Thermal Storage Design: Part A

Deep Learning in
Scientific Computing
Due date: June 15th, 2023

Training and testing data can be found on the Moodle page: `https://moodle-app2.let.ethz.ch/course/view.php?id=19789`

IMPORTANT INFORMATION

To get ECTS credits for the course Deep Learning in Scientific Computing you need to submit and obtain a passing grade on the project. Part A consists of three tasks on different topics. For each task, you will be asked to train a learning model and provide predictions on a testing set. **The final submission consists of the predictions files, the code and a report of maximum 2000 characters per task** where you succinctly describe the procedure followed in each task. The submissions have to be collected in a zip folder named as *yourfirstname_yoursecondname_yourleginumber.zip*.

john_doe_111111.zip should include

report1.txt
report2.txt
report3.txt
code1.py
....
pred1.txt
pred2.txt
pred3.txt

Project Description

The main objective of the project is to apply machine learning algorithms to solve various tasks related to the preliminary design of a *thermal energy storage*.

The device is used in solar power plants to store thermal energy during the *charging phase* and release it for production of electricity during the *discharging phase*. The thermal energy is stored due to the interaction of a fluid and a solid phase. During the charging state the fluid is injected at high temperature from one end of the storage and heats the solid up. In contrast, during the discharging phase the reverse process occurs: cold fluid flows from the opposite end and absorbs heat from the solid. Between charging and discharging *idle phases* take place, where no fluid enters the thermal storage.

Therefore, at any instant of time the thermal storage can be in one of the following states:

1. Charging;
2. Idle between charging and discharging;
3. Discharging;
4. Idle between discharging and charging;

Together the four states establish a *cycle* and the same process is repeated for several cycles until the thermal storage reaches a periodic or stationary regime.

Mathematical Model (1D problem)

The thermal storage is modeled by a cylinder with length L and diameter D and it is assumed that temperature variation occurs only along the axis of the cylinder (see Figure 1 for a schematic representation of the thermal storage).

The temperature evolution of the solid and fluid phases, T_s and T_f, is described by a system of two linear *reaction-convection-diffusion* equations:

$$\begin{aligned}\varepsilon\rho_f C_f \frac{\partial T_f}{\partial t} + \varepsilon\rho_f C_f u_f(t)\frac{\partial T_f}{\partial x} &= \lambda_f \frac{\partial^2 T_f}{\partial x^2} - h_v(T_f - T_s) \quad x \in [0, L],\ t \in [0, T],\\ (1-\varepsilon)\rho_s C_s \frac{\partial T_s}{\partial t} &= \lambda_s \frac{\partial^2 T_s}{\partial x^2} + h_v(T_f - T_s) \quad x \in [0, L],\ t \in [0, T],\end{aligned} \tag{1}$$

with ρ being the density of the phases, C the specific heat, λ the diffusivity, ε the solid porosity, u_f the fluid velocity entering the thermal storage and h_v the heat exchange coefficient between solid and fluid. The fluid velocity is assumed to be uniform along the cylinder and varying only in time: $u_f = u$ during charging, $u = 0$ during idle and $u_f = -u$ during discharging, with u being a positive constant.

The system of equations has to be augmented with suitable initial and boundary conditions:

Neumann Boundary conditions

$$T_f(x, t = 0) = T_s(x, t = 0) = T_0,\ x \in [0, L] \tag{2}$$

$$\frac{\partial T_s(x,t)}{\partial x}\Big|_{x=0} = \frac{\partial T_s(x,t)}{\partial x}\Big|_{x=L} = 0,\ t \in [0, T] \tag{3}$$

The boundary conditions for the fluid instead will be different according to the current state of the thermal storage:

- **Charging State:** Dirichlet Neumann

$$T_f(0,t) = T_{hot}, \qquad \frac{\partial T_f(x,t)}{\partial x}\Big|_{x=L} = 0, \qquad t \in [0,T] \tag{4}$$

- **Discharging State:**

$$\frac{\partial T_f(x,t)}{\partial x}\Big|_{x=0} = 0, \qquad T_f(L,t) = T_{cold}, \qquad t \in [0,T] \tag{5}$$

- **Idle Phase:**

$$\frac{\partial T_f(x,t)}{\partial x}\Big|_{x=0} = 0, \qquad \frac{\partial T_f(x,t)}{\partial x}\Big|_{x=L} = 0, \qquad t \in [0,T] \tag{6}$$

