Thermal Modeling Using Physics-Informed Neural Networks

by
Zach Kutschke

Submitted to the
Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

February 2021

Signature of Author _____
Department of Mechanical Engineering
December 7, 2020

Certified by _____
Tomasz Wierzbicki
Professor of Mechanical Engineering
Thesis Supervisor

Accepted by _____

Thermal Modeling Using Physics-Informed Neural Networks

by
Zach Kutschke

Submitted to the Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of
Bachelor of Science in Mechanical Engineering

ABSTRACT

Finite element methods are the industry standard for analyzing a plethora of modeling problems. However, these methods may become highly computationally expensive or even nonconvergent in the face of more complicated loading situations and/or coupled multi-physics problems. For instance, battery and electric car manufacturers alike may wish to model failure via internal short circuit (ISC) of a battery pack. Such a model may consist of the battery's mechanics, thermal properties, and electrochemistry, all of which are coupled together and would likely cause issues for a standard finite element analysis (FEA) package.

In order to supplement the accuracy of FEA with further robustness and speed, this project sought to explore the use of physics-informed neural networks (PINNs) to solve coupled multi-physics problems specifically related to lithium-ion battery internal short circuits. PINNs were generated for thermal models of increasing complexity up to a 2-dimensional unsteady anisotropic model. The accuracy for most PINNs was comparable to their respective analytical and FEA results.

Thesis Supervisor: Tomasz Wierzbicki
Title: Professor of Mechanical Engineering

Acknowledgements

I would like to thank the Impact and Crashworthiness Lab as a whole for taking me in over the past several months and including me in their research and discussions. I feel I have gained an immense amount from this project and from the lab as a whole. I would also like to thank Juner Zhu, Wei Li, and Tomasz Wierzbicki for all of the help and support they had given me to pursue this project, especially in a period of time as difficult as the one we found ourselves in.

1. Introduction

1.1 Motivation

The use of lithium-ion (Li-ion) batteries and battery packs has been on a sharp rise primarily with increases in the production of electric vehicles. Of prime concern to battery and vehicle designers and manufacturers is the safety and potential failure mechanisms of these batteries, since failure could result in an uncontrollable release the high quantity of energy stored in them.

Li-ion cells are generally sensitive to mechanical, thermal, and electrical mistreatment. Severe deformations, overheating, and overcharging/overdrawing have been known to induce rapid and often catastrophic failure typically resulting in smoke, flames, chemical leakage/spraying, and potentially even explosions. There may be thousands of individual cells in the large format packs found in electric vehicles, vastly increasing the likelihood of incorporating a cell that is defective or was damaged during some manufacturing/handling step. Such cells are severely problematic for a pack since if they overheat (say, due to an internal short circuit) and the physical pack carrying the cells isn't designed properly, then the heat produced by the single cell may induce thermal runaway whereby neighboring cells undergo thermal failure and so on until the entire pack has essentially caught fire [1].

The physics surrounding these batteries and their failure modes is typically very complex. An in-depth analysis of the mechanics of a singular cell may involve some combination of plate bending and poromechanical theory. Furthermore, these mechanics are likely influenced by the cell's electrochemical and thermal states, both of which are also interdependent making the complete analysis very complicated.

The finite element method/analysis (FEM/FEA) is one of the most widely used algorithms for solving such mechanics and multi-physics problems, though with increasingly complex and interconnected loading situations come FEA simulations that either take on the order of days to complete or are entirely nonconvergent. In order to get around the computational expense and potential convergence errors that accompany the use of FEA on large/complicated simulations, it may be worthwhile to explore the use of machine learning techniques and solvers.

1.2 Physics Informed Neural Networks

Machine learning is a subset of artificial intelligence and broadly involves taking a statistical model of a set of data to be analyzed and refining it based on some set of training data. Neural networks are a subfield within machine learning techniques with the distinguishing feature being in the structure of the data being fed into the model and the processing. A classical machine learning algorithm relies on supervised learning, whereby all of the data being fed into the

algorithm has to be labeled such that it may learn to classify data based directly on what's fed into it. Conversely, neural networks may use unsupervised training where the data contains no labels. Instead the algorithm finds patterns and clusters data together according to those patterns. With supervised learning, a relatively accurate model can usually be achieved with less data than with unsupervised learning, though typically the process of labeling data is a very laborious [2].

Taking a closer look at neural networks, they are structured such that data is fed into several nodes dubbed neurons as shown in Figure 1. Each path between neurons has its own weight that the neuron's output is multiplied by. Each neuron in turn sums all of the weighted inputs it sees, adds a bias to that sum, and then passes the entire linear combination through an activation function. True to its name, this activation function takes the neuron output and typically maps it either to a [0,1] or [-1,1] range with an emphasis on either end of the range.

