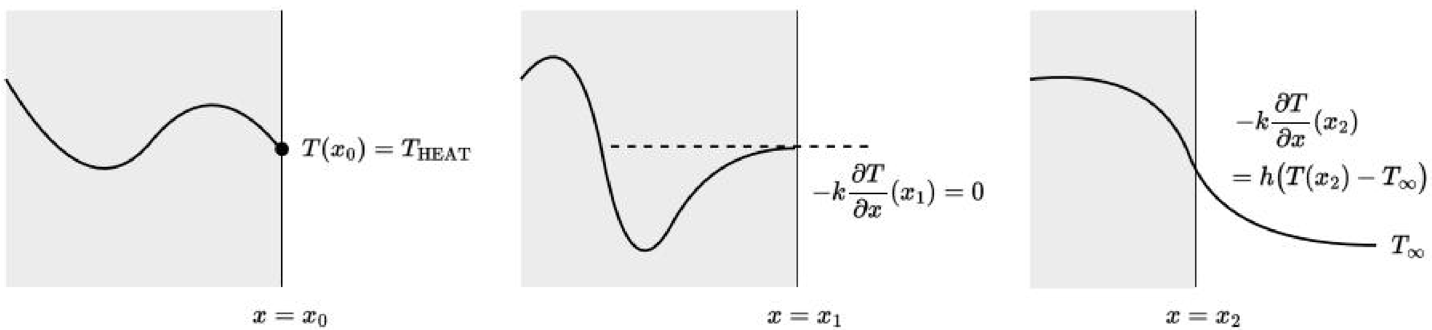
A Unified Hard-Constraint Framework for Solving Geometrically Complex PDEs

NeurIPS 2022

Many problems are characterized by partial differential equations (PDEs) with the solution constrained by boundary conditions (BCs)

commonly used BCs

① Dirichlet
② Neumann
③ Robin



Dirichlet BC **Neumann BC** **Robin BC**

Physics-Informed Neural Networks (PINNs)

$$\begin{aligned} \Delta u(x_1, x_2) &= 0, & x_1 &\in (0,1], x_2 \in [0,1], \\ u(x_1, x_2) &= g(x_2), & x_1 &= 0, x_2 \in [0,1], \end{aligned}$$

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\mathcal{F}}(\boldsymbol{\theta}) + \mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta}) \triangleq \frac{1}{N_f}\sum_{i=1}^{N_f}\left|\Delta \hat{u}(x_{f,1}^{(i)}, x_{f,2}^{(i)}; \boldsymbol{\theta})\right|^2 + \frac{1}{N_b}\sum_{i=1}^{N_b}\left|\hat{u}(0, x_{b,2}^{(i)}; \boldsymbol{\theta}) - g(x_{b,2}^{(i)})\right|^2,$$

$\mathcal{L}_{\mathcal{F}}$ convergeres faster than $\mathcal{L}_{\mathcal{B}}$, leading to solutions which does not satisfy the BCs

Hard-Constraint Methods

$$\hat{u}(\boldsymbol{x}; \boldsymbol{\theta}) = u^{\partial\Omega}(\boldsymbol{x}) + l^{\partial\Omega}(\boldsymbol{x})\mathrm{NN}(\boldsymbol{x}; \boldsymbol{\theta}),$$

$$l^{\partial\Omega}(\boldsymbol{x}) = \begin{cases} 0 & \text{if } \boldsymbol{x} \in \partial\Omega, \\ > 0 & \text{otherwise.} \end{cases}$$

$$\hat{u}(x_1, x_2; \boldsymbol{\theta}) = g(x_2) + x_1 \mathrm{NN}(x_1, x_2; \boldsymbol{\theta}).$$

hard to directly extend this method to cases of Robin BCs, $u^{\partial\Omega}(\boldsymbol{x})$ is hard to obtain

