Three-dimensional Collocation-based Robust Variational Physics Informed Neural Networks

Tomasz Służalec[1][0000-0001-6217-4274], Marcin Łoś[1][0000-0002-8426-6345], and Maciej Paszyński[1][0000-0001-7766-6052]

AGH University of Krakow, Poland
sluzalec@agh.edu.pl, los@agh.edu.pl, paszynsk@agh.edu.pl

Abstract. Physics Informed Neural Networks (PINNs) are an increasingly popular approach to utilizing tools and infrastructure developed for neural network training to solve PDEs. While the mainstream approach is based on strong formulations, Variational PINNs (VPINNs) have been proposed to tackle problems with lower regularity. Their issues with robustness, which manifest as a disconnect between the value of the loss function used for training and the error in the relevant Sobolev norm , can be alleviated by employing Robust Variational PINNs (RVPINNs) at the expense of efficiency due to the cost of integrating terms involving the neural network and factorizing the Gram matrix. Collocation-based Robust Variational PINNs (CRVPINN) aim to regain the efficiency of PINNs while retaining robustness by applying the RVPINN framework to variational formulations based on discrete grids and finite difference approximations. Here, we extend the CRVPINN method to 3D and validate it on Poisson and advection-diffusion problems.

Keywords: Collocation method · Robust loss · Physics Informed Neural Networks · Advection-diffusion problem

1 Introduction

The rapid development of deep learning has significantly influenced numerical methods for solving Partial Differential Equations (PDEs) (see, e.g., [3], [5]). Physics-Informed Neural Networks (PINNs) [7] incorporate physical constraints into neural network training by minimizing strong-form residuals at collocation points. While flexible and mesh-free, classical PINNs often suffer from stability and convergence issues, which become more pronounced in three-dimensional problems due to increased computational costs and ill-conditioning. Variational Physics-Informed Neural Networks (VPINNs) [4] address some of these limitations by constructing loss functions based on weak formulations. However, the original VPINN framework lacks robustness at the discrete level. Robust VPINNs (RVPINNs) [8] resolve this issue by introducing Gram-matrix-weighted norms that ensure consistency between discrete and continuous formulations. Despite their theoretical soundness, RVPINNs require costly numerical

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_53

integration and dense matrix operations, limiting their scalability in higher dimensions. In [12], we introduced Collocation-Based Robust Variational PINNs (CRVPINNs), combining point-collocation with a Gram-matrix-weighted loss and LU-based solver strategy. The method was validated in two spatial dimensions on Laplace, advection–diffusion, Stokes, Navier–Stokes, and linear elasticity problems, demonstrating improved efficiency while preserving robustness.

We extend the CRVPINN framework to 3D PDEs. By discretizing the weak formulation directly at collocation points, we obtain a sparse Gram matrix resembling finite-difference operators [10]. Its one-time LU factorization enables efficient evaluation of the robust loss via forward and backward substitutions, yielding a scalable and stable formulation suitable for 3D problems.

2 Mathematical formulation of 3D CRVPINN

We extend the CRVPINN method [12] to 3D. Let us consider the unit cube $\Omega = (0,1)^3$ and a set of uniformly distributed collocation points

$$\Omega_h \coloneqq \{(ih, jh, kh) : 0 \leq i, j, k \leq N\}, \tag{1}$$

where $h = 1/N$ is the point spacing for some fixed grid size N. Our discrete formulations will be built using subspaces of the space of real functions on Ω_h:

$$D_h \coloneqq \{u \colon \Omega_h \to \mathbb{R}\} \cong \mathbb{R}^{(N+1)^3}. \tag{2}$$

Mimicking the continuous theory, we can endow it with a scalar product and its induced norm:

$$(u,v)_h \coloneqq h^3 \sum_{\boldsymbol{x} \in \Omega_h} u(\boldsymbol{x}) v(\boldsymbol{x}), \qquad \|u\|_h^2 \coloneqq (u,u)_h \,. \tag{3}$$

These make D_h a $(N+1)^3$-dimensional Hilbert space with a natural orthogonal basis $\left\{\delta^{ijk}\right\}$ given by

$$\delta^{ijk}(\boldsymbol{x}) = \begin{cases} 1 & \text{if } \boldsymbol{x} = (ih, jh, kh) \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

