Decomposing physics-informed neural networks

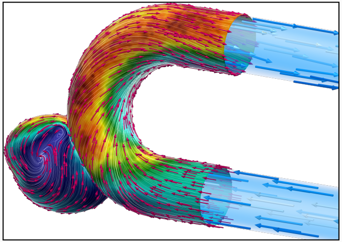
Alexander Heinlein[1]

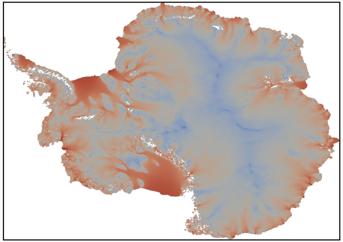
Dutch Computational Science Day, Utrecht, November 10, 2023

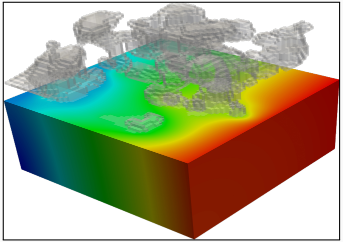
[1]Delft University of Technology

Based on joint work with Victorita Dolean (University of Strathclyde & University Côte d'Azur) and Sid Mishra and Ben Moseley (ETH Zürich)

Scientific Machine Learning in Computational Science and Engineering







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 (SciML)

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

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

Solve the boundary value problem

$$\begin{aligned} \Delta\Psi_t(\boldsymbol{x},\boldsymbol{\theta})+1 &= 0, \quad \text{on } [0,1], \\ \Psi_t(0,\boldsymbol{\theta})=\Psi_t(1,\boldsymbol{\theta}) &= 0, \end{aligned}$$

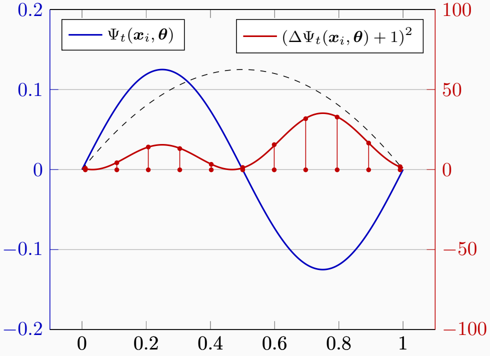
via a **collocation approach**:

$$\min_{\boldsymbol{\theta}} \sum_{\boldsymbol{x}_i} \left(\Delta\Psi_t(\boldsymbol{x}_i,\boldsymbol{\theta})+1\right)^2$$

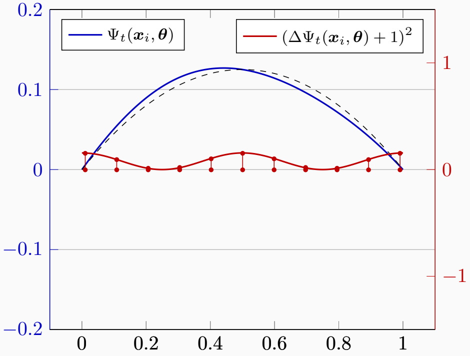
Boundary conditions

The boundary conditions can be **enforced explicitly**, for instance, via the ansatz:

$$\Psi_t(\boldsymbol{x},\boldsymbol{\theta}) = \sin(\pi \boldsymbol{x}) \cdot F(\boldsymbol{x}, N(\boldsymbol{x},\boldsymbol{\theta}))$$



$(\Delta\Psi_t(\boldsymbol{x}_i,\boldsymbol{\theta})+1)^2 >> 0$



$(\Delta\Psi_t(\boldsymbol{x}_i,\boldsymbol{\theta})+1)^2 \approx 0$

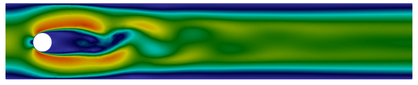
Physics-Informed Neural Networks (PINNs)

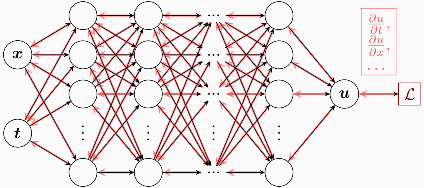
Physics-informed neural networks (PINNs) by Raissi et al. (2019) are based on the approach by Lagaris et al. (1998). The main novelty of PINNs is the use of a **hybrid loss function**:

$$\mathcal{L} = \omega_{\text{data}} \mathcal{L}_{\text{data}} + \omega_{\text{PDE}} \mathcal{L}_{\text{PDE}},$$

