Physics-Informed Machine Learning to Accelerate Unsteady Computational Fluid Dynamics

Joongoo Jeon [a*], Juhyeong Lee [b], Ricardo Vinuesa [c]
[a]*Department of Nuclear Engineering, Seoul National University, Seoul 08826, Republic of Korea*
[b]*Department of Nuclear Engineering, Hanyang University, Seoul 04763, Republic of Korea*
[c]*FLOW, Engineering Mechanics, KTH Royal Institute of Technology, SE-100 44 Stockholm, Sweden*
[*]Corresponding author: jgjeon@snu.ac.kr

1. Introduction

The importance of computational fluid dynamics (CFD) for thermal-hydraulic analysis continues to grow in the nuclear industry. Despite rapid advancements in the performance of central processing units (CPUs), the computational cost of CFD remains unrealistic in many cases. This is especially true for turbulent or reacting flows with grid and timestep in scales less than a millimeter and a millisecond, respectively. Tolias et al. identified that approximately 10–100 h of CPU time per second of physical time was required to simulate unsteady hydrogen deflagration based on a domain size of 20.0×14.4×12.0 m [1]. In terms of their commercial application to the nuclear industry, these costly simulations are impractical, given that hour-based accident analysis is required in the regulatory guide [2].

To reduce computational cost, this work focuses on the application of machine learning techniques, which are in the forefront as the most innovative technology [3]. As Vinuesa et al. noted [4], one of the main objectives of recent machine learning studies in fluid mechanics is to accelerate CFD simulations. Although the application of convolutional layers in flow field predictions has been studied, the need to develop a unique network design fitted for CFD has emerged. Notably, the convolutional neural networks (CNNs) specialized in general image processing require many images to train the parameters inside the kernels [5]. Recently, Jeon et al. developed a neural network model introducing the finite volume method with a unique network architecture. The finite volume method network (FVMN), considering the nature of the CFD flow field where the identical governing equations are applied to all grid points, can predict the future fields with only two previous fields.

However, the numerical algorithm of FVM for handling governing equations is difficult to be implemented sufficiently into the network design only by the proposed input/output system. In this paper, we introduced a physics-informed loss function calculating the balance of each governing equation by summation of physical fluxes, to the network to enhance the network prediction accuracy. Additionally, a cross-coupling strategy of machine learning and CFD using the physics-informed loss function was proposed to overcome the long-term prediction limitation of data-driven approaches. TensorFlow, an open-source machine learning software, was used to develop the network model.

2. Physics-Informed Machine Learning

2.1 Physics-Informed Loss Function

Recently, the FVMN model was developed to accelerate unsteady CFD simulations by introducing the FVM principles in the network design, as shown in **Fig. 1(a)** [6]. The basic procedure is similar to that of previous multi-layer perceptron (MLP), but the developed network model employs tier-input and derivative-output systems. The CFD time series fields at times t and (t+1) are used as input and output variables in the training process, respectively. It should be noted that, considering the nature of the CFD flow field where the identical governing equations are applied to all grid points, the grid-based training approach was adopted in this network design. This approach bears a strong advantage over CNNs' image-based training approach in terms of accelerating unsteady CFD simulations. More details on the FVMN model can be found in Ref. [6].

The loss function in the previous network (**Fig. 1(a))** is defined based on the mean square error (MSE) between the predicted value and ground truth value as shown **Eq. (1)**. Note that 80% of the total grid points (14,400) were used for the training loss and 20% (3,600) were used for the validation loss for checkpoints.

$$L = L_{mse} = \frac{1}{n}\sum_{k=1}^{n}(Z^k - Z^k(\theta))^2 \quad (1)$$

On the contrary, **Fig. 1(b)** shows the newly developed FVMN model design with a physics-informed loss function as shown **Eq. (2)**. The residual ε_j^k for each grid k and each governing equation j is iteratively calculated to optimize the parameters in the mini-batch. In other words, The residual is calculated by the summation of physical fluxes of each governing equation, continuity and the Navier-Stokes equation (x-momentum and y-momentum).

