1st Serbian International Conference on Applied Artificial Intelligence (SICAAI)
Kragujevac, Serbia, May 19-20, 2022

DATA-DRIVEN SOLUTION OF 1-D STEFAN PROBLEM

Srđan T. Nikolić, Miloš R. Ivanović, Ana M. Kaplarević-Mališić

Faculty of Science, University of Kragujevac, Radoja Domanovića 12, 34000 Kragujevac, Serbia
e-mail: srdjan.nikolic@pmf.kg.ac.rs, mivanovic@kg.ac.rs , ana@kg.ac.rs

Abstract

The Stefan problem which describes phase changes along with heat conduction is used for formulating a wide range of problems in science and engineering. Such direct and inverse formulations of non-linear problems are successfully solved using numerical techniques. Solving inverse problems demands calculations that are extremely time-consuming, because dealing with unknown model parameters involves an iterative procedure. Therefore, accuracy is often substituted for efficiency. Physics-informed neural networks offer a different approach for solving and discovering problems described by partial differential equations, which have the potential to enable fast calculations while preserving accuracy. In this paper, we propose a data-driven solution of the direct Stefan problem with the aim to determine its accuracy and potential of replacing well-established numerical methods. A physics-informed neural network method is applied in solving a direct 1D Stefan problem with time dependent Dirichlet boundary conditions that describe a melting process. Computational results of the proposed approach are compared to analytical and finite difference solutions, for various PINN architectures. The applied methodology exhibited sufficient stability and good average accuracy.

Keywords: Stefan problem, physics-informed neural network, predictive modeling, non-linear partial differential equations

1. Introduction

The process of heat conduction with a change of phase occurs in a wide range of real-world phenomena in which the phase changes between states, such as liquid, solid, and vapor. The material is assumed to undergo a phase change with a moving boundary. Moving boundary problems that are inherently nonlinear require solving the heat equation in an unknown region, which must also be determined as a part of the solution.There is a substantial number of well-established numerical methods used for solving diverse types of Stefan problems reported in [1]. In this paper, we address this problem by employing physics-informed neural networks.

Physics-informed neural networks (PINN) are artificial neural networks capable of solving supervised learning tasks respecting a certain physical phenomenon which is formulated using nonlinear partial differential equations [2,3]. Besides solving nonlinear differential equations, PINN can be very efficient in their discovery, namely, in determining unknown parameters when experimental data is provided. In such a case, the PINN model is trained to follow experimental data and respect given physical laws. A trained model predicts behavior of modeled systems directly, unlike standard numerical procedures whose calculations are iterative and time-consuming. The first step of introducing PINNs in production is determining their effectiveness (accuracy) in solving a concrete problem.

In this paper we present a methodology for using a PINN approach in solving direct the 1-D Stefan problem describing a melting process, with the aim of determining a temperature distribution in liquid phase along with moving boundary location. We tested various network architectures of PINN to assess its capability of providing sufficient stability and accuracy compared to the analytical solution, but also to the numerical solution presented in [1]. Additionally, we implemented an early stopping approach for halting a training process if it does not improve model performance after an arbitrary number of epochs, defined by the *patience* parameter [4]. To determine the optimal value of *patience*, we conducted experiments for different patience values on each network architecture.

2. PINN solution of 1-D direct Stefan problem

Physics informed neural network (PINN) is a machine learning method for solving partial differential equation problems which acts as a collocation solver. The training data is obtained by random selection of collocation points distributed over the domain and then collocated to the solution through a loss function. The original approach of the method is given in [2,3].

In this paper, we present a PINN solution of the Stefan problem formulated in [1], where one dimensional phase change is demonstrated by a semi-infinite solid, like a thin block of ice occupying $0 \leqslant x < \infty$, on solidification temperature. We assume the temperature at $x = 0$, which is the fixed boundary of the thin block of ice, increases exponentially with time. The temperature of the entire solid phase is assumed to be at melting point. At time t_0, temperature distribution in the liquid phase must be determined along with the location of the free boundary $s(t_0)$, where $x < s(t_0)$. Over a time interval (t_0, t_1), where $t_1 > t_0$, the part of the thin block of ice has been melted, from position $s(t_0)$ to position $s(t_1)$.