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

$$\begin{aligned} \mathcal{L}_{\text{data}} &= \frac{1}{N_{\text{data}}} \sum_{i=1}^{N_{\text{data}}} \left(u(\hat{\mathbf{x}}_i, \hat{\mathbf{t}}_i) - u_i \right)^2, \\ \mathcal{L}_{\text{PDE}} &= \frac{1}{N_{\text{PDE}}} \sum_{i=1}^{N_{\text{PDE}}} \left(\Delta u(\mathbf{x}_i, \mathbf{t}) + 1 \right)^2. \end{aligned}$$





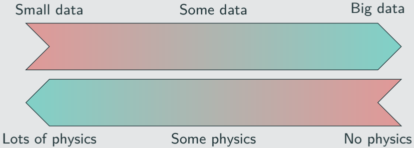
Advantages

- **"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

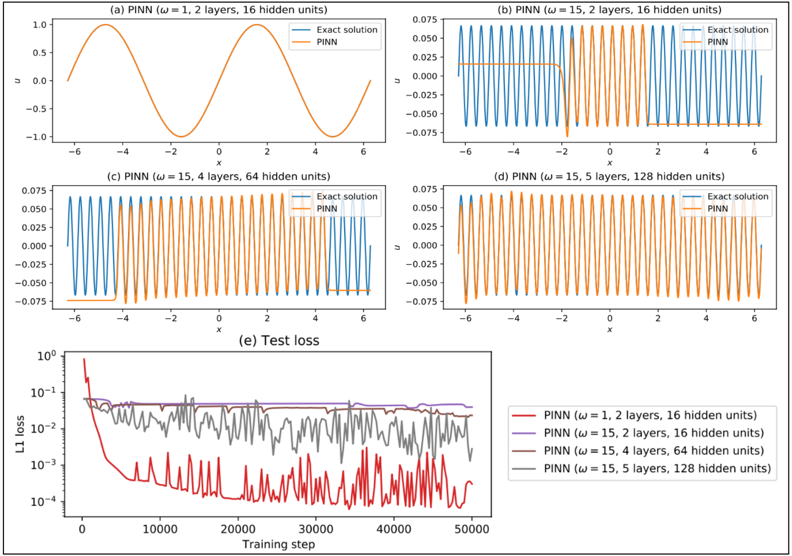


- **Known solution values** can be included in $\mathcal{L}_{\text{data}}$
- **Initial and boundary conditions** are also included in $\mathcal{L}_{\text{data}}$

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

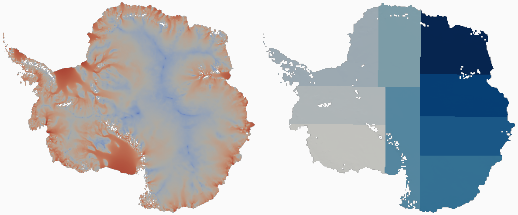
Scaling issues

- Large computational domains
- Small frequencies

(related to so-called spectral bias)

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

Domain Decomposition Methods



Images based on Heinlein, Perego, Rajamanickam (2022)

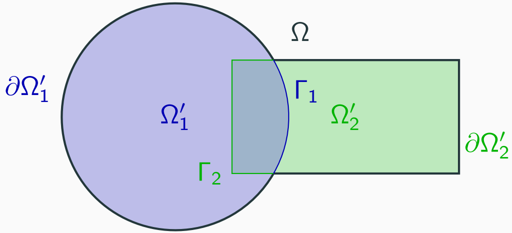
Historical remarks: The **alternating Schwarz method** is the earliest **domain decomposition method (DDM)**, which has been invented by **H. A. Schwarz** and published in **1870**:

- Schwarz used the algorithm to establish the **existence of harmonic functions** with prescribed boundary values on **regions with non-smooth boundaries**.

