Deep Learning Applications in Scientific Computing

Thermal Storage Design
Assignment Report

Timothy Stroschein
Seminar for Applied Mathematics

June 16, 2023

Contents

1 Introduction

This report presents an exploration of an energy storage device model, where energy exchange occurs between a solid and a fluid medium. Specifically, the fluid flows into a cylindrical device, with two distinctive states identified: charging (hot fluid flows from left to right) and discharging (cold fluid flows from the opposite direction). These states are separated by idle periods when the fluid is at rest. The system undergoes repeated cycles of these states, each for an equal duration.

The behaviors of this model can be effectively described by reaction-convection-diffusion equations, with adjustments made to the fluid velocity term corresponding to each state. For the solid component, the boundary conditions are defined by homogeneous Neumann conditions, reflecting the assumption of interaction exclusively with the fluid. The charging state is defined by Dirichlet boundary conditions at $x = 0$, implying the entry of hot fluid. Similarly, during the discharging phase, Dirichlet boundary conditions are applicable for the fluid at $x = L$. Moreover, we simplify our model under the assumption of one-dimensional system behavior due to cylindrical symmetry.

This simple model finds a wide range of applications and facilitates a blend of physical and data-driven analysis. In the ensuing sections, we delve into the study of this device's aspects in three distinct contexts, using various deep learning methods.

2 Task 1: PINNs for solving PDEs

The first task involves the determination of temperature distributions of the solid and fluid during the initial charging phase. This evolution is represented by two coupled reaction-convection-diffusion equations, subject to both Neumann and Dirichlet boundary conditions. Given the temperature of the fluid at the left boundary $T_f(0, t) = f(t)$, the model is fully defined by the boundary conditions and the PDEs.

Physics-Informed Neural Networks (PINNs) present a robust solution for such problems. By incorporating the governing equations (PDEs and boundary conditions) into the loss function of a deep learning model, PINNs facilitate the prediction of system behaviors directly from physical principles. This method is particularly effective for systems, like ours, that are governed by complex or coupled PDEs.

The script `Pinns.ipynb` can be easily adjusted to this setup. The `add spatial boundary points` function is adapted to implement Neumann boundary condition for T_s and Dirichlet and Neumann boundary conditions at $x = 0$ and $x = 1$ for T_f respectively. We train a single NN with two outputs (T_f, T_s) and adjust the pde residual and boundary points according to the given PDE. Fig 1 depicts the results of the trained model.

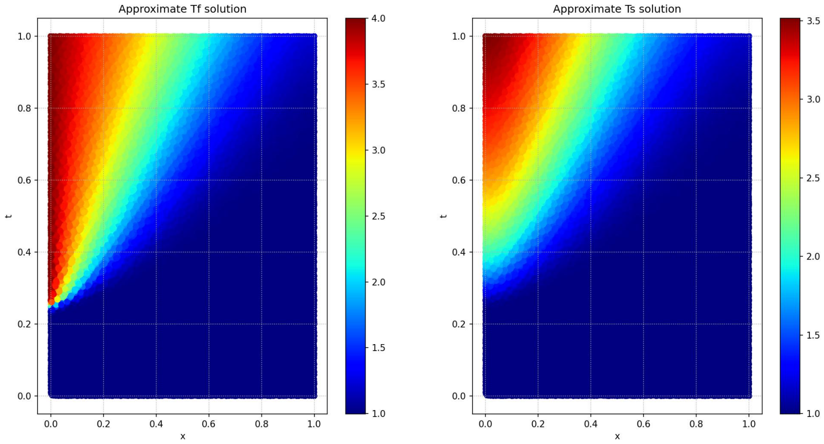

Figure 1: Solution of the trained model on points sampled from a Sobol sequence. The left plot depicts the learned solution for the temperature of the fluid T_f and the right plot for the temperature of the solid T_s.

3 Task 2: PDE-Constrained Inverse Problem

The second task investigates the inverse problem of determining the function $T_s(t, x)$, given the noiseless measurement points for temperature distribution $T_f(t, x)$ of the fluid. We consider two full cycles of the heat storage device. So the temperature of the fluid system is described by eight concatenated reaction-convection-diffusion equations, with fluid velocity and boundary conditions corresponding to the state of the device. This represents a classic inverse problem where we aim to extract unknown information from known data through knowledge of the underlying pysical model.

