Proceedings of the 12nd International Symposium on Heating, Ventilation and Air Conditioning (ISHVAC 2021) Seoul, Korea, 24-26 November 2021

A Novel Physics-informed Algorithm for Training AI Models to Predict Indoor Airflow

Cary A. Faulkner[1], Dominik S. Jankowski[2], Wangda Zuo[1,3,*], Phillip Epple[4], Michael D. Sohn[5]

[1] University of Colorado Boulder, Boulder, CO, U.S.A.
[2] HySON Institut, Sonneberg, Germany
[3] National Renewable Energy National Laboratory, Golden, CO, U.S.A.
[4] Coburg University of Applied Sciences, Coburg, Germany
[5] Lawrence Berkeley National Laboratory, Berkeley, CA, U.S.A.

[*] *Corresponding email: Wangda.Zuo@colorado.edu*

ABSTRACT

An artificial intelligence (AI) model trained by Computational Fluid Dynamics (CFD) simulations can be used to accelerate indoor airflow prediction. However, generating CFD data to train the AI model is time consuming. This work proposes a novel physics-informed algorithm to generate CFD training data for the AI model. The algorithm minimizes the amount of training data needed while maintaining the training quality by dynamically selecting training data points based on critical physical outputs. The proposed algorithm is then evaluated using the classical lid-driven cavity flow with three different training output thresholds. The results show that the algorithm can reduce the amount of training data by up to 97% and training time by up to 85% compared to the conventional uniform data generation approach. The resulting trained AI model can be used to explain the distribution of velocity magnitude within 5% error compared to the CFD model.

KEYWORDS: Artificial intelligence, computational fluid dynamics, physics-informed algorithm, indoor airflow.

1 INTRODUCTION

Stratified indoor airflow simulation is needed for many applications to understand how the airflow distribution affects the indoor environment. For example, Han et al. (2021) studied the stratified airflow and thermal environment resulting in data centers in order to improve the operation energy efficiency. Tian et al. (2019) used indoor airflow simulation to optimize thermostat placement in an office to balance energy efficiency and thermal comfort. More recently, researchers have studied the spread of COVID-19 virus indoors to improve indoor air quality and reduce risk of infection (Bhattacharyya, 2020; Mirzaie, 2021).

All these studies use Computational Fluid Dynamics (CFD) models to numerically solve the governing equations of fluid motion. While this technique is suitable for certain applications, it can also be excessively computationally expensive for many applications. This includes long-duration evaluations (e.g., annual simulations) or optimal design problems that require many simulation realizations. CFD is also too slow for applications that require faster than real time simulations, such as emergency management scenarios.

Artificial intelligence (AI) approaches trained with CFD data show promise for fast simulation of indoor airflow. Zhou and Ooka (2020) used deep learning trained with CFD data to predict isothermal airflow in an office room. Cao and Ren (2018) used an artificial neural network (ANN) approach trained by CFD simulations to predict CO_2 concentration for online ventilation control. While promising, however, generating CFD data to train the AI model is often time consuming due to the computational expense of running CFD simulations and the need to run many simulations to provide sufficient training data for the AI modeling. A significant advancement would be a method to reduce the time required to generate the training data using CFD. Hanna et al. (2020) used coarse grids to run faster CFD simulations to provide training data for two machine learning models. Cao and Ren (2018) used linear superposition of CFD simulations to predict CO_2 concentration from multiple sources, which increased the training dataset without the need to run additional CFD simulations.

While studies show the ability to accelerate the training process when using CFD, they are often case dependent and cannot be applied generally to a wide range of indoor airflow studies. To address this gap, we propose a physics-informed algorithm to strategically generate training data to train an AI model that can be applied to a wide variety of physical scenarios. The physics-informed algorithm is inspired by the In-Situ Adaptive Tabulation (ISAT) algorithm (Pope, 1997), which has been used to accelerate stratified indoor airflow predictions in the literature (Tian et al., 2018). The proposed physics-informed algorithm decides which training datapoints are needed within a given input range based on how significantly critical outputs change with the varying input. The algorithm is evaluated for a benchmark isothermal airflow case using a Conditional Generative Adversarial Network (CGAN) (Mirza and Osindero, 2014) AI model.

The remainder of this paper is organized as follows. First, the physics-informed algorithm is outlined in Section 2. Next, the benchmark isothermal airflow case is described in Section 3. The CGAN AI model is then detailed in Section 4. We show the results of using the physics-informed algorithm to train the AI model for isothermal airflow prediction in Section 5. Finally, conclusions are made in Section 6.

2 PHYSICS-INFORMED ALGORITHM

Figure 1(a) shows a flowchart describing the physics-informed algorithm and Figure 1(b) provides an example of how training data can be generated using the algorithm. First, initial grid points $(x_1, \dots, x_N)$ within the training domain are generated. In Figure 1(b), this represents the black markers generated in the input range [*a, b*]. The algorithm then checks the critical output, *g(x)*, at the point between the first two grid points (e.g., the first two black markers in Figure 1(b)). If the change in this output compared to the same output at the neighboring grid points exceeds a defined threshold, ε, then a new grid point is added. This can be described as:

$$|g((x_{n+1} + x_n)/2) - g(x_{n+1})| > \varepsilon, or\ |g((x_{n+1} + x_n)/2) - g(x_n)| > \varepsilon, \quad (1)$$

where $g(x)$ is the critical output at location x, x_n and x_{n+1} are the grid points being checked, $(x_{n+1}+x_n)/2$ is the midpoint between the two grid points, and ε is the defined threshold to determine whether a new grid point should be added. For example, the first blue marker is added between the first two black markers in Figure 1(b). The process continues until the final grid point is reached (e.g., the last black marker at *b* in Figure 1(b)). Next, the resolution increases and new grid points are checked nearby the previously added grid points. For example, the red markers are added nearby the blue markers in Figure 1(b). The process continues until either no new grid points are added or a minimum resolution is reached.

This algorithm offers two potential benefits. First, it can reduce the amount of training data generated, thus decreasing the time needed to run simulations (e.g., computationally expensive CFD simulations). For example, some studies may generate a large, uniformly distributed dataset to train their AI model for simplicity. However, this algorithm can selectively decide which data points are needed within the domain and thus potentially reduce the amount of training data generated compared to a large, uniform dataset. Second, it can improve the training time or accuracy of the AI model by providing a diverse dataset representing the variety of possible outputs across the input domain. For a large, uniformly distributed

training dataset, the AI model may become overfitted to predict similar outputs seen within the training data. By avoiding redundant data points, the algorithm can avoid such overfitting and better learn the diverse possibility of outputs within the input domain.

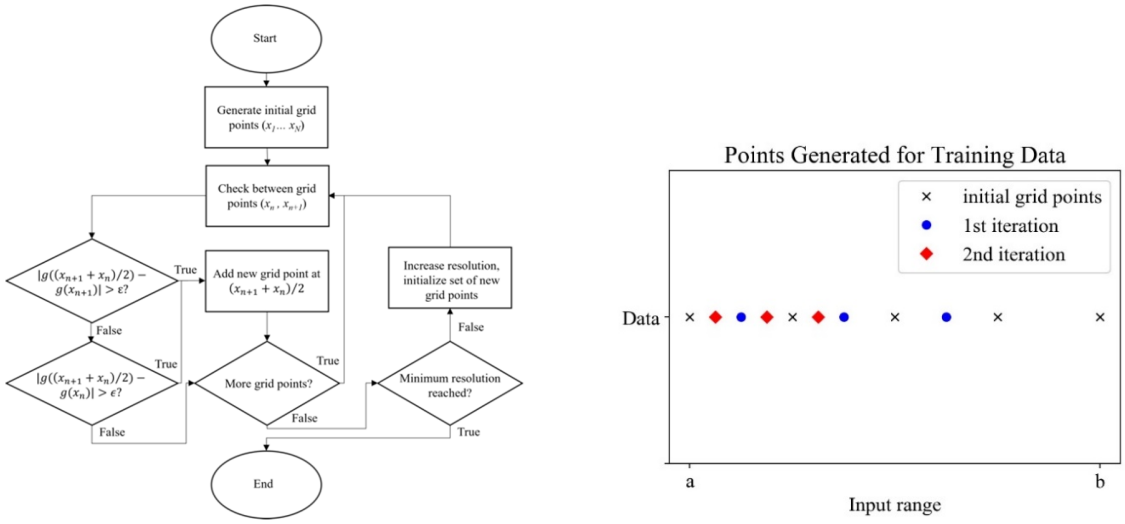


Figure 1. (a) Flowchart of physics-informed algorithm and (b) example output of training data points using the physics-informed algorithm.

3 CASE DESCRIPTION

Figure 2(a) shows the isothermal lid-driven cavity case used for this study. This is a classical flow case that has been used in many studies (Ghia et al., 1982; Hanna et al., 2020). This two-dimensional flow has three fixed walls (left, right, and bottom) and a lid that moves at constant velocity, U_0. The motion of the lid shears the fluid in the cavity and causes a recirculation pattern. The three walls and lid all have the same length, L. The lid-driven cavity flow can be characterized by the Reynolds (Re) number:

$$Re = \frac{U_0 L}{\nu}, \tag{2}$$

where Re is the Reynolds number, U_0 is the constant velocity of the lid, L is the length of the lid, and ν is the kinematic viscosity of the fluid.

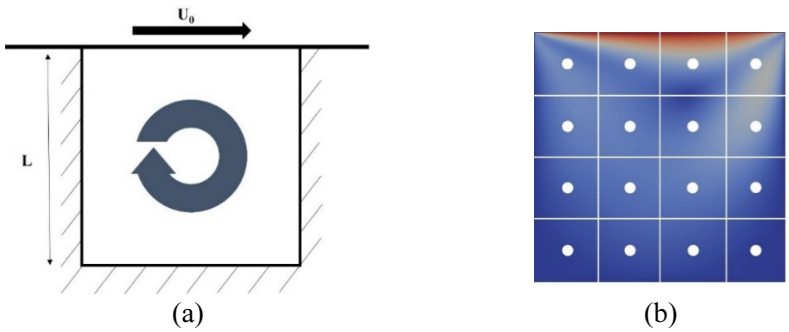


Figure 2. (a) Diagram of lid-driven cavity case and (b) locations of critical outputs for the physics-informed algorithm.

For this study, the AI model receives the Re number as the input and outputs the velocity magnitude contour. A simplified CFD model known as Fast Fluid Dynamics (Zuo and Chen, 2009) is used to simulate the flow. The flow pattern within the cavity can vary greatly depending on the Re number. The Re number is varied in this study by holding U_0 and L constant and varying v. The input domain for the AI model is $Re \in [100, 10000]$. The critical outputs for the physics-informed algorithm are shown with the 16 points on the grid in Figure 2(b).

4 AI MODEL DESCRIPTION

A CGAN AI model is used to predict the lid-driven cavity velocity magnitude contour based on the Re number for this study. The CGAN model is chosen due to its ability to accurately and efficiently predict distribution of image data based on a categorical input. Mokhtar et al. (2020) demonstrated the use of a CGAN model trained by CFD simulations to predict pedestrian wind flow around different geometries. Figure 3 describes the CGAN architecture. The CGAN uses two competing neural networks: the generator (G) and discriminator (D). The generator receives an input vector and attempts to output a realistic image based on this input. For this case, it attempts to output the lid-driven cavity velocity magnitude contour. The discriminator receives "real" training data images and "fake" images produced by the generator and attempts to classify each received image as real or fake. Both the generator and discriminator receive a label which corresponds to the input Reynolds number for the data point. Initially, the generator is not properly trained and is unable to produce realistic images. As the training process continues, the generator learns based on feedback from the discriminator and is able generate more realistic images. Eventually, the generator produces images that the discriminator cannot distinguish from the real images, and the discriminator essentially "flips a coin" to guess if they are real or fake.

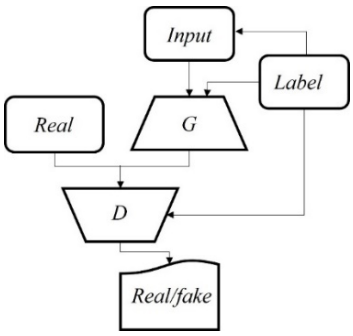
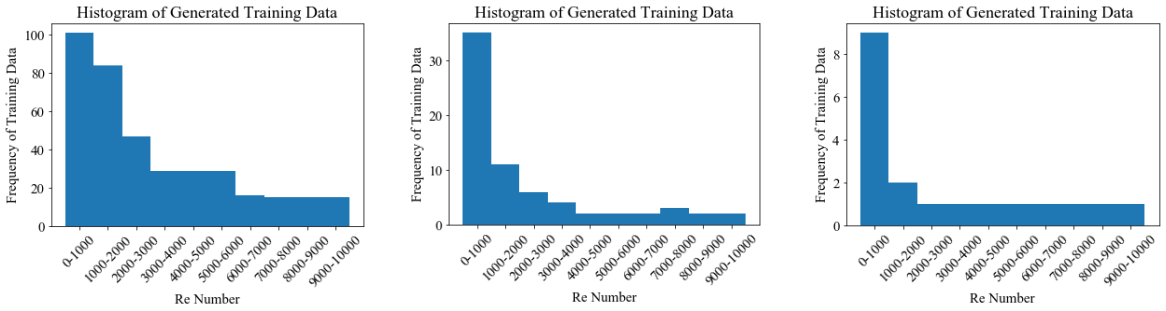


Figure 3. Conditional Generative Adversarial Network (CGAN) architecture.

The CGAN in this study uses Convolutional Neural Networks (CNNs) (LeCun et al., 1989) for the generator and discriminator. This type of neural network was chosen due to their ability to process images, including those generated from CFD simulations (Guo et al., 2016). The CGAN is trained until the generator produces images for a few representative labels with less than 5% error. The error is calculated by comparing the output of the AI model to the data generated from the CFD simulation. We also verify the accuracy of the model by comparing the centerline (x = ½) velocity for the output of the AI model with the CFD data, as shown in Figure 5. The representative labels chosen for this study are $Re = [100, 500, 1000, 10000]$ and demonstrate the variation of flow pattern with Re number within the domain.

5 RESULTS

Three different thresholds, ε, were used to generate the training data using the physics-informed algorithm. Figure 4 shows histograms with the distribution of training data within the input range for the three threshold values. The smaller thresholds result in more training data since the algorithm must add new data points for less significant changes in the critical outputs. These training data sets were compared against a baseline of a uniformly distributed data set, which included data points from Re number 100 to 10000 by step of 5 (e.g., $Re = [100, 105, 110, \ldots, 10000]$).



(a) ε = 0.001 m/s (b) ε = 0.01 m/s (c) ε = 0.05 m/s

Figure 4. Distribution of training data generation for different thresholds.

Figure 5 shows an example of the AI model prediction capability once fully trained. The AI model is trained until it produces images for the reference labels with less than 5% error for all four training data sets. Thus, the AI model behaves similarly once trained for all the training data sets. Table 1 summarizes the training results for the three different thresholds and the baseline. The results show that using the physics-informed algorithm can reduce the training time by up to 85% and reduce the number of training data points by up to 97% compared to the baseline. Loosening the threshold of the physics-informed algorithm from ε = 0.001 m/s to ε = 0.01 m/s or ε = 0.05 m/s can reduce the time to sufficiently train the AI model. However, there seems to be an optimal point when further increasing the threshold begins to increase the training time, as seen by the increase in training time when loosening the threshold from ε = 0.01 m/s to ε = 0.05 m/s.

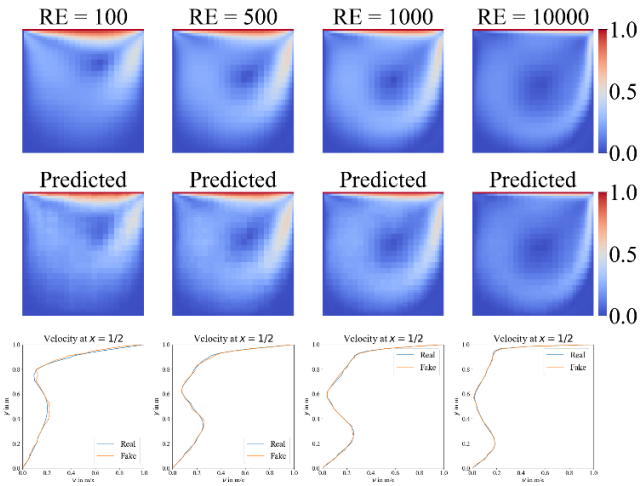


Figure 5. Example results of the fully trained AI model.

Table 1. Results for training

Case	Number of training data points	Training Time
Baseline	1981	72 min
ε = 0.001 m/s	380	19 min
ε = 0.01 m/s	69	11 min
ε = 0.05 m/s	19	13 min

6 CONCLUSION

We developed and evaluated a physics-informed algorithm that strategically reduces the training data for an AI model. The algorithm was demonstrated for an isothermal lid-driven cavity case with a CGAN AI model. The results show that use of the physics informed algorithm can reduce the training time by up to 85% and reduce the number of training data points needed by up to 97% compared to a baseline of uniformly generated training data. The accuracy of the AI model remains the same for all the cases and can predict the flow with less than 5% error. The proposed physics-informed algorithm shows promise for training other physical-based AI models, since this approach is not specific to indoor airflow prediction

trained by CFD simulations. The algorithm can be used to efficiently train AI models for real-time prediction or optimization of physical scenarios.

7 ACKNOWLEDGEMENTS

This research was supported in part by the U.S. Defense Threat Reduction Agency and performed under U.S. Department of Energy Contract No. DE-AC02-05CH11231. This research was also supported in part by the Europe-Colorado Mobility Program.

REFERENCES

Bhattacharyya, S., Dey, K., Paul, A. R., & Biswas, R. (2020). A novel CFD analysis to minimize the spread of COVID-19 virus in hospital isolation room. Chaos, Solitons & Fractals, 139, 110294.

Cao, S. J., & Ren, C. (2018). Ventilation control strategy using low-dimensional linear ventilation models and artificial neural network. Building and Environment, 144, 316-333.

Ghia, U. K. N. G., Ghia, K. N., & Shin, C. T. (1982). High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. Journal of computational physics, 48(3), 387-411.

Guo, X., Li, W., & Iorio, F. (2016, August). Convolutional neural networks for steady flow approximation. In Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (pp. 481-490).

Han, X., Tian, W., VanGilder, J., Zuo, W., & Faulkner, C. (2021). An open source fast fluid dynamics model for data center thermal management. Energy and Buildings, 230, 110599.

Hanna, B. N., Dinh, N. T., Youngblood, R. W., & Bolotnov, I. A. (2020). Machine-learning based error prediction approach for coarse-grid Computational Fluid Dynamics (CG-CFD). Progress in Nuclear Energy, 118, 103140.

LeCun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W., & Jackel, L. (1989). Handwritten digit recognition with a back-propagation network. Advances in neural information processing systems, 2.

Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.

Mirzaie, M., Lakzian, E., Khan, A., Warkiani, M. E., Mahian, O., & Ahmadi, G. (2021). COVID-19 spread in a classroom equipped with partition–A CFD approach. Journal of Hazardous Materials, 420, 126587.

Mokhtar, S., Sojka, A., & Davila, C. C. (2020). Conditional generative adversarial networks for pedestrian wind flow approximation. Proceedings of SimAUD, