HIGH SCHOOL EDITION
Journal of Student Research
Volume 12 Issue 2 (2023)

Solving Partial Differential Equations for Physical and Chemical Problems Using Physics-Informed Neural Networks

Xiaorui Yang[1] and Haotian Chen[#]

[1]Beijing National Day School
[#]Advisor

ABSTRACT

Numerous physical and chemical problems at a high school level can be described by ordinary differential equations (ODEs) and partial differential equations (PDEs). However, the underlying equations troubled high school students because they often lack advanced mathematical skills, such as discrete calculus. Our goal is not to elaborate on those skills, but to offer a shortcut to the solution. In this paper, we demonstrated the use of Physics-Informed Neural Networks (PINNs), a neural network which solves the PDEs by incorporating the PDEs into the loss functions. The heat transfer equation and second order chemical kinetics are the two chosen model problems for high school seniors. Using PINNs, we were able to solve these two problems without recurring to university math. Hence, we strongly recommend peers to employ this method for physical or chemical problems for high school students and beyond.

Introduction

With the explosive growth of data and computing resources, Machine Learning (ML) algorithms represented by deep learning have generated revolutionary achievements in many disciplines, and text recognition is a prominent example. The Transformer (Vaswani et al., 2017), for instance, is a network architecture that is based on attention mechanisms and the encoder-decoder framework. Through two machine translation tasks, the model has been shown to be more parallelizable and to require significantly less training time. The model achieved 28.4 BLEU (bilingual evaluation understudy) on the WMT14 (Ninth Workshop on Machine Translation) English to German translation task and 41.0 BLEU on the English to French translation task after training on eight GPUs.

In the field of Computer Vision (CV), Swin Transformer (Liu et al., 2021) is a newly-proposed vision Transformer that is adapted from the previously mentioned Transformer for text recognition. Its representation is computed with shifted windows, producing higher efficiency by limiting self-attention computation and allowing cross-window connections. Its performance demonstrates compatibility with a wide range of vision tasks, including image classification, with 87.3 top-1 accuracy on ImageNet-1K.

In addition, this major accomplishment has greatly contributed to the domain of healthcare (Mohamadou, Halidou, & Kapen, 2020). The healthcare industry has been largely benefiting from the sharp rise of machine learning algorithms. In particular, certain algorithms can help diagnose and predict diseases by identifying patterns through continuous data training. COVID-19 (Ciotti et al., 2020), for instance, is a highly infectious disease caused by the SARS-CoV-2 virus. Starting a global epidemic, the virus may trigger side effects such as diabetes, heart disease, and asthma (Kim, Marrast, & Conigliaro, 2020). Presently, machine learning algorithms are used around the world to tackle this disease. For instance, meta-classifier (Classification via Regression (CR)) (Arpaci, Huang, Al-Emran, Al-Kabi, & Peng, 2021) techniques are utilized to predict

positive and negative cases with an accuracy of 84.21%. In addition, neural networks have also accelerated the detection of lung cancer from Computer Topography (CT) images. A recent report using a robust feature extraction method and a feed-forward neural network achieved an overall accuracy of 96.67% on 5 types of images (Miah & Yousuf, 2015). In short, conventional machine learning algorithms have demonstrated their effectiveness in the healthcare domain and many others.

However, a large quantity of data is essential for the training of the previously mentioned Machine Learning algorithms, which can be too costly or even impossible to obtain. For example, when optimizing high-entropy alloys, from equiatomic binary alloys to quinary alloys (Kaufmann & Vecchio, 2020) obtaining experimental data to a find correlation between metal composition and catalytic efficiency becomes a high-dimensional problem. In some academic disciplines such as drug discovery, design, and data-driven optimization of catalyst, etc, data generation, data curation, and data cleaning can be quite expensive or infeasible. In general, in a "small data" regime, data-driven neural networks are inevitably quantified by “small data,” and extrapolations are generally less acceptable.

A conventional neural network is often described as a black box model. Though neural networks are claimed to be able to approximate any function suggested by the universal approximation theorem (Lu & Lu, 2020) they failed to provide any insights empowering the understanding of the function being approximated – which makes the underlying difficult to rationalize.

