Domain Decomposition for Randomized Neural Networks

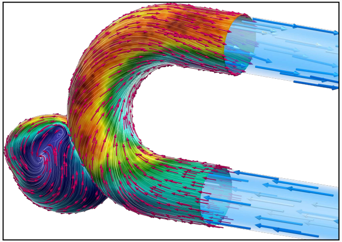
Alexander Heinlein[1]

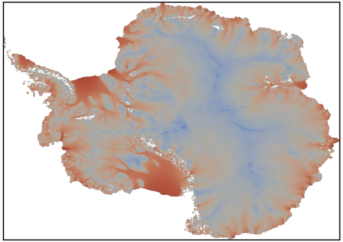
95th Annual Meeting of the International Association of Applied Mathematics and Mechanics (GAMM 2025), Poznan, Poland, April 7-11, 2025

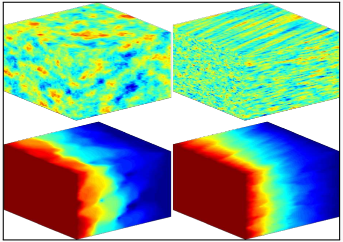
[1]Delft University of Technology

Based on joint work with **Siddhartha Mishra** (ETH Zürich) and **Yong Shang** and **Fei Wang** (Xi'an Jiaotong University)

Scientific Computing and Machine Learning







Numerical methods

Based on physical models

+ Robust and generalizable
− Require availability of mathematical models

Machine learning models

Driven by data

+ Do not require mathematical models
− Sensitive to data, limited extrapolation capabilities

Scientific machine learning

Combining the strengths and **compensating the weaknesses** of the individual approaches:

numerical methods **improve** machine learning techniques
machine learning techniques **assist** numerical methods

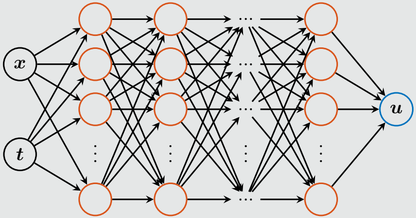
Randomized Neural Networks (RaNNs)

Neural networks

A standard **multilayer perceptron (MLP)** with L hidden layers is a **parametric** model of the form

$$u(\boldsymbol{x}, \boldsymbol{\theta}) = F_{L+1}^{\boldsymbol{A}} \cdot F_L^{\boldsymbol{W}_L, \boldsymbol{b}_L} \circ \ldots \circ F_1^{\boldsymbol{W}_1, \boldsymbol{b}_1}(\boldsymbol{x}),$$

where $\boldsymbol{A}$ is linear, and the ith hidden layer is nonlinear $F_i^{\boldsymbol{W}_i, \boldsymbol{b}_i}(\boldsymbol{x}) = \sigma(\boldsymbol{W}_i \cdot \boldsymbol{x} + \boldsymbol{b}_i)$.



In order to optimize the loss function

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}),$$

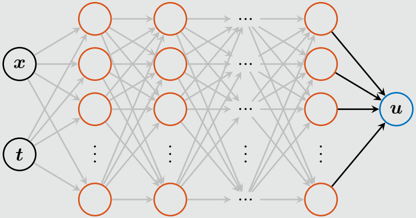
all parameters $\boldsymbol{\theta} = (\boldsymbol{A}, \boldsymbol{W}_1, \boldsymbol{b}_1, \ldots, \boldsymbol{W}_L, \boldsymbol{b}_L)$ are **trained**.

Randomized neural networks

In **randomized neural networks (RaNNs)** as introduced by **Pao and Takefuji (1992)**,

$$u(\boldsymbol{x}, \boldsymbol{A}) = F_{L+1}^{\boldsymbol{A}} \cdot F_L^{W_L, b_L} \circ \ldots \circ F_1^{W_1, b_1}(\boldsymbol{x}),$$

the weights in the hidden layers are randomly initialized and **fixed**; only $\boldsymbol{A}$ is trainable.



The model is linear with respect to the trainable parameters $\boldsymbol{A}$, and the optimization problem reads

$$\min_{\boldsymbol{A}} \mathcal{L}(\boldsymbol{A}).$$

This can **simplify the training process**.

Physics-Informed Neural Networks (PINNs)

In the **physics-informed neural network (PINN)** approach introduced by **Raissi et al. (2019)**, a **neural network** is employed to **discretize a partial differential equation**

$$\mathcal{N}[u] = f, \quad \text{in } \Omega.$$

PINNs use a **hybrid loss function**:

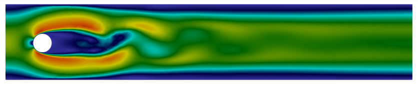
$$\mathcal{L}(\boldsymbol{\theta}) = \omega_{\text{data}} \mathcal{L}_{\text{data}}(\boldsymbol{\theta}) + \omega_{\text{PDE}} \mathcal{L}_{\text{PDE}}(\boldsymbol{\theta}),$$

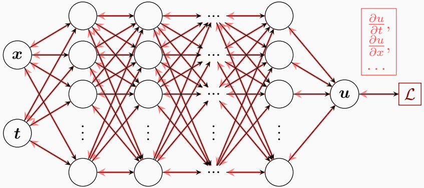
where ω_{data} and ω_{PDE} are **weights** and

$$\mathcal{L}_{\text{data}}(\boldsymbol{\theta}) = \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \left(u(\hat{\mathbf{x}}_i, \boldsymbol{\theta}) - u_i \right)^2,$$

$$\mathcal{L}_{\text{PDE}}(\boldsymbol{\theta}) = \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} \left(\mathcal{N}[u](\mathbf{x}_i, \boldsymbol{\theta}) - f(\mathbf{x}_i) \right)^2.$$

See also **Dissanayake and Phan-Thien (1994); Lagaris et al. (1998)**.





