

## Preliminary Analysis for Applicability of Physics Informed Neural Networks (PINNs) on Water Flow Analysis in Tanks

Cheolwoong Kim <sup>a</sup>, Joon Young Bae <sup>a</sup>, Chang Hyun Song <sup>a</sup>, Jinho Song <sup>a</sup>, Joongoo Jeon <sup>c,d\*</sup>, and Sung Joong Kim <sup>a,b\*</sup>

<sup>a</sup> Department of Nuclear Engineering, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul 04763, Republic of Korea

<sup>b</sup> Institute of Nano Science and Technology, Hanyang University, 222 Wangsimni-ro, Seongdong-gu, Seoul 04763, Republic of Korea

<sup>c</sup> Graduate School of Integrated Energy-AI, Jeonbuk National University, 567, Baekje-daero, Deokjin-gu, Jeonju-si, Jeollabuk-do 54896, Republic of Korea

<sup>d</sup> Department of Quantum System Engineering, Jeonbuk National University, 567, Baekje-daero, Deokjin-gu, Jeonju-si, Jeollabuk-do 54896, Republic of Korea

\* Corresponding author: [jgjeon41@jbnu.ac.kr](mailto:jgjeon41@jbnu.ac.kr), [sungikim@hanyang.ac.kr](mailto:sungikim@hanyang.ac.kr)

**\*Keywords : PINNs, MELCOR, Governing Equation**

### 1. Introduction

The growing capability of the Artificial Intelligence (AI) has drawn attention to application of neural networks in various fields. [1] Among numerous types of AI, Physics Informed Neural Networks (PINNs) is one of the neural networks that has recently grabbed attention. [2,3] The concept of PINN was first suggested by M. Raissi in 2019 for the neural networks that incorporate physical governing equation as a loss function, thereby improving the performance of the neural network with only a few provided data. [4]

PINN is applied in a wide range of studies such as fluid mechanics, chemical engineering, and bio-medics to name a few. [2,5] The concept of PINN is explained in figure 1. An ordinary neural network calculates the loss function from the difference between the obtained data and desired data. In such cases, providing more data allows model to train itself more, resulting in higher accuracy. Unlike any ordinary neural network, PINN utilizes equation loss for training the model. Therefore PINN demonstrates its strength in the field where the data acquisition is limited. The insufficient amount of data is fulfilled with governing equation as a loss function. PINNs is able to make precise predictions with relatively small amount of information. [3,6]

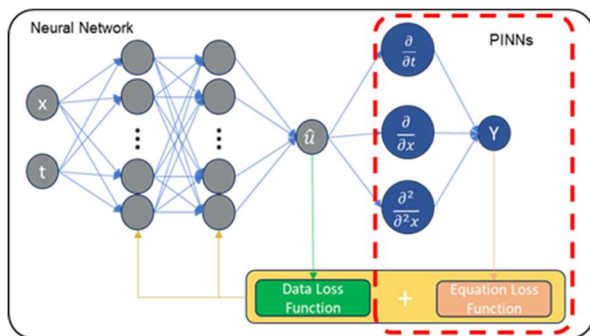


Fig. 1. The basic diagram for PINNs: Neural Network with governing equation as a loss function.

In the field of nuclear power plants, PINNs hold great promise in solving thermal-hydraulic phenomena.

Several studies have been conducted in application of PINNs to the nuclear reactor behavior [7] or the simulation of the accidents. [8] With the effort to bridge the bond between the AI and MELCOR, this study demonstrates the capability of calculating the mass flow of the water in the six tanks with Python based code. The developed model utilizes functional based programming. The goal of this study is to analyze the validation of applying governing equation to the code thereby suggesting the possibility of application of PINN to MELCOR. The effect of the governing equation is to be analyzed and the sensitivity of each factor is discussed.

### 2. Methodology

#### 2.1. Scenario Description

Figure 2 shows the modeling of the scenario. All the control volumes (CVs) have the height of 2 m and area of 50 m<sup>2</sup>, aligned with the elevation difference of 1.8 m. The CVs are connected with the 0.1 m length and 0.1 m radius of pipe at the bottom or top of the tanks. Initially the highest tank (CV01) is filled with water, the rest of the tanks are empty. CVs are connected with the atmosphere on the top, thereby reduce the effects of the pressure difference. As the simulation starts, the water naturally flows downward due to gravitational force.