$$\begin{aligned} L &= w_1 L_{mse} + w_2 L_{residual} \\ &= w_1 \cdot \frac{1}{n}\sum_{i=1}^{3}\sum_{k=1}^{n}\left(Z_i^k - Z_i^k(\theta)\right)^2 + w_2 \cdot \frac{1}{n}\sum_{j=1}^{2}\sum_{k=1}^{n}\left(\varepsilon_j^k\right)^2 \end{aligned} \quad (2)$$

Because all variables (p, u, v) were predicted in the unified framework to compute the residuals during network training, the MSE loss function also summed squared error of each grid k and each variable i. w_1 and w_2 are weighting factors to adjust the scale of each governing equation ($w_1 = 1, w_2 = 0.5$ in this study). Additionally, the continuity residual was scaled by 1/10 to consider the residual of the continuity and the Navier-

Stokes equation together. The residuals of 100 % grid points were averaged and used for training loss $L_{residual}$.. Although the total loss function including L_{mse} has a supervised manner, $L_{residual}$ can be utilized as an unsupervised monitoring function in the prediction dataset. In MSE function, same as network (a) 80% of the total grid points was used for the training loss and 20% was used for the validation loss for checkpoints.

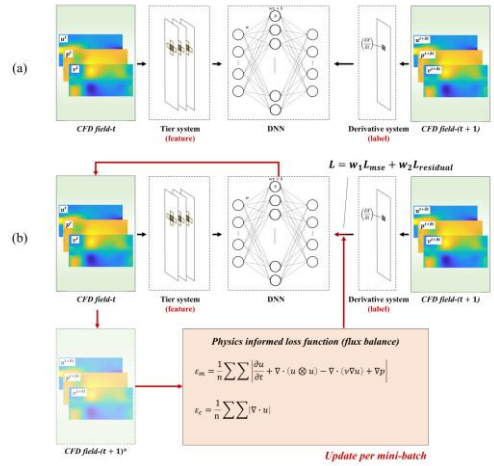

Fig. 1. FVMN model with (a) MSE loss function. (b) physics-informed loss function.

2.2 Laminar Flow Dataset

To investigate the efficacy of the physics-informed loss function, a 2D incompressible counterflow dataset was generated by the OpenFOAM code. The OpenFOAM solver was icoFoam, which is transient solver for incompressible laminar flow of Newtonian fluids (**Eqs. (3) and (4)**). The geometry of the simulation was a simple rectangle as shown in **Fig. 2(a)**. The mesh was constructed by 20,000 hexahedral cells. Through the overall same mesh quality, the maximum non-orthogonality was 0 and maximum skewness was 6.66e-14. The inlet 1 located at the middle of left wall was assigned to 2 m/s of positive x-axis velocity (u) and 1 m/s of negative x-axis velocity in inlet 2 (nonsymmetric inlet condition). As the working fluid, a virtual fluid with a kinematic viscosity of $0.01 m^2/s$ was modelled. The simulation time was 4 seconds, and each timestep was equally controlled as 0.01 s.

$$\nabla \cdot u = 0 \tag{3}$$

$$\frac{d(u)}{dt} + \nabla \cdot (u \otimes u) - \nabla \cdot (\nu \nabla u) = -\nabla p \tag{4}$$

Fig. 2(b) shows pressure and velocity fields according to time transient; p represents pressure field, u and v each represents the x-velocity and y-velocity. Because the velocity of inlet 1 was twice than velocity of inlet 2, this counterflow showed biased results from left side to the right side and y-velocity showed symmetric value along central axis ($y = 0$). The 1.00-1.11 s timeseries dataset, which was one of the most unsteady phases, was used for network performance evaluation. The input and output variables for the FVMN were extracted from the ML domain ($n = 180 \times 100$).