To simplify notation, we shall write u_{ijk} to denote $u(ih, jh, kh)$. Derivatives are replaced with finite difference approximations:

$$\begin{gathered} \nabla^+ u_{ijk} \coloneqq \left(\nabla_x^+ u_{ijk}, \nabla_y^+ u_{ijk}, \nabla_z^+ u_{ijk}\right), \; \nabla^- u_{ijk} \coloneqq \left(\nabla_x^- u_{ijk}, \nabla_y^- u_{ijk}, \nabla_z^- u_{ijk}\right) \\ Dcu_{ijk} \coloneqq \left(\nabla_x^c u_{ijk}, \nabla_y^c u_{ijk}, \nabla_z^c u_{ijk}\right) \end{gathered} \tag{5}$$

where

$$\begin{aligned} \nabla_x^+ u_{ijk} &= \frac{u_{i+1,j,k} - u_{ijk}}{h}, & \nabla_x^- u_{ijk} &= \frac{u_{ijk} - u_{i-1,j,k}}{h}, \\ \nabla_y^+ u_{ijk} &= \frac{u_{i,j+1,k} - u_{ijk}}{h}, & \nabla_y^- u_{ijk} &= \frac{u_{ijk} - u_{i,j-1,k}}{h}, \\ \nabla_z^+ u_{ijk} &= \frac{u_{i,j,k+1} - u_{ijk}}{h}, & \nabla_z^- u_{ijk} &= \frac{u_{ijk} - u_{i,j,k-1}}{h}, \\ \nabla_\alpha^c u &= \frac{1}{2}\left(\nabla_\alpha^+ u + \nabla_\alpha^- u\right) \end{aligned} \tag{6}$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_53

for these indices (i, j, k) for which the right-hand side is defined, and zero otherwise. We can define a degenerate scalar product and a corresponding seminorm, mimicking the continuous H_0^1 concepts:

$$\begin{aligned}(u,v)_{\nabla,h} &\coloneqq \left(\nabla_x^+ u, \nabla_x^+ v\right)_h + \left(\nabla_y^+ u, \nabla_y^+ v\right)_h + \left(\nabla_z^+ u, \nabla_z^+ v\right)_h \\ \|u\|_{\nabla,h}^2 &\coloneqq (u,u)_{\nabla,h} = \left\|\nabla_x^+ u\right\|_{\nabla,h}^2 + \left\|\nabla_y^+ u\right\|_{\nabla,h}^2 + \left\|\nabla_z^+ u\right\|_{\nabla,h}^2 .\end{aligned} \tag{7}$$

When restricted to the subspace

$$D_{0,h} \coloneqq \left\{u \in D_h : u|_{\partial\Omega_h} = 0\right\}, \quad \partial\Omega_h \coloneqq \partial\Omega \cap \Omega_h \tag{8}$$

of functions that vanish on the discrete boundary points, these two become a scalar product and a norm, respectively.

Advection-diffusion formulation. In our numerical examples, we solve the advection-diffusion equations of the form

$$-\epsilon \Delta u + \beta \cdot \nabla u = f \tag{9}$$

with Dirichlet boundary conditions. The corresponding discrete problem posed in D_h spaces can be formulated as: find $u \in D_{0,h}$ such that

$$-\epsilon \Delta_h u + \beta \cdot \nabla^c u = f \tag{10}$$

where $\nabla^c = \frac{1}{2}\left(\nabla^+ + \nabla^-\right)$ is the central difference operator. By forming $(\cdot,\cdot)_h$-product with testing functions $v \in D_{0,h}$, we obtain a discrete variational formulation: find $u \in D_{0,h}$ such that

$$\underbrace{\epsilon\left(\nabla^+ u, \nabla^+ v\right)_h + \left(\beta \cdot \nabla^c u, v\right)_h}_{b(u,v)} = \underbrace{(f,v)_h}_{l(v)} \tag{11}$$

holds for all $v \in D_{0,h}$. We can prove that the above problem is well-posed by showing that b is bounded and coercive with respect to the $\|\cdot\|_{\nabla,h}$ norm and invoking the Lax-Milgram theorem [1].