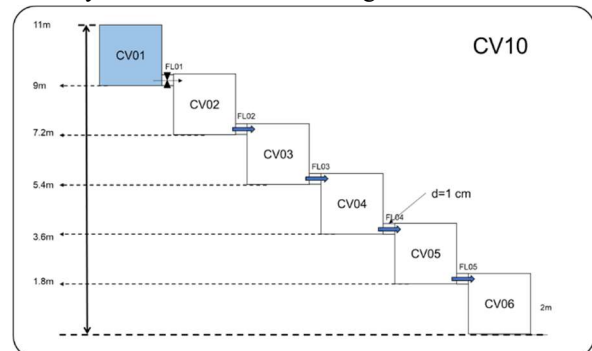


Fig. 2. The modeling of the scenario, 6 water tanks with 5 flow paths where only CV01 water tank is filled.

## 2.2. Calculation

MELCOR's governing equations have been adopted to calculate the mass and velocity of the water flow within the tanks. [9] The velocity is calculated through the following equation.

$$\alpha_{j,\varphi} \rho_{j,\varphi} L_j \frac{\partial v_{j,\varphi}}{\partial t} = \alpha_{j,\varphi} (P_i - P_k) + \alpha_{j,\varphi} (\rho g \Delta z)_{j,\varphi}^n + \alpha_{j,\varphi} \Delta P + \alpha_{j,\varphi} \rho v_{j,\varphi} (\Delta v)_{j,\varphi} - \frac{1}{2} K^*_{j,\varphi} \alpha_{j,\varphi} \rho_{j,\varphi} |v_{j,\varphi}| v_{j,\varphi} - \alpha_{j,\varphi} \alpha_{j,-\varphi} f_{2,j} L_{2,j} (v_{j,\varphi} - v_{j,-\varphi}) \quad (1)$$

Equation (1) is an alternate form of Navier-Stokes equation which is acceleration equation for fluid in the flow path. The equation brings out the velocity term by taking integral on both sides. Equation (2) is the result of the integration on equation (1).

$$v_{j,\varphi}^n = v_{j,\varphi}^{o+} + \frac{\Delta t}{\rho_{j,\varphi} L_j} (P_i^n + \Delta P_j - P_k^n + (\rho g \Delta z)_{j,\varphi}^n + v_{j,\varphi}^o (\rho \Delta v)_{j,\varphi}^o) - \frac{K^*_{j,\varphi} \Delta t}{2 L_j} (|v_{j,\varphi}^n + v_{j,\varphi}^o| v_{j,\varphi}^n - |v_{j,\varphi}^o| v_{j,\varphi}^o) - \frac{\alpha_{j,-\varphi} f_{2,j} L_{2,j} \Delta t}{\rho_{j,\varphi} L_j} (v_{j,\varphi}^n - v_{j,-\varphi}^n) \quad (2)$$

The nomenclature for each of the variable is described in Table I.

Table I: Nomenclature of the equation

| Variable             | Meaning  |
|----------------------|--|
| $v_{j,\varphi}^n$    | New velocity (from iteration)                    |
| $v_{j,\varphi}^{o+}$ | Old velocity                                     |
| $\varphi$            | Phase (liquid, air, solid)                       |
| $\rho$               | Density  |
| $L_j$                | Inertial length of the pipe                      |
| $\Delta z$           | Height of the water                              |
| $K^*_{j,\varphi}$    | Net form- and wall- loss coefficient             |
| $f_{2,j}$            | Momentum exchange coefficient                    |
| $L_{2,j}$            | Effective length over the interphase force       |
| $v_{j,\varphi}$      | Tangent linearization                            |
| $v_{j,\varphi}^{n-}$ | $v_{j,\varphi}^n$ old value inside the iteration |

Each term is explained in Table II.