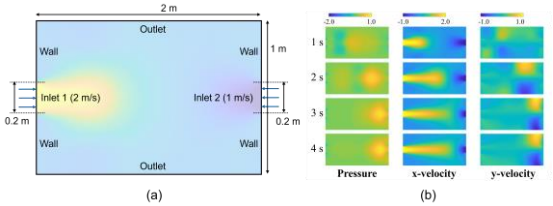

Fig. 2. Laminar flow CFD dataset from the counterflow simulation. (a) computational domain and boundary conditions with nonslip walls. (b) time series until formation of stabilized counterflow.

2.3 Efficacy of Physics-Informed Loss Function

We investigated the efficacy of the physics-informed loss function by comparing the performance of network (a) and (b). A single time series was used for network training and 10 time series data were used for performance evaluation. More specifically, the training/validation dataset was 80/20% of the 1.00–1.01 s time series data and test dataset (prediction dataset) was the 1.01-1.11 s time series data. The hyperparameters were optimized by identical manner in Ref. [6]: 2 hidden layers, 64 node/layer, mean square error (MSE) and ReLU function. The training process was terminated when each loss function converged.

Fig. 3 compares the converged value of each loss between cases. Network (a)-double training data is the identical network condition as network (a), but the training dataset size is doubled (1.00-1.02 s). When the residual term was included in the loss function (network (b)), the total residual converged to a lower value without compromising the training/validation error. Intriguingly, the magnitude of the residual was even smaller than network(a)-double which was trained by the double dataset size. It means that providing the governing equation information in the loss function can increase network performance as much as increasing the amount of training data. Although the training/validation losses in the training dataset did not have significant difference in all cases, the reduced residual can affect the accuracy of the test dataset (1.01-1.11 s).

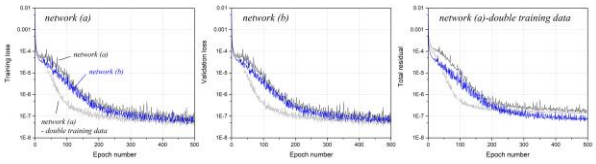

Fig. 3. Comparison of each loss according to the network model (logscale). When the residual term was included in the loss function (network (b)), the total residual converged to a lower value without compromising the training/validation error.

Fig. 4(left) shows the variation of network accuracy in the test dataset (1.01-1.11 s time series). Because the ground truth velocity values of some fields are close to zero, the network accuracy was compared based on the absolute error. First, the maximum absolute error of each variable gradually increased with timestep. In network (a), the u-maximum error of about 0.03 m/s in the last timestep was an error of 1% based on the maximum velocity in the domain. Considering that the network predicted a dataset 10 times the training dataset, the FVMN shows good agreement with the ground truth data. In network (b), the u-maximum error of about 0.015 m/s in the last timestep was about half of the error in network (a). Although the p-maximum error (Pa) increased as compared to network (a), the overall accuracy was improved by the physics-informed loss function.

Fig. 4(right) shows the variation of residuals of the governing equations such as continuity and Navier-Stokes equation. The residuals can be calculated in an unsupervised manner without a ground truth data. Interestingly, the trained network (b) considering the physical flux balance in training process always showed a lower residual value than network (a) for the test dataset. Especially, the residual in the Navier-Stokes equation (momentum balance equation) where all variables were included in the equation showed a significant difference. We concluded that the physics-loss function can prevent the non-physical overfitting problem where the network was trained to reduce the difference between the predicted value and ground truth while ignoring the physical flux balance. For this reason, the p-maximum error in network (b) was slightly larger than the u-maximum error in network (a), but it was reversed by a large difference (**Fig. 14(left)**). In general CFD simulations, an elevated residual value can reduce the reliability of simulation results. The reduction of residuals contributed to the improvement of the multistep time series prediction accuracy, but it seems not a remarkable improvement due to the sensitivity for each variable.

More importantly, in both networks the residuals increased with time and the maximum error increased accordingly. Without CFD ground truth, we can estimate the network accuracy trend over timestep by calculating the residual in an unsupervised manner. Additionally, CFD field prediction for longer time series may be feasible if we can prevent the increase of the residual with timestep. This applicability of the physics-informed loss function provides the feasibility of the ML-CFD cross coupling strategy in the next section. Consequently, two intriguing scientific observations were clarified in this section: (1) the physic-informed loss function can prevent non-physical overfitting problem and ultimately reduces the error in test dataset (2) observing the calculated residuals in an unsupervised manner can indirectly monitor the network accuracy.

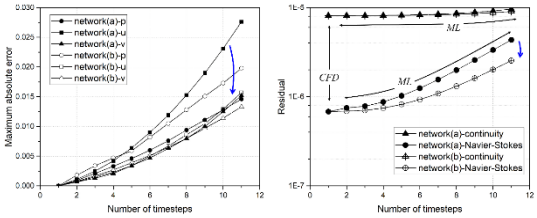

Fig. 4. Variation of maximum absolute error of each variable (left) and calculated residual of each governing equation (continuity and the Navier-Stokes equation) in ML prediction time series (right, logscale).

3. Strategy to Couple ML and CFD

3.1 Computational Framework with OpenFOAM

Our hypothesis is that continuous CFD time-series predictions are possible if the parameters in the network model are periodically updated with the latest CFD time-series data. In this study, a cross-coupling strategy was proposed and verified based on this novel concept. As shown **Fig. 5**, the CFD calculation for a single step is initially conducted by solving the first-principles. The number of time steps included in the training dataset may depend on the complexity of each simulation. In the next step, the calculated CFD time-series data are used to train the constructed neural networks to determine the model parameters. The trained networks can then quickly predict the multi-step CFD data without having to perform a costly simulation. The ML calculation reverts to the CFD calculation when the residual of the predicted time series reaches the tolerance level. The residual represents the size of the conservation error in the governing equations.

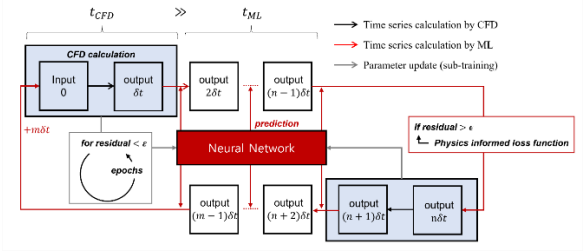

Fig. 5. Concept of the ML-CFD cross coupling framework accompanied by periodic parameter updates through the physics-informed loss function. Notably. the computation time of machine learning for one timestep t_{ML} is much smaller than that of CFD t_{CFD} (about 10 times faster in the 500,000 grid condition).

This method seems feasible because an acceptable residual range exists even in conventional CFD simulations. Reducing the conservation error below the allowable range again, the intermediate CFD simulation calculates the variable fields of the next timestep. These latest CFD results are used as training dataset to update the neural-network parameters. The evolved neural networks again accelerate the CFD simulation by

calculating the multistep data; this process is repeated until the calculation is completed. The key advantage of this cross-coupling strategy is that it can be applied individually to each simulation. Although some network optimization is required depending on the simulation complexity, network training using numerous simulation results is completely unnecessary.

3.2 Feasibility of Long-Term Prediction

Fig. 6(left) shows the efficacy of the intermediate CFD simulation in the cross-coupling strategy Before entering the second CFD zone, the maximum absolute errors in the ML and the ML-CFD cases were the same. The entry point was assumed to be 1.06 s when the residual scale changed to 10^{-6}. Because the strategy monitored residuals of each ML time series. this entry point may change depending on the tolerance setting. Notably, the increased error started to diminish as the computation enters the CFD zone. As about three time series were calculated by the first principles, the error values were converged to the initial ML time series error level. As shown in **Fig. 6 (right)**, the residual increased at the first timestep after entering the CFD simulation zone, but then effectively returned to the initial range (10^{-7}).

