RePIT-v1.0: A Software for Hybrid ML-CFD Computation

Shilaj Baral [a], Sangam Khanal [a], Joongoo Jeon [a,b,c*], Youngkyu Lee [d], George Em Karniadakis [d]
[a]*Graduate School of Integrated Energy-AI, Jeonbuk National University*
[b]*Department of Quantum System Engineering, Jeonbuk National University*
[c]*Department of Applied Plasma and Quantum Beam Engineering, Jeonbuk National University*
[d]*Division of Applied Mathematics, Brown University*
[*]*Corresponding author: jgjeon41@jbnu.ac.kr*

Keywords : computational fluid dynamics, scientific machine learning, hybrid solver, RePIT, simulation acceleration

1. Introduction

In recent years, the shift can be seen towards small modular reactors (SMRs) which offer enhanced safety, modular design, and more flexible power generation. Either to enhance the efficiency or increase the safety of any type of nuclear reactor, understanding heat and fluid flow within them is essential. There are several existing traditional system analysis codes like RELAP5 and TRACE, which are essential, but often rely on simplified models and may not capture intricate local phenomena. Whereas, Computational Fluid Dynamics (CFD) addresses this by using numerical methods to simulate detailed fluid flow and heat transfer scenarios.

However, CFD's computational demands are significant; for instance, simulating one second of physical time in a turbulent jet scenario can require 72 hours of CPU time, even when parallelized across 32 cores [1]. To mitigate these challenges, Artificial Intelligence (AI) has emerged as a promising solution, offering faster inference once models are trained. Yet, AI models often struggle to maintain accuracy over extended simulations due to error accumulation [2].

Hybrid approaches, such as the residual-based physics-informed transfer learning (RePIT) [3] strategy and a hybrid, iterative, numerical, transferable solver (HINTS) [4], integrate AI with traditional numerical methods to enhance simulation accuracy and computational efficiency. The RePIT strategy alternates between AI predictions and CFD computations, monitoring residuals to maintain accuracy, achieving up to 1.9 times faster computations without compromising precision.

While the RePIT strategy significantly accelerates CFD simulations using AI, it still requires manual tasks like data conversion, transfer learning execution, and OpenFOAM integration. This study aims to automate the cross-computation process, making it adaptable to various CFD applications and neural network architectures. Additionally, it seeks to eliminate the need for ground truth data in boundary fields and optimize two key hyperparameters: transfer learning epochs, which reduce parameter update time, and residual thresholds, which extend ML prediction intervals before switching back to CFD. By fine-tuning these parameters, the study aims to balance computational efficiency and accuracy, validating an automated, scalable framework for hybrid CFD-AI simulations.

2. Methods and Results

In this section the concept and components of the framework are discussed. The section talks about dataset choice, vanilla RePIT, automatic RePIT, bc-enforcement and analytical capabilities of the RePIT framework.

2.1 CFD Dataset

Natural convection was chosen to evaluate the framework's predictability in a hybrid strategy. Temperature differences on the left (307.75 K) and right walls (288.15 K) drive circulation, with hot air rising due to buoyancy and cool air sinking due to gravity. Based on previous studies, the most chaotic flow occurs between 10s-20s, stabilizing afterward. This time range was selected for analysis within a 40000-cell domain, with probes T1-3, B1-3 monitoring key flow regions for top and bottom adiabatic walls respectively. A clearer depiction of the domain and setup is demonstrated in Fig. 1. The dataset was generated using buoyantPimpleFoam.

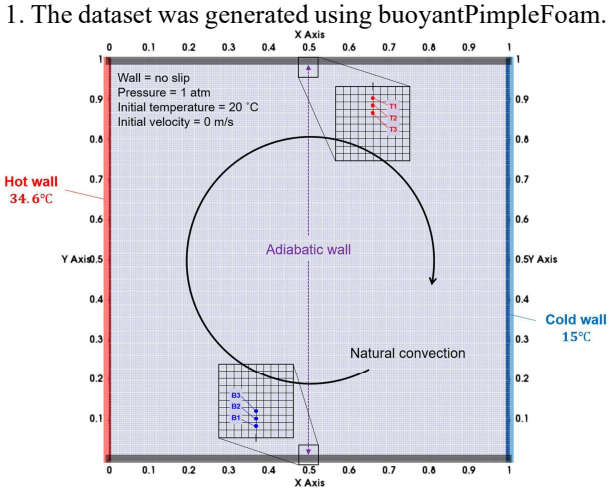

Fig. 1. Schematic representation for mesh and boundary conditions of natural convection [3]

2.2 Vanilla RePIT

The original study on this hybrid solver uses finite volume method network (FVMN) [5] as a neural network and OpenFOAM as a CFD solver. The FVMN was chosen because it introduces finite volume principles in the network architecture. This has enabled the FVMN model to predict longer timesteps from a relatively small

dataset in comparison with traditional neural networks. The RePIT has used residual monitoring to define the cross point between ML and CFD. This residual threshold is scaled and calculated using first principles. After ML prediction exceeds this threshold a transfer learning is done to inform the pre-trained model about the changed dynamics. This strategy has accelerated the simulation by 1.9x as compared to OpenFOAM. The workflow is shown in Fig. 2.

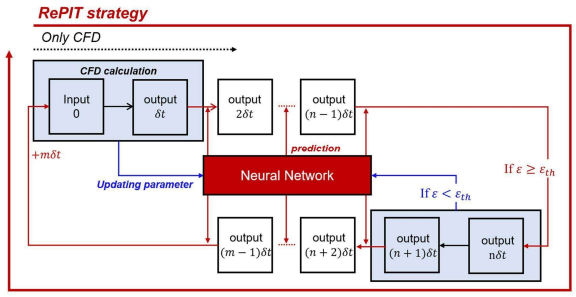

Fig. 2. RePIT strategy workflow [3]

2.3 RePIT Framework

The framework extends the RePIT strategy by enabling start to end automation of the algorithm. It is designed to be modular and open-source, allowing users to easily modify, extend functionalities, and contribute collaboratively. The complete flowchart is represented in Fig. 3. And, the simple algorithm structure is represented in the Algorithm 1 where we can see inside the time loop ML section and CFD section are coupled together. It operates through a simple configuration file, where users specify parameters like simulation time, neural network architecture, domain information, and more. The automation process integrates:

- running OpenFOAM via python,
- exchanging data between the AI model and OpenFOAM, and
- connecting the different modules for automation.

To achieve full automation, python's scripting capabilities have been widely used. A wrapper was created to run the OpenFOAM from a python script using built-in modules like subprocess. The open-source Ofpp module was used to extract OpenFOAM data to numpy whereas a script has been made to convert the numpy back to OpenFOAM format. And, to connect everything as a whole, a separate script has been made. The integrity of the framework is well represented by the results in the following sections.

2.4 BC-enforced RePIT

The original study focused on proving that hybrid computation enables acceleration, so it used the CFD solver for boundary layer cell values because their calculation is computationally inexpensive. Now, as shown in Fig. 4, boundary layers are enforced, eliminating the need for solver-calculated values. The original study used to include boundary values to the model output but now the boundary values are enforced in the model input and as an output the whole computational domain is predicted from the neural network. In doing so, the model accuracy has not been altered as compared to the previous case. So, the results shown in the following section are all using the BC enforcement technique.

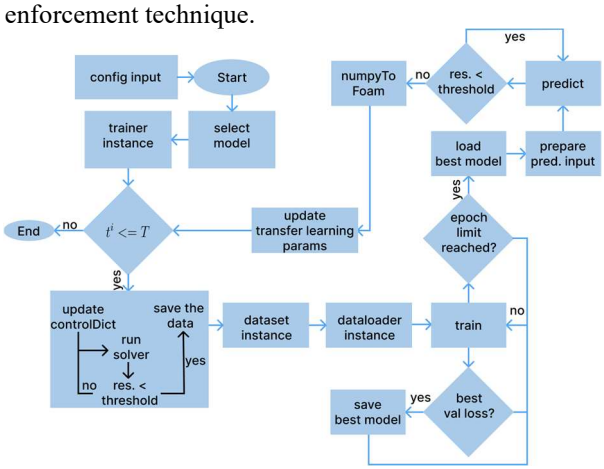

Fig. 3. Flowchart representing the RePIT-framework

Algorithm 1 RePIT Framework

Input: initial neural network parameters θ_0, ϵ_{th}
for $t = 0, T$ **do**
  $\boldsymbol{T_{CFD}} = t + n_{CFD} * \delta t$
  while $t \leq \boldsymbol{T_{CFD}}$ **do**
    Run the OpenFOAM solver and get u^t, p^t
    Update time: $t = t + \delta t$
  end while
  Optimize surrogate loss L w.r.t θ
    $\theta^* = arg\ min_\theta \sum([\hat{u}, \hat{p}] - [u, p])^2$
  while $\epsilon \leq \epsilon_{th}$ **do**
    $u^{new}, p^{new} \leftarrow u^t, p^t$
    $u^t, p^t = NN(u^{new}, p^{new}; \theta)$
    Update time: $t = t + \delta t$
    Update continuity error ϵ
  end while
end for