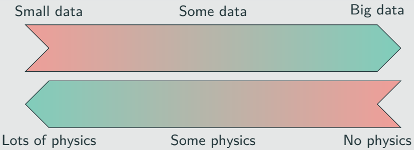
Advantages

- **"Meshfree"**
- **Small data**
- **Generalization properties**
- **High-dimensional problems**
- **Inverse** and **parameterized problems**

Drawbacks

- **Training cost** and **robustness**
- **Convergence not well-understood**
- **Difficulties with scalability** and **multi-scale problems**

Hybrid loss



- **Known solution values** can be included in $\mathcal{L}_{\text{data}}$
- **Initial and boundary conditions** are also included in $\mathcal{L}_{\text{data}}$

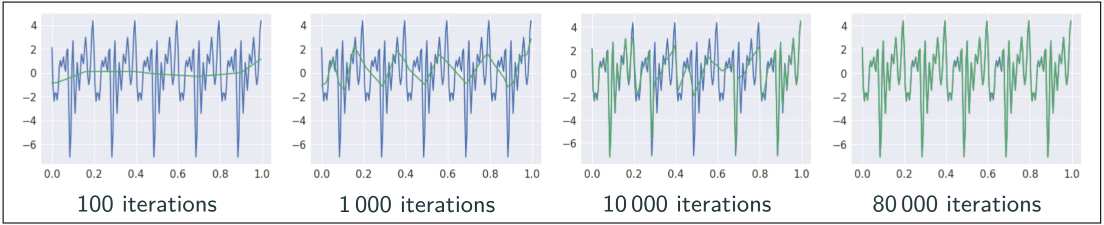
Estimate of the generalization error (Mishra and Molinaro (2022))

The generalization error (or total error) satisfies

$$\mathcal{E}_G \leq C_{\text{PDE}} \mathcal{E}_T + C_{\text{PDE}} C_{\text{quad}}^{1/p} N^{-\alpha/p}$$

- $\mathcal{E}_G = \mathcal{E}_G(\boldsymbol{X}, \theta) := \|\mathbf{u} - \mathbf{u}^*\|_V$ **general. error** (V Sobolev space, $\boldsymbol{X}$ training data set)
- $\mathcal{E}_T$ **training error** (l^p **loss** of the residual of the PDE)
- N **number of the training points** and α **convergence rate of the quadrature**
- C_{PDE} and C_{quad} **constants** depending on the **PDE**, **quadrature**, and **neural network**

Rule of thumb: **"As long as the PINN is trained well, it also generalizes well"**



Rahaman et al., *On the spectral bias of neural networks*, **ICML (2019)**

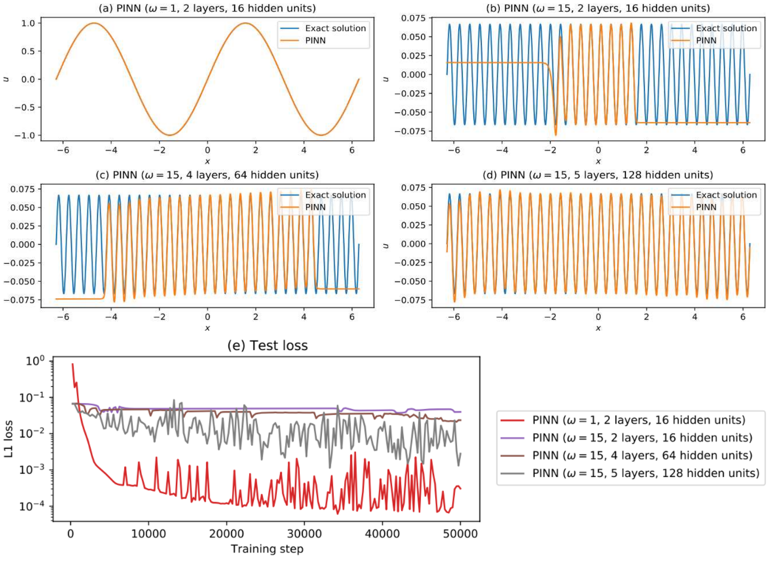
Related works: **Cao et al. (2021)**, **Wang, et al. (2022)**, **Hong et al. (arXiv 2022)**, **Xu et al (2024)**, ...

Scaling of PINNs for a Simple ODE Problem

Solve

$$
\begin{aligned}
u' &= \cos(\omega \boldsymbol{x}), \\
u(0) &= 0,
\end{aligned}
$$

for different values of ω using **PINNs with varying network capacities**.



(a) 321 free parameters

(d) 66 433 free parameters

Scaling issues

- Large computational domains
- Small frequencies

Cf. **Moseley, Markham, and Nissen-Meyer (2023)**

Scaling of PINNs for a Simple ODE Problem

Solve

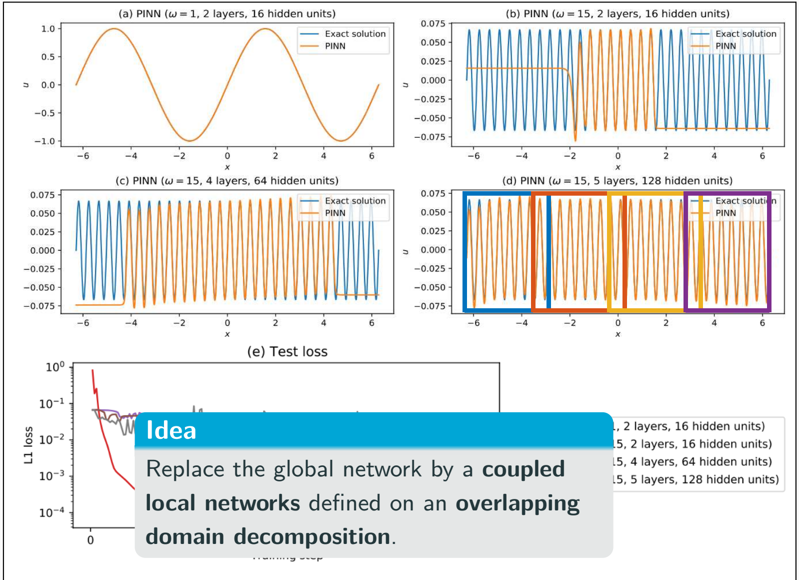
$$
\begin{aligned}
u' &= \cos(\omega \boldsymbol{x}), \\
u(0) &= 0,
\end{aligned}
$$

for different values of ω using **PINNs with varying network capacities**.

Scaling issues

- Large computational domains
- Small frequencies

Cf. Moseley, Markham, and Nissen-Meyer (2023)



Idea

Replace the global network by a **coupled local networks** defined on an **overlapping domain decomposition**.