We conclude that the error-recovery effect was obtained by adjusting residuals of the mass and momentum conservation equations. For time-series data after 1.06 s, the suggested strategy showed much higher accuracy than the single-training approach. Because the error with the ground truth did not diverge in the unsteady simulation, the cross-coupling strategy can achieve partial acceleration in the ML prediction zone.

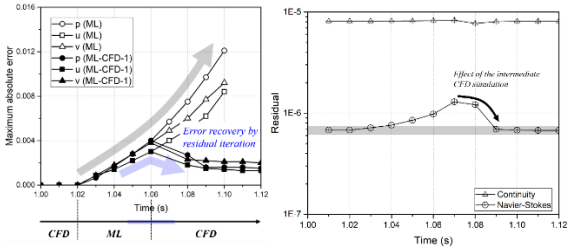

Fig. 6. Comparison of maximum absolute error of each variable (left) and calculated residual of each governing equation (right) in ML and ML-CFD time series. As the residual of the Navier-Stokes equations was stabilized by the intermediate CFD simulation, the error-recovery effect was observed.

3.3 Acceleration Performance

At the 20,000 grid points condition, the average times for ML and CFD calculations were 0.012 and 0.033 s, respectively. In the case with 500,000 grid points with the same boundary/initial condition, the average times for ML and CFD calculations were 0.035 and 0.55 s, respectively. In this case, the ML can compute the CFD time series 15 times faster. Finally, about 2.0 times acceleration effect was certificated in the assumed 1:1 ML-CFD cross computation condition considering the parameter-updating time. The CPU and GPU systems are an Intel Xenon Gold (56 cores/112 threads) 6258R Processor (2.70 GHz) and an NVIDIA TESLA A100 (CUDA core, 40GB MEM), respectively.

4. Conclusions

In this study, a laminar-flow CFD dataset was used to investigate the feasibility of the proposed ML-based strategy for CFD acceleration. The developed physics-informed loss function can not only increase the accuracy, but also play the role of accuracy-monitoring function. Because it was a relatively simple simulation which does not include turbulence or reaction models, the original CFD simulation was already fast. In calculations where the nonlinearity of the governing equations is more prominent, the CFD simulation time will raise sharply compared to the parameter-updating time and the computation time in machine-learning frameworks. In other words, the acceleration performance of this strategy will be greater for turbulent- or reacting-flow CFD problems. Observing the variation in acceleration performance according to the CFD complexity is our future work.

Acknowledgement

The authors would like to acknowledge the support from the Nuclear Safety Research Program through the Korea Foundation of Nuclear Safety (KoFONS) using the financial resource granted by the Nuclear Safety and Security Commission (NSSC) of the Republic of Korea (Grant No. 1805001-0522-CG100).

REFERENCES

[1] I. Tolias, et al., Numerical simulations of vented hydrogen deflagration in a medium-scale enclosure, Journal of Loss Prevention in the Process Industries 52 (2018) 125–139.
[2] N.K. Kim, et al., Systematic hydrogen risk analysis of OPR1000 containment before RPV failure under station blackout scenario, Annals of Nuclear Energy 116 (2018) 429–438.
[3] R. Vinuesa, et al., The role of artificial intelligence in achieving the Sustainable Development Goals, Nature Communications 11(1) (2020) 1-10.
[4] R. Vinuesa, S.L. Brunton, The potential of machine learning to enhance computational fluid dynamics, arXiv preprint arXiv:2110.02085 (2021).
[5] S. Lee, D. You, Data-driven prediction of unsteady flow over a circular cylinder using deep learning, Journal of Fluid Mechanics 879 (2019) 217–254.
[6] J. Jeon, J. Lee, S.J. Kim, Finite volume method network for acceleration of unsteady computational fluid dynamics: non-reacting and reacting flows, International Journal of Energy Research, In Press, 10.1002/er.7879.