, болашақ бағыттар мен соңғы үрдістер, сондай-ақ дәлдік пен жинақталу мәселелері қарастырылады. PINN әдісінің эволюциясына арналған шолу, оның ішінде Physics-Informed Kolmogorov-Arnold Networks (PIKANs) сияқты жаңа архитектуралар және олардың биомедицина, механика, геофизика және басқа салалардағы қолданылуы [9]-да талқыланады. Жасанды нейрондық желілерді (artificial neural networks – ANN) қолдану арқылы дифференциалдық теңдеулерді шешу әдісі [10]-да сипатталады. [11] мақалада авторлар дәстүрлі машиналық оқыту әдістерін физикалық заңдармен біріктіру арқылы күрделі жүйелерді модельдеу мен болжау үшін қолданылатын тәсілді сипаттайды, бұл тәсіл әсіресе дифференциалдық теңдеулермен сипатталатын физикалық процестерге бағытталған.

Осы мақалада біз өзіне-өзі ұқсас шешімдерге негізделген PINN әдісін пайдалана отырып, жылу және газ динамикасының ішінара туынды теңдеулерін шешудің тиімді жолдарын қарастырамыз. Бұл тәсілдің артықшылығы – ол аналитикалық шешімге жақын, яғни өзіне-өзі ұқсастық қасиеті бар шешімдерді нейрондық желі арқылы қалпына келтіруге мүмкіндік береді. Осы бағыттағы зерттеулер PINN әдісінің теориялық және қолданбалы мүмкіндіктерін кеңейтіп, оны нақты физикалық процестерді модельдеуде кеңінен қолдануға жол ашады.

Материалдар мен әдістер

Өзіне-өзі ұқсастық ұғымы

Өзіне-өзі ұқсас шешім – уақыт немесе кеңістік масштабталғанда шешімнің формасы өзгермей, тек үлкейіп немесе кішірейіп отыратын сипат. Жалпы түрде шешім келесідей болады:

$$u(x, t) = t^{\alpha} * f\left(\frac{x}{t^{\beta}}\right),$$

мұндағы $f(\cdot)$ – өзіне-өзі ұқсас профиль, ал α , β – масштабтау көрсеткіштері [12].

Жылу теңдеуіне PINN қолдану

Зерттеуімізде тәжірибе ретінде жылу өткізгіштік және газ динамикасы теңдеулерін шешу үшін PINN әдісін қолдануды көрсетеміз. Бұл әдістің негізінде нейрондық желіні тек қана деректерге емес, сонымен қатар физикалық заңдарға – дифференциалдық теңдеулерге бағынатын түрде үйрету қағидасы қалай болатынын нақтылаймыз.

Жылу теңдеуі үшін модель

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}, \quad u(x, 0) = \delta(x).$$

Аналитикалық шешімі:

$$u(x, t) = \frac{1}{\sqrt{4\pi t}} * \exp\left(-\frac{x^2}{4t}\right).$$

x – кеңістік координатасы;

t – уақыт;

$\xi = \frac{x}{\sqrt{t}}$ – өзіндік ұқсастық айнымалысы;

$u(x, t)$ – температура функциясы;

$f(\xi)$ – PINN арқылы үйрелінетін өзіндік ұқсас функция.

$$u_{\theta}(x, t) = \frac{1}{\sqrt{t}} * NN_{\theta}(\xi).$$

PINN шығын функциясы

$$u_{\theta}(x, t) = \frac{1}{\sqrt{t}} f_{\theta}(\xi), \quad \xi = \frac{x}{\sqrt{t}},$$

$$\mathcal{L}_{\text{heat}} = \frac{1}{N} \sum_{i=1}^N \left| \frac{\partial u_{\theta}}{\partial t}(x_i, t_i) - \frac{\partial^2 u_{\theta}}{\partial x^2}(x_i, t_i) \right|^2.$$

MAE (Mean absolute error):