Problem Setup

$$\mathcal{F}[\boldsymbol{u}(\boldsymbol{x})] = \mathbf{0}, \qquad \boldsymbol{x} = (x_1, \ldots, x_d) \in \Omega,$$

$$\mathcal{F} = (\mathcal{F}_1, \ldots, \mathcal{F}_N) \qquad \boldsymbol{u}(\boldsymbol{x}) = (u_1(\boldsymbol{x}), \ldots, u_n(\boldsymbol{x}))$$

for each $u_j, j = 1, \ldots, n$, suitable boundary conditions (BCs)

$$a_i(\boldsymbol{x})u_j + b_i(\boldsymbol{x})(\boldsymbol{n}(\boldsymbol{x}) \cdot \nabla u_j) = g_i(\boldsymbol{x}), \quad \boldsymbol{x} \in \gamma_i, \quad \forall i = 1, \ldots, m_j,$$

Dirichlet BC if $a_i \equiv 1$; $b_i \equiv 0$, a Neumann BC if $a_i \equiv 0$; $b_i \equiv 1$, and a Robin BC otherwise.

at least $(\sum_{j=1}^{n} m_j + N)$ terms

Reformulating PDEs via Extra Fields

$$p_j(x) = (p_{j1}(x), \cdots, p_{jd}(x)) = \nabla u_j, j = 1, \cdots, n$$

$$\tilde{\mathcal{F}}[\boldsymbol{u}(\boldsymbol{x}), \boldsymbol{p}_1(\boldsymbol{x}), \ldots, \boldsymbol{p}_n(\boldsymbol{x})] = \mathbf{0}, \qquad \boldsymbol{x} \in \Omega,$$

$$\boldsymbol{p}_j(\boldsymbol{x}) = \nabla u_j, \qquad \boldsymbol{x} \in \Omega \cup \partial\Omega, \qquad \forall j = 1, \ldots, n,$$

$$a_i(\boldsymbol{x})u_j + b_i(\boldsymbol{x})(\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{p}_j(\boldsymbol{x})) = g_i(\boldsymbol{x}), \qquad \boldsymbol{x} \in \gamma_i, \qquad \forall i = 1, \ldots, m_j.$$

$$\begin{aligned} \Delta u(x) &= -a^2 \sin ax, && x \in (0, 2\pi), \\ u(x) &= 0, && x = 0 \vee x = 2\pi, \end{aligned}$$

$$\begin{aligned} \nabla p(x) &= -a^2 \sin ax, && x \in (0, 2\pi), \\ p(x) &= \nabla u(x), && x \in (0, 2\pi), \\ u(x) &= 0, && x = 0 \vee x = 2\pi. \end{aligned}$$

$$a_i(\boldsymbol{x}) u_j + b_i(\boldsymbol{x}) \big(\boldsymbol{n}(\boldsymbol{x}) \cdot \boldsymbol{p}_j(\boldsymbol{x}) \big) = g_i(\boldsymbol{x}), \qquad \boldsymbol{x} \in \gamma_i, \qquad \forall i = 1, \ldots, m_j.$$

⇓ carefully chosen basis ***B**(x)*

$$\tilde{\boldsymbol{p}}_j^{\gamma_i}(\boldsymbol{x}; \boldsymbol{\theta}_j^{\gamma_i}) = \boldsymbol{B}(\boldsymbol{x}) \mathrm{NN}_j^{\gamma_i}(\boldsymbol{x}; \boldsymbol{\theta}_j^{\gamma_i}) + \tilde{\boldsymbol{n}}(\boldsymbol{x}) \tilde{g}_i(\boldsymbol{x}),$$

A Unified Hard-Constraint Framework

$$(\hat{u}_j, \hat{\boldsymbol{p}}_j) = l^{\partial\Omega}(\boldsymbol{x}) \mathrm{NN}_{\mathrm{main}}(\boldsymbol{x}; \boldsymbol{\theta}_{\mathrm{main}}) + \sum_{i=1}^{m_j} \exp\left[-\alpha_i l^{\gamma_i}(\boldsymbol{x}) \right] \tilde{\boldsymbol{p}}_j^{\gamma_i}(\boldsymbol{x}; \boldsymbol{\theta}_j^{\gamma_i}), \quad \forall j = 1, \ldots, n$$

$$\alpha_i = \frac{\beta_s}{\min_{\boldsymbol{x} \in \partial\Omega \setminus \gamma_i} l^{\gamma_i}(\boldsymbol{x})},$$

final loss function

$$\mathcal{L} = \frac{1}{N_f}\sum_{k=1}^{N_f}\sum_{j=1}^{N}\left|\tilde{\mathcal{F}}_j[\hat{\boldsymbol{u}}(\boldsymbol{x}^{(k)}), \hat{\boldsymbol{p}}_1(\boldsymbol{x}^{(k)}), \ldots, \hat{\boldsymbol{p}}_n(\boldsymbol{x}^{(k)})]\right|^2 + \frac{1}{N_f}\sum_{k=1}^{N_f}\sum_{j=1}^{n}\left\|\hat{\boldsymbol{p}}_j(\boldsymbol{x}^{(k)}) - \nabla\hat{u}_j(\boldsymbol{x}^{(k)})\right\|_2^2,$$

why it works?