Idea

Decomposing a large **global problem** into smaller **local problems**:

- **Better robustness** and **scalability** of numerical solvers
- **Improved computational efficiency**
- Introduce **parallelism**



Finite Basis Physics-Informed Neural Networks (FBPINNs)

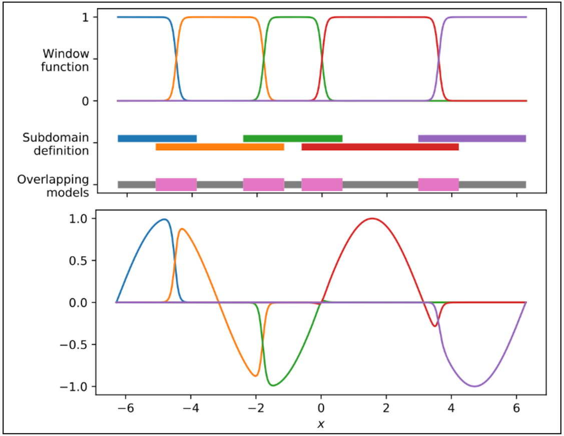
Finite basis physics informed neural network (FBPINNs) method introduced in Moseley, Markham, and Nissen-Meyer (2023) use the **network architecture**

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

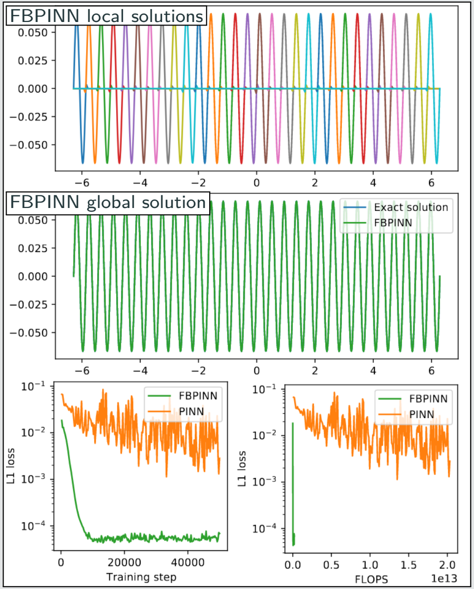
and the **loss function**

$$\mathcal{L}(\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_J) = \frac{1}{N} \sum_{i=1}^{N} \left(n[\mathcal{C} \sum_{\boldsymbol{x}_i \in \Omega_j} \omega_j u_j](\boldsymbol{x}_i, \boldsymbol{\theta}_j) - f(\boldsymbol{x}_i) \right)^2.$$

using **window functions** ω_j with $\operatorname{supp}(\omega_j) \subset \Omega_j$ and $\sum_{j=1}^{J} \omega_j \equiv 1$ on Ω.



PINN vs FBPINN (Moseley et al. (2023))



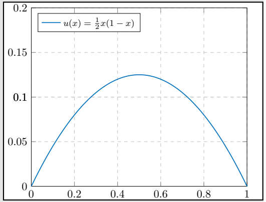
Scalability of FBPINNs

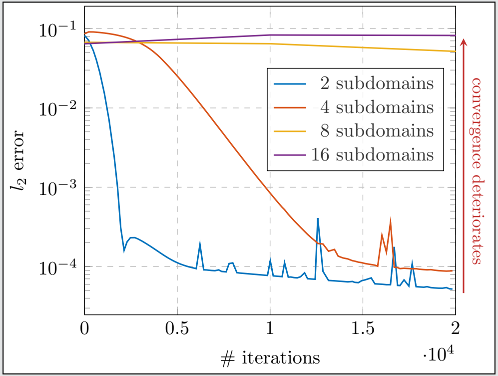
Consider the **simple boundary value problem**