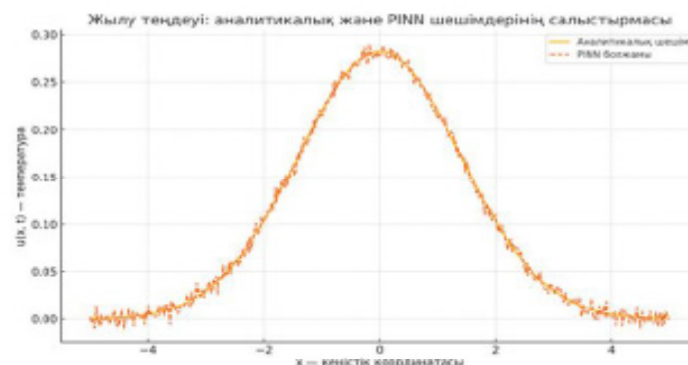
$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |U_{\text{pred}}(x_i) - U_{\text{true}}(x_i)|.$$

RMSE (Root mean square error):

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (U_{\text{pred}}(x_i) - U_{\text{true}}(x_i))^2}.$$

x	$U_{\text{pred}}(x) - U_{\text{true}}(x)$
-5	0
-2	+0.02
0	0
2	-0.02
5	0

Төмендегі 1-суретте жылу теңдеуі үшін аналитикалық және PINN шешімдерінің салыстырмасы ұсынылған.



Сурет 1 – Жылу теңдеуі: аналитикалық және PINN шешімдерінің салыстырмасы

Нәтижелер мен талқылау

1D Эйлер теңдеулері және өзіне өзі ұқсастық

Эйлер теңдеулерінің 1 өлшемді формасы:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} = 0$$

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} = 0$$

$$\frac{\partial E}{\partial t} + \frac{\partial((E + p)u)}{\partial x} = 0.$$

Соққы түтігі бастапқы шарттары:

$$(\rho, u, p)(x, 0) = \begin{cases} (1.0, 0, 1.0), & \text{егер } x < 0 \\ (0.125, 0, 0.1), & \text{егер } x > 0 \end{cases}$$

PINN әдісіне негізделген шығын функциясы

1D Эйлер теңдеулері үшін PINN архитектурасы t және x айнымалыларын кіріс ретінде қабылдап, тығыздық $\rho(t, x)$, жылдамдық $u(t, x)$ және қысым $p(t, x)$ функцияларын болжайды. Бұл функциялардың көмегімен келесі қалдықтар анықталады:

$$\mathcal{R}_1(t, x) = \frac{\partial \rho_\theta}{\partial t} + \frac{\partial(\rho_\theta u_\theta)}{\partial x},$$

$$\mathcal{R}_2(t, x) = \frac{\partial(\rho_\theta u_\theta)}{\partial t} + \frac{\partial(\rho_\theta u_\theta^2 + p_\theta)}{\partial x};$$

$$\mathcal{R}_3(t, x) = \frac{\partial E_\theta}{\partial t} + \frac{\partial((E_\theta + p_\theta)u_\theta)}{\partial x},$$

мұндағы, $E_\theta = \frac{p_\theta}{\gamma - 1} + \frac{1}{2} \rho_\theta u_\theta^2$ – жалпы энергия.

Осы қалдықтарды ескере отырып, шығын функциясы келесі түрде жазылады:

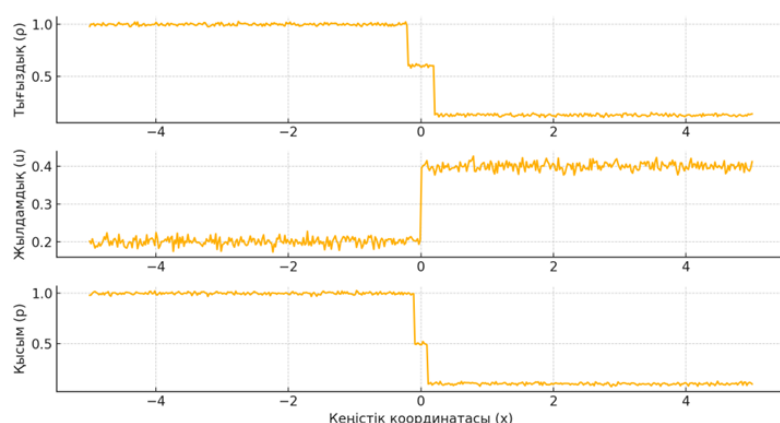
$$\mathcal{L}(\theta) = \mathcal{L}_d + \lambda_p \cdot \mathcal{L}_{phys},$$