(a) 321 free parameters

(d) 66 433 free parameters

Finite Basis Physics-Informed Neural Networks (FBPINNs)

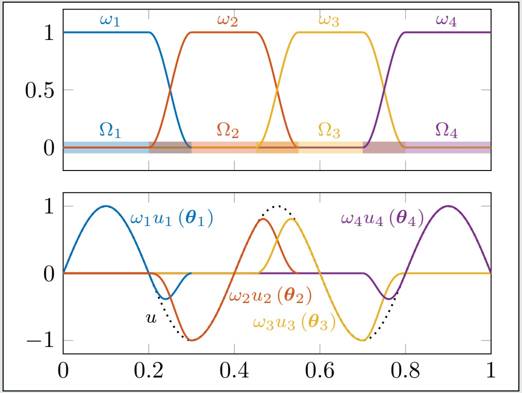
FBPINNs (Moseley, Markham, Nissen-Meyer (2023))

FBPINNs employ the **network architecture**

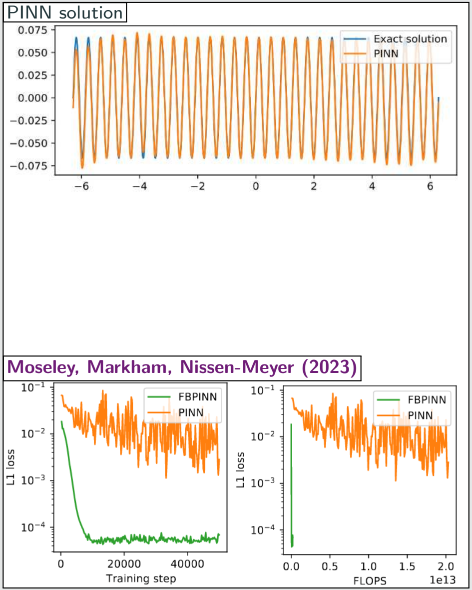
$$u(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_J) = \sum\nolimits_{j=1}^{J} \omega_j u_j(\boldsymbol{\theta}_j)$$

and the **loss function**

$$\mathcal{L} = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left(n[\sum\nolimits_{\mathbf{x}_i \in \Omega_j} \omega_j u_j](\mathbf{x}_i, \boldsymbol{\theta}_j) - f(\mathbf{x}_i) \right)^2.$$



1D single-frequency problem



Finite Basis Physics-Informed Neural Networks (FBPINNs)

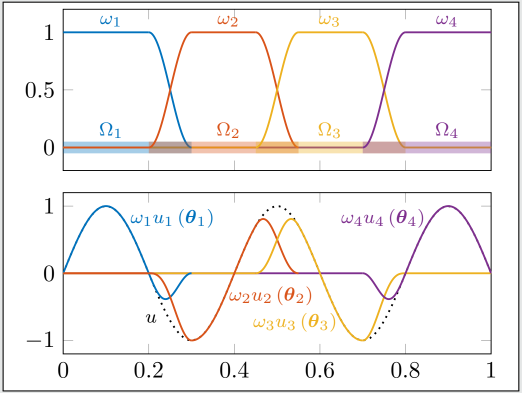
FBPINNs (Moseley, Markham, Nissen-Meyer (2023))

FBPINNs employ the **network architecture**

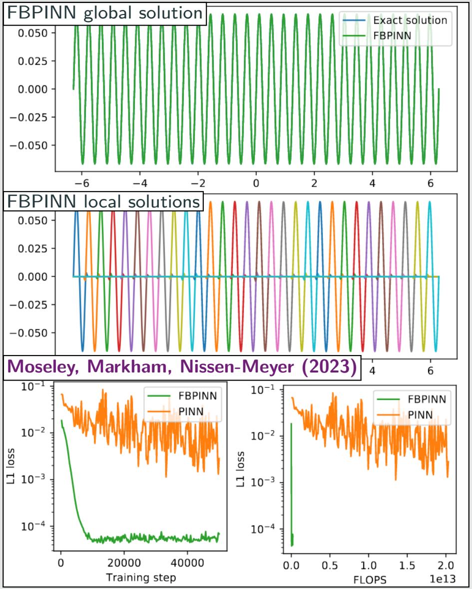
$$u(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_J) = \sum\nolimits_{j=1}^{J} \omega_j u_j\left(\boldsymbol{\theta}_j\right)$$

and the **loss function**

$$\mathcal{L} = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left(n[\sum\nolimits_{\mathbf{x}_i \in \Omega_j} \omega_j u_j](\mathbf{x}_i, \boldsymbol{\theta}_j) - f(\mathbf{x}_i) \right)^2.$$

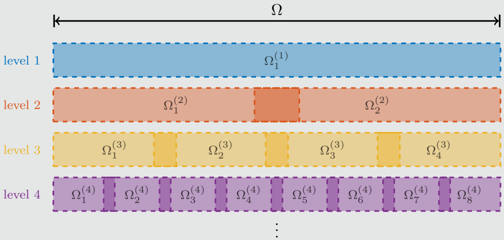


1D single-frequency problem



Multi-level FBPINNs (ML-FBPINNs)

ML-FBPINNs (**Dolean, Heinlein, Mishra, Moseley (2024)**) are based on a **hierarchy of domain decompositions:**



This yields the **network architecture**

$$u\left(\boldsymbol{\theta}_1^{(1)}, \ldots, \boldsymbol{\theta}_{J^{(L)}}^{(L)}\right) = \sum\nolimits_{l=1}^{L} \sum\nolimits_{i=1}^{N^{(l)}} \omega_j^{(l)} u_j^{(l)}\left(\boldsymbol{\theta}_j^{(l)}\right)$$

and the **loss function**

$$\mathcal{L} = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left(n[\sum\nolimits_{\boldsymbol{x}_i \in \Omega_j^{(l)}} \omega_j^{(l)} u_j^{(l)}](\boldsymbol{x}_i, \boldsymbol{\theta}_j^{(l)}) - f(\boldsymbol{x}_i)\right)^2.$$

Multi-Frequency Problem

Let us now consider the **two-dimensional multi-frequency Laplace boundary value problem**

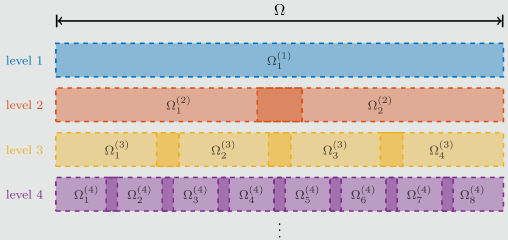
$$\begin{aligned} -\Delta u &= 2 \sum_{i=1}^{n} (\omega_i \pi)^2 \sin(\omega_i \pi x) \sin(\omega_i \pi y) && \text{in } \Omega, \\ u &= 0 && \text{on } \partial\Omega, \end{aligned}$$

with $\omega_i = 2^i$.