For this combined setting of physical knowledge and available measurement data, PINNs represent a suitable architecture to learn the solid temperature and we can adjust the PINN from the previous task. A measurement loss on the noiseless data is added and the PDE and boundary residuals are adjusted to a single PDE, corresponding to the first output T_f of the NN. However, according to the input time, t different fluid velocity U_f in the PDE and spatial boundary conditions for T_f apply. One option to handle this adjustment is the usage of boolean tensors and element-wise multiplication. F.e. in `add spatial boundary points` boolean tensors are used to create an appropriate tensor of boundary points for a given tensor of input points. Further element-wise multiplication with a boolean tensor is used to apply Neumann or Dirichlet boundary conditions according to the given input tensor in `apply boundary conditions`. Similarly, a boolean velocity tensor is used to calculate the PDE residual in `compute pde residual`. To ensure sufficient expressiveness and resolution, a network with 12 hidden layers of 160 neurons is used and 1024 interior points are sampled for the physics loss.

By minimizing the total loss we expect the PINN to learn the solid temperature distribution T_s such that the given PDE indeed describes the evolution of T_f in agreement with the measured data T_f. The learned fluid and solid temperature are depicted in fig.2.

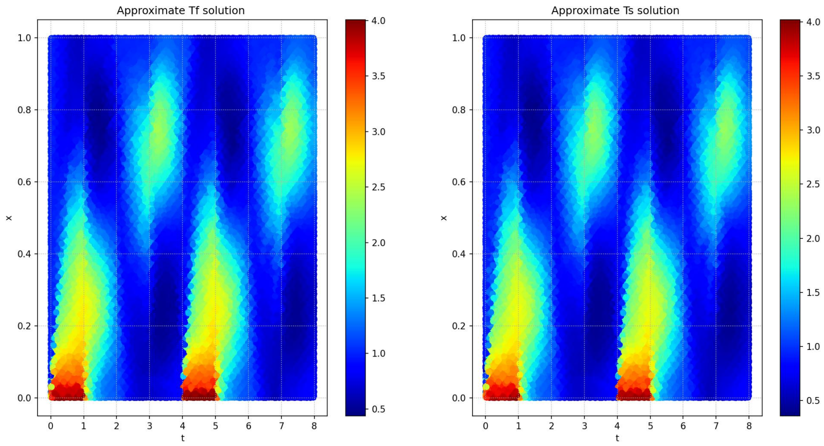

Figure 2: The left plot depicts the learned fluid temperature T_f over two cycles, and the right of the solid T_s. The temperature T_s is solely learned as PDE constrained inverse problem while learning of T_f includes training on noiseless data.

4 Task 3: Time Series Forecasting with Neural Operator

In this task, we are given noiseless measurements of T_f and T_s at $x = 0$ over multiple cycles during the time $t \in [0, T]$. Our goal is to predict the systems behavior during $[T, T_{end}]$. This is a quintessential example of time series forecasting, in which historical data are used to predict future behavior.

Under the mild assumption, that the Dirichlet boundary conditions during the charging and discharging state remain the same for every cycle, the operator describing the evolution of the heat storage device is periodic in time with periodicity corresponding to the time required to complete one cycle. From the periodic pattern of the given data one can see that during $[0, T]$, the system undergoes 6 cycles where a single cycle has duration ΔT. Further, we have $T_{end} - T \leq \Delta T$ and thus predicting the temperature progression for one additional cycle suffices. Therefore, we aim to learn the operator $\mathcal{D}$ that describes the evolution during a single cycle. More Specifically we want to learn $\mathcal{D}$ as a mapping between functions, where the input functions describe the previous state of the system. In addition, we also want our numerical operator also to include the time window as input. Thus we have

$$\mathcal{D}(\mathbf{t}^{(i)}, T_f^{(i)}, T_s^{(i)}) = (T_f^{(i+1)}, T_s^{(i+1)}) \tag{1}$$

where $\mathbf{t}^{(i)}$ is the vector of time steps during which the measurements were taken and $T_f^{(i)}$ is the temperature distribution during the i-th cycle. Finally, note that the above considerations imply that $\mathcal{D}$ should be equivariant under time translations.