Collocation Variational Physics Informed Neural Networks. In the PINN approach, the neural network is used as the direct representation of the solution of a PDE. We seek to solve scalar problems posed on a three-dimensional domain, so our neural network accepts a vector (x, y, z) as input and outputs a single value:

$$u_\theta(x,y,z) = NN(x,y,z) = \sigma_n(A_n\sigma_{n-1}(\cdots\sigma_1(A_1\begin{bmatrix}x\\y\\z\end{bmatrix} + \beta_1)\cdots) + \beta_n) \tag{12}$$

where A are layer weights, σ are nonlinear activation functions, and β are biases. Following the CRVPINN approach introduced in [12], we train the neural network by minimizing a loss function constructed as the norm of the Riesz

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_53

representative of the residual of the variational formulation from eq. (11). More precisely, the loss function is defined as

$$\mathcal{L}(\theta) = \|r(\theta)\|^2_{\nabla,h} \tag{13}$$

where $r(\theta) \in D_{0,h}$ is the unique element such that $(r(\theta), v)_{\nabla,h} = b(u_\theta, v) - l(v)$ for all $v \in D_{0,h}$. Applying the theory from [9] to the variational formulation eq. (11), we obtain the following bounds on the true error in terms of value of the loss function. A practical way of evaluating the robust loss is by computing $\mathcal{L}(\theta) = \boldsymbol{r}(\theta)^T \boldsymbol{G} \boldsymbol{r}(\theta) = \mathrm{RES}(\theta)^T \boldsymbol{G}^{-1} \mathrm{RES}(\theta)$. The following theorem highlights what robustness means here.

Theorem 1 (Robustness). *Let the loss function $\mathcal{L}$ be given by eq. (13). Then*

$$\frac{1}{\mu}\sqrt{\mathcal{L}(\theta)} \leq \|u_\theta - u_{EXACT}\|_{\nabla,h} \leq \frac{1}{\alpha}\sqrt{\mathcal{L}(\theta)}, \tag{14}$$

where u_{EXACT} denotes the exact solution of the variational formulation eq. (11), and μ, α are the boundedness and coercivity constants of b.

Let $\boldsymbol{r}(\theta)$ denote the vector of coefficients of $r(\theta)$, and $\mathrm{RES}(\theta)$ denote the vector of values of $b(u_\theta, v) - l(v)$, where v ranges over the basis functions of $D_{0,h}$. We can compute $\boldsymbol{r}(\theta)$ by solving

$$\boldsymbol{G}\boldsymbol{r}(\theta) = \mathrm{RES}(\theta) \tag{15}$$

where $\boldsymbol{G}$ is the Gram matrix of the scalar product that makes b bounded and coercive. In the case of advection-diffusion equations, $(\cdot,\cdot)_{\nabla,h}$ satisfies that condition, so we employ the Gram matrix of this scalar product in our numerical tests. Since

$$\begin{aligned}
\left(u, \delta^{ijk}\right)_{\nabla,h} &= \left(\nabla_x^+ u, \nabla_x^+ \delta^{ijk}\right)_h = -\left(\nabla_x^- \left(\nabla_x^+ u\right), \delta^{ijk}\right)_h \\
&= -\left(\Delta_h u, \delta^{ijk}\right)_h = -h^3 \left(\Delta_h u\right)_{ijk} \\
&= h\big[6u_{ijk} - (u_{i+1,j,k} + u_{i-1,j,k} + u_{i,j+1,k} \\
&\quad + u_{i,j-1,k} + u_{i,j,k+1} + u_{i,j,k-1})\big]
\end{aligned} \tag{16}$$

the entries of the Gram matrix can be computed as

$$G^{pqr}_{ijk} = h \begin{cases} 6 & \text{if } (i,j,k) = (p,q,r) \\ -1 & \text{if } p = i \pm 1,\ (q,r) = (j,k) \\ -1 & \text{if } q = j \pm 1,\ (p,r) = (i,k) \\ -1 & \text{if } r = k \pm 1,\ (p,q) = (i,j) \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

Then, the loss function can be computed as

$$\mathcal{L}(\theta) = \boldsymbol{r}(\theta)^T \boldsymbol{G} \boldsymbol{r}(\theta) = \mathrm{RES}(\theta)^T \boldsymbol{G}^{-1} \mathrm{RES}(\theta) \tag{18}$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_53

To achieve efficient evaluation, we assemble the Gram matrix and compute its Cholesky factorization before the training process. Since the matrix does not change during training, this is a one-time upfront cost. Furthermore, since $\boldsymbol{G}$ is symmetric and positive definite, factorizing it is easier than factorizing the matrix of the bilinear form b.