$$-u'' = 1 \quad \text{in } [0,1],$$
$$u(0) = u(1) = 0,$$

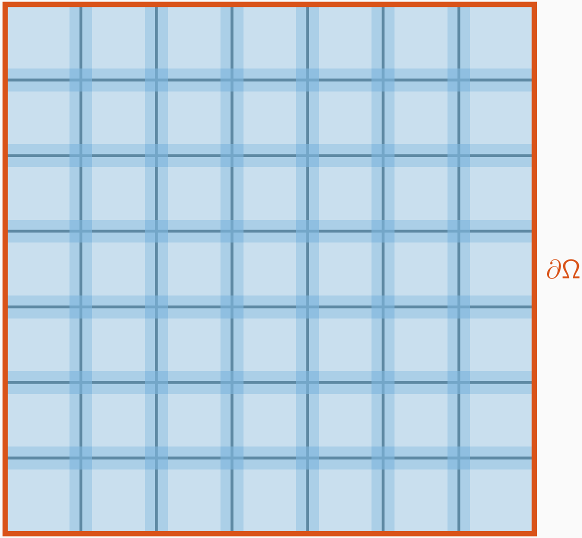
which has the **solution**

$$u(x) = 1/2\,x(1-x).$$

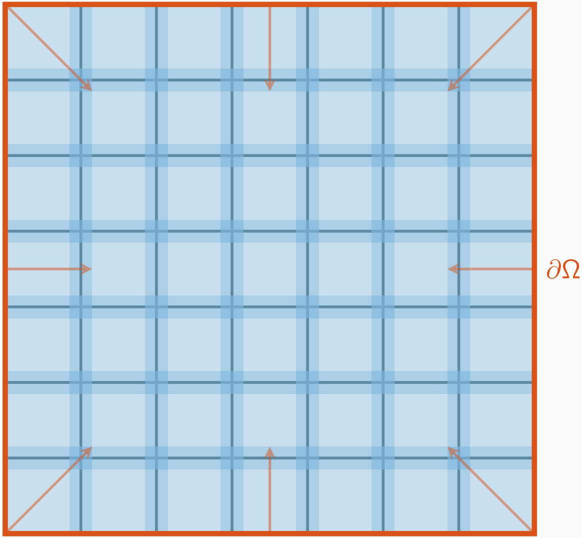




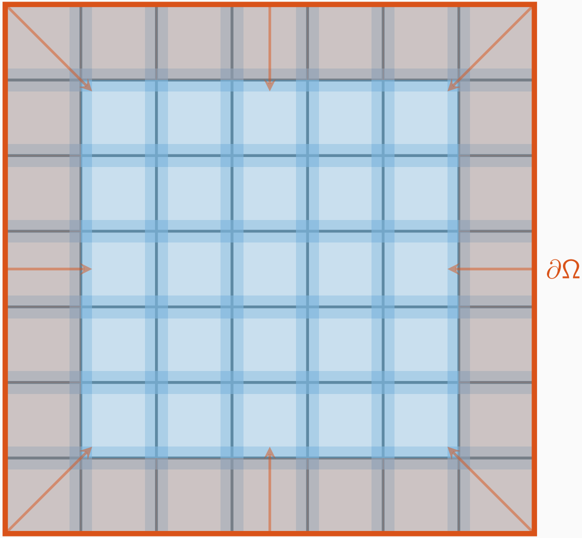
Transport of Information in the One-Level FBPINN Algorithm



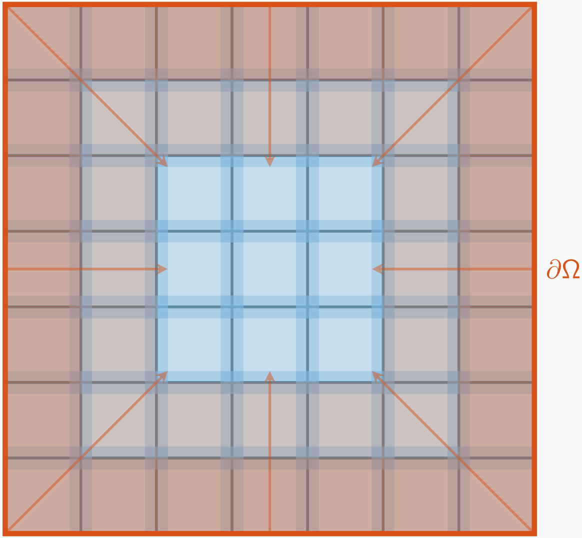
Transport of Information in the One-Level FBPINN Algorithm