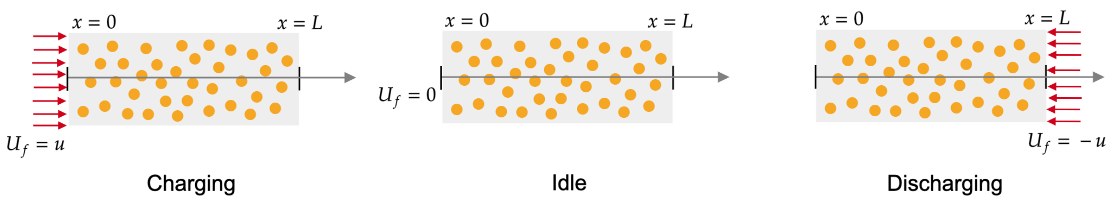

Figure 1: Schematic representation of the thermal storage

Task 1: PINNs for solving PDEs

In this task we aim at solving the system of equations 1 with physics informed neural networks. In particular we are interested in the solution of the system during the charging phase of the first cycle.

To this end, consider the *non-dimensional* set of equations:

$$\begin{aligned} \frac{\partial \bar{T}_f}{\partial t} + U_f \frac{\partial \bar{T}_f}{\partial x} &= \alpha_f \frac{\partial^2 \bar{T}_f}{\partial x^2} - h_f(\bar{T}_f - \bar{T}_s) \quad x \in [0,1],\ t \in [0,1], \\ \frac{\partial \bar{T}_s}{\partial t} &= \alpha_s \frac{\partial^2 \bar{T}_s}{\partial x^2} + h_s(\bar{T}_f - \bar{T}_s) \quad x \in [0,1],\ t \in [0,1], \end{aligned} \tag{7}$$

with the following initial and boundary conditions:

Initial

$$\bar{T}_f(x, t=0) = \bar{T}_s(x, t=0) = T_0, \qquad x \in [0,1],$$

Boundary

$$\begin{aligned} \frac{\partial \bar{T}_s}{\partial x}\Big|_{x=0} = \frac{\partial \bar{T}_s}{\partial x}\Big|_{x=1} = \frac{\partial \bar{T}_f}{\partial x}\Big|_{x=1} &= 0, \qquad t \in [0,1], \\ \bar{T}_f(x=0,t) &= \frac{T_{hot} - T_0}{1 + \exp\big(-200(t-0.25)\big)} + T_0, \qquad t \in [0,1]. \end{aligned} \tag{8}$$

The values of the constants are:

$$\begin{aligned} &\alpha_f = 0.05 \quad h_f = 5 \quad T_{hot} = 4 \quad U_f = 1 \\ &\alpha_s = 0.08 \quad h_s = 6 \quad T_0 = 1 \end{aligned} \tag{9}$$

Approximate the solution of the system of PDEs (1) with a physics informed neural network (Pinns). To this end, you can either use:

1. a two-outputs neural network $(t, x) \mapsto (\bar{T}_s^\theta, \bar{T}_s^\theta)$ with tunable parameters θ, or
2. two distinct neural networks $(t, x) \mapsto \bar{T}_f^{\theta_f}$ and $(t, x) \mapsto \bar{T}_s^{\theta_s}$ with distinct sets of tunable parameters θ_s and θ_f.

The python script *Pinns.ipynb* can be easily modified to address the task. You can follow the steps below:

1. initialize the approximate neural network solution in the *Pinns* class;
2. implement the functions *add_interior_points*, *add_temporal_boundary_points*, and *add_spatial_boundary_points*;
3. implement the function *appy_initial_condition*;
4. implement the function *appy_boundary_conditions*;

   in tutorial only dirichlet, now also Neumann boundary conditions. You can get those via autodifferentiation
5. implement the function *compute_pde_residuals*;
6. train the model.

Once the model is trained, make predictions of your model on the data stored in *Task1/TestingData.txt* and save them in *yourfirstname_yoursecondname_yourleginumber/Task1.txt*. **The format of the file has to be the same as the file Task6/SubExample.txt.**

Hint: in the function *appy_boundary_conditions* you need to implement Neumann boundary conditions at $x = 0$ for the solid and $x = 1$ for both the phases. The network derivative at $x = 0$ and $x = 1$ with respect to t and x can be computed as done in the *compute_pde_residuals* function.