мұнда $\mathcal{L}_d$ – мәліметтерге (data) негізделген (мысалы, MSE), ал $\mathcal{L}_{phys}$ – физикалық қалдықтардан тұратын бөлік:

$$\mathcal{L}_{phys} = \frac{1}{N_f} \sum_{i=1}^{N_f} (\mathcal{R}_1^2(t_i, x_i) + \mathcal{R}_2^2(t_i, x_i) + \mathcal{R}_3^2(t_i, x_i)),$$

мұндағы N_f – физикалық нүктелер саны, ал λ_p – физикалық заңдарды сақтау салмағы. Бұл функция PINN моделінің физикалық заңдарды сақтап шешім үйренуіне мүмкіндік береді.

Эйлер теңдеуі үшін соққы түтігі жағдайында PINN нәтижесі 2-суретте ұсынылған.

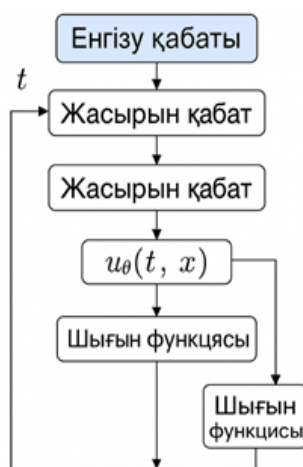


Сурет 2 – PINN нәтижесі: Эйлер тендеуі (соққы түтігі жағдайы)

PINN моделі және шығын функциялары

PINN архитектурасы

PINN архитектурасы – көпқабатты перцептрон (MLP) түрінде құрылады. Мысал ретінде, 3 жасырын қабатты нейрондық желі қарастырайық (3-сурет). Оның кірісінде x және t айнымалылары (кеңістік пен уақыт), ал шығысында $u_\theta(t, x)$ шамасы беріледі, мұндағы θ – модель параметрлері (салмақтар мен ығысу).



Сурет 3 – PINN моделі архитектурасы (Қосымша А негізінде)

Желі қабаттарының жалпы сипаттамасы:

- ♦ Кіріс қабаты: x, t мәндерін қабылдайды.
- ♦ Жасырын қабаттар: активация функциясы ретінде $\tanh$ немесе $\sin$ пайдаланылады.
- ♦ Шығыс қабаты: $u_\theta(t, x)$ – шешімнің жуықталған мәні.

Шығын функциялары

PINN-нің негізгі ерекшелігі – оның шығын функциясы физикалық қалдықтарды да қамтиды. Жалпы шығын функциясы келесі түрде жазылады:

$$\mathcal{L}(\theta) = \lambda_d \mathcal{L}_d + \lambda_p \cdot \mathcal{L}_p. \quad (1)$$

Мұндағы:

- ♦ $\mathcal{L}_d$ – мәліметтерге негізделген шығын (мысалы, нақты өлшемдермен салыстыру);
- ♦ $\mathcal{L}_n$ – физикаға негізделген қалдықтар: $\|\mathcal{N}[u_\theta] - f\|^2$;
- ♦ λ_d, λ_p – салмақ параметрлері.