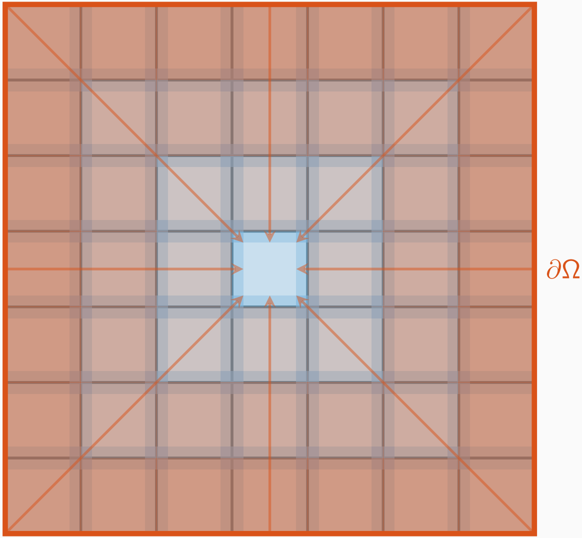
Transport of Information in the One-Level FBPINN Algorithm



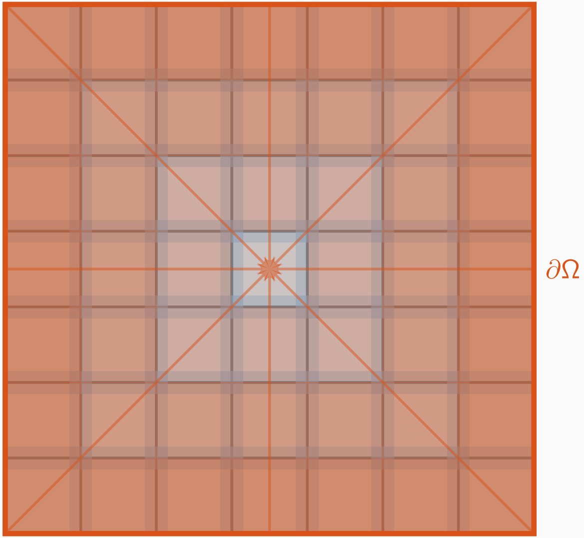
Transport of Information in the One-Level FBPINN Algorithm



Transport of Information in the One-Level FBPINN Algorithm



Transport of Information in the One-Level FBPINN Algorithm



Information (in particular, boundary data) is **only exchanged via the overlapping regions**, leading to **slow convergence** $\rightarrow$ establish a **faster / global transport of information**.

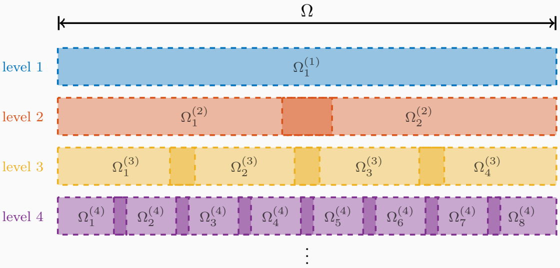
Multi-Level FBPINN Algorithm

We introduce a **hierarchy of L overlapping domain decompositions**

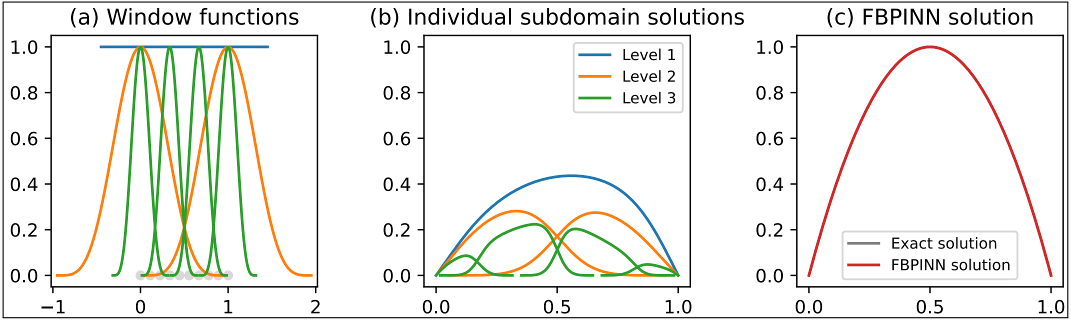
$$\Omega = \bigcup_{j=1}^{J^{(l)}} \Omega_j^{(l)}$$

and corresponding window functions $\omega_j^{(l)}$ with

$$\operatorname{supp}\left(\omega_j^{(l)}\right) \subset \Omega_j^{(l)} \text{ and } \sum_{j=1}^{J^{(l)}} \omega_j^{(l)} \equiv 1 \text{ on } \Omega.$$