Furthermore, traditional data-driven neural networks do not entirely satisfy the natural laws of physics described by the partial differential equations (PDEs) and are thus not capable of generating accurate results under the constraints of physics.

As discussed above, conventional machine learning methods are typically less robust and ignorant of the underlying physics laws. This would result in the need for larger quantity of data and often requires complex neural network structures. Hence, physics-informed neural networks (PINNs) (Raissi, Perdikaris, & Karniadakis, 2019), a pioneering universal function approximator that runs under the regularization of physical models described by partial differential equations (PDEs), are utilized in various academic disciplines. PINNs solve problems solely by constraining the neural network with PDEs so that experimental data is no longer mandatory. Conventionally, PDEs are solved using the finite difference (FD) and finite element (FE) methods, which require lengthy discretization of the system and a strong background in discrete calculus. PINNs, on the other hand, allow a discretization-free solution to PDEs, significantly lowering the barrier for solving PDEs, especially for high school students.

Power system engineers have benefited from PINNs’ comparable advantages of less training data requirement, a simpler structure, and higher accuracy. In the paper “Physics-Informed Neural Networks for Power Systems” (Misyris, Venzke, & Chatzivasileiadis, 2020), PINNs are reported to have an enhanced efficiency of 87 times faster duration when determining the frequency and rotor angle.

When solving inverse problems in relation to 3D supersonic flows, wake flows, and biomedical flows, PINN has proven its effectiveness in the domain of computational fluid dynamics (Cai, Mao, Wang, Yin, & Karniadakis, 2022). Problems such as high dimensionality, noisy data, and mesh generation are solved. It is able to integrate the experimental data under the formulations of the Navier Stokes Equation (NSE) for both imcompressible and compressible flows.

In this paper, we first demonstrate finite difference solution to heat transfer equation, which require lengthy derivation and the explicit finite difference method are not unconditional stable. Then, a PINNs solution to the same equation is proposed, although it requires a little bit more time to train the neural network, the discretization-free solution provided by PINNs saves a huge amount of time on mathematical derivations, allowing scientists to focus on setting up the science problem without having to excessively delve into mathematical textbooks. Furthermore, the PINN solution showed comparable accuracy with the FD method, and we suggest that PINNs provide an easier and possibly faster solution to physical problems that can be described with PDEs. After solving the heat transfer equation as an example of a high school physics problem, we further

apply PINNs to second order chemical kinetics like dimerization reaction to show that "physics-informed" neural networks can also be "chemistry-informed."

Theory

Heat Transfer Equation

$$\frac{\partial T}{\partial t} = \alpha\left(\frac{\partial^2 T}{\partial x^2}\right) \tag{1}$$

Were α is the thermal diffusivity ($m^2\ s^{-1}$). Solving a partial differential equation like Equation 1 conventionally involves the finite difference method (FDM) (Özişik, Orlande, Colaço, & Cotta, 2017) or the finite element method (FEM) (Wilson & Nickell, 1966). Applying FDM and FEM from scratch is often challenging for high school students without previous exposure to discrete calculus and programming. Commercial software like COMSOL is thus available, but its high demands for computational resources and meshing may still be a challenge. Its price tag may prevent high school students from utilizing it. To showcase the advantage of discretization-free simulation using PINNs, the problem setup is introduced here, followed by a finite difference solution using the explicit method and then the PINNs method for comparison in the next two sections. Note that for simplicity of illustration, we assume that all variables are dimensionless.

The initial condition at $t = 0$ is:

$$T = \pi \sin(x)\,, x \in [0,1] \tag{2}$$

The boundary conditions at $x = 0$ (left) or 1 (right) represents a heat sink:

$$T = 0, t \in [0,1] \tag{3}$$

The problem setup implies that the correct solution should observe the gradual dissipation of heat from high temperature to low temperature (the boundary), and as $t \to \infty$, $T = 0$ in all the spatial domains.