Физикалық қалдықтар $\mathcal{L}_p$ – жылу немесе басқа теңдеулердің шешіміндегі теңсіздіктер арқылы анықталады. Мысалы, жылу теңдеуі үшін жоғарыда берілген формасын тағы да бір еске саламыз:

$$\mathcal{L}_p = \frac{1}{N_f} \sum_{i=1}^{N_f} \left| \frac{\partial u_\theta}{\partial t}(t_i, x_i) - \alpha \frac{\partial^2 u_\theta}{\partial x^2}(t_i, x_i) \right|^2. \quad (2)$$

Мұнда α – диффузия коэффициенті, N_f – физикалық нүктелер саны. Модельдің басты артықшылықтары:

- ♦ Шешімнің физика заңдарын бұзбай үйренуі;
- ♦ Деректер аз болған жағдайда да жұмыс істей алуы;
- ♦ Бастапқы және шекаралық шарттарды оңай енгізу мүмкіндігі;
- ♦ Жоғары өлшемді есептерге бейімдеуге болатын икемді құрылым.

Қорытынды

Осы мақалада біз PINN әдісі арқылы жылу теңдеуі және газ динамикасы теңдеулерін (Эйлер теңдеулері) шешудің мүмкіндіктерін қарастырдық. PINN – бұл физикалық жүйелерді модельдеудің жаңа парадигмасы болып табылатынын дәлелдедік. Ол сандық әдістер мен машиналық оқытудың үйлесімін қамтамасыз етіп, дәстүрлі дискретизация әдістеріне балама ұсынатынын көрсеттік. Өзіне өзі ұқсас шешімдер PINN моделінің тиімділігін арттырып, оқу уақытын қысқартатынын көрсетілді. Бұл тәсіл әсіресе аналитикалық шешімі жоқ күрделі дербес туындылы дифференциалдық теңдеулер үшін пайдалы екенін анықтадық. PINN әдісінің қазіргі таңда әлеуеті өте жоғары, алайда ол да жетілдіруді қажет ететінін мойындауымыз керек. Болашақ зерттеулерге келесі бағыттарды ұсынуға болады:

- ♦ Мультифизикалық есептер: PINN бірнеше теңдеулерден тұратын жүйелерге (мысалы, кеуекті серпімді, көп фазалы ағын) кеңейтілуі мүмкін [13];
- ♦ Есептің өлшемін үлкейту: Қазіргі жұмыс бір немесе екі өлшемге негізделген. Алдағы мақсат – үш өлшемді кеңістіктік және уақыттық есептерге бейімдеу;
- ♦ Адаптивті торлармен біріктіру: Тор тығыздығын қалдық мәніне қарай өзгертіп отыратын PINN түрін жасау – есептеу тиімділігін арттыруға сеп болады;
- ♦ Шығын функциясын автоматты теңшеу: λ_d, λ_p сияқты гиперпараметрлерді автоматты түрде реттейтін стратегияларды зерттеу керек;
- ♦ Сирек мәліметтер жағдайы: Реалистік жағдайда деректер өте аз және толық емес болады. Мұндай жағдайда, PINN-ді күшейту үшін генеративті модельдермен (мысалы, VAE, GAN) біріктіру даму мүмкіндігі бар бағыт.

Нәтижелер көрсеткендей, PINN арқылы алынған шешімдер жоғары дәлдікке ие. Сонымен қатар, оның шешімдері аналитикалық түрде алынатын шешімдерге жақын және сәйкес келетіндігі сандық тәжірибелердің графикалық салыстырулары арқылы дәлелденді. Қорытындылай келе, PINN – ғылым мен техникадағы көпқырлы күрделі есептерді шешуде болашақтың қуатты құралдарының бірі бола алады деп айта аламыз.