Task 2: PDE-Constrained Inverse Problem

Let us now consider the non-dimensional equation governing the fluid temperature only:

System goes through two. cycles of length 4

$$\frac{\partial \bar{T}_f}{\partial t}(x, t) + U_f(t)\frac{\partial \bar{T}_f}{\partial x}(x, t) = \alpha_f \frac{\partial^2 \bar{T}_f}{\partial x^2} - h_f(\bar{T}_f(x, t) - \bar{T}_s(x, t)) \quad x \in [0, 1], \boxed{t \in [0, 8]}. \tag{10}$$

You go through chargin g-idle-dischar ge-idle each with time=1

Observe that compared to Task 1, the time horizon is 8. In this time frame, 2 cycles of length $T_c = 4$ are realized. Each cycle starts with the charging phase where the velocity of the fluid is $U_f = 1$, followed then by an idle phase ($U_f = 0$), a discharging phase ($U_f = -1$) and finally again an idle phase ($U_f = 0$). Each phase has length 1. The fluid velocity during the two cycle frame $U_f = U_f(t)$ is plotted in the left panel Figure 2.

Either through PINN or finite difference solver in pytorch methods

In this task we aim at solving a PDE-constrained inverse problem. In particular we would like to infer the solid temperature $\bar{T}_s = \bar{T}_s(x,t)$, given noiseless measurements of the fluid temperature $\bar{T}_f$ in the entire domain. The fluid temperature is plotted in the right panel of Figure 2.

The initial condition for the fluid temperature is:

$$\bar{T}_f(x, t=0) = T_0, \; x \in [0,1] \tag{11}$$

Instead, the boundary conditions are defined below:

- **Charging State:**

$$\bar{T}_f(0,t) = T_{hot}, \qquad \frac{\partial \bar{T}_f(x,t)}{\partial x}\Big|_{x=1} = 0, \qquad t \in [0,8] \tag{12}$$

- **Discharging State:**

$$\frac{\partial \bar{T}_f(x,t)}{\partial x}\Big|_{x=0} = 0, \qquad \bar{T}_f(1,t) = T_{cold}, \qquad t \in [0,8] \tag{13}$$

- **Idle Phase:**

$$\frac{\partial \bar{T}_f(x,t)}{\partial x}\Big|_{x=0} = 0, \qquad \frac{\partial \bar{T}_f(x,t)}{\partial x}\Big|_{x=1} = 0, \qquad t \in [0,8] \tag{14}$$

The values of the constants are:

$$\alpha_f = 0.005 \quad h_f = 5 \quad T_{hot} = 4 \quad T_{cold} = T_0 = 1 \tag{15}$$

One order of magnitude lower than in Task 1

The data measurements are stored in the file **Task1/DataSolution.txt**. The first and the second column correspond to the x and t-coordinates (first and second column, respectively) where the fluid temperature (third column) is recorded.

Provide the values of the inferred solid temperature at the same time-space coordinates, and save them in the file *yourfirstname_yoursecondname_yourleginumber/Task2.txt*. **The format of the file has to be the same as the file Task2/SubExample.txt**. Make sure the first column correspond to time t, the second to space x and the last to the solid temperature T_s.

