

# $\pi^3$ Course: Introduction to TorchPhysics for Solving PDEs and Parameter Identification Problems (A3)

## Further Information - Second Part of the Course



July 19-21, 2023

Janek Gödeke

## General Information

### Background

Many problems in application and industry are concerned with solving partial differential equations. There are several Deep Learning approaches, such as PINNs (Physics Informed Neural Networks) or DeepONets (Deep Operator Networks), which turned out to be very promising in this field.

This course gives an introduction to TorchPhysics, a Python library that can be used not only for solving PDEs, but also for solving parameter identification problems, so inverse problems.

TorchPhysics was originally developed by Nick Heilenkötter and Tom Freudenberg, as part of a seminar project<sup>1</sup> at the University of Bremen, in cooperation with the Robert Bosch GmbH. It provides a very intuitive and flexible way to translating the mathematical problem into executable code.

For further information, see also <https://torchphysics.readthedocs.io/en/latest/>.

### Contents

#### Theoretical Part

- Quick introduction into PDEs and parameter identification problems
- Overview about existing Deep Learning methods for solving these problems
- Overview about the basic concept and functionalities of TorchPhysics

#### Practical Part

- Getting started by solving easy problems, e.g. with PINNs
- Solving more complex problems with DeepONets

### Learning Objectives

The participants will get an overview of different Deep Learning methods for solving PDEs and parameter identification problems. Further, they will get in touch with the TorchPhysics library, which can be used for solving those problems conveniently.

### Prior Knowledge

Basic programming skills in Python.

### Technical Requirements

- Own Laptop
- Connection to the Wifi of the University Bremen via eduroam (see <https://www.uni-bremen.de/en/zfn/wifi/instructions-wifi> for more information)

---

<sup>1</sup> <http://www.math.uni-bremen.de/zetem/cms/detail.php?template=modellierungsseminar>

## Further Questions

In case of any questions, please write an email to [janek-goedeke@uni-bremen.de](mailto:janek-goedeke@uni-bremen.de).

## Installation Instructions

### 1. Install Python and (Mini)Conda

See also the description of the first part of the course:

[https://www.uni-bremen.de/fileadmin/user\\_upload/fachbereiche/fb3/rtg-pi3/courses/3rd\\_cohort/Details-1stPart-A3\\_2023\\_01.pdf](https://www.uni-bremen.de/fileadmin/user_upload/fachbereiche/fb3/rtg-pi3/courses/3rd_cohort/Details-1stPart-A3_2023_01.pdf)

### 2. Open a Terminal or Anaconda Prompt

Open a terminal or Anaconda Prompt (Windows) and create a new conda environment by running

```
conda create --name tp
```

### 3. Activate new Environment

Activate the new environment by `conda activate tp`

### 4. Install Jupyter with conda

Before installing TorchPhysics, please install Jupyter with conda:

```
conda install -c anaconda jupyter
```

### 5. Install TorchPhysics

Install TorchPhysics via `pip install torchphysics[all]`

This will automatically install all other required packages, namely

- Numpy
- Matplotlib
- PyTorch Lightning
- Shapely
- Scipy
- PyTorch
- Trimesh

Now you should be able to start e.g. Jupyter lab, by typing `jupyter-lab` into the terminal or Anaconda Prompt (Windows). Please test whether the TorchPhysics installation has been successful, see the instructions in the next paragraph.

## Test Successful Installation

### 1. Open a terminal or Anaconda Prompt

Open a terminal or Anaconda Prompt (Windows) and start Jupyter lab.

### 2. Download Jupyter Notebook

Download the Jupyter notebook `Installation_Test.ipynb`, which you can find on the course website:

<https://www.uni-bremen.de/rtg-pi3/qualification/study-program/schedule-3rd-cohort/a3-introduction-to-deep-learning-pytorch-and-torchphysics>

### 3. Run Jupyter Notebook

Run all cells and check whether the plot at the end of the notebooks appears.

In case of any problems, please write an email to [janek-goedeke@uni-bremen.de](mailto:janek-goedeke@uni-bremen.de).