1. Қосымша А

PINN арқылы өзіндік ұқсас шешімді үйрену – толық Python коды (жылу теңдеуі мысалында)

```
import torch
import torch.nn as nn
import torch.autograd as autograd
import matplotlib.pyplot as plt
# Құрылғыны анықтау
```

```
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")

# PINN моделі
class SelfSimilarPINN(nn.Module):
    def __init__(self):
        super().__init__()
        self.net = nn.Sequential(
            nn.Linear(1, 50),
            nn.Tanh(),
            nn.Linear(50, 50),
            nn.Tanh(),
            nn.Linear(50, 1) )

    def forward(self, x, t):
        xi = x / torch.sqrt(t + 1e-5)
        f = self.net(xi)
        u = f / torch.sqrt(t + 1e-5)
        return u

# Оқу деректерін генерациялау
def generate_training_data(n=1000):
    x = torch.linspace(-5, 5, n).reshape(-1, 1)
    t = torch.linspace(0.1, 1.0, n).reshape(-1, 1)
    x, t = torch.meshgrid(x.squeeze(), t.squeeze(), indexing='ij')
    return x.reshape(-1, 1).to(device), t.reshape(-1, 1).to(device)

# PDE қалдығын есептеу
def pde_residual(model, x, t):
    x.requires_grad = True
    t.requires_grad = True
    u = model(x, t)
    u_t = autograd.grad(u, t, grad_outputs=torch.ones_like(u), retain_graph=True, create_graph=True)[0]
    u_x = autograd.grad(u, x, grad_outputs=torch.ones_like(u), retain_graph=True, create_graph=True)[0]
    u_xx = autograd.grad(u_x, x, grad_outputs=torch.ones_like(u_x), retain_graph=True, create_graph=True)[0]
    return u_t - u_xx

# Модель және optimizer
model = SelfSimilarPINN().to(device)
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)

# Оқу деректері
x_train, t_train = generate_training_data(200)

# Оқыту циклі
loss_history = []
for epoch in range(1000):
    optimizer.zero_grad()
    residual = pde_residual(model, x_train, t_train)
    loss = torch.mean(residual ** 2)
```



```

loss.backward()
optimizer.step()
loss_history.append(loss.item())
if epoch % 100 == 0:
    print(f'Epoch {epoch}, Loss: {loss.item():.6f}')

```

Шығын функция графигі

```

plt.plot(loss_history)
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.title("Training Loss")
plt.grid(True)
plt.show()

```

2. Қосымша В

PINN арқылы Эйлер теңдеуін шешу(Self-Similar Shock Tube) — Python коды

```

import torch
import torch.nn as nn
import torch.autograd as autograd
import matplotlib.pyplot as plt
device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
gamma = 1.4
# Желі блогы
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.net = nn.Sequential(
            nn.Linear(1, 64),
            nn.Tanh(),
            nn.Linear(64, 64),
            nn.Tanh(),
            nn.Linear(64, 1) )
    def forward(self, xi):
        return self.net(xi)

# PINN моделі: rho, u, p
class EulerPINN(nn.Module):
    def __init__(self):
        super(EulerPINN, self).__init__()
        self.rho_net = Net()
        self.u_net = Net()
        self.p_net = Net()
    def forward(self, x, t):
        xi = x / (t + 1e-5)
        rho = self.rho_net(xi)
        u = self.u_net(xi)
        p = self.p_net(xi)
        return rho, u, p
# PDE қалдықтары
def euler_residuals(model, x, t):
    x.requires_grad = True