Given the plot on the right infer the temperature of the solid

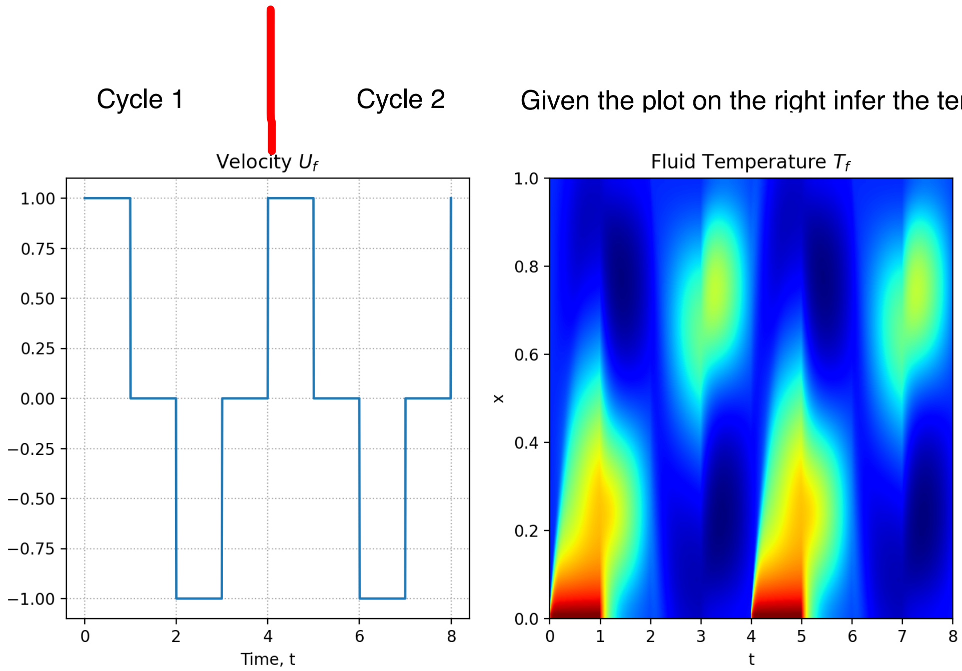

Figure 2: Task 2. Velocity snapshot U_f (left) and fluid temperature T_f measurements data (right).

Hint: there are many possible inverse algorithms which could be used. A couple of ideas include 1) using a PINN, or 2) writing a FD solver and differentiating through it to optimise for the temperature values.

Task 3: Time Series Forecasting with Neural Operator

Let us assume that **noiseless** measurements of the fluid and solid temperature $T^0_{f,i}, T^0_{s,i}$, $i = 0, ..., 210$, are taken at the top end of the storage $x = 0$ during the entire process. In this task we are interested in **future** (or **out of samples**) predictions of the fluid and solid temperature. In other words, the training data consists of time measurements t_i registered in a frame $[0, T]$, $T = 520000s$ and we would like to forecast the fluid and solid temperature in the frame $[T, T_{end}]$, $T_{end} = 602168s$.

Train a learning model **based on Neural Operators** with the data given in **Task3/TrainingData.txt** and make predictions on the test set **Task3/TestingData.txt**. The first column of the training data contains the time frames t_i, $i = 0, ..., 210$, the second and the third column the fluid and solid temperatures $T^0_{f,i}$, $T^0_{s,i}$, $i = 0, ..., 210$, respectively. See Figure 3 for a schematic representation of the training data.

Afterwards, save your predictions in the file *yourfirstname_yoursecondname_yourleginumber/Task3.txt*. **The format of the file has to be the same as the file Task3/SubExample.txt.**

Hint: observe that $t_{i+1} - t_i = \Delta t = const$, for all, $i = 0, ..., 209$.

Neural Operator: f(x) -> f(x)
but here we are only given a sequence of measurements -> process the measurements to get a function representation

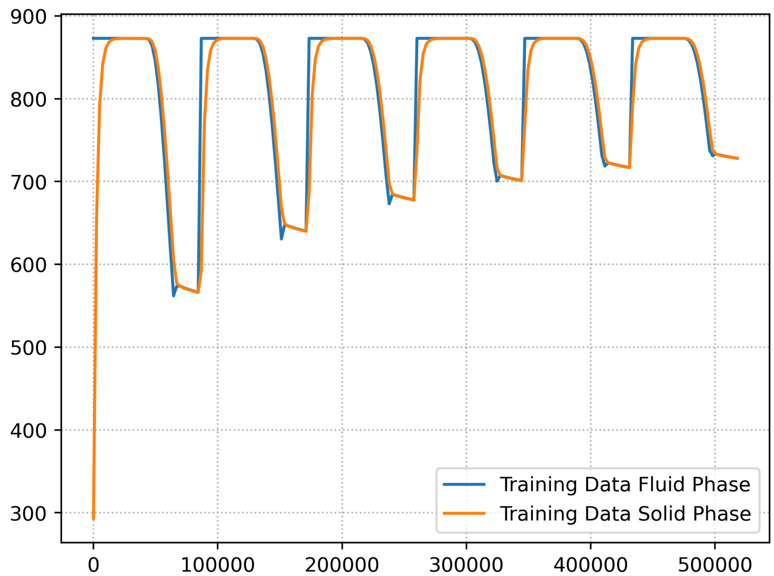

Figure 3: Task 3 training data