For increasing values of n, we obtain the **analytical solutions**:

Multi-level FBPINNs (ML-FBPINNs)

ML-FBPINNs (**Dolean, Heinlein, Mishra, Moseley (2024)**) are based on a **hierarchy of domain decompositions:**



This yields the **network architecture**

$$u\left(\theta_1^{(1)}, \ldots, \theta_{J^{(L)}}^{(L)}\right) = \sum\nolimits_{l=1}^{L} \sum\nolimits_{i=1}^{N^{(l)}} \omega_j^{(l)} u_j^{(l)}\left(\theta_j^{(l)}\right)$$

and the **loss function**

$$\mathcal{L} = \frac{1}{N} \sum\nolimits_{i=1}^{N} \left(n[\sum\nolimits_{x_i \in \Omega_j^{(l)}} \omega_j^{(l)} u_j^{(l)}](\boldsymbol{x}_i, \boldsymbol{\theta}_j^{(l)}) - f(\boldsymbol{x}_i)\right)^2.$$

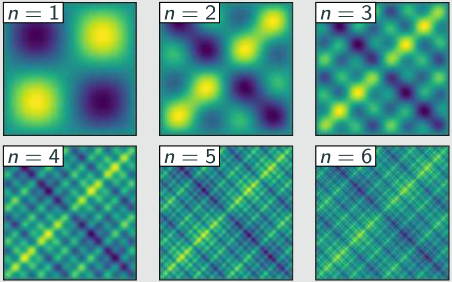
Multi-Frequency Problem

Let us now consider the **two-dimensional multi-frequency Laplace boundary value problem**

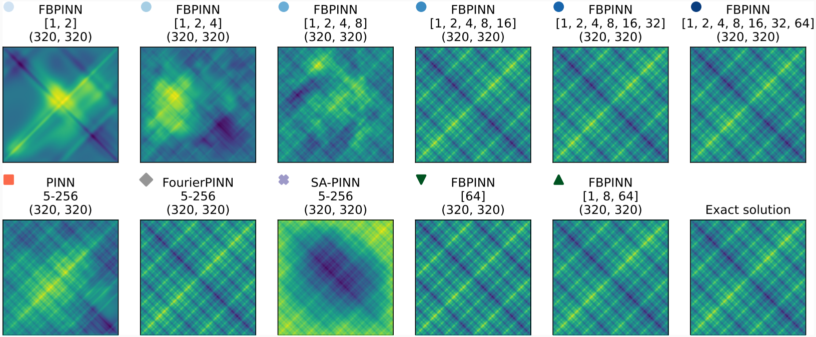
$$\begin{aligned} -\Delta u &= 2 \sum_{i=1}^{n} (\omega_i \pi)^2 \sin(\omega_i \pi x) \sin(\omega_i \pi y) && \text{in } \Omega, \\ u &= 0 && \text{on } \partial\Omega, \end{aligned}$$

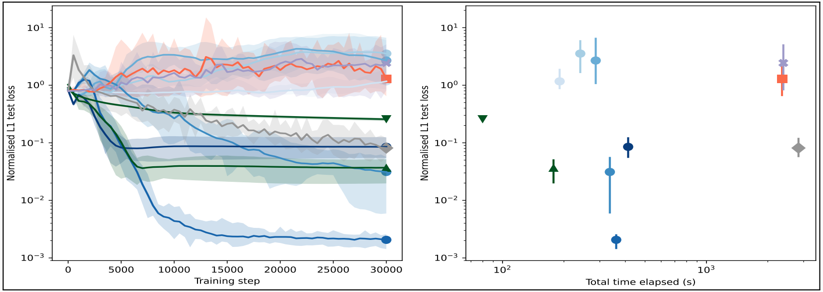
with $\omega_i = 2^i$.

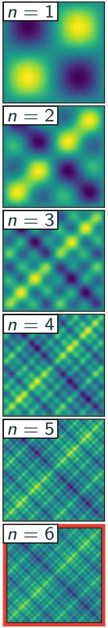
For increasing values of n, we obtain the **analytical solutions**:



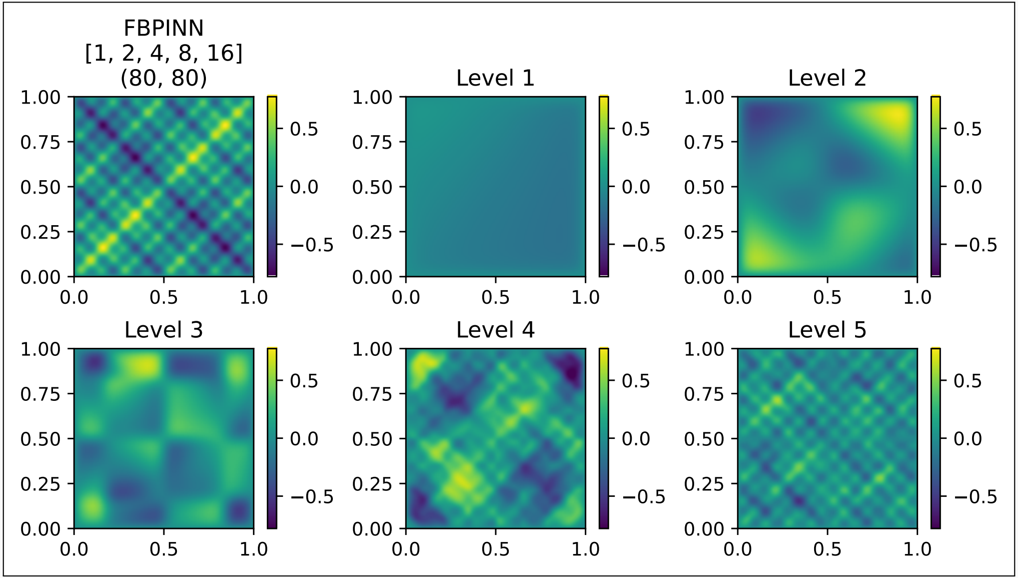
Multi-Level FBPINNs for a Multi-Frequency Problem – Strong Scaling