Three-dimensional Laplace problem with sine functions as the right hand side.

In the strong formulation, we seek a solution u on $\Omega \subset (0,1)^3$ such that

$$-\Delta u = f \tag{19}$$

In our examples, we will calculate the manufactured solution with Dirichlet boundary conditions. For a given $u(x,y,z) = \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$, we obtain the right hand side

$$f(x,y,z) = -\Delta u(x,y,z) = -12\pi^2 \sin(2\pi x)\sin(2\pi y)\sin(2\pi z) \tag{20}$$

Then our residual in the CRVPINN is a vector defined as follows:

$$\mathrm{RES}(\Theta)_{ijk} = \Delta u(\boldsymbol{x}_{ijk}) + f(\boldsymbol{x_{ijk}}), \quad 0 < i,j,k < N. \tag{21}$$

The last layer of the neural net is the calculation presented in [11], which enforces Dirichlet boundary conditions. The loss in the CRVPINN is calculated as (18).

Three-dimensional Laplace problem with exp-sin-sin as the right-hand side. The second model problem taken into account is related to the Laplace problem eq. (19). We construct the manufactured solution that includes the sine and exponential functions $u(x,y,z) = -e^{\pi(x-2y)} sin(2\pi x) sin(\pi y) sin(\pi t)$. The interpretation is that we have a peak spot with high values in one corner of the cube. Then we can calculate the right hand side of the problem, $f = \Delta u$, namely

$$f = e^{\pi(x-2y)}\pi^2(4\cos(\pi y)\sin(2\pi x) + (-4\cos(2\pi x) + \sin(2\pi x))\sin(\pi y))\sin(\pi z) \tag{22}$$

Then, the residual is defined this way

$$\mathrm{RES}(\Theta)_{ijk} = \Delta u(\boldsymbol{x}_{ijk}) + f(\boldsymbol{x_{ijk}}), \quad 0 < i,j,k < N. \tag{23}$$

The computation of the CRVPINN loss is as follows (18).

Eriksson-Johnson model problem [2]. We consider the advection vector in the y direction; then $\beta = [0,1,0]$, $\epsilon = 0.1$, and our formula in 3D is as follows:

$$\frac{\partial u}{\partial y} + \epsilon\frac{\partial^2 u}{\partial x^2} + \epsilon\frac{\partial^2 u}{\partial y^2} + \epsilon\frac{\partial^2 u}{\partial y^2} = 0 \tag{24}$$

We also define the specific boundary conditions. There are zero Dirichlet boundary conditions on 5 planes, and there is a source of pollution on one plane.

$$f_3 = \begin{cases} \sin(\pi x)\sin(\pi y), & \text{on} \quad (x,y,0) \quad x \in (0,1), y \in (0,1) \\ 0, & \text{on} \quad (x,0,z),(0,y,z) \quad x,y,z \in (0,1) \\ 0, & \text{on} \quad (1,y,z),(x,1,z),(x,y,1) \quad x,y,z \in (0,1) \end{cases} \tag{25}$$

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_53

We introduce the shift for the boundary conditions.

$$u_{\text{shift}}(x,y,z) = \sin(\pi x)(1-y)\sin(\pi z) \tag{26}$$

We can clearly see that $u(x,y,z) - u_{\text{shift}}(x,y,z) = 0 \quad \text{on } \partial\Omega$. We will find a solution for the shifted u, namely $NN(x,t,z) = u(x,y,z) - u_{\text{shift}}(x,y,z)$, so our neural net approximating the solution can be enforced to have zero Dirichlet boundary conditions [11]. Now we can define the residual for $0 < i,j,k < N$,

$$\text{RES}(\Theta)_{ijk} = \frac{\partial u}{\partial y}(\boldsymbol{x_{ijk}}) - \epsilon \Delta u(\boldsymbol{x_{ijk}}) + \frac{\partial u_{\text{shift}}}{\partial y}(\boldsymbol{x_{ijk}}) - \epsilon \Delta u_{\text{shift}}(\boldsymbol{x_{ijk}}). \tag{27}$$

