Seattle SciML Summer School: Case Study Two
An Introduction to Fourier Neural Operators (FNO)

Chris Budd OBE (University of Bath) and
Chaoyu Liu (University of Cambridge)

June 10, 2025

1 Introduction

This case study introduces the **Fourier Neural Operator** (FNO) [LKA+21], a neural operator-based method for learning mappings between function spaces. FNOs have shown excellent performance on learning solutions to PDEs, especially those with complex geometries or parametric coefficients. In this exercise, we will train an FNO model from the repository

`https://github.com/neuraloperator/neuraloperator`

to learn the solution operator of a two-dimensional linear elliptic PDE: **Darcy flow**. If time permits you can look at some other challenging examples of applications of the FNO methodology.

2 The 2D Darcy Flow Problem

We consider the linear elliptic PDE given by

$$-\nabla \cdot (a(x)\nabla u(x)) = f(x), \quad x \in [0,1]^2, \quad u|_{\partial\Omega} = 0. \tag{1}$$

which describes the flow of ground water, or oil, through fractured rock Here,

- $a(x)$ is a spatially varying permeability coefficient.
- $f(x)$ is a known source term (typically constant).
- $u(x)$ is the unknown pressure field to be learned.

We assume $a(x)$ is sampled from a Gaussian random field. The goal of this problem is to learn the mapping from $a(x)$ to $u(x)$, by training the FNO over a dataset of pre-computed solutions.

3 Training FNO on Darcy Flow

1. Open a **Colab notebook**.
2. Enable GPU support via `Runtime` → `Change runtime type` → `T4 GPU` → `Save`.
3. Clone the following GitHub repository:
   !git clone https://github.com/NeuralOperator/neuraloperator
4. Navigate to the folder `fourier_neural_operator/`:

   ```
   %cd neuraloperator
   ```

5. Install the necessary dependencies using:

   ```
   !pip install -r requirements.txt
   ```

6. After installing the dependencies, install the package in editable mode using:

```
!pip install -e .
```

7. Open the train_darcy.py in the Python-Examples Git-Hub folder and copy its content and replace neuraloperator/scripts/train_darcy.py in the FNO repository.
8. Train the FNO model using the training script:

```
!PYTHONPATH=$(pwd) python scripts/train_darcy.py
```

   Note the change in the loss during training.

9. Visualise the predicted pressure field and compare it with the ground truth:

```
from IPython.display import Image
Image("/content/neuraloperator/input_prediction_truth.png")
```

10. Having trained the FNO you are now able to test it on different data sets (which are stored in the repository).
    - Open the test_darcy.py in the Python-Examples Git-Hub folder and copy its content and copy it into neuraloperator/scripts/test_darcy.py in the FNO repository.
    - The variable `idx` on line 122 of test_darcy.py refers to the data set of the function $a(x)$ that you use as input. Set this to 1
    - Run the FNO on the test data via:

      ```
      !PYTHONPATH=$(pwd) python scripts/test_darcy.py
      ```

    - Display image 1 using:

      ```
      from IPython.display import Image
      Image("/content/neuraloperator/input_prediction_truth_1.png")
      ```

    - Repreat for other inages by changing the vakue of `idx`

4 Further Exploration

- Locate the configuration file for FNO applied to Darcy flow in the config folder and adjust the number of training epochs.
- Experiment with different numbers of Fourier modes and network layers to observe their impact on training performance and generalisation.

5 Advanced Exercises

- Look at some of the other examples in neuraloperator/scripts/ For example Burgers equation.
- Try training the FNO on the 2D Allen–Cahn equation [GTWJ24], which models phase separation and takes the form:

$$\begin{aligned} u_t &= u - u^3 + \epsilon^2 \Delta u, \quad && \mathbf{x} \in [0,1]^2,\ t \in (0,T), \\ u(0,\mathbf{x}) &= u_0(\mathbf{x}), && \mathbf{x} \in [0,1]^2, \end{aligned} \tag{2}$$

  where $u_0 \in L^2_{\text{per}}([0,1]^2, R)$ is the initial condition, and $\epsilon > 0$ is a small parameter that controls the interface thickness between phases. We consider periodic boundary conditions, and a suitable choice for ϵ is 0.01. The dataset can be downloaded at https://drive.google.com/file/d/152GSSpGoG-2udgVoNtXEknhasd_lVkYb/view?usp=sharing

- *Advanced* Try using an FNO to solve one of the problems posed in a recent Maths4DL Hackathon. See `challenge1.pdf` in the Git-Hub Case Studies folder.

References

[GTWJ24] y. Geng, y. Teng, Z. Wang, and L. Ju. A deep learning method for the dynamics of classic and conservative allen-cahn equations based on fully-discrete operators. *J. Comp. Phys.*, 2024.

[LKA+21] Zongyi Li, Nikola Kovachki, Kamyar Azizzadenesheli, Burigede Liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations*, 2021.