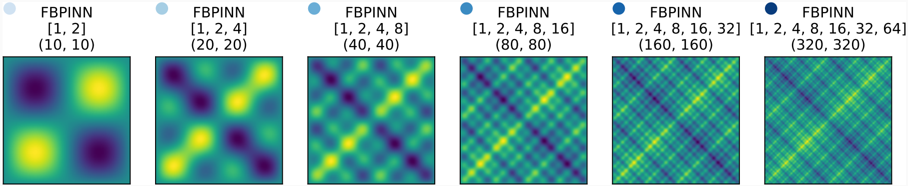


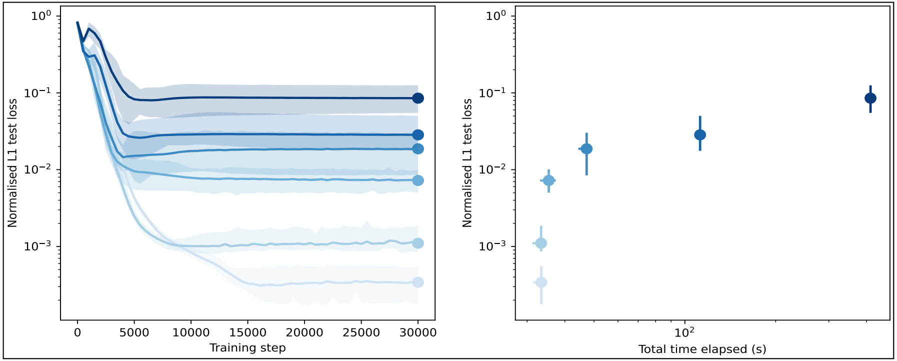
Multi-Frequency Problem – What the FBPINN Learns

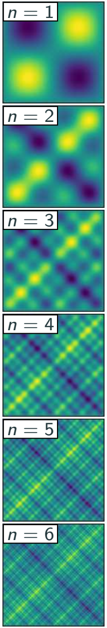


Cf. **Dolean, Heinlein, Mishra, Moseley (2024)**.

Multi-Level FBPINNs for a Multi-Frequency Problem – Weak Scaling







→ Details and results for the Helmholtz equation can be found in **Dolean, Heinlein, Mishra, Moseley (2024)**.

Physics-Informed Randomized Neural Networks (PIRaNNs)

Physics-informed randomized neural networks (PIRaNNs) make use of the aforementioned linearization of the model with respect to the trainable parameters as well as the fact that RaNNs retain **universal approximation properties**, as shown in **Igelnik and Pao (1995)**.

Consider a **linear differential operator** $\mathcal{A}$. Then, solving the PDE

$$\mathcal{A}[u] = f, \quad \text{in } \Omega.$$

using PIRaNNs yields the **linear equation system**

$$\mathcal{A}[u](\boldsymbol{x}_i) = f(\boldsymbol{x}_i), \quad i = 1, \ldots, N_{\text{PDE}},$$

where N_{PDE} is the number of **collocation points**.

Enforcement of boundary conditions

We construct u to explicitly satisfy BCs:

$$u(\boldsymbol{x}, \boldsymbol{A}) = G(\boldsymbol{x}) + L(\boldsymbol{x})\mathcal{N}(\boldsymbol{x}, \boldsymbol{A})$$

- $\mathcal{N}$ is a **feedforward neural network** with **trainable parameters $\boldsymbol{A}$**
- G and L are **fixed functions**, chosen such that u satisfies the boundary conditions

The resulting linear equation system

$$\boldsymbol{H}\boldsymbol{A} = \boldsymbol{f}$$

generally does **not have a unique solution**. In fact, $\boldsymbol{H}$ is typically **rectangular**, **dense**, and **ill-conditioned**.

Solving the system using least squares corresponds to applying the **classical PINN loss function to the RaNN model** u. As we will see, this approach offers a **potentially efficient alternative**.

Randomized Neural Networks – (Non-Exhaustive) Literature Overview

Randomized neural networks

- **RaNNs**: Pao, Takefuji (1992); Pao Park, Sobajic (1994); Igelnik, Pao (1995)
- **Extreme Learning Machines (ELMs)**: Huang, Zhu, Siew (2006); Liu, Lin, Fang, Xu (2014); Gallicchio, Scardapane (2020); Calabrò, Fabiani, Siettos (2021); Ni, Dong (2023); Wang, Dong (2024)

Domain decomposition for neural networks and randomized neural networks

- **cPINNs, XPINNs**: Jagtap, Kharazmi, Karniadakis (2020); Jagtap, Karniadakis (2020)
- **Schwarz it. for PINNs or DeepRitz (D3M, DeepDDM, etc):**: Li, Tang, Wu, Liao (2019); Li, Xiang, Xu (2020); Mercier, Gratton, Boudier (arXiv 2021); Dolean, H., Mercier, Gratton (arXiv 2024); Li, Wang, Cui, Xiang, Xu (2023); Sun, Xu, Yi (arXiv 2023, 2024); Kim, Yang (2023, 2024, 2024)
- **FBPINNs, FBKANs**: Moseley, Markham, Nissen-Meyer (2023); Dolean, H., Mishra, Moseley (2024, 2024); H., Howard, Beecroft, Stinis (acc. 2024); Howard, Jacob, Murphy, H., Stinis (arXiv 2024)
- **DD for RaNNs, ELMS, Random Feature Method**: Dong, Li (2021); Dang, Wang (2024); Sun, Dong, Wang (2024); Sun, Wang (2024); Chen, Chi, E, Yang (2022); Shang, H., Mishra, Wang (acc. 2025)

An overview of the state-of-the-art in 2024:

A. Klawonn, M. Lanser, J. Weber

Machine learning, domain decomposition methods – a survey

Computational Science and Engineering. 2024

Domain Decomposition-Based PIRaNNs

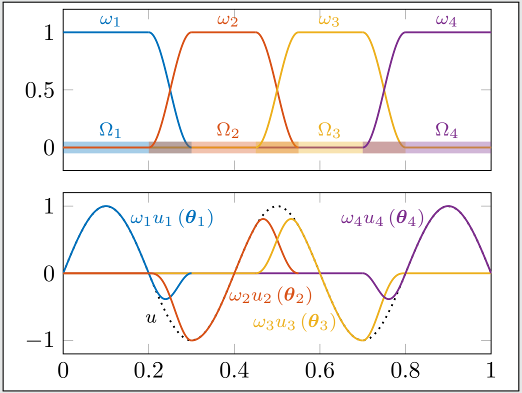
FBPINNs (Moseley, Markham, Nissen-Meyer (2023))

FBPINNs employ the **network architecture**

$$u(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_J) = \sum_{j=1}^{J} \omega_j u_j(\boldsymbol{\theta}_j)$$

and the **loss function**

$$\mathcal{L} = \frac{1}{N} \sum_{i=1}^{N} \left(n[\sum_{\boldsymbol{x}_i \in \Omega_j} \omega_j u_j](\boldsymbol{x}_i, \boldsymbol{\theta}_j) - f(\boldsymbol{x}_i) \right)^2.$$

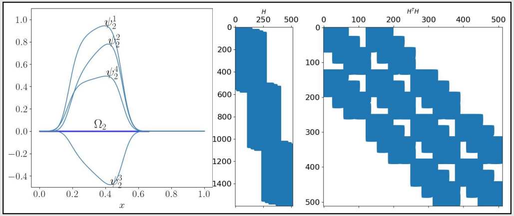


Domain decomposition for RaNNs

We employ the FBPINNs approach; cf. **Shang, Heinlein, Mishra, Wang (acc. 2025)**. This is closely related to the **random feature method (RFM)** by **Chen, Chi, E, Yang (2022)**. In particular, we solve

$$\mathcal{A}[\sum_{j=1}^{J} \omega_j u_j(\boldsymbol{A}_j)](\boldsymbol{x}_i) = f(\boldsymbol{x}_i),$$

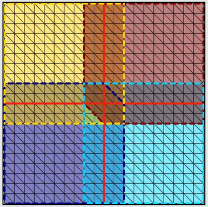
for $i = 1, \ldots, N_{\text{PDE}}$; the boundary condtions are incorporated directly into the u_j.



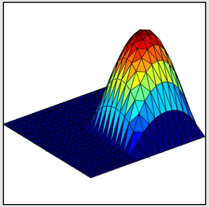
The hidden weights are randomly initialized, the resulting matrices $\boldsymbol{H}$ and $\boldsymbol{H}^\top \boldsymbol{H}$ are block-sparse.

One-level Schwarz preconditioner

Overlap $\delta = 1h$



Solution of local problem



Based on an **overlapping domain decomposition**, we define a **one-level Schwarz operator** for $K := H^\top H$

$$M_{\text{OS-1}}^{-1} K = \sum_{i=1}^{N} R_i^\top K_i^{-1} R_i K,$$

where R_i and $R_i^\top$ are restriction and prolongation operators corresponding to Ω_i', and $K_i := R_i K R_i^\top$.

Here, the matrix K_i could be singular in which case we use a **pseudo inverse** K_i^+ instead of K_i^{-1}.

We also consider restricted and scaled additive Schwarz preconditioners; cf. **Cai, Sarkis (1999)**.

Singular Value Decomposition

As discussed before, on each subdomain Ω_j, the RaNN is

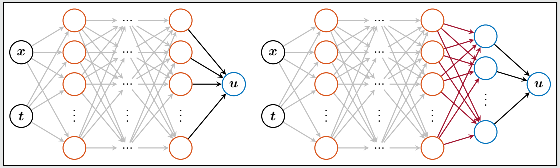
$$\begin{aligned} u_j(\boldsymbol{x}, \boldsymbol{A}_j) &= F_{L+1}^{\boldsymbol{A}} \cdot F_L^{W_L, b_L} \circ \ldots \circ F_1^{W_1, b_1}(\boldsymbol{x}) \\ &= \boldsymbol{A}_j \begin{bmatrix} \Phi_1(\boldsymbol{x}) & \cdots & \Phi_k(\boldsymbol{x}) \end{bmatrix}^\top, \end{aligned}$$

where k is the width of the last hidden layer and the Φ_l are the randomized basis functions.

Consider a **reduced SVD** $\boldsymbol{\Phi} = \boldsymbol{U\Sigma V}^\top$, where the entries of the matrix are $\boldsymbol{\Phi}_{i,l} = \Phi_l(\boldsymbol{x}_i)$. Then, we consider

$$\hat{u}_j(\boldsymbol{x}, \boldsymbol{A}_j) = \boldsymbol{A}_j \hat{\boldsymbol{V}}^\top \begin{bmatrix} \Phi_1(\boldsymbol{x}) & \cdots & \Phi_k(\boldsymbol{x}) \end{bmatrix}^\top,$$

where $\hat{\boldsymbol{V}}^\top$ is obtained by omitting the right singular vectors corresponding to small singular values.



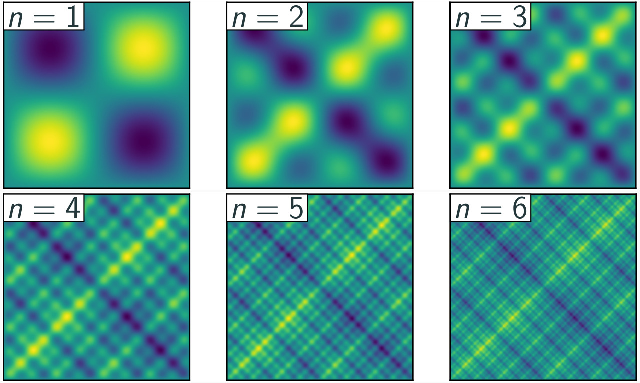
Multi-Frequency Problem

Let us now consider the **two-dimensional multi-frequency Laplace boundary value problem**