Our loss for this problem also comes in the form of (18).The exact solution to this problem is as follows.

$$u_{\text{exact}}(x,y,z) = \frac{e^{\alpha_1(1-y)-e^{\alpha_2(1-y)}}}{e^{-\alpha_1}-e^{-\alpha_2}}\sin(\pi x)\sin(\pi y) \tag{28}$$

where $\alpha_1 = \frac{1-\sqrt{1+(4\pi\epsilon)^2}}{4\epsilon}, \quad \alpha_2 = \frac{1+\sqrt{1+(4\pi\epsilon)^2}}{4\epsilon}$.

3 Numerical results

The solution was obtained by our CRVPINN code written in python and pytorch. We employ a fully connected neural network with 2 layers and 100 neurons per layer. The input to the neural network is (x,y,z); the output is $u(x,y,z)$. We selected the activation function according to [6]. The last layer applied the pinning technique to obtain specific boundary conditions [11]. We use the Adam optimizer and a different number of epochs in our numerical experiments. Our code can be run on Google Colab Pro with an A100 graphics card to perform fast GPU computations. For the first problem, we used $20\times20\times20$ mesh; for the second and third problems, we used $40\times40\times40$ mesh. To obtain higher accuracy, we need to either increase the number of collocation points or consider non-uniform adaptive distributions, particularly for the Eriksson-Johnson problem.

4 Conclusions

We extended the Collocation Robust Variational Physics-Informed Neural Network framework to 3D problems. The proposed approach preserves the computational efficiency of standard PINNs while significantly improving robustness through a carefully designed loss function. Numerical experiments confirm that the resulting loss provides a reliable indicator of the true discretization error. The proposed methodology is general and can be applied to a wide range of three-dimensional PDEs. For each new problem, the key step is the design of a suitable inner product that reflects the structure of the weak formulation. This positions CRVPINNs as a robust and flexible alternative to standard PINNs, closely related in spirit to residual minimization and Petrov–Galerkin techniques, while remaining well suited for high-dimensional problems.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_53