This yields the ***L*-level FBPINN algorithm**:



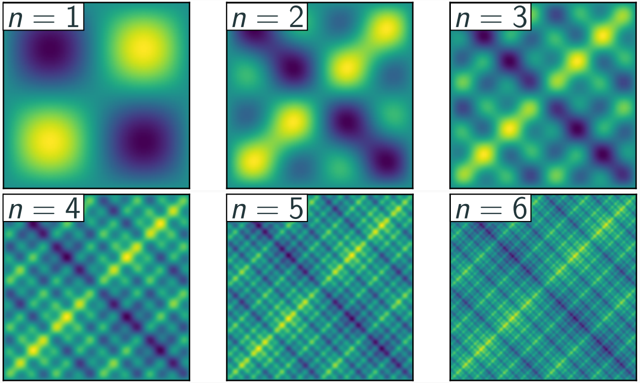
Multi-Frequency Problem

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

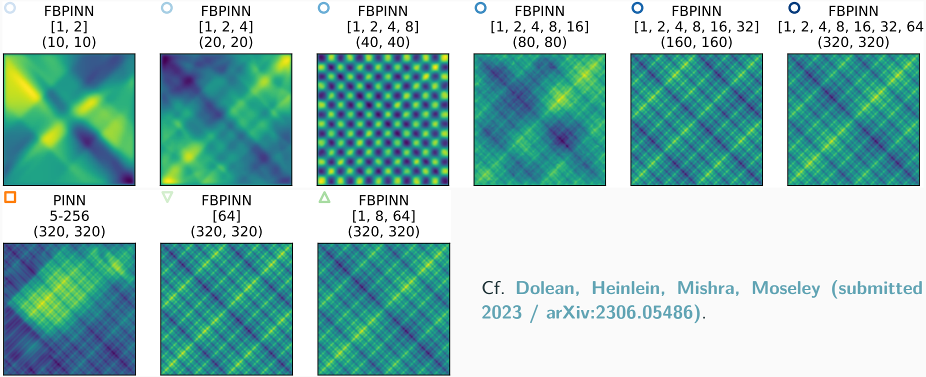
$$
\begin{aligned}
-\Delta u &= 2\sum_{i=1}^{n} (\omega_i \pi)^2 \sin(\omega_i \pi \boldsymbol{x}) \sin(\omega_i \pi \boldsymbol{y}) && \text{in } \Omega = [0,1]^2, \\
u &= 0 && \text{on } \partial\Omega,
\end{aligned}
$$

with $\omega_i = 2^i$.

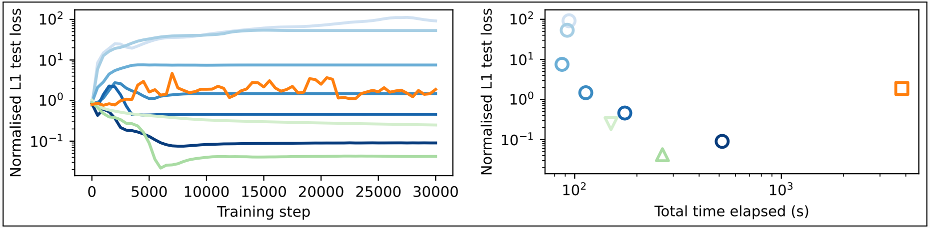
For increasing values of n, we obtain the **analytical solutions**:

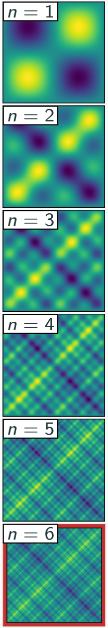


Multi-Level FBPINNs for a Multi-Frequency Problem – Strong Scaling

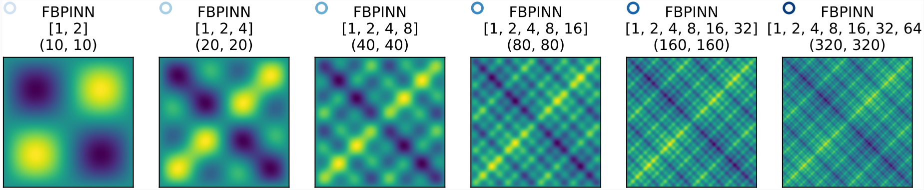


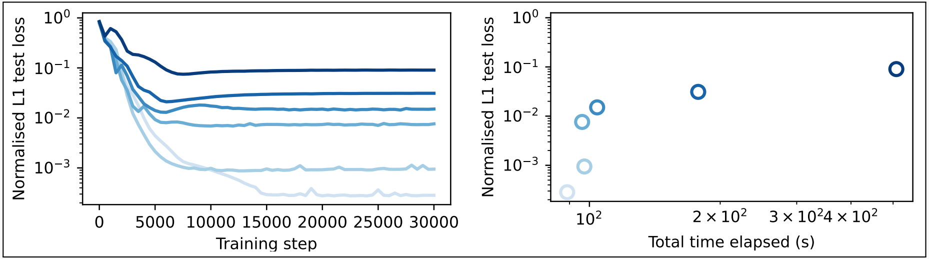
Cf. **Dolean, Heinlein, Mishra, Moseley (submitted 2023 / arXiv:2306.05486)**.

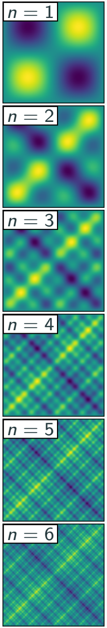




Multi-Level FBPINNs for a Multi-Frequency Problem – Weak Scaling

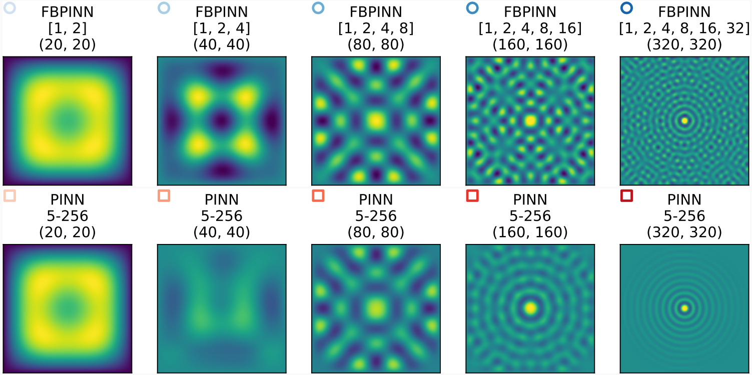


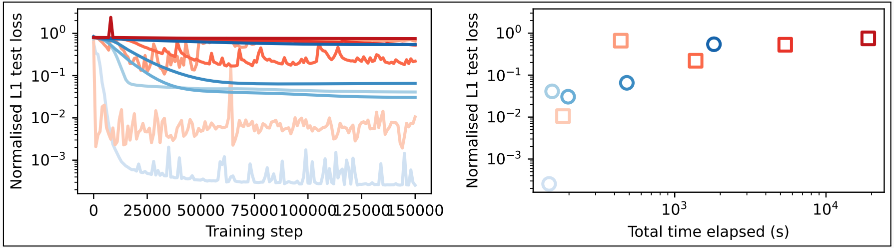


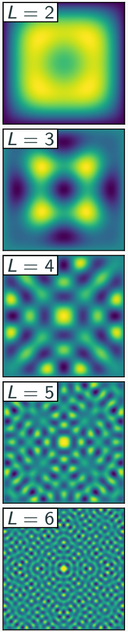


Cf. **Dolean, Heinlein, Mishra, Moseley (submitted 2023 / arXiv:2306.05486)**.

Multi-Level FBPINNs for the Helmholtz Problem – Weak Scaling







Workshop – Scientific Machine Learning

Dates: December 6 – 9, 2023 **Location:** CWI, Amsterdam

Invited speakers

- Machine-learning-enhanced numerical methods, physics-informed machine learning
  - Jan Hesthaven (EPFL Lausanne),
  - Stefania Fresca (Politecnico di Milano)
- Scientific computing for machine learning
  - Sid Mishra (ETH Zurich)
  - Andrea Walther (Humboldt University Berlin)
- Scientific machine learning in applications
  - Dirk Hartmann (Siemens)
  - Elías Cueto (Universidad de Zaragoza)

Organizers

- Benjamin Sanderse (CWI)
- Mengwu Guo (University of Twente)
- Alexander Heinlein (TU Delft)