```



```
t.requires_grad = True
rho, u, p = model(x, t)
m = rho * u
E = p / (gamma - 1) + 0.5 * rho * u**2
rho_t = autograd.grad(rho, t, grad_outputs=torch.ones_like(rho), retain_graph=True, create_graph=True)[0]
m_t = autograd.grad(m, t, grad_outputs=torch.ones_like(m), retain_graph=True, create_graph=True)[0]
E_t = autograd.grad(E, t, grad_outputs=torch.ones_like(E), retain_graph=True, create_graph=True)[0]
rho_x = autograd.grad(rho, x, grad_outputs=torch.ones_like(rho), retain_graph=True, create_graph=True)[0]
m_x = autograd.grad(m, x, grad_outputs=torch.ones_like(m), retain_graph=True, create_graph=True)[0]
p_x = autograd.grad(p, x, grad_outputs=torch.ones_like(p), retain_graph=True, create_graph=True)[0]
u_x = autograd.grad(u, x, grad_outputs=torch.ones_like(u), retain_graph=True, create_graph=True)[0]
E_x = autograd.grad(E, x, grad_outputs=torch.ones_like(E), retain_graph=True, create_graph=True)[0]

# Қалдықтар
mass_eq = rho_t + m_x
momentum_eq = m_t + autograd.grad(m * u + p, x, grad_outputs=torch.ones_like(p), retain_graph=True, create_graph=True)[0]
energy_eq = E_t + autograd.grad((E + p) * u, x, grad_outputs=torch.ones_like(E), retain_graph=True, create_graph=True)[0]
return mass_eq, momentum_eq, energy_eq

# Деректер генерациясы
def generate_data(n=1000):
    x = torch.linspace(-1, 1, n).reshape(-1, 1).to(device)
    t = torch.ones_like(x) * 1.0
    return x, t

# Модель және optimizer
model = EulerPINN().to(device)
optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)
x_train, t_train = generate_data(1000)

# Оқыту циклі
loss_history = []
for epoch in range(1000):
    optimizer.zero_grad()
    mass_eq, mom_eq, energy_eq = euler_residuals(model, x_train, t_train)
    loss = torch.mean(mass_eq**2) + torch.mean(mom_eq**2) + torch.mean(energy_eq**2)
    loss.backward()
    optimizer.step()
    loss_history.append(loss.item())
    if epoch % 100 == 0:
        print(f'Epoch {epoch}: Loss = {loss.item():.6f}')
```

```
# Шығын функция графигі
plt.plot(loss_history)
plt.xlabel("Epoch")
plt.ylabel("Loss")
plt.title("PINN Training Loss (1D Euler, self-similar)")
plt.grid(True)
plt.show()
```

ӘДЕБИЕТТЕР

- 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. <https://doi.org/10.1016/j.jcp.2018.10.045>.
- 2 Raissi M., Perdikaris P., Daryakenari N.A., Karniadakis G.E. Physics-Informed Neural Networks and Extensions. – 2024. <https://doi.org/10.48550/arXiv.2408.16806>.
- 3 Meirmanov A. Mathematical models for poroelastic flows. – Paris: Atlantis Press, Atlantis Studies in Differential Equations, 2014. <https://doi.org/10.2991/978-94-6239-015-7>.
- 4 Fefferman C.L. Existence and smoothness of the Navier–Stokes equation. – Clay Mathematics Institute, 2017. – P. 1–6. URL: <https://www.claymath.org/wp-content/uploads/2022/06/navierstokes.pdf>.
- 5 Конюхов И.В., Конюхов В.М., Черница А.А., Дюсенова А. Особенности применения физически информированных нейронных сетей для решения обыкновенных дифференциальных уравнений // *Компьютерные исследования и моделирование*. – 2024. – Т. 16. – № 7. – С. 1621–1636. <https://doi.org/10.20537/2076-7633-2024-16-7-1621-1636>.
- 6 Meirmanov A., Nurtas M. Mathematical models of seismic in composite media: elastic and poro-elastic components // *Electronic Journal of Differential Equations*. – 2016. – No. 184. – P. 1–22. URL: <http://ejde.math.unt.edu>.
- 7 Lawal Z.K., Yassin H., Lai D.T.C., Idris A.C. Physics-Informed Neural Network (PINN) Evolution and Beyond: A Systematic Literature Review and Bibliometric Analysis // *Big Data and Cognitive Computing*. – 2022. – Vol. 6. – No. 4. – Article 140. <https://doi.org/10.3390/bdcc6040140>.
- 8 Cuomo S., Schiano di Cola V., Giampaolo F., Rozza G., Raissi M., Piccialli F. Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next. – 2022. <https://doi.org/10.48550/arXiv.2201.05624>.
- 9 Toscano J.D., Oommen V., Varghese A.J., Zou Z., Daryakenari N.A., Wu C., Karniadakis G.E. From PINNs to PIKANs: Recent Advances in Physics-Informed Machine Learning. – 2024. <https://doi.org/10.48550/arXiv.2410.13228>.
- 10 Lagaris I.E., Likas A., Fotiadis D.I. Artificial neural networks for solving ordinary and partial differential equations // *IEEE Transactions on Neural Networks*. – 1998. – Vol. 9. – No. 5. – P. 987–1000.
- 11 Karniadakis G.E., Kevrekidis I.G., Lu L., Perdikaris P., Wang S., Yang L. Physics-informed machine learning // *Nature Reviews Physics*. – 2021. – Vol. 3. – P. 422–440.
- 12 Barenblatt G.I. Scaling, self-similarity, and intermediate asymptotics. – Cambridge: Cambridge University Press, 1996.
- 13 Krasnozhonov R. and Nurtas M. Modeling the Propagation of Acoustic Waves in an Elastic Medium Using Physics-Informed Neural Networks, 2025 IEEE 5th International Conference on Smart Information Systems and Technologies (SIST), Astana, Kazakhstan, 2025, pp. 1–7. <https://doi.org/10.1109/SIST61657.2025.11139217>.