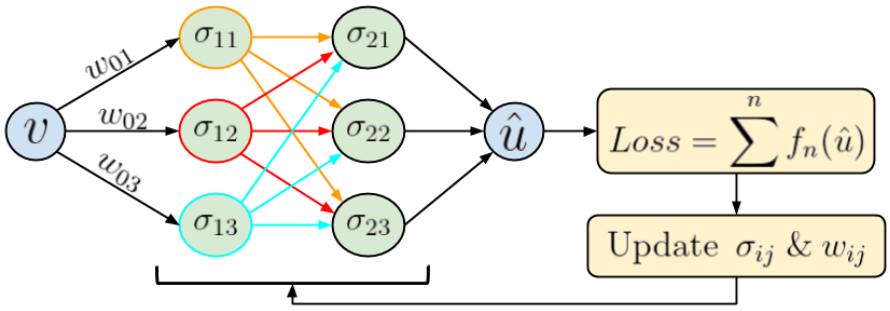

Figure 1. General form of a neural network during training. This network is 3 neurons wide and 2 deep, though they can be larger or smaller with layers of varying width. There are weights assigned to every edge between each neuron σ_{ij}, input v, output $\hat{u}$ though they're only shown on the first 3 edges for clarity.

Prior to training, the weights and biases are initialized by sampling from a specified probabilistic distribution. During training, data is sent through the network and culminates to an estimated output, which is then sent through a loss function. The model's entire job revolves around adjusting each of the weights and biases in the neural network to minimize the total loss calculated, though the means it actually goes about doing this is out of the scope of this project.

The total loss may either be constructed from known data by comparing the model output to experimental data, or it may be formed with a loss function or collection of functions. The latter of these forms the basis of physics informed neural networks (PINNs). If there is sufficient information known about the physics of the situation, then those equations and requirements can be used to tune the neural network. This typically takes the form of using the overall partial differential equations (PDEs) and accompanying initial and boundary conditions of the physics involved.

2. Battery Physics and Modelling Goals

A Li-ion cell may be generalized as consisting of 5 repeating layers of material wrapped together in different configurations as such as the cylindrical battery shown in Figure 2. An internal short circuit (ISC) in the context of Li-ion batteries is roughly defined as anytime the anode and cathode are in direct contact with one another. Given this broad definition, ISC can be shown to be an almost universal feature in cell failures, whether due to mechanical deformation or puncturing bringing the anode and cathode together, electrical overcharging/overdrawing causing dendritic growth that then pierces the separator layer, or excess heat collapsing the separator [3]. While direct external damage to a cell may be an obvious failure mode, Li-ion batteries may also contain manufacturing defects (namely material contamination) that can put a cell at risk for spontaneous ISC [3]. These spontaneous failures are far harder to screen out and directly protect against, necessitating some form of preventative design on part of the entire pack and cell layout.

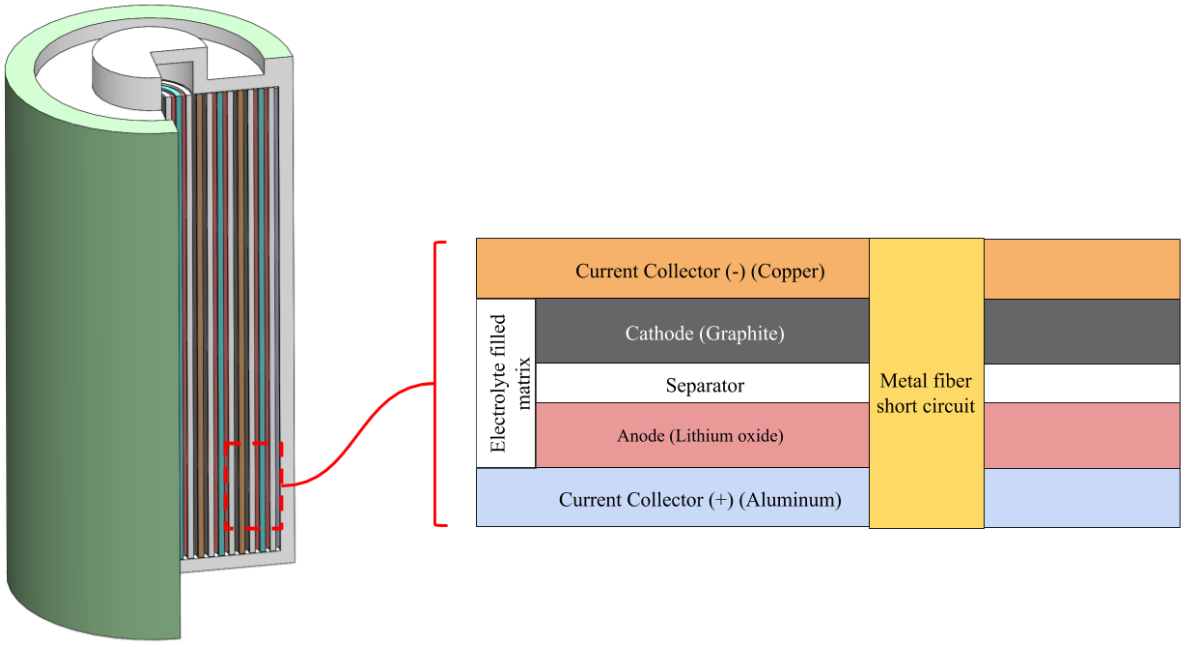

Figure 2. Cross section of a singular layer of a typical Li-ion cell with a manufacturing contamination in the form of a metal fiber shorting between the current collectors.