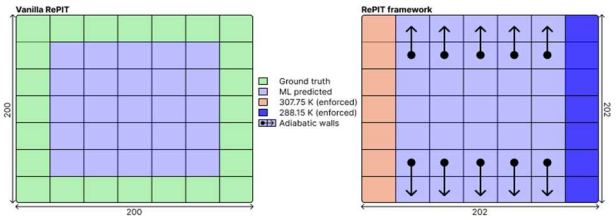

Fig. 4. Illustration of bc-enforcement in RePIT-framework

2.5 Hyper-parameters analysis

The total of the imbalances at every computational grid point is what CFD solutions use to determine residuals. However, the absence of a universal reference point makes it difficult to evaluate convergence using raw residuals. Solvers usually scale residuals using their

initial values as a baseline in order to overcome this. A similar strategy has been adopted in the original study of RePIT, scaling the residuals in the time series predicted by machine learning in relation to those from the initial training dataset. This scaled residual serves as the basis for determining the point at which ML and CFD calculations switch. The term scaled residual and relative residual threshold have been used interchangeably in this report.

Two parameters that account for the acceleration performance are the scaled residual limit (that defines the switching point for ML and CFD) and the number of transfer learning epochs (that determines for how long to keep on updating parameters based on the latest CFD data). Their relation with the parameters affecting the acceleration performance have been shown in Table 1. Changing these two, an in-depth analysis has been performed to figure out what can be the best combination for these two.

The analysis is made possible because of the automation framework proposed here. From Fig. 5 we can see that even the prediction with a scaled residual value set to 100 is performing better in a hybrid approach than the single training approach for a longer time frame. But, comparing the results when the scaled residual limits are 5 and 100 in the hybrid-computation, the former case is showing higher fidelity to the ground truth value. This can be further validated by Fig. 6 which shows the comparison between ground truth and case 1 from Table II in the probe locations (T1-3 and B1-3) for the whole simulation time frame. Comparing Fig. 6 with Fig. 7, it is conspicuous that hybrid approach is able to control the divergence that is seen in single training approach. This is also the testament that the framework is working effectively. As seen in Fig. 8, the intermediate CFD computation is bringing down the residual value to an acceptable range as soon as the ML-model crosses the threshold. Because of this monitoring, the simulation through hybrid computation is made possible. The rising edges in the plot represent ML-prediction while the trailing edges represent solver calculation.

Table I: Relation of hyper-parameters with acceleration parameters

↑ scaled residual	↓ n_{CFD}; ↑ n_{ML}	↑α
↑ transfer learning epochs	↑ t_{up}	↓α

The α means the acceleration performance (represented by Eqn. 1) of the framework compared to the traditional solver, n_{CFD} is the number of CFD time steps calculated by solver in cross-computation, n_{ML} is number of ML predicted time steps, t_{up}, t_{ML}, and t_{CFD} are the time taken per time step for parameter update, ML prediction and CFD simulation respectively.

$$\alpha = \frac{N * t_{CFD}}{n_{CFD} * t_{CFD} + n_{ML}(t_{ML} + t_{up})} \quad (1)$$

Table II: Acceleration performance analysis

Epochs	Residual limit	CFD (s)	ML+CFD (s)	α
2	5	438.43	251.16	1.74
2	10	424.42	217.11	1.95
2	100	448.44	149.95	2.98
10	5	442.44	386.6	1.14
10	10	443.44	343.94	1.28
10	100	448.44	263.25	1.7

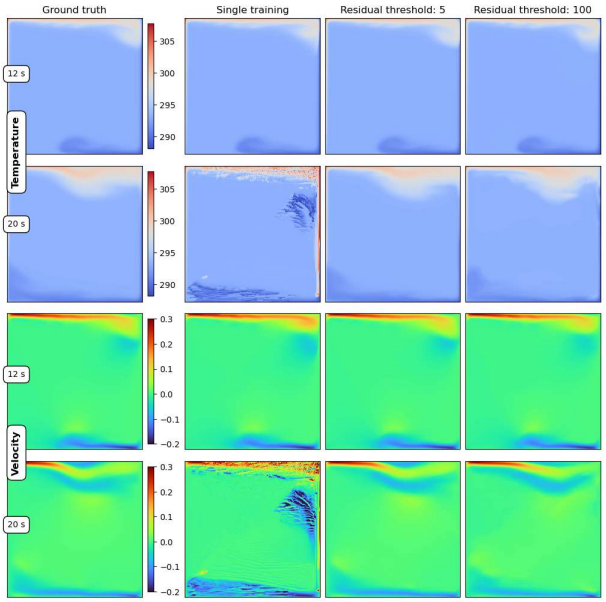

Fig. 5. Comparison of single training and framework predicted results with ground truth for temperature and velocity fields using only two transfer learning epochs in hybrid approach.