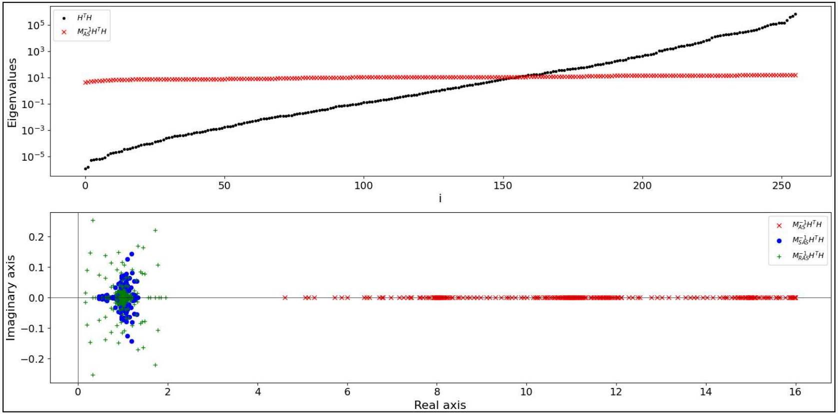
$$
\begin{aligned}
-\Delta u &= 2 \sum_{i=1}^{n} (\omega_i \pi)^2 \sin(\omega_i \pi x) \sin(\omega_i \pi y) && \text{in } \Omega = [0,1]^2, \\
u &= 0 && \text{on } \partial\Omega,
\end{aligned}
$$

with $\omega_i = 2^i$.

For increasing values of n, we obtain the **analytical solutions**:



Results for the Multi-Frequency Problem (n=2)



	$M^{-1}=I$		$M^{-1}=M_{AS}^{-1}$		$M^{-1}=M_{RAS}^{-1}$		$M^{-1}=M_{SAS}^{-1}$	
	iter	e_{L^2}	iter	e_{L^2}	iter	e_{L^2}	iter	e_{L^2}
CG	>2000	$1.95\cdot10^{-2}$	8	$5.03\cdot10^{-3}$	—	—	—	—
CGS	>2000	$2.63\cdot10^{-2}$	4	$5.04\cdot10^{-3}$	24	$5.03\cdot10^{-3}$	6	$5.04\cdot10^{-3}$
BICG	>2000	$1.03\cdot10^{-2}$	8	$5.08\cdot10^{-3}$	32	$5.05\cdot10^{-3}$	11	$5.09\cdot10^{-3}$
GMRES	>2000	$8.68\cdot10^{-2}$	13	$5.07\cdot10^{-3}$	31	$5.06\cdot10^{-3}$	11	$5.08\cdot10^{-3}$

4×4 subdomains; DoF $=256$; $N=1600$; $\boldsymbol{\theta}^0\in\mathcal{U}(-1,1)$; stop.: $\|\boldsymbol{M}^{-1}\boldsymbol{r}^k\|_{L^2}/\|\boldsymbol{M}^{-1}\boldsymbol{r}^0\|_{L^2}\leq10^{-5}$

Results for the Multi-Frequency Problem (n=2) – Initialization & Activation

Now, we inverstigate the effect of different sampling strategies for the 40×40 collocation points:

- **uniformly-spaced** points in the domain,
- **Gauss–Legendre quadrature** points, and
- random sampling from a **uniform distribution** $\mathcal{U}(0,1)$.

as well as tanh and sigmoid activation functions:

collocation points	activation	$M^{-1} = I$			$M^{-1} = M_{AS}^{-1}$		
		κ	iter	e_{L^2}	κ	iter	e_{L^2}
uniformly-spaced	tanh	10^{11}	> 2000	$1.15{\cdot}10^{-2}$	3.5	8	$5.02{\cdot}10^{-3}$
	sigmoid	10^{15}	> 2000	$4.29{\cdot}10^{-2}$	3.6	8	$3.38{\cdot}10^{-3}$
quadrature points	tanh	10^{12}	> 2000	$2.51{\cdot}10^{-2}$	28.3	21	$9.03{\cdot}10^{-3}$
	sigmoid	10^{15}	> 2000	$4.55{\cdot}10^{-2}$	49.6	22	$5.86{\cdot}10^{-3}$
random samples	tanh	10^{12}	> 2000	$5.67{\cdot}10^{-2}$	38.3	22	$3.18{\cdot}10^{-2}$
	sigmoid	10^{15}	> 2000	$7.17{\cdot}10^{-2}$	46.7	25	$1.68{\cdot}10^{-2}$

4×4 subdomains; $N = 1600$; $\boldsymbol{\theta}^0 \in \mathcal{U}(-1,1)$; stop.: $\|\boldsymbol{M}^{-1}\boldsymbol{r}^k\|_{L^2} / \|\boldsymbol{M}^{-1}\boldsymbol{r}^0\|_{L^2} \leq 10^{-5}$

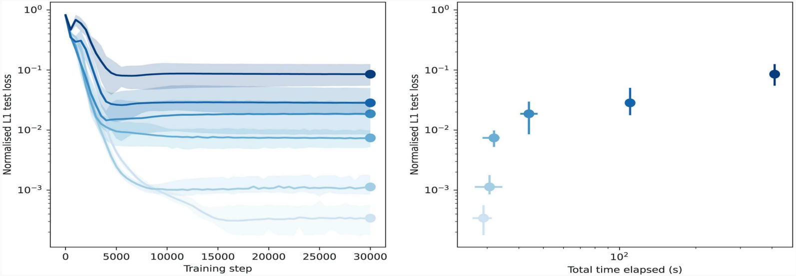
Results for the Multi-Frequency Problem (n=2) – Effect of the SVD

We now investigate the effect of omitting right singular vectors associated with singular values below a varying tolerance τ.

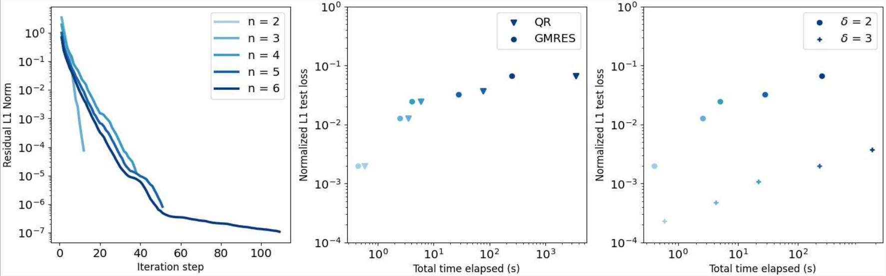
τ	DoF	M^{-1}	σ_{min}	σ_{max}	iter	e_{L^2}
10^{-4}	512	I	10^{-10}	10^{6}	>2000	$3.72 \cdot 10^{-2}$
		M_{AS}^{-1}	10^{-6}	10^{6}	27	$5.46 \cdot 10^{-5}$
		M_{SAS}^{-1}	10^{-7}	10^{5}	30	$5.49 \cdot 10^{-5}$
10^{-3}	436	I	10^{-8}	10^{5}	>2000	$3.75 \cdot 10^{-2}$
		M_{AS}^{-1}	10^{-5}	10^{5}	16	$1.28 \cdot 10^{-4}$
		M_{SAS}^{-1}	10^{-6}	10^{4}	18	$1.28 \cdot 10^{-4}$
10^{-2}	335	I	10^{-5}	10^{5}	>2000	$4.51 \cdot 10^{-2}$
		M_{AS}^{-1}	10^{-3}	10^{4}	14	$7.14 \cdot 10^{-4}$
		M_{SAS}^{-1}	10^{-4}	10^{3}	13	$7.11 \cdot 10^{-4}$
10^{-1}	212	I	10^{-3}	10^{6}	>2000	$5.01 \cdot 10^{-2}$
		M_{AS}^{-1}	10^{-2}	10^{3}	12	$7.13 \cdot 10^{-3}$
		M_{SAS}^{-1}	10^{-3}	10^{2}	11	$7.10 \cdot 10^{-3}$

4×4 subdomains; $N = 1600$; $\boldsymbol{\theta}^0 \in \mathcal{U}(-1,1)$; stop.: $\|\boldsymbol{M}^{-1}\boldsymbol{r}^k\|_{L^2} / \|\boldsymbol{M}^{-1}\boldsymbol{r}^0\|_{L^2} \leq 10^{-5}$

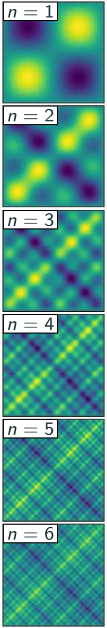
Results for the Multi-Frequency Problem



Multi-level FBPINNs; cf. **Dolean, Heinlein, Mishra, Moseley (2024)**



DD-PIRaNNs; cf. **Shang, Heinlein, Mishra, Wang (acc. 2025)**

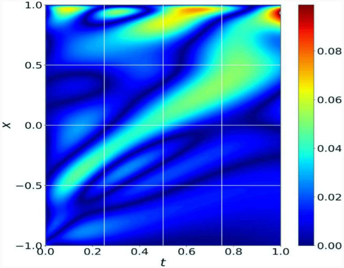


Numerical Results for 1D Advection-Diffusion Problem

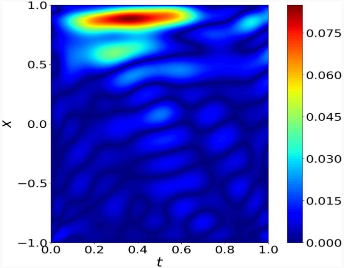
Given $\Omega = (-1, 1)$, we consider a **one-dimensional advection-diffusion equation**

$$
\begin{aligned}
\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} &= \kappa \frac{\partial u^2}{\partial x^2} && \text{in } \Omega \times I, \\
u(-1, t) = u(1, t) &= 0 && \text{in } I, \\
u(x, 0) &= -\sin(\pi x) && \text{in } \Omega,
\end{aligned}
$$

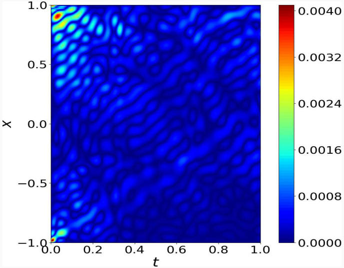
with $I = (0, 1)$. Here, $\kappa = 0.1/\pi$ is the diffusivity coefficient, complicating the solution near $x = 1$ due to the no-slip boundary; see **Kharazmi, Zhang, Karniadakis (2021)**.



hp-**VPINNs**: error results from **Kharazmi, Zhang, Karniadakis (2021)**



DD-PIRaNNs (weak bc. enf.): error; relative L^2 error of 2.88×10^{-2}; 0.12 s



DD-PIRaNNs (hard bc. enf.): error; relative L^2 error of 7.40×10^{-4}; 0.08 s

CWI Research Semester Programme:

Bridging Numerical Analysis and Scientific Machine Learning: Advances and Applications

Co-organizers: Victorita Dolean (TU/e), Alexander Heinlein (TU Delft), Benjamin Sanderse (CWI), Jemima Tabbeart (TU/e), Tristan van Leeuwen (CWI)

- **Autumn School** (October 27–31, 2025):
  - Chris Budd (University of Bath)
  - Ben Moseley (Imperial College London)
  - Gabriele Steidl (Technische Universität Berlin)
  - Andrew Stuart (California Institute of Technology)
  - Andrea Walther (Humboldt-Universität zu Berlin)
  - Ricardo Baptista (University of Toronto)
- **Workshop** (December 1–3, 2025):
  - 3 days with plenary talks (academia & industry) and an industry panel
  - Confirmed plenary speakers:
    - Marta d'Elia (Atomic Machines)
    - Benjamin Peherstorfer (New York University)
    - Andreas Roskopf (Fraunhofer Institute)

Centrum Wiskunde & Informatica

Join us for inspiring talks, hands-on sessions, and industry collaboration!

Summary

Physics-informed neural networks and randomized neural networks

- **PINNs** incorporate physics the loss, improving accuracy and generalization. **RaNNs** use random weights and biases, while only training the last layer. This can **simplify the training** and **reduce computational cost**.
- Both approaches suffer from **ill-conditioning** leading to **a challenging training process**.
- **Not in this talk:** extension to Kolmogorov–Arnold networks (KANs) and neural operators

Domain decomposition architectures and preconditioning

- **Domain decomposition-based architectures** **improve the scalability of PINNs** to **large domains** / **high frequencies**, **keeping the complexity of the local networks low**.
- **Preconditioning**, using a combination of **Schwarz preconditioning** and **SVD**, can **improve the conditioning of the RaNN problem** and **significantly reduce the number of iterations** needed for convergence.

Thank you for your attention!

Topical Activity Group

Scientific Machine Learning