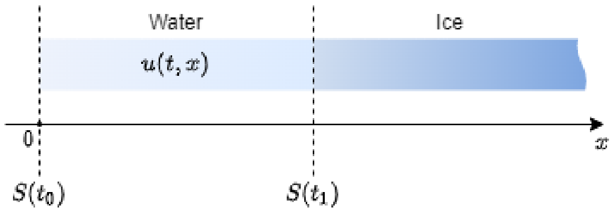

Figure 1. Schematic view of the 1-D Stefan problem.

The temperature distribution $u(x, t)$ in the liquid phase region $0 \leqslant x \leqslant s(t)$ is given by the heat equation

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \tag{1}$$

under the following boundary conditions:

$$\begin{aligned} u(x,t) &= e^{\alpha t}, \quad x = 0, \qquad t > 0 \\ u(x,t) &= 1, \qquad x = s(t), \quad t > 0, \end{aligned} \tag{2}$$

where α is a physical parameter combining density, specific heat, and thermal conductivity. The location of the moving boundary complies to Stefan condition: $\frac{1}{\alpha}\frac{ds}{dt} = -\frac{\partial u}{\partial x}, x = s(t), t > 0$. In the general case, the initial condition is given by $s(0) = 0$. The exact solution for this specific problem is known: $u(x,t) = e^{\alpha t - x}, s(t) = \alpha t$.

Solving this problem following the PINN approach assumes constructing two neural networks. The first approximating temperature distribution function $u(x, t)$ and the second approximating the function of free boundary location $s(t)$. Approximate solutions are differentiated with respect to their variables for values defined in set of collocation points selected from the domain $[0, T] \times \mathcal{D}$, where $\mathcal{D} \subset \mathbb{R}^d$ is a bounded domain, and T denotes the final time. Loss function consists of terms used for measuring satisfaction of (1) and (2) by neural network approximations of u and s at collocation points, where terms include a given partial differential equation and both the initial and boundary conditions along the domain boundary.

Specifying the model for training included determining the following components of the composite loss function:

- Differential operator $L1 = \frac{\partial u}{\partial t} - \alpha \frac{\partial^2 u}{\partial x^2}$, as an implicit formulation of the conduction equation (3).
- Condition $C1 = \left(\frac{1}{\alpha}\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x}\right)(1 + sgn(x - (s - \delta))(1 - sgn(x - s))$, where δ represents the width of the region taken by a loss component. It quatifies the Stefan condition in a narrow region of the phase change (free boundary).
- Initial contition $C2 = (1 - sgn(t - (t_0 + \delta)))(s - s_0)$, where δ represents the width of the region taken by this loss component.
- Derichlet condition $C3 = (1 - sgn(x - (0 + \delta))(u - e^{\alpha t})$.
- Boundary condition $C4 = (1 - sgn(x - (s + \delta)))(1 + sgn(x - s))(u - 1)$.

We took the hyperbolic tangent as the activation function for all neurons in both neural networks. The training data set is based on a generated 300×300 mesh grid of collocation points. For the learning process, the Adam algorithm is used, along with MSE as an error measure. Various network architectures were considered, which is discussed in the next section, along with the number of epochs driven by the early stopping mechanism. The batch size is set to 256. Following recommendation given in [1], this relatively large value has been chosen to avoid the insufficient number of collocation points needed for boundary condition check. All calculations have been conducted using SciANN, a Python library abstracting Keras and TensorFlow [4]. The PINN implementation of the 1-D Stefan problem is available on GitHub[1].

3. Results and Discussion

Neural network architectures converge differently, so there is no mechanism to determine ANN architecture of hidden layers that best suits this specific kind of problem. We benchmarked the proposed PINN solution of the 1-D Stefan problem with various neural network architectures in order to explore how they affect accuracy and training time. We carried out variants with 1, 2, 3 and 4 layers in combination with 20, 30 and 40 neurons per layer to conclude how the architecture affects the rate of convergence. The learning rate was set to 0.02. The physical system is observed in the interval $t_0 = 0$ till $t_{max} = 1$. The α parameter was set to 1. We conducted all runs on the GeForce RTX 3080 graphics controller.

For each ANN architecture, we conducted training, with a limit of 10000 epochs, for different values of the patience parameter. We calculated the average RMSE across all benchmarked network architectures, together with standard deviation and average training time. The results are shown in Figure 2, with Stefan's boundary position, $s(t)$, acting as the single representative accuracy indicator. The very last value of patience (10000) is depicted for the sake of ensuring the early stopping mechanism works.

The application of the numerical methodology presented in [1], for identical setup, results in approximately 0.005 RMSE value, which aligns with the average RMSE of PINN solution. We can conclude that the PINN accuracy is in line with existing numerical solutions. It should be emphasized that certain PINN architectures exhibit even better accuracy than the numerical solution [1]. For example, PINN consists of 2 layers with 30 neurons reached RMSE ≈ 0.004.

[1] https://github.com/srdjan034/PINN-Stefan-problem

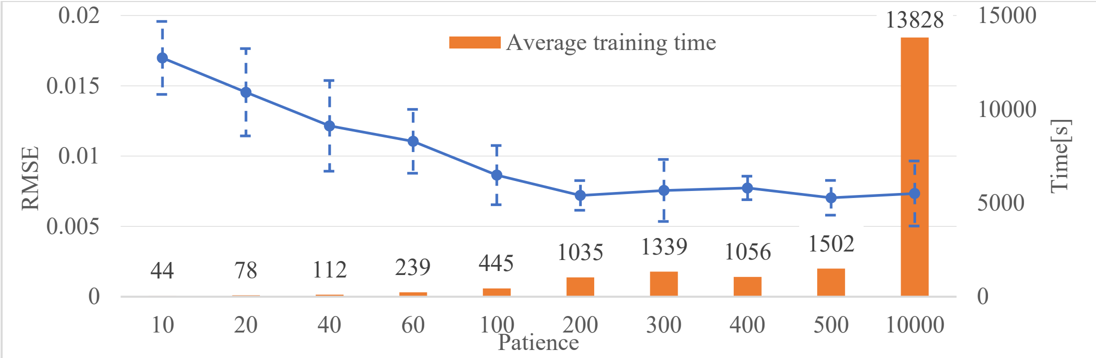

Figure 2. Average RMSE of various ANN architectures and average training time as a functions of patience parameter.

It can be noted that the early stopping mechanism significantly improves training time, while not affecting the average RMSE for *patience* higher than 200. However, the value of the standard deviation of RMSE (varying PINN architectures) is not negligible, justifying further research towards searching for the best network architecture for a specific problem.

4. Conclusions

We presented the PINN solution of a direct 1D Stefan problem with time dependent Dirichlet boundary conditions describing a melting process. The applied methodology exhibited stability and accuracy comparable to the accuracy of the existing numerical methods, offering more flexibility in various application areas. The results of all experiments confirmed that the PINN solution can be used effectively and efficiently for the selected problem, which makes further investigation towards the performance of the PINN method in solving inverse Stefan problem grounded on solid assumptions.

Acknowledgment: Part of this research was supported by The Ministry of Education, Science and Technological Development in Serbia (contract 451-03-68/2022-14/ 200122), Horizon 2020 programme (SILICOFCM Project, No 777204; SGABU Project, No 952603), EIT's HEI Initiative SMART-2M project, supported by EIT RawMaterials, funded by the European Union.

References

[1] Ivanovic, M., Svicevic, M., Savovic, S., Numerical solution of Stefan problem with variable space grid method based on mixed finite element/finite difference approach, INTERNATIONAL JOURNAL OF NUMERICAL METHODS FOR HEAT & FLUID FLOW, vol. 27 br. 12, str. 2682-2695, 2017.

[2] Raissi, M., Perdikaris, P. and Karniadakis, G.E., Physics informed deep learning (Part I): Data-driven solutions of nonlinear partial differential equations, arXiv:1711.10561, 2017.

[3] Raissi, M., Perdikaris, P., and Karniadakis, G. E., "Physics Informed Deep Learning (Part II): Data-driven Solutions of Nonlinear Partial Differential Equations", 1711.10566, 2017.

[4] Ehsan Haghighat, Ruben Juanes, SciANN: A Keras/TensorFlow wrapper for scientific computations and physics-informed deep learning using artificial neural networks, Computer Methods in Applied Mechanics and Engineering, Volume 373, 113552, 2021.