REFERENCES

- 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*, 378, 686–707 (2019). <https://doi.org/10.1016/j.jcp.2018.10.045>.
- 2 Raissi, M., Perdikaris, P., Daryakenari, N.A., Karniadakis, G.E. Physics-Informed Neural Networks and Extensions, 2024. <https://doi.org/10.48550/arXiv.2408.16806>.

- 3 Meirmanov, A. Mathematical models for poroelastic flows. Paris: Atlantis Press, Atlantis Studies in Differential Equations, 2014. <https://doi.org/10.2991/978-94-6239-015-7>.
- 4 Fefferman, C.L. Existence and smoothness of the Navier–Stokes equation. Clay Mathematics Institute, pp. 1–6 (2017). URL: <https://www.claymath.org/wp-content/uploads/2022/06/navierstokes.pdf>.
- 5 Konyukhov, I.V., Konyukhov, V.M., Chernitsa, A.A., Dyusenova, A. Osobennosti primeneniya fizicheskii informirovannykh neyronnykh setey dlya resheniya obyknovennykh differentsialnykh uravneniy. Kompyuternye issledovaniya i modelirovanie, 16(7), 1621–1636 (2024). <https://doi.org/10.20537/2076-7633-2024-16-7-1621-1636>.
- 6 Meirmanov, A., Nurtas, M. Mathematical models of seismic in composite media: elastic and poroelastic components. Electronic Journal of Differential Equations, 184, 1–22 (2016). URL: <http://ejde.math.unt.edu>.
- 7 Lawal, Z.K., Yassin, H., Lai, D.T.C., Idris, A.C. Physics-Informed Neural Network (PINN) Evolution and Beyond: A Systematic Literature Review and Bibliometric Analysis. Big Data and Cognitive Computing, 6(4), article 140 (2022). <https://doi.org/10.3390/bdcc6040140>.
- 8 Cuomo, S., Schiano di Cola, V., Giampaolo, F., Rozza, G., Raissi, M., Piccialli, F. Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next, 2022. <https://doi.org/10.48550/arXiv.2201.05624>.
- 9 Toscano, J.D., Oommen, V., Varghese, A.J., Zou, Z., Daryakenari, N.A., Wu, C., Karniadakis, G.E. From PINNs to PIKANs: Recent Advances in Physics-Informed Machine Learning, 2024. <https://doi.org/10.48550/arXiv.2410.13228>.
- 10 Lagaris, I.E., Likas, A., Fotiadis, D.I. Artificial neural networks for solving ordinary and partial differential equations. IEEE Transactions on Neural Networks, 9(5), 987–1000 (1998).
- 11 Karniadakis, G.E., Kevrekidis, I.G., Lu, L., Perdikaris, P., Wang, S., Yang, L. Physics-informed machine learning. Nature Reviews Physics, 3, 422–440 (2021).
- 12 Barenblatt, G.I. Scaling, self-similarity, and intermediate asymptotics. Cambridge: Cambridge University Press, 1996.
- 13 Krasnozhonov, R. and Nurtas, M. Modeling the Propagation of Acoustic Waves in an Elastic Medium Using Physics-Informed Neural Networks, 2025 IEEE 5th International Conference on Smart Information Systems and Technologies (SIST), Astana, Kazakhstan, 2025, pp. 1–7, <https://doi.org/10.1109/SIST61657.2025.11139217>.