Table II: Velocity terms Description

| No. | Term  | Explanation                                      |
|-----|---|--|
| 1   | $P_i^n - P_k^n$   | Pressure difference                              |
| 2   | $\Delta P_j$  | Pump head pressure                               |
| 3   | $(\rho g \Delta z)_{j,\varphi}^n$   | Static head                                      |
| 4   | $v_{j,\varphi}^o (\rho \Delta v)_{j,\varphi}^o$                             | Advection of momentum                            |
| 5   | $\frac{K^*_{j,\varphi} \Delta t}{2 L_j}$                                    | Net form- and wall-loss coefficient              |
| 6   | $\frac{\alpha_{j,-\varphi} f_{2,j} L_{2,j} \Delta t}{\rho_{j,\varphi} L_j}$ | Interphase force coefficient (momentum exchange) |

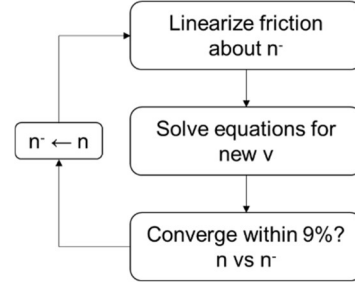


Fig. 3. Iterations of the velocity calculation

The calculation for the new velocity term in frictional loss and momentum exchange terms is done explicitly. The pressure and advection of momentum terms are calculated implicitly. From the equation (2) and table I, shows that the velocity is calculated semi-implicitly. Due to the form and wall loss term, the internal iteration must be considered. [9] The iteration is explained in figure 3 and the logic is embodied in developed model to calculate the new velocity. When the water level of CV05 and CV06 meet, the calculation is terminated.

## 3. Results and Discussion

### 3.1. Water flow rate

Figure 4 present the result of the MELCOR calculation flow rate of water through each flow path 1 to 5. From the calculation with the equation (2), the result of the python calculation is demonstrated in Figure 4.

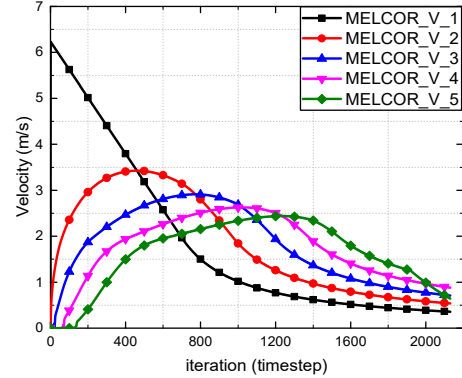


Fig. 4. Output of the calculation from MELCOR for the water flow rate for the flow path pipe.

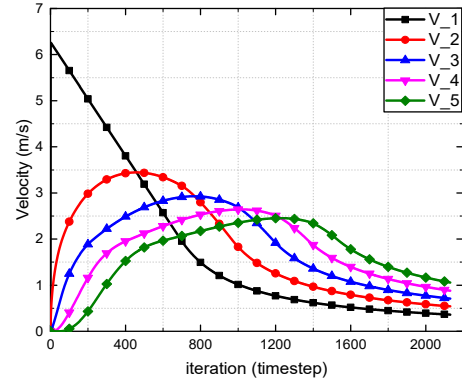


Fig. 5. Output of the calculation from developed model for the water flow rate for the flow path pipe

Table III: Errors analysis for velocity

| Method       | MAE      | MSE        |
|--------------|----------|------------|
| Liquid Level | 0.002925 | 7.80894e-5 |

Table III shows the error between the two-calculation method. The error methods, Mean Absolute Error (MAE) and Mean Squared Error (MSE) is calculated by the following equation.

$$MAE = \frac{1}{n} \sum_n |y - \hat{y}| \quad (3)$$

$$MSE = \frac{1}{n} \sum_n (y - \hat{y})^2 \quad (4)$$

The MAE and MSE are widely used to suggest the indicator of the error. The developed model for the mass flow calculation shows high accuracy to the value obtained from MELCOR.

### 3.2. Liquid Level

The level of liquid is obtained from the velocity, by calculating the amount of flowing in and flowing out. Due to the gravitational force acting on the water, CV01 through CV 05 acts as a donor, having the mass flowing out. As the five water tanks bring out the mass, CV02 to CV06 are receivers, so the mass flowing in needs to be considered. The following figures show the following liquid level behavior.

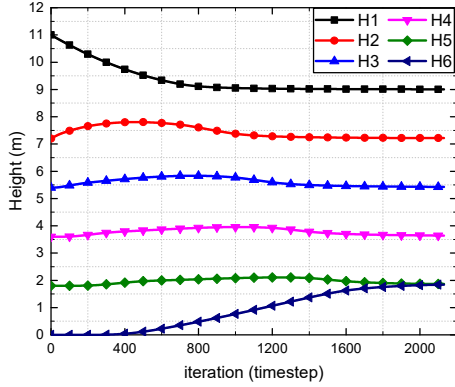


Fig.5. Output of the calculation from MELCOR for the water level in Control Volumes

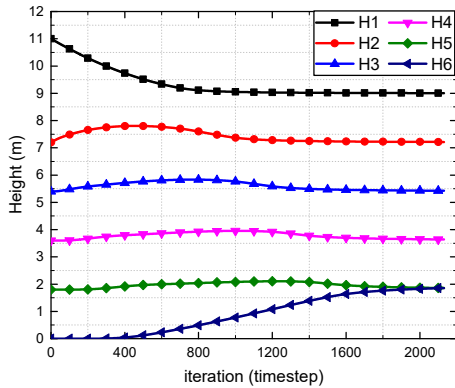


Fig. 6. Result of the calculation from developed model for the water level in Control Volumes

Table IV: Errors analysis for liquid level

| Method   | MAE      | MSE     |
|----------|----------|---------|
| Velocity | 0.013417 | 0.00156 |

### 3.3. Sensitivity of the factors

Velocity terms in Table II are analyzed. The conditions and the chosen coefficient for the analysis is described in table V. For the analysis, flow rate through FL02 was chosen.

Table V: Velocity terms and description for variable selection

| No. | Term  | Description   |
|-----|---|---|
| 1   | $P_i^n - P_k^n$ :   | Set to be equal for this model  |
| 2   | $\Delta P_j$ :  | Not regarded in this scenario   |
| 3   | $(\rho g \Delta z)_{j,\varphi}^{\tilde{n}}$ :                                 | Calculates the velocity out. No variation is valid.                   |
| 4   | $v_{j,\varphi}^o (\rho \Delta v)_{j,\varphi}^o$ :                             | Calculates the velocity difference in FL. Not regarded in this study. |
| 5   | $\frac{K_{j,\varphi}^* \Delta t}{2L_j}$ :                                     | <b>Regarded coefficient for variation</b>                             |
| 6   | $\frac{\alpha_{j,-\varphi} f_{z,j} L_{z,j} \Delta t}{\rho_{j,\varphi} L_j}$ : | <b>Regarded coefficient for variation</b>                             |

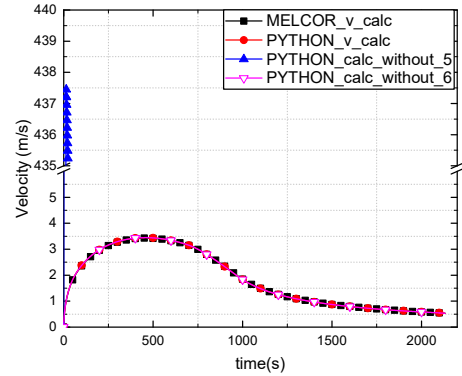


Fig. 7. Result of velocity for four cases.

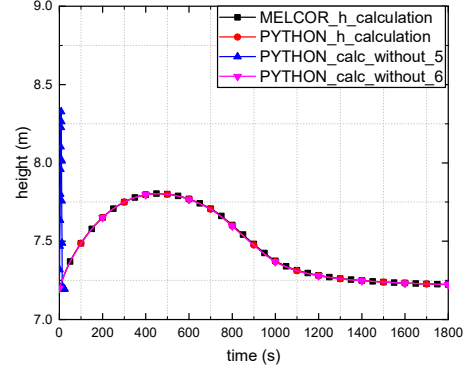


Fig. 8. Output of the calculation in height in four cases

Figure 7 and 8 demonstrates the water velocity and height of four different cases. The cases include the calculation of MELCOR, developed model, developed

model without coefficient  $\mathbf{K}$  (number 5 of Table V), and developed model without coefficient  $\mathbf{f}$  (number 6 of Table V). The result shows that velocity term heavily relies on  $\mathbf{K}$  value which is the wall loss or form loss coefficient. Without the  $\mathbf{K}$  value given, the calculation ends within 19 iteration and the value of the velocity reach over 437m/s. Due to fast flow out rate, the height of the water remaining also run out within a short period of time. From Fig. 7 and 8, the result of MELCOR calculation, Python calculation, and model calculation without  $\mathbf{f}$  are overlapped both in velocity and height, the coefficient  $\mathbf{f}$  does not give impact on the velocity nor the height of the model calculation. From Table V, term number 4 is not regarded in this study due to the trivial difference for the front and back of the flow path.