The modelling framework setup in this project is with ISC failures in mind, specifically spontaneous ISC from contamination in the form of metal fibers shorting layers of the cell. Insufficient progress was made towards a PINN for the electrochemical interaction within a cell, so the physics and equations will be discussed here only as they may inform some of the inputs used for the thermal modelling. The electric potential/ charge potential across the current collectors

were assumed to be constant in order to simplify finding rough estimates for the heat flux and temperatures to expect [4]:

$$I_{SC} = \frac{\phi_s|_{ca,cc} - \phi_s|_{an,cc}}{R_{SC}} \tag{1}$$

$$R_{SC} = \frac{L_{SC}}{\sigma_{SC}\pi r_{SC}^2} \tag{2}$$

$$Q_{SC} = I_{SC}^2 R_{SC} \tag{3}$$

Where $\phi_s|_{ca,cc}$ and $\phi_s|_{an,cc}$ are the surface potentials at the cathode and anode current collectors and the SC subscript denotes properties of the ICS fiber. From this and some preliminary models made in COMSOL, it seemed reasonable to set the ISC fiber interface as a surface temperature in the range of 400-600K.

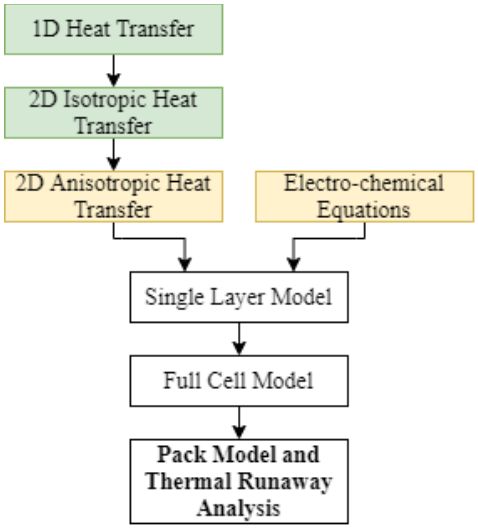

Figure 3. Overall flow of modelling progression and milestones towards a full pack model for thermal runaway analysis using coupled thermal and electrochemical networks.

The goal of this project was to create a PINN based model to simulate thermal runaway and preventative strategies in a Li-ion battery pack. As such, separate neural networks were built up for the thermal and electrochemical interactions in the cells and pack (though as mentioned the latter was not sufficiently progressed to report on). The thermal model was built by sequentially adding layers of complexity to the physics being modelled since tuning the neural network to each situation turned out to be an arduous task.

3. 1D Unsteady Heat Transfer

The starting point of this project was to successfully model the transient response of a semi-infinite 1-dimensional slab of material instantaneously exposed to a constant temperature surface with a PINN.

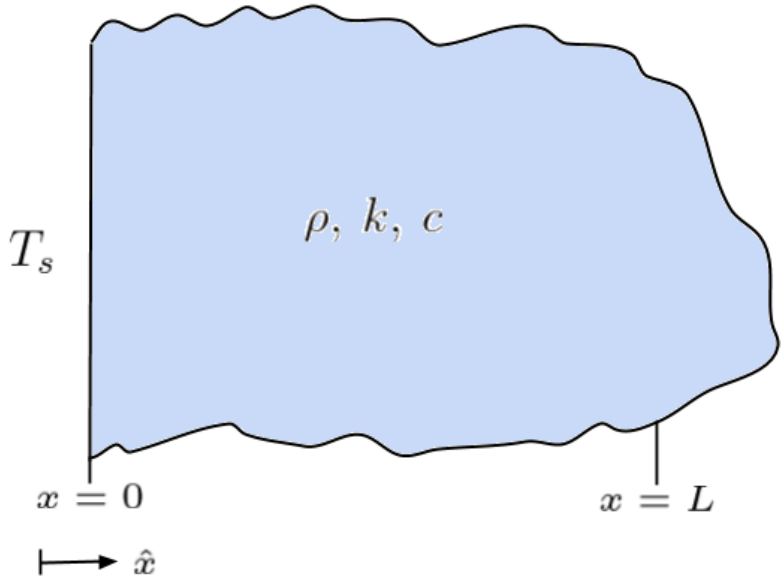

Figure 4. Overview of the 1D semi-infinite slab model. The side at $x = 0$ is instantaneously exposed to temperature T_s.

A 5x5 (5 neuron wide and 5 neuron deep) neural network was setup using the PyTorch Python module. The network's weights were initialized using an 'Xavier uniform' distribution, which is essentially a uniform distribution sampled from a range that's a function of the preceding neuron layer's width. This has been shown to help with convergence in deep networks (deep meaning greater than 2 layers thick) over typical uniformly or normally distributed initialization values since their interaction with the chosen activation function is less prone to prematurely saturating neurons [5]. The biases were initialized with a normal distribution. The loss functions used were simply taken for the 1-dimensional form of the heat equation and its initial and boundary conditions:

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T = \alpha \frac{\partial^2 T}{\partial x^2} \tag{4}$$

$$T(t = 0, x) = T_0 \tag{5}$$

$$T(t, x = 0) = T_s \tag{6}$$

$$T(t, x = L) = T_0 \tag{7}$$

Where L was taken far enough away such that the hypothetical material remained unaffected during the period of interest, and with $T_0 = 293\text{K}$ and $T_s = 500\text{K}$. Even after working out a number of initial bugs to include not properly normalizing the input and output of the PINN to a [0,1] range and various errors in assembling training data around the initial and boundary conditions, the initial results for this model were not at all ideal. After 3000 epochs the results looked similar to Figure 6. This initial attempt was done with a tanh activation function, the next optimization taken was to change this to something that might be more suitable to the network given that the inputs and output had been normalized to a [0,1] range. Both the sigmoid and ReLU functions were used and also found to perform quite poorly.

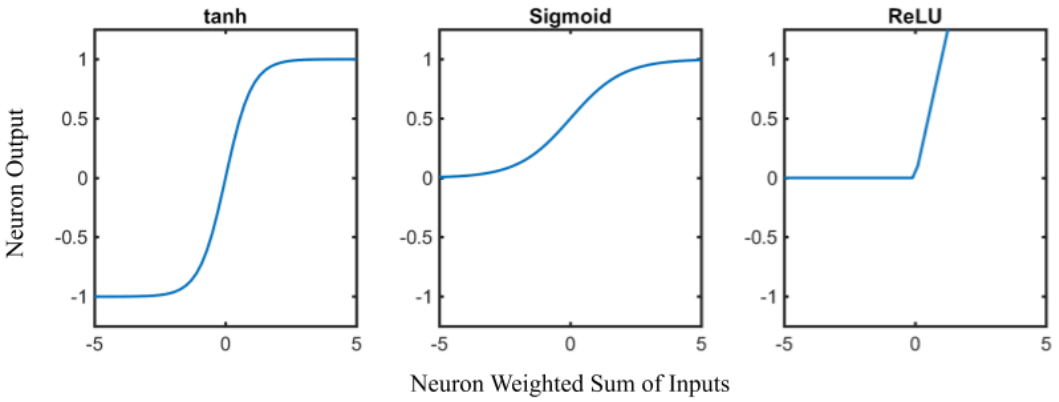

Figure 5. Different activation functions. The sigmoid function is defined as $S(x) = [1 + e^{-x}]^{-1}$, and the Rectified Linear Unit (ReLU) function attenuates all inputs less than 0 and then assumes a direct linear relation thereafter.

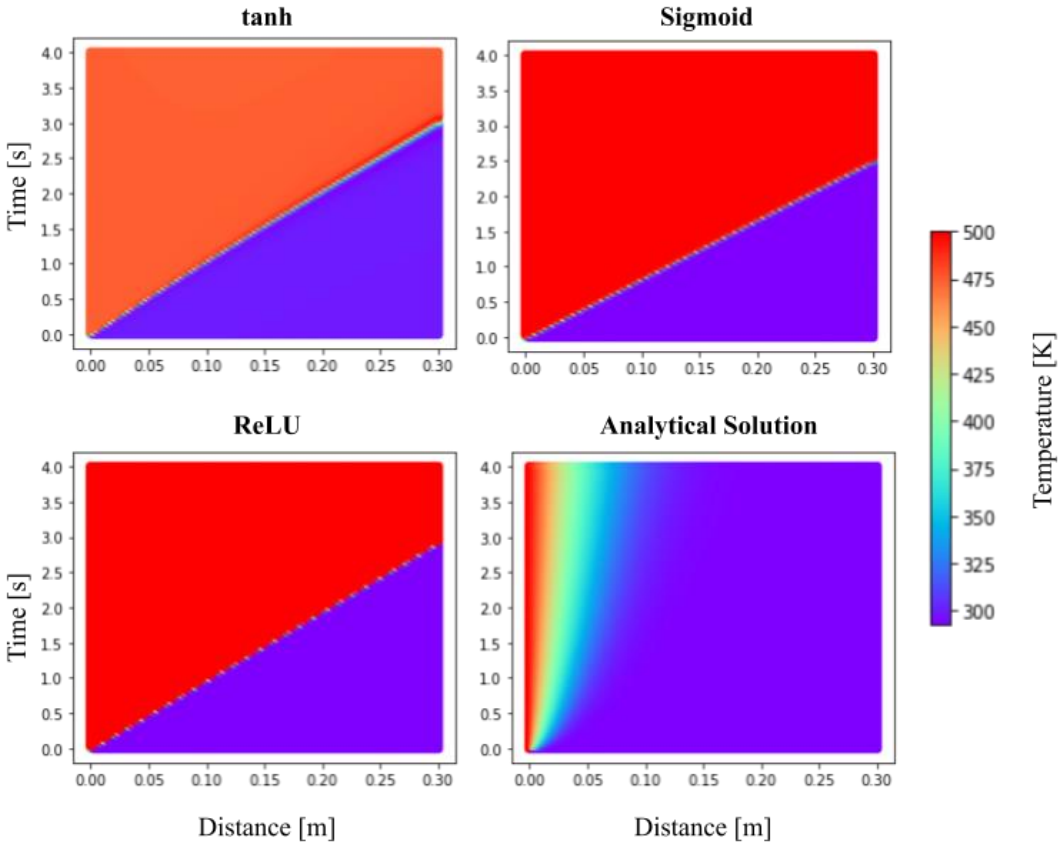

Figure 6. The initial attempts with at the semi-infinite 1D heat transfer model with three different activation functions shows almost no gradient and only seems to have roughly met the boundary conditions.

In addition to changing the activation function, different distributions of biases and weights were tried and different distributions of training data were tried (both on their own and in combinations) to no avail. Another loss component based on the total energy change of the system was also attempted using the heat flux into the model:

$$\Delta U = \int_{t_0}^{t_f} \dot{Q}_{SC}\, dt = A \int_{t_0}^{t_f} \dot{q}(x = 0, t)\, dt$$

$$\Delta U = m\, c\, \Delta T = A\, c\, \rho \int_0^{\mathrm{L}} (\mathrm{T(x, t = t_f)} - \mathrm{T_0})\, \mathrm{dx}$$

$$\int_{t_0}^{t_f} \dot{q}(x = 0, t)\, dt = \; c\, \rho \int_0^{\mathrm{L}} (\mathrm{T(x, t = t_f)} - \mathrm{T_0})\, \mathrm{dx} \qquad (8)$$

This was inspired by previous work on applying PINN to plate bending that had suggested that using a set of energy-based loss functions may be more efficient and accurate than PDE-based functions [6]. This approach did not work well here but was implemented on a global scale in addition to PDE-based loss functions instead of alone and in a localized fashion as it had been in the aforementioned study.

Next, an attempt was made at weighting some of the various loss functions in the hopes that the unsavory portions of the previous results could be tuned. The lack of a visible gradient and the overextension of the 500K region led to incremental order of magnitude upregulation of the PDE loss component (4) and similar downregulation of the loss component from the boundary condition at $x = 0$ (6). The sigmoid activation function was kept since it seemed to make the most logical sense given the range of the network inputs and outputs. The results continually converged onto the analytical solution as shown in Figure 7.

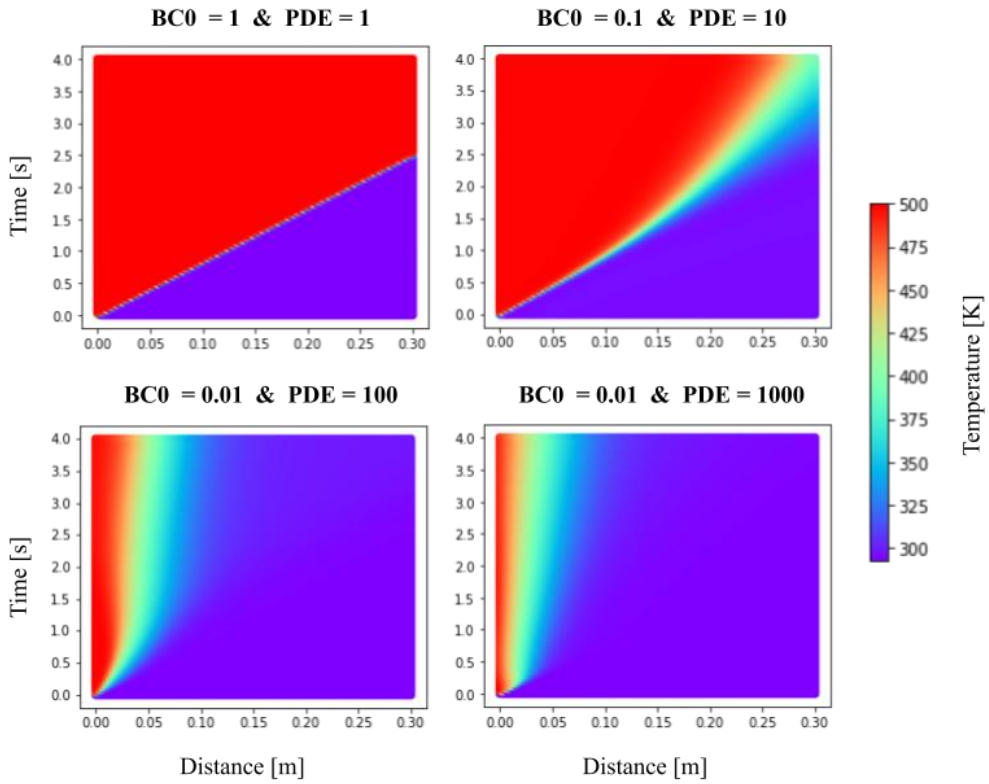

Figure 7. Resultant temperature distributions with different weights on the $x = 0$ boundary condition loss component (BC0) and the overall PDE loss component.

4. 2D Unsteady Heat Transfer

4.1 2D Isotropic Heat Transfer

The 2-dimensional isotropic model was taken as a slab of infinite depth but finite height and width, and also with a heat transfer pathway applied to one of its sides. The heat transfer coefficient was mostly set to 2,000 $\frac{\mathrm{W}}{\mathrm{m^2K}}$ so as to force a visibly noticeable gradient to appear in the transient response.

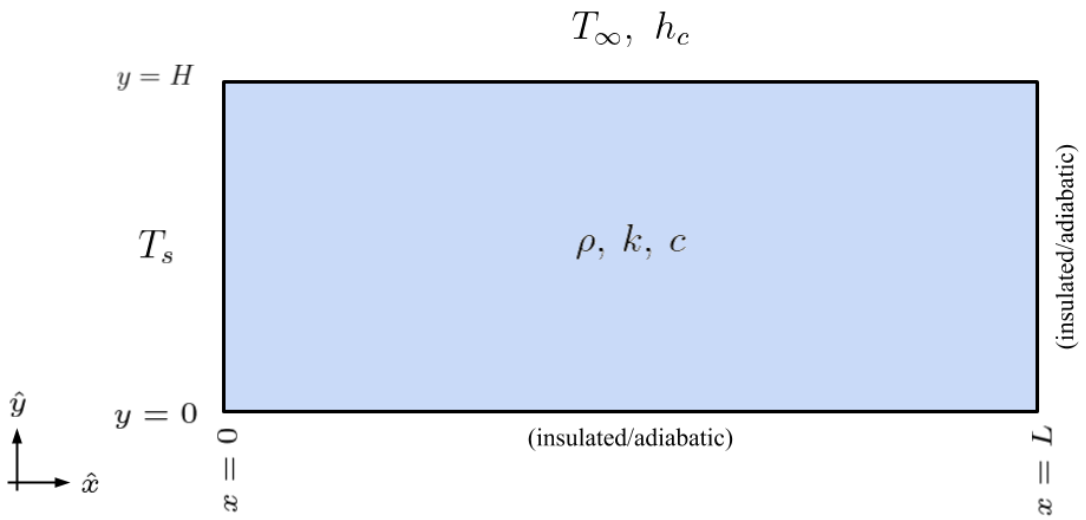

Figure 8. Overview of the 2D isotropic semi-infinite slab model. The side at $x = 0$ is instantaneously exposed to temperature T_s and the side at $y = H$ is exposed to a fluid at T_∞ with heat transfer coefficient h_c. The other sides are insulated and the model extends into the page infinitely.

The loss functions were set once again to the overall PDE, initial, and boundary conditions. The other network/model parameters were kept the same as they were in the 1-dimensional model.

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \tag{9}$$

$$T(t = 0, x, y) = T_0 \tag{10}$$

$$T(t, x = 0, y) = T_s \tag{11}$$

$$\left. \frac{\partial \mathrm{T}}{\partial \mathrm{x}} \right|_{\mathrm{t,x=L,y}} = 0 \tag{12}$$

$$\left.\frac{\partial \mathrm{T}}{\partial \mathrm{y}}\right|_{\mathrm{t,x,y=0}} = 0 \tag{13}$$

$$\left.\frac{\partial \mathrm{T}}{\partial \mathrm{y}}\right|_{\mathrm{t,x,y=H}} = \mathrm{h_c}(T(t,x,y=H) - T_\infty) \tag{14}$$

The results of this model were not nearly as promising as with the 1-dimensional model. The initial attempt without loss weighting did show a clearer gradient but its shape/direction was not totally correct when compared to a COMSOL model of the same situation.

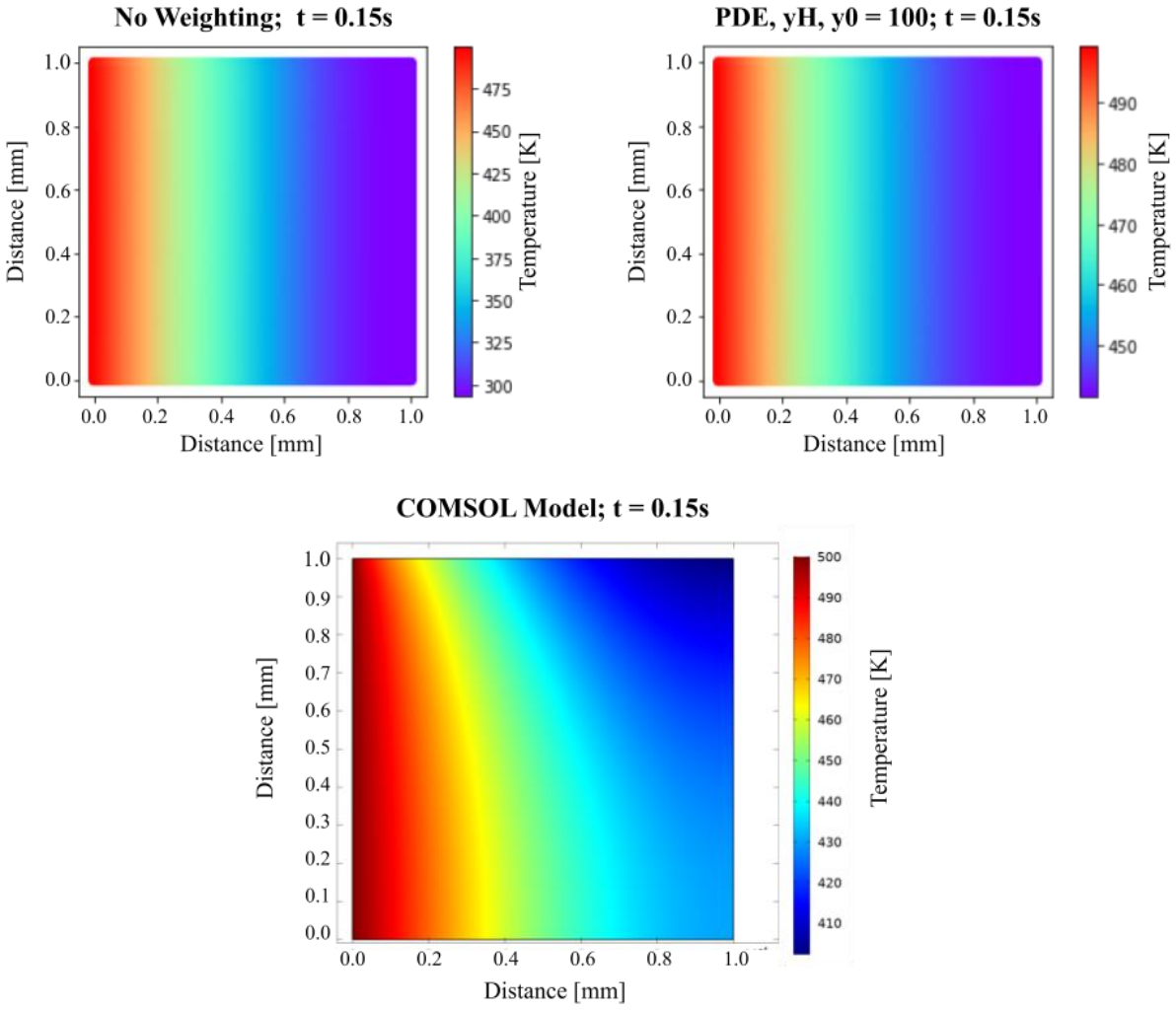

Figure 9. Temperature distributions at $t = 0.15s$. There was little difference seen in the gradient going from no loss weighting to weighting the overall PDE by 100, the boundary condition at $y = H$ (yH) by 100, and the boundary condition at $y = 0$ by 100 though the temperature did shift favorably.

Enforcing weights on the overall PDE as well as on the $y = 0$ (13) and $y = H$ (14) boundary conditions did not seem to correct for this or really produce much of any noticeable effect in the model other than to shift the overall temperatures somewhat closer to the COMSOL model. Shifting the learning rate, network size, and training data volume also had little to no positive effect of the predicted distribution.

4.2 2D Anisotropic Heat Transfer

Though the isotropic 2-dimensional model did not show very accurate results, a 2-dimensional anisotropic model was still attempted. The model was meant to be a fairly close approximate of a single layer in a Li-ion battery. The layout is shown in Figure 10, and the material properties for each layer roughly corresponds to those in actual Li-ion batteries [7].

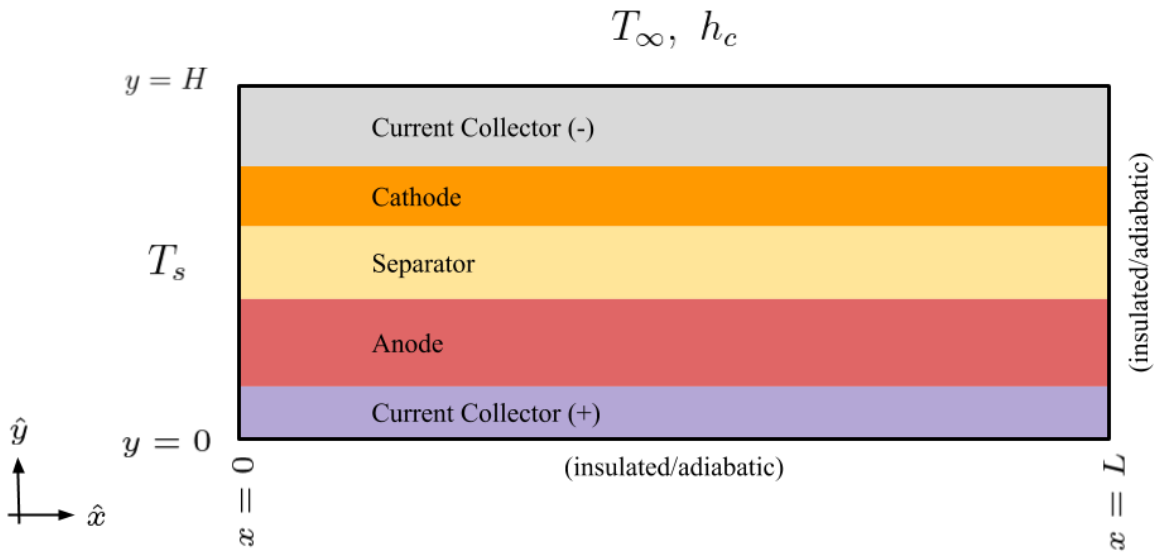

Figure 10. Overview of the 2D anisotropic semi-infinite slab model. The side at $x = 0$ is instantaneously exposed to temperature T_s and the side at $y = H$ is exposed to a fluid at T_∞. The other sides are insulated and the model extends into the page infinitely.