- lower derivatives result in less accumulation of back-propagation, and thus stabilize the training process
- reduce the number of loss terms, alleviating the unbalanced competition between loss terms

evaluation

mean absolute error (MAE) and mean absolute percentage error (MAPE)
replace MAPE with weighted mean absolute percentage error (WMAPE)
to avoid the "division by zero"

$$\mathrm{WMAPE} = \frac{\sum_{i=1}^{n} |\hat{y}_i - y_i|}{\sum_{i=1}^{n} |y_i|},$$

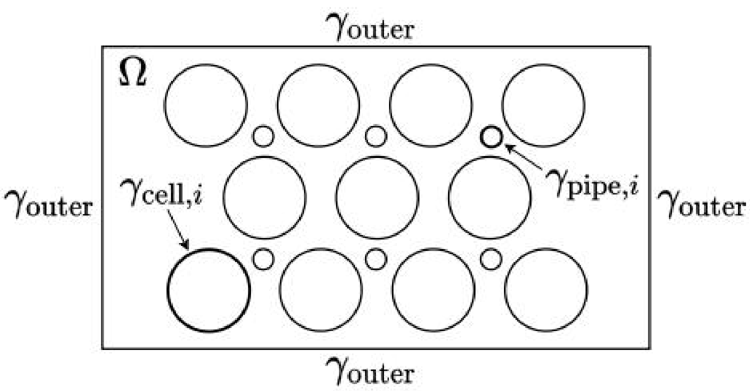
baselines

- PINN
- PINN-LA & PINN-LA-2
- xPINN& FBPINN
- PFNN & PFNN-2

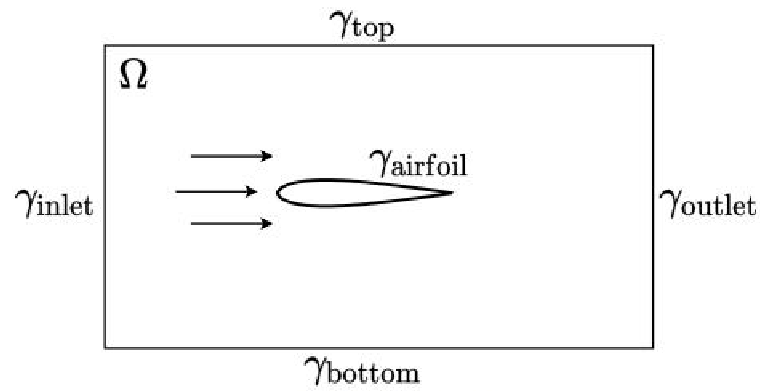
dataset

two real-world problems and high-dimensional equation

Experiments



(a) A 2D battery pack



(b) An airfoil

Table 1: Experimental results of the simulation of a 2D battery pack

	MAE of T				MAPE of T			
	$t=0$	$t=0.5$	$t=1$	average	$t=0$	$t=0.5$	$t=1$	average
PINN	0.1283	0.0457	0.0287	0.0539	128.21%	11.65%	4.47%	24.82%
PINN-LA	0.0918	0.0652	0.0621	0.0661	91.72%	19.13%	11.96%	27.06%
PINN-LA-2	0.1062	0.0321	**0.0211**	0.0402	106.05%	8.94%	4.09%	19.76%
FBPINN	0.0704	0.0293	0.0249	0.0343	70.33%	8.13%	5.87%	14.74%
xPINN	0.2230	0.1295	0.1515	0.1454	222.83%	30.28%	20.25%	54.70%
PFNN	**0.0000**	0.3036	0.4308	0.2758	**0.02%**	79.64%	84.60%	68.29%
PFNN-2	**0.0000**	0.3462	0.5474	0.3215	**0.02%**	66.06%	90.21%	59.62%
HC	**0.0000**	**0.0246**	0.0225	**0.0221**	**0.02%**	**5.38%**	**3.77%**	**5.10%**