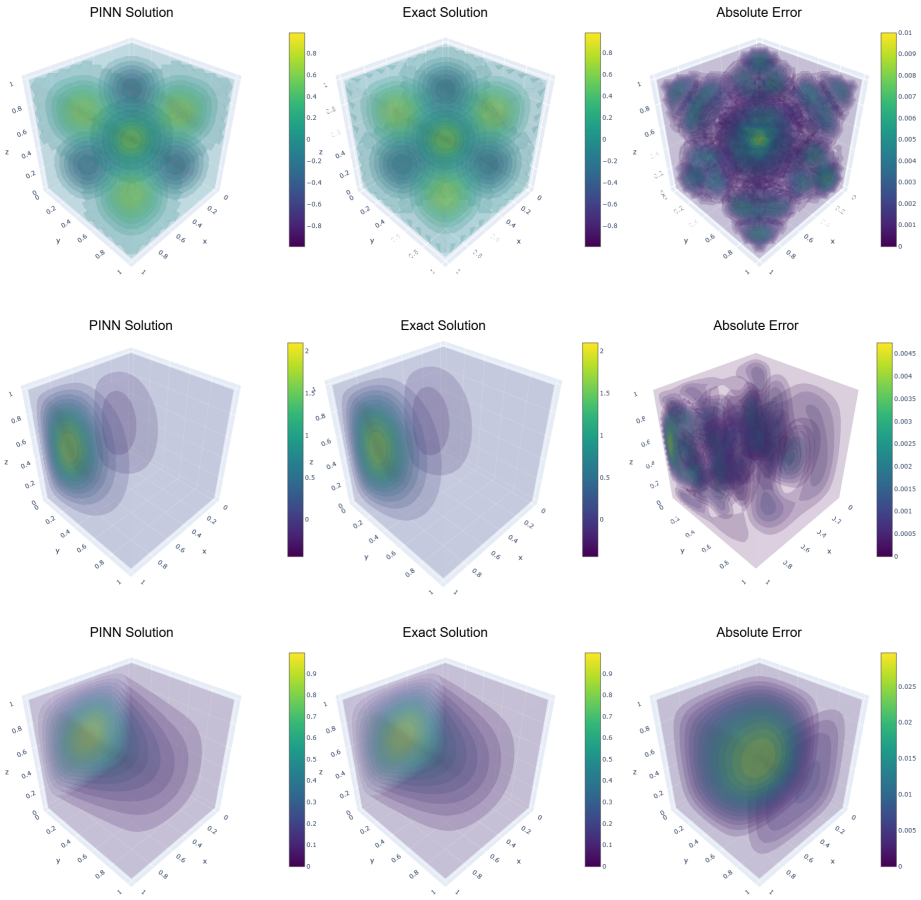

Fig. 1. First row: Laplace problem with sin-sin-sin. Second row: Laplace problem with sin-exp-sin. Third row: Eriksson-Johnson problem. First solumn: CRVPINN solution. Second column: Exact solution. Third column: point-wise error.

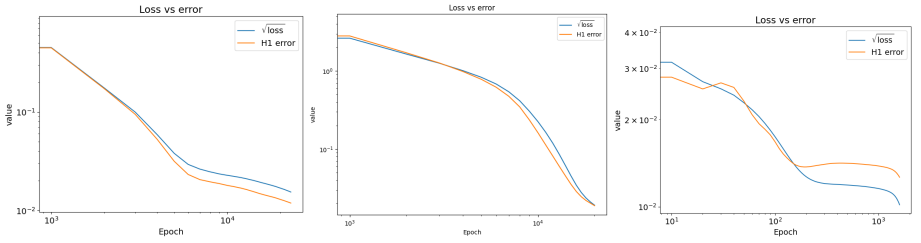

Fig. 2. Convergence of training for the Laplace problem with sin-sin-sin, the Laplace problem with sin-exp-sin, and the Eriksson-Johnson problem. Comparison of the square root of the robust loss and the H^1 norm of the solution (see Theorem 1).

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_53

Acknowledgments This work has been supported by the National Science Centre, Poland grant no. 2025/57/B/ST6/00058.

References

1. Ciarlet, P.G.: The finite element method for elliptic problems. No. v. 4 in Studies in mathematics and its applications, North-Holland Pub. Co. ; sole distributors for the U.S.A. and Canada, Elsevier North-Holland, Amsterdam ; New York : New York (1978)
2. Eriksson, K., Johnson, C.: Adaptive streamline diffusion finite element methods for stationary convection-diffusion problems. Mathematics of Computation **60**(201), 167–188 (1993)
3. Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A.r., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N., et al.: Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. IEEE Signal processing magazine **29**(6), 82–97 (2012)
4. Kharazmi, E., Zhang, Z., Karniadakis, G.E.: Variational physics-informed neural networks for solving partial differential equations. arXiv preprint arXiv:1912.00873 (2019)
5. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Communications of the ACM **60**(6), 84–90 (2017)
6. Maczuga, P., Paszyński, M.: Influence of activation functions on the convergence of physics-informed neural networks for 1d wave equation. In: International Conference on Computational Science. pp. 74–88. Springer (2023)
7. 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)
8. Rojas, S., Maczuga, P., Muñoz-Matute, J., Pardo, D., Paszyński, M.: Robust variational physics-informed neural networks. Computer Methods in Applied Mechanics and Engineering **425**, 116904 (2024)
9. Rojas, S., Maczuga, P., Muñoz-Matute, J., Pardo, D., Paszyński, M.: Robust Variational Physics-Informed Neural Networks. Computer Methods in Applied Mechanics and Engineering **425**, 116904 (May 2024)
10. Shin, D., Strikwerda, J.C.: Inf-sup conditions for finite-difference approximations of the stokes equations. The ANZIAM Journal **39**(1), 121–134 (1997)
11. Sun, L., Gao, H., Pan, S., Wang, J.X.: Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. Computer Methods in Applied Mechanics and Engineering **361**, 112732 (2020)
12. Łoś, M., Służalec, T., Maczuga, P., Vilkha, A., Uriarte, C., Paszyński, M.: Collocation-based robust variational physics-informed neural networks (CRVPINNs). Computers & Structures **316**, 107839 (Sep 2025)

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_53