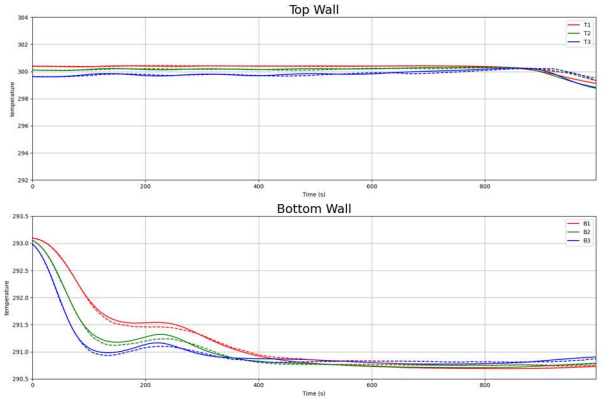

Fig 6. Comparison of predicted results and ground truth values at probe locations for temperature profile [epochs: 2; scaled residual limit: 5]

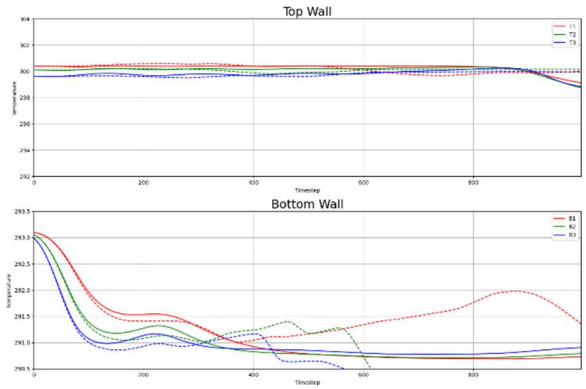

Fig 7. Comparison of single training predictions with ground truth values at probe locations for temperature profile.

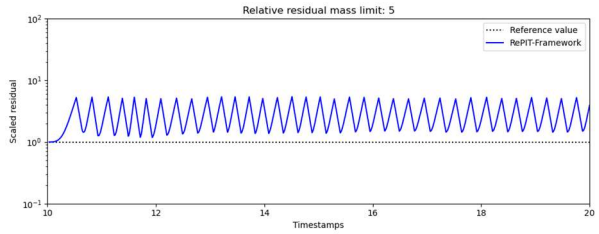

Fig 8. Scaled residual vs timestamps representing residual monitoring in hybrid-approach [epochs: 2; scaled residual limit: 5]

3. Conclusions

In order to improve long-term efficiency, this study integrates OpenFOAM with neural networks and uses transfer learning to create the first completely automated ML-CFD cross-calculation framework in history. The enforcement of boundary knowledge in the pre-processing step enables the scalable use of FVMN in a hybrid framework for 2D cases, which is one of the notable additions of this study. According to the results, computation is greatly accelerated without appreciably sacrificing accuracy when the transfer learning epoch is set to two. Additionally, a thorough examination of residual thresholds shows that moderate values (such as 5 or 10) provide steady and reliable findings, but extremely high thresholds (such as 100) can induce flow field anomalies. The framework strikes the ideal balance between simulation speed and accuracy by carefully adjusting these hyperparameters.

In order to facilitate real-time modeling and complex fluid dynamics applications, this study lays the groundwork for next-generation hybrid learning-based CFD simulations. Future plans call for using this framework to analyze the CFD of the Small Modular Reactor (SMR) system, showcasing its applicability to actual technical issues. Future studies will investigate operator learning strategies in this cross-computation framework, such as combining DeepONet with an FVMN-based branch network which we believe can open exciting avenues in the realm of hybrid solvers in CFD.

ACKNOWLEDGEMENT

This work was supported by the Nuclear Safety Research Program through the Regulatory Research Management Agency for SMRs (RMAS) and the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea. (No. RS-2024-00509653) and the National Research Council of Science & Technology (NST) grant by the Korea government (MIST) (No. GTL24031-000).

REFERENCES

[1] Jeon, Joongoo, et al. "Identification of Hydrogen Flammability in steam generator compartment of OPR1000 using MELCOR and CFX codes." Nuclear Engineering and Technology 51.8 (2019): 1939-1950.
[2] Vinuesa, Ricardo, and Steven L. Brunton. "Enhancing computational fluid dynamics with machine learning." Nature Computational Science 2.6 (2022): 358-366.
[3] Jeon, Joongoo, et al. "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 (2024): 124900.
[4] Zhang, Enrui, et al. "Blending neural operators and relaxation methods in PDE numerical solvers." Nature Machine Intelligence (2024): 1-11.
[5] Jeon, Joongoo, et al. "Finite volume method network for the acceleration of unsteady computational fluid dynamics: Non-reacting and reacting flows." International Journal of Energy Research 46.8 (2022): 10770-10795.