### 3.4. Computation time

Time consumption is one of the most important factors to evaluate the performance of the modeling. The calculation cost for the MELCOR and developed model is shown in Table VI.

Table VI: Time cost of the two calculation methods

| Method  | MELCOR | Developed model |
|---------|--------|-----------------|
| Time(s) | 0.6875 | 0.16451         |

The developed model's time spent for calculating is 17 times less than running with MELCOR. This is due to the simplified model developed through python. To be more specific, when MELCOR runs calculation, it takes the energy, temperatures, and other terms into account. Whereas for the developed model in this case scenario, only the mass and the momentum are regarded. Such difference brings faster calculation in developed model.

## 4. Conclusions

In this study, preliminary analysis of PINN applicability to MELCOR was analyzed. Governing equation takes big portions in PINN hence the physics equation from the MELCOR manual was adopted. With the code generated from Python, the comparison between the two could be obtained. The followings are the key findings and the future works.

- (1) By constructing model with the main governing equation, the possibility of developing model to the PINN was assessed.
- (2) With certain boundary conditions and initial conditions, MELCOR code can be simulated with PINN.
- (3) For the future work, composing a MELCOR governing equation based neural network and running the system code to train the model is expected.

## Acknowledgement

This work was supported by the Innovative Small Modular Reactor Development Agency grant funded by the Korea Government (MSIT) (No. RS-2023-00259516). Additionally the authors would like to acknowledge the support from the Hyundai Motor Group through the Future POC program (Research Collaboration-Innovation Technology for Vehicle)

## REFERENCES

- [1] B. Huang and J. Wang, "Applications of Physics-Informed Neural Networks in Power Systems - A Review," in IEEE Transactions on Power Systems, vol. 38, no. 1, pp. 572-588, Jan. 2023, doi: 10.1109/TPWRS.2022.3162473.
- [2] Jeon, J., Lee, J., & Kim, S. J. (2022). Finite volume method network for the acceleration of unsteady computational fluid dynamics: Non-reacting and reacting flows. International Journal of Energy Research, 46(8), 10770-10795.
- [3] Jeon, J., Lee, J., Vinuesa, R., & Kim, S. J. (2024). Residual-based physics-informed transfer learning: A hybrid method for accelerating long-term CFD simulations via deep learning. International Journal of Heat and Mass Transfer, 220, 124900.
- [4] M. Raissi, P. Perdikaris, G.E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, Journal of Computational Physics, Volume 378, 2019, Pages 686-707, ISSN 0021-9991,
- [5] Karniadakis, G.E., Kevrekidis, I.G., Lu, L. et al. Physics-informed machine learning. Nat Rev Phys 3, 422-440 (2021). <https://doi.org/10.1038/s42254-021-00314-5>
- [6] Jochen Stiasny, Spyros Chatzivasileiadis, Physics-informed neural networks for time-domain simulations: Accuracy, computational cost, and flexibility, Electric Power Systems Research, Volume 224, 2023, 109748, ISSN 0378-7796
- [7] Prantikos, K., Chatzidakis, S., Tsoukalas, L.H. et al. Physics-informed neural network with transfer learning (TL-PINN) based on domain similarity measure for prediction of nuclear reactor transients. Sci Rep 13, 16840 (2023).
- [8] Federico Antonello, Jacopo Buongiorno, Enrico Zio, "Physics informed neural networks for surrogate modeling of accidental scenarios in nuclear power plants", Nuclear Engineering and Technology, Volume 55, Issue 9, 2023, 3409-3416, ISSN 1738-5733
- [9] L.L. Humphries, R.K. Cole, D.L. Louie, V.G. Figueroa, M. F. Y. (2005). MELCOR computer code manual: Reference Manual, version 1.8.6.
- [10] L.L. Humphries, R.K. Cole, D.L. Louie, V.G. Figueroa, M. F. Y. (2005). MELCOR computer code manual: Primer and Users Guide, version 1.8.6.