Solving Heat Transfer Equation Using Finite Difference Method

The explicit method discretizes the heat conduction equation:

$$\frac{T_i^k - T_i^{k-1}}{\Delta t} = \alpha\frac{T_{i-1}^{k-1} - 2T_i^{k-1} + T_{i+1}^{k-1}}{(\Delta x)^2} \tag{4}$$

where T_i^k represents the temperature at spatial step i and temporal step k. Δt and Δx are temporal and spatial step sizes.

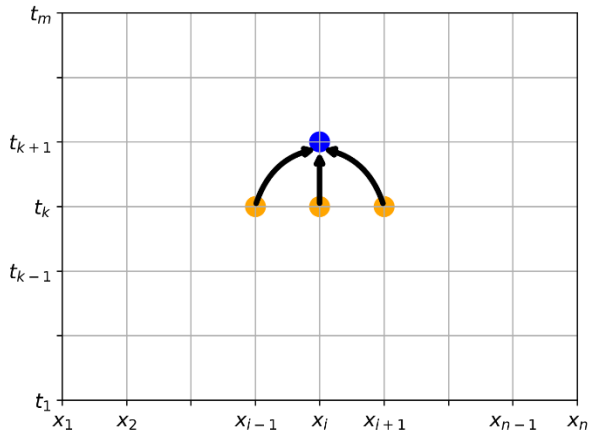

Figure 1. The scheme of explicit method. The solution at T_i^{k+1} (blue dot) depends on T_{i-1}^k, T_i^k and T_{i+1}^k (orange dots) where $k+1$ stands for the next time step.

Unfortunately, for explicit method to converge, Δt and Δx must satisfy the condition below (Thomée, 1990):

$$\lambda = \frac{\Delta t}{(\Delta x)^2} < 0.5 \tag{5}$$

In simulation, we use $\Delta x = 0.025$ and $\Delta t = 0.0002$ so that $\lambda = 0.32$. If the condition is not met, the solution will be unstable and diverging from the true solution as the temperature will oscillate widely in both spatial and temporal domain, possibly resulting in negative temperature which is certainly unrealistic. Consequently, to maintain accuracy, Δt must decrease with decreasing Δx, making simulation very inefficient as one may need small Δx to accurately represent a continuous function. To solve Equation 4, it is rearranged to the following form,

$$T_i^k - T_i^{k-1} = \alpha \frac{\Delta t}{(\Delta x)^2}\left(T_{i-1}^{k-1} - 2T_i^{k-1} + T_{i+1}^{k-1}\right) \tag{6}$$

which is then ultimately transformed to the following form as: $\lambda = \frac{\Delta t}{(\Delta x)^2}$.

$$T_i^k = T_i^{k-1} + \alpha\lambda\left(T_{i-1}^{k-1} - 2T_i^{k-1} + T_{i+1}^{k-1}\right) \tag{7}$$