¹Нуртас М.,

PhD, ассоциированный профессор, ORCID ID: 0000-0003-4351-0185,

e-mail: m.nurtas@iitu.edu.kz

^{1*}Абдикаликова З.Т.,

PhD, ассоциированный профессор, ORCID ID: 0000-0003-0131-4469,

*e-mail: z.abdikalikova@iitu.edu.kz

¹Адиат Л.,

магистр, лектор, ORCID ID: 0009-0006-2969-1695,

e-mail: l.adiat@iitu.edu.kz

¹Международный университет информационных технологий, г. Алматы, Казахстан

МЕТОД ФИЗИЧЕСКИ ОБОСНОВАННОЙ НЕЙРОННОЙ СЕТИ (PINN), ОСНОВАННЫЙ НА САМОПОДОБНЫХ РЕШЕНИЯХ

Аннотация

При численном решении дифференциальных уравнений в частных производных, описывающих сложные физические процессы, такие как теплопроводность и газовая динамика, зачастую требуется значительная вычислительная мощность. Для решения этих сложностей в последние годы особое внимание науки и техники привлекают физически информированные нейронные сети (Physics-Informed Neural Networks, PINN). В данной статье рассматривается задача нахождения решений уравнений теплопроводности и газовой динамики с помощью метода PINN. В отличие от традиционных численных методов, метод физиче-

ски информированных нейронных сетей позволяет решать задачи, внедряя физические законы в структуру нейронной сети. То есть решение подчиняется не только данным, но и самому уравнению. В статье описываются архитектура метода PINN, структура функций потерь и их связь с уравнением теплопроводности и уравнениями Эйлера на конкретных примерах. Кроме того, анализируются механизмы введения начальных и граничных условий, а также факторы, влияющие на устойчивость и точность решений. Полученные результаты демонстрируют эффективность PINN и возможность их применения в будущем для решения сложных многомерных и многофазных задач. Также предложены исследования, направленные на ускорение вычислительного процесса и повышение стабильности PINN.

Ключевые слова: PINN, самоподобие, уравнение теплопроводности, уравнение Эйлера, ударная труба (shock tube).

¹**Nurtas M.,**

PhD, Associate Professor, ORCID ID: 0000-0003-4351-0185,

e-mail: m.nurtas@iitu.edu.kz

^{1*}**Abdikalikova Z.T.,**

PhD, Associate Professor, ORCID ID: 0000-0003-0131-4469,

*e-mail: z.abdikalikova@iitu.edu.kz

¹**Adiat L.,**

MSc, Lecturer, ORCID ID: 0009-0006-2969-1695,

e-mail: l.adiat@iitu.edu.kz

¹International University of Information Technologies, Almaty, Kazakhstan

PHYSICS-INFORMED NEURAL NETWORK (PINN) METHOD BASED ON SELF-SIMILAR SOLUTIONS

Abstract

In the numerical solution of partial differential equations that describe complex physical processes such as heat conduction and gas dynamics, substantial computational resources are often required. To address these challenges, Physics-Informed Neural Networks (PINNs) have gained increasing attention in recent years within the fields of science and engineering. This paper investigates the application of the PINN methodology to obtain solutions to the heat conduction and gas dynamics equations. Unlike traditional numerical approaches, the physics-informed neural network framework incorporates governing physical laws directly into the neural network architecture. Consequently, the solution is constrained not only by data but also by the underlying differential equations. The paper presents the architecture of the PINN framework and details the structure of loss functions, demonstrating their relationship with the heat equation and the Euler equations using specific examples. Furthermore, the implementation of initial and boundary conditions is discussed, along with an analysis of factors influencing the stability and accuracy of the obtained solutions. The results highlight the efficiency of PINNs and demonstrate their potential for solving complex multiphase and high-dimensional problems in the future. Additionally, current research directions aimed at accelerating the computational process and enhancing the robustness of PINNs are outlined.

Keywords: PINN, self-similarity, heat equation, Euler equation, shock tube.

Мақаланың редакцияға түскен күні: 17.10.2025