The only difference in the loss functions was that PDE functions for each region n within the body were added. The overall geometry was also changed to more closely match those of an actual battery. The other network parameters from the isotropic model and 1-dimensional model were still retained here.

$$\left.\frac{\partial T}{\partial t}\right|_n = \alpha_n \nabla^2 T_n = \alpha_n \left(\left.\frac{\partial^2 T}{\partial x^2}\right|_n + \left.\frac{\partial^2 T}{\partial y^2}\right|_n \right) \tag{15}$$

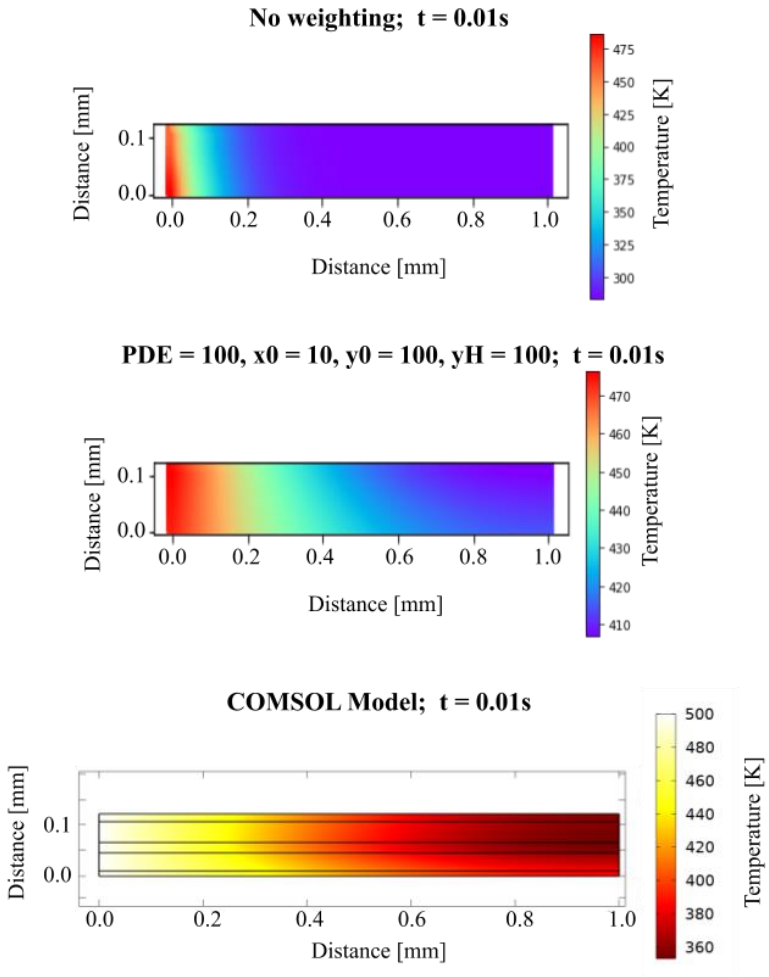

Figure 11. Temperature distributions at $t = 0.01s$. There did seem to be benefit in going from no loss weighting to weighting the overall PDE by 100, the boundary condition at $y = H$ (yH) by 100, the boundary condition at $y = 0$ by 100 and the boundary condition at $x = 0$ by 10.

Surprisingly, with some adjustment to the loss weights the gradient of the PINN model looks much closer to the COMSOL model than the weighted isotropic PINN model. The temperatures are still slightly off, though with some further tuning it seems probable that the model could form a good fit.

5. Conclusion

In this project, physics informed neural networks were established for thermal modelling situations of increasing complexity in the hopes of modelling an internal short circuit in a lithium-ion battery. The governing partial differential equations and their initial/boundary conditions were utilized as loss functions for each model. Though it was found that the neural networks were initially very limited in their ability to accurately capture the full features and values of the temperature gradients present in the models, it was found that weighting various components of the loss functions during training led to accurate estimations in most networks. These components had to be weighted by nearly 1000x in some cases in order to gain the desired level of accuracy. These networks thus far seem to still have a lot of potential in supplementing FEA in modelling complex and/or multi-physics models and simulations, but the process of tuning them to the model is still not entirely obvious.

References

[1] Feng, X.; He, X.; Ouyang, M.; Lu, L.; Wu, P.; Kulp, C.; Prasser, S.; 2015, "Thermal runaway propagation model for designing a safer battery pack with 25 Ah $LiNi_xCo_yMn_zO_2$ large format lithium ion battery," Applied Energy, Vol. 154, pp 74-91.

[2] Kavlakoglu, E., 2020, "AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?." From https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks.

[3] Feng, X.; Ouyang, M.; Liu, X.; Lu, L.; Xia, Y.; and He, X.; 2018, "Thermal runaway mechanism of lithium ion battery for electric vehicles: A review," Energy Storage Materials, Vol. 10, pp 246-267.

[4] Kim, J.; Mallarapu, A.; and Santhanagopalan S.; 2020, "Transport Processes in a Li-ion Cell during an Internal Short-Circuit," Journal of Electrochemical Society, Vol. 167 No. 9.

[5] Glorot, X. and Bengio, Y., 2010, "Understanding the difficulty of training deep feedforward neural networks," Journal of Machine Learning Research, Vol. 9, pp249-256.

[6] Li, W.; Bazant, M.; and Zhu, J.; 2020, "A Physics-Guided Neural Network Framework for Elastic Plates: Comparison of Governing Equations-Based and Energy-Based Approaches,"

[7] Gu, W. B. and Wang, C. Y., 2000, "Thermal-Electrochemical Coupled Modeling of a Lithium-Ion Cell," Department of Mechanical Engineering & Pennsylvania Transportation Institute, Pennsylvania State University.