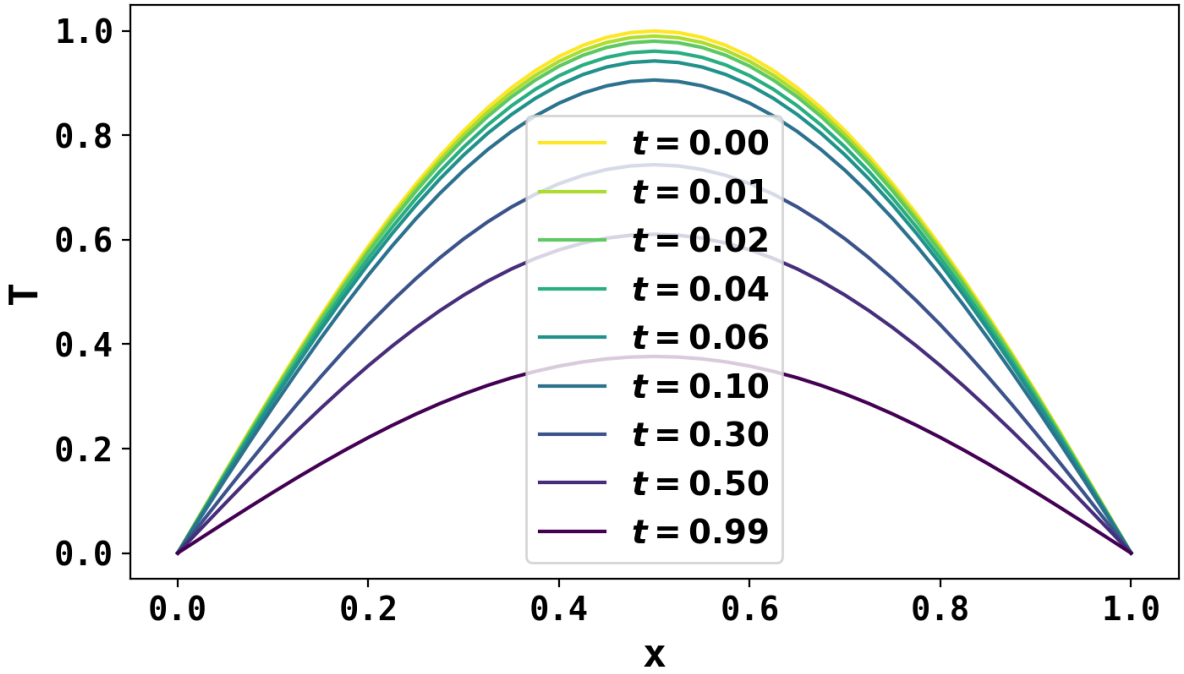

Figure 2. Solution of heat transfer equation using explicit finite element method which shows temporal dissipation of heat where $\alpha = 0.1$.

The temporal evolution of temperature is shown in Figure 2 as the solution to the heat transfer equation where the heat diffusivity is in dimensionless and $\alpha = 0.1$. With the progressing of time, the sinusoidal curves are becoming flatter, suggesting transfer of heat from high temperature to low temperature.

Solving Heat Transfer Equation Using PINNs

The heat transfer equation can also be solved using PINNs without any prior knowledge of discretization. Training of PINNs for the heat transfer equation requires prediction of the temperature evolution $T(t, x)$ as a function of time, t, and spatial coordinate, x. Note that these parameters are dimensionless. The heat transfer equation with its boundary and initial conditions is formulated in the following form:

$$\frac{\partial T}{\partial t} - \alpha\left(\frac{\partial^2 T}{\partial x^2}\right) = 0 \; on \; \mathcal{T} \times \Omega_x \tag{8}$$

$$T = 0 \; on \; \mathcal{T}, x = 0 \tag{9}$$

$$T = 0, \;\; on \; \mathcal{T}, x = 1 \tag{10}$$

$$C = \pi \sin(x) \, , on \; t = 0, x \in [0,1] \tag{11}$$

where $\mathcal{T} \in [0,1]$ and $\Omega_x \in [0,1]$ represents the temporal and one-dimensional spatial coordinates respectively. Equation 8 represents the heat transfer equation, Equation 9 and Equation 10 are the left and right boundaries respectively and Equation 11 is the initial condition at $T = 0$. Since there are four equations, four training datasets are required to train PINNs. To enforce Equation 8, a set of N collocation points $\{t_i, x_i\}_{i=1}^{N}$ is generated

using a random uniform distribution throughout the temporal spatial domain. Similarly, N collocation points are generated for the left and right boundaries, which are: $\{t_i, x = 0\}_{i=1}^{N}$ and $\{t_i, x = 1\}_{i=1}^{N}$. Similarly, the datasets for Equation 11 are $\{t = 0, x_i\}_{i=1}^{N}$. Figure 3 illustrates the four sets of collocation points generated using uniform random distribution.