Table 2: Experimental results of the simulation of an airfoil

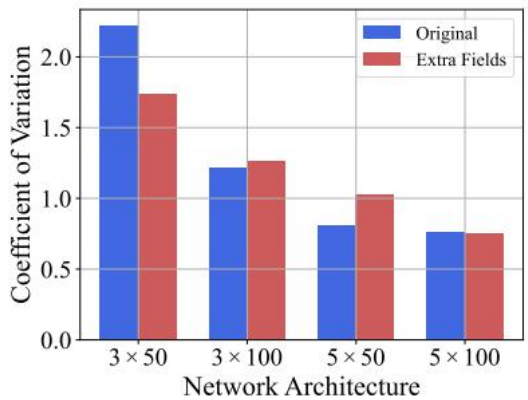
	MAE			WMAPE		
	u_1	u_2	p	u_1	u_2	p
PINN	0.4682	0.0697	0.3883	0.5924	1.1979	0.3539
PINN-LA	0.4018	0.0595	0.2652	0.5084	1.0225	0.2418
PINN-LA-2	0.5047	0.0659	0.2765	0.6385	1.1325	0.2521
FBPINN	0.4058	0.0563	0.2665	0.5134	0.9676	0.2429
xPINN	0.7188	0.0583	1.1708	0.9095	1.0029	1.0672
HC	**0.2689**	**0.0435**	**0.2032**	**0.3402**	**0.7474**	**0.1852**

Table 3: Experimental results of the high-dimensional heat equation

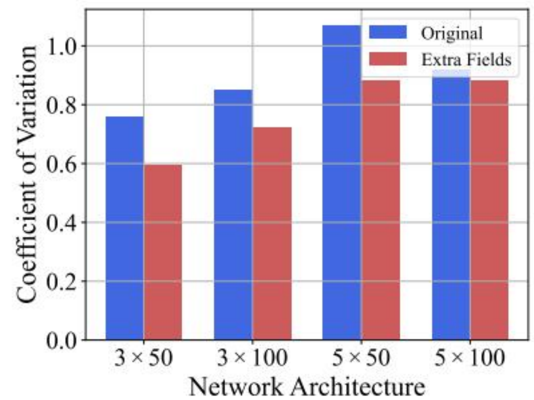
	MAE of u				MAPE of u			
	$t = 0$	$t = 0.5$	$t = 1$	average	$t = 0$	$t = 0.5$	$t = 1$	average
PINN	0.0219	0.0428	0.1687	0.0582	1.43%	1.70%	4.04%	1.99%
PINN-LA	0.0085	0.0149	0.0727	0.0235	0.55%	0.59%	1.73%	0.78%
PINN-LA-2	0.0122	0.0274	0.1495	0.0466	0.79%	1.08%	3.57%	1.49%
PFNN	**0.0000**	0.1253	0.3367	0.1425	**0.00%**	5.02%	8.19%	4.64%
HC	**0.0000**	**0.0029**	**0.0043**	**0.0026**	**0.00%**	**0.12%**	**0.11%**	**0.10%**

Ablation study

Extra fields



(a) Poisson's equation



(b) Schrödinger equation

Hyper-parameters of Hardness

Table 4: The MAE / MAPE of T on different β_s and β_t

	$\beta_t = 1$	$\beta_t = 2$	$\beta_t = 5$	$\beta_t = 10$
$\beta_s = 1$	0.3492 / 48.21%	0.3539 / 48.56%	0.3226 / 44.64%	0.2889 / 40.69%
$\beta_s = 2$	0.2800 / 40.30%	0.1670 / 26.16%	0.2140 / 31.72%	0.1619 / 25.20%
$\beta_s = 5$	0.1878 / 28.68%	0.1195 / 19.68%	0.0542 / 10.35%	**0.0221 / 5.10%**
$\beta_s = 10$	0.1896 / 29.15%	0.1104 / 18.70%	0.0517 / 10.56%	0.0329 / 8.15%