To learn this operator we divide the data into five input-output pairs

$$\left[(\mathbf{t}^{(i)}, T_f^{(i)}, T_s^{(i)}), (, T_f^{(i+1)}, T_s^{(i+1)})\right] \quad i = 1 \dots 5$$

I choose to use to use two Fourier Neural Operators (FNO) with a single output to learn this operator. The FNO is a general operator capable of effectively learning a wide range of operators as most real-world PDE problems indeed have a translational invariant greens function, and leverages at the same time various advantages from Fourier analysis. [1]

In our case, the FNO is especially suitable, as its architecture inherently implies translational equivariance as a consequence of its convolutional integral form.[1] Thus the Fourier structure of the

[1]For clarity let T_a be the translation operator. Then we have

$$T_a[(A\vec{v})(y) + (\kappa \star v)(y)] = (AT_a\vec{v})(y) + (\kappa \star T_a v)(y)$$

FNO is advantageous to learn the strong periodic structure of the measured data.

With the trained operator we can predict the temperature during $[T, T_{end}]$ based on the data we have for $[T - \Delta T, T]$. The prediction of the trained model is depicted in fig.3.

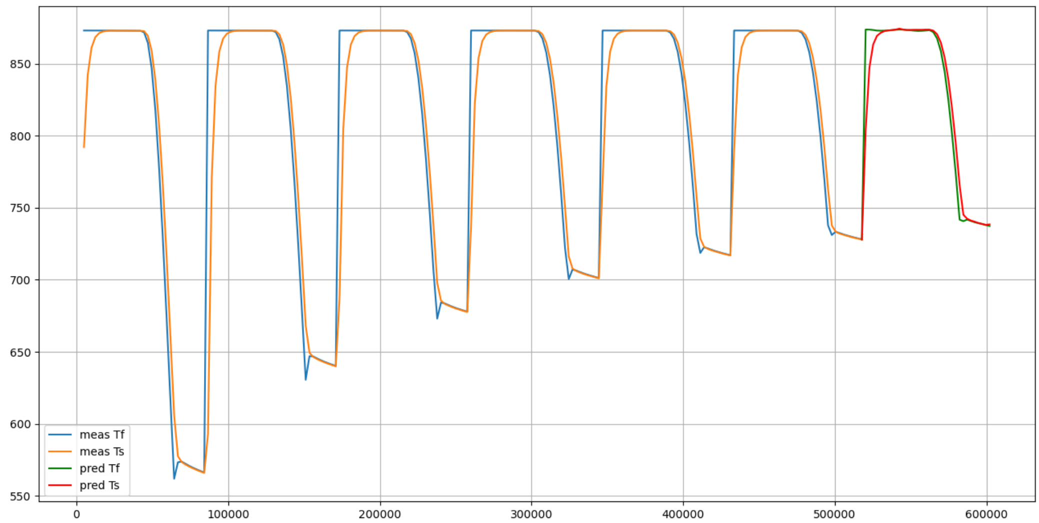

Figure 3: Solid and fluid temperature at x=0 as a function of time. The blue line depicts the noiseless data for the fluid and the orange for the solid temperature. The green line depicts the prediction of fluid temperature during the the time $[T, T_{end}]$ and the red line of the solid temperature.

5 Conclusion

This report provides an examination of a thermal energy storage device from multiple deep learning perspectives. The blend of data-driven and physics-based approaches has proven instrumental in addressing tasks with varying requirements. The PINN's ability to solve PDEs, the neural network's adeptness in inverse problem solving, and the Neural Operator's capacity for time series forecasting all underscore the efficacy of deep learning methodologies in scientific computing.

While the proposed solutions have shown promising results, further research is required to refine these methods. Potential improvements could involve exploring different network architectures, fine-tuning hyperparameters, and integrating additional physical constraints into the model.

We also discerned a considerable advantage in Task 3 through incorporating underlying symmetries into the architecture to learn an operator. A more profound understanding of this approach and the ability to incorporate increasingly complex symmetries could serve as a crucial advancement for operator learning architectures. Such an advancement could potentially equip us with the means to tackle and effectively solve more complex and intricate problems with enhanced efficiency and accuracy.

References

[1] Zongyi Li et al. "Fourier Neural Operator for Parametric Partial Differential Equations". In: *CoRR* abs/2010.08895 (2020). arXiv: `2010.08895`. URL: `https://arxiv.org/abs/2010.08895`.

where we used that A is a linear map only acting on the components and change of variables in the convolution. Since the activation functions act component-wise, this implies equivariance of the FNO under translation.