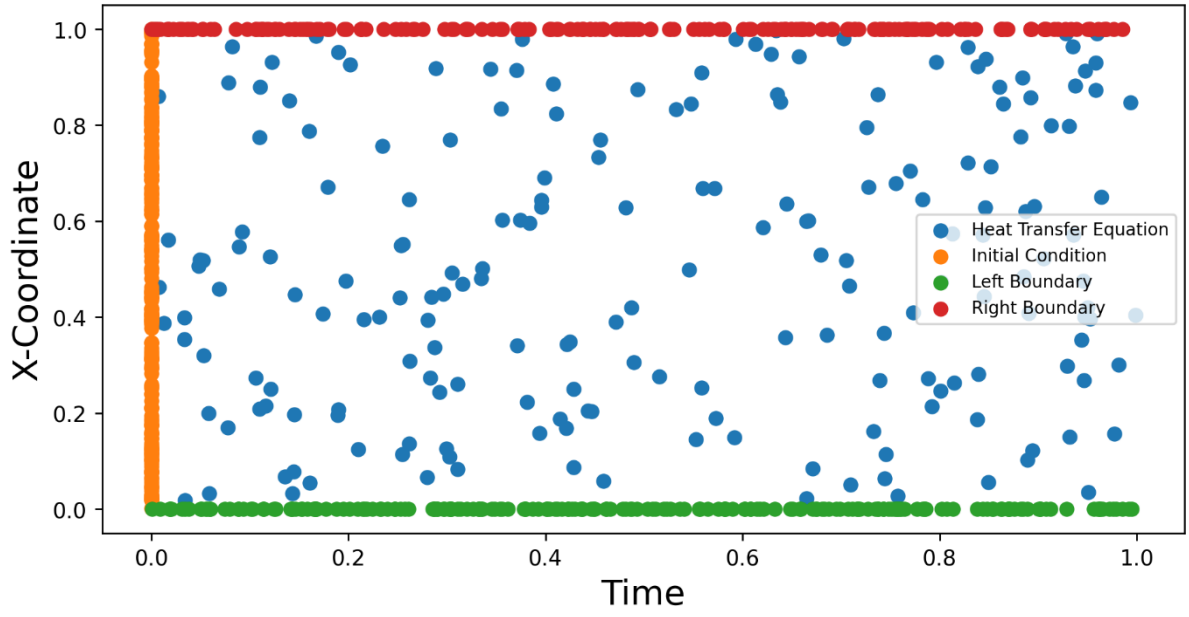

Figure 3. An illustration of collocation points used to solve the heat transfer equation. There are four sets of collocation points to enforce the heat transfer equation, initial condition, and boundary conditions.

Unlike conventional data driven neural network trained to predict $T(t, x)$ from enormous amount of known temperature at $\mathcal{T} \times \Omega_X$, with PINNs, a fully connected neural network is used to approximate the known concentration at the initial and boundary conditions. And more importantly, Equation 8 must be satisfied for every collocation point inside the $\mathcal{T} \times \Omega_X$ domain. To enforce the physical laws and conditions described in Equations 8 to 11, a loss function $\mathcal{T}$ is constructed as a linear combination of four mean square error (MSE) functions:

$$\mathcal{L} = w_1 MSE_f + w_2 MSE_{ini} + w_3 MSE_{bnd_1} + w_4 MSE_{bnd_2} \tag{12}$$

where w_j are weights of each loss function, which are hyperparameters for training to condition each MSEs with difference numerical scales. In this paper, all the weights are set to 1. The four MSEs are defined as:

$$MSE_f = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\partial T_i}{\partial t_i} - \alpha\left(\frac{\partial^2 T_i}{\partial x_i^2}\right)\right)^2 \tag{13}$$

$$MSE_{ini} = \frac{1}{N}\sum_{i=1}^{N}(T_i - 1)^2 \tag{14}$$

$$MSE_{bnd_1} = \frac{1}{N}\sum_{i=1}^{N}(T_i)^2 \tag{15}$$

$$MSE_{bnd_2} = \frac{1}{N}\sum_{i=1}^{N}(T_i)^2 \tag{16}$$

where MSE_f records the error of enforcing the heat transfer equation, MSE_{ini} records the error of enforcing initial conditions and the other two equations record the error of enforcing boundary conditions. When the errors are combined linearly to form $\mathcal{L}$, $\mathcal{L}$ then represents the "global" error of solving PDE. The error is then minimized with Adam optimizer (learning rate = 10^{-3}) to find the optimal solution and a mini-batch training is used to stabilize the training process.

After training, the neural network can predict $T(t,x)$ on the $\mathcal{T} \times \Omega_x$ domain. The predicted concentration profile is shown in the results section.

Second Order Chemical Kinetics

Not only to solve physics problem like heat transfer equation, PINNs can also solve chemistry problem like high order chemical kinetics. This section illustrates PINNs' solution to second order chemical kinetics to present PINNs as a future candidate for high school level chemical kinetics problems. Second order chemical kinetics, for example, dimerization of 2,5-dimethyl-3,4-diphenylcyclopentadienone as shown in Figure 4, example of $A + A \rightarrow B$, is solved using PINNs. The second order kinetics is described using an ordinary differential equation (ODE):

$$\frac{d[A]}{dt} = -k[A]^2 \tag{17}$$

where $[A]$ is the concentration of A and k is the second order reaction rate constant. Note A, t, and k are dimensionless. The equation is solved by rearrangement and integration of the form:

$$\int_{[A]_0}^{[A]_t} \frac{1}{[A]^2}\, d[A] = -k \int_0^t dt \tag{18}$$

So that the concentration at any time t is:

$$[A]_t = \frac{1}{\frac{1}{[A]_0} + kt} \tag{19}$$

which is used to validate the PINNs solution.

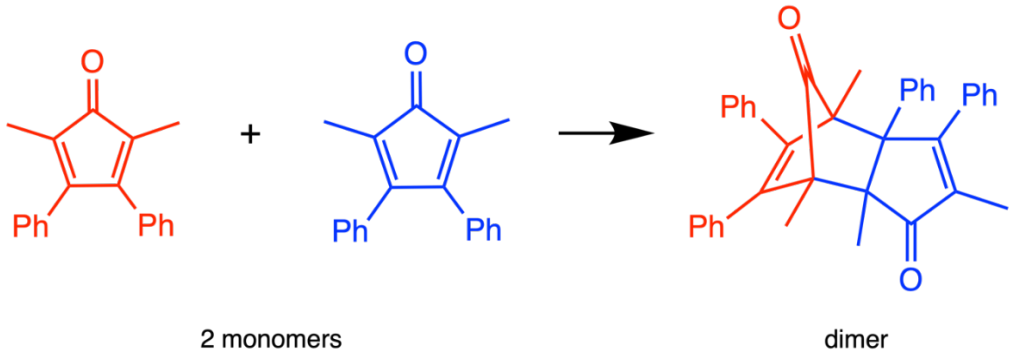

Figure 4. Dimerization of 2,5-dimethyl-3,4-diphenylcyclopentadienone, an example of second order chemical kinetics.

To solve the second order kinetics using PINNs, we incorporate the chemical kinetics and initial condition into the loss functions:

$$MSE_f = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\partial[A]}{\partial t} + k[A]^2\right)^2 \tag{20}$$

$$MSE_{ini} = \frac{1}{N}\sum_{i=1}^{N}([A] - 1) \tag{21}$$

Thus, the global loss function for the second order chemical kinetics:

$$\mathcal{L} = w_1 MSE_f + w_2 MSE_{ini} \tag{22}$$

Simulation Methods

The PINNs programs used as better alternatives to solve the heat transfer equation and chemical kinetics problems were written in Python 3.8.16 using the TensorFlow 2.92 library for neural networks. Both networks were trained for 400 epochs. The heat transfer equation was solved using the explicit finite difference method and the PINNs method. The second order chemical kinetic was solved using *odeint* in scipy library and PINNs. The former methods were the validation tool for the latter one, which also enabled the comparison of implementation time by the user. Figure 4 was created using Chemdraw 2020 version, and the other shown diagrams were drawn using Matplotlib.

Results and Discussion

PINNs Solution to Heat Transfer Equation

The solution of the heat transfer equation is shown in Figure 5. The contour plot at the top shows the temporal spatial evolution of temperature. The three concentration profiles at the bottom of Figure 5 show that the sinusoidal curves become flatter with the passage of time, suggesting that heat is dissipated from high temperatures to low temperatures, which agrees well with the results shown in Figure 2.

As both methods provide the correct solution, the merit of PINNs lies within the simplicity of implementation. In comparison with the finite difference method, PINNs saved a huge amount of time on mathematical derivations by providing a discretization-free solution to partial differential equations. Such an advantage is essential for high school students, as discrete calculus and numeric methods are only covered at the university level. Finding an alternative to PDEs is thus essential. Moreover, PINNs can solve high dimensional problems without significantly changing the implemented code. Fortunately, the recent appearance of PINN provides a viable and more accessible solution to PDEs, as it requires almost no background in advanced mathematics.

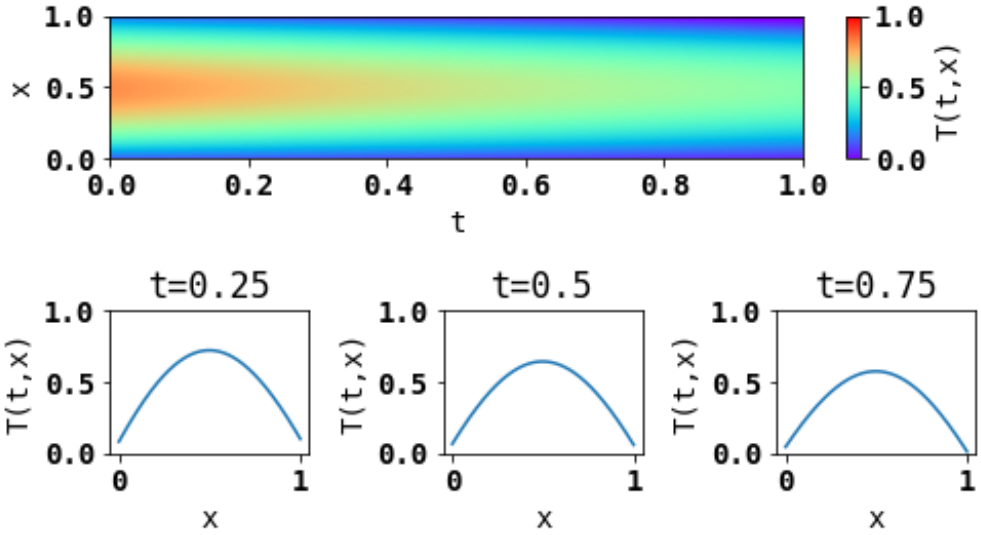

Figure 5. (Top) Temporal-spatial evolution of temperature solved by PINNs when $\alpha = 0.1$. (Bottom) The concentration profile at three time points: t=0.25, 0.5 and 0.75.

PINNs Solution to Chemical Kinetics

The PINN solution and the analytical solution to chemical kinetics are shown in Figure 6 when the second order reaction rate constant is $k = 0.1$. The figure showed that PINN mostly agreed with the analytical solution at the start and end of simulation, while slightly deviates from the analytical solution when $0 < t < 1$. In general, PINNs has successfully predicted the temporal evolution of concentration for a second order kinetics. Thus, PINNs are also viable tools to solve high school chemistry problems, especially nonlinear chemical kinetics such as the second order reaction shown in Figure 6.

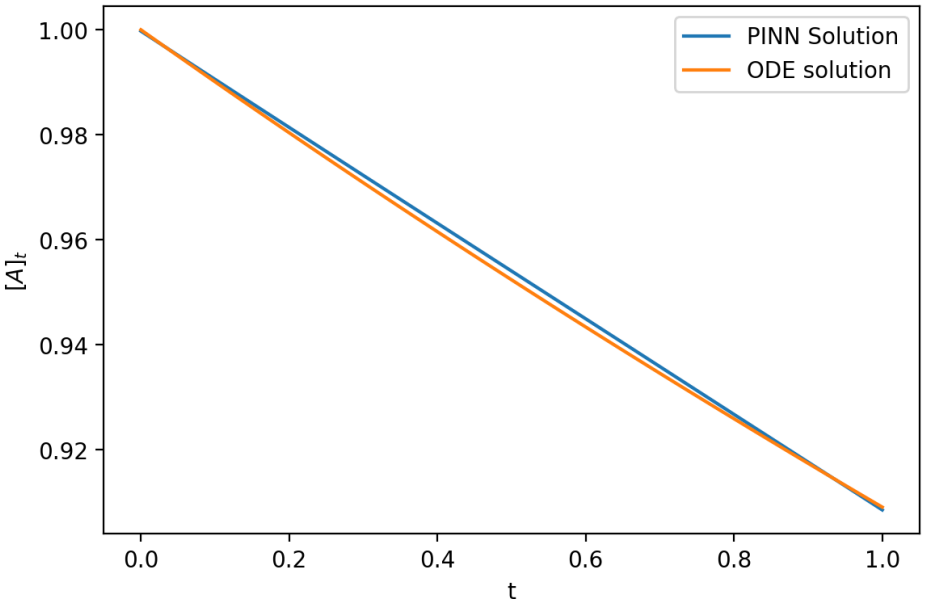

Figure 6. Solution to second order chemical kinetics when $k = 0.1$. The PINN and ODE solution are shown in blue and orange respectively.

Conclusion

In this paper, we demonstrated the applications of PINNs to solve physical and chemical problems regularized by partial and ordinary differential equations. The 1D temporal-spatial evolution of heat transfer and second order chemical kinetics are solved successfully. The successes highlight the applicability of PINNs in most scientific disciplines, which can be described with PDEs. PINNs is still at its infant age, and constitute a growingly active area of research (Cuomo et al., 2022). The paper offers readers the starting points of physics-informed machine learning for science, and we highly recommend applying this method in other scientific contexts.

Acknowledgment

I would also like to thank my family members for their continuous support over the period.

References

Arpaci, I., Huang, S., Al-Emran, M., Al-Kabi, M. N., & Peng, M. (2021). Predicting the COVID-19 infection with fourteen clinical features using machine learning classification algorithms. *Multimedia Tools and Applications, 80*(8), 11943-11957.

Cai, S., Mao, Z., Wang, Z., Yin, M., & Karniadakis, G. E. (2022). Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mechanica Sinica*, 1-12.

Ciotti, M., Ciccozzi, M., Terrinoni, A., Jiang, W.-C., Wang, C.-B., & Bernardini, S. (2020). The COVID-19 pandemic. *Critical reviews in clinical laboratory sciences, 57*(6), 365-388.

Cuomo, S., Di Cola, V. S., Giampaolo, F., Rozza, G., Raissi, M., & Piccialli, F. (2022). Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next. *arXiv preprint arXiv:2201.05624*.

Kaufmann, K., & Vecchio, K. S. (2020). Searching for high entropy alloys: A machine learning approach. *Acta Materialia, 198*, 178-222.

Kim, E. J., Marrast, L., & Conigliaro, J. (2020). COVID-19: magnifying the effect of health disparities. *Journal of general internal medicine, 35*(8), 2441-2442.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., . . . Guo, B. (2021). *Swin transformer: Hierarchical vision transformer using shifted windows.* Paper presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision.

Lu, Y., & Lu, J. (2020). A universal approximation theorem of deep neural networks for expressing probability distributions. *Advances in neural information processing systems, 33*, 3094-3105.

Miah, M. B. A., & Yousuf, M. A. (2015). *Detection of lung cancer from CT image using image processing and neural network.* Paper presented at the 2015 International conference on electrical engineering and information communication technology (ICEEICT).

Misyris, G. S., Venzke, A., & Chatzivasileiadis, S. (2020). *Physics-informed neural networks for power systems.* Paper presented at the 2020 IEEE Power & Energy Society General Meeting (PESGM).

Mohamadou, Y., Halidou, A., & Kapen, P. T. (2020). A review of mathematical modeling, artificial intelligence and datasets used in the study, prediction and management of COVID-19. *Applied Intelligence, 50*(11), 3913-3925.

Özişik, M. N., Orlande, H. R., Colaço, M. J., & Cotta, R. M. (2017). *Finite difference methods in heat transfer*: CRC press.

Raissi, M., Perdikaris, P., & Karniadakis, G. E. (2019). Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics, 378*, 686-707.

Thomée, V. (1990). Finite difference methods for linear parabolic equations. *Handbook of numerical analysis, 1*, 5-196.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., . . . Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems, 30*.

Wilson, E. L., & Nickell, R. E. (1966). Application of the finite element method to heat conduction analysis. *Nuclear engineering and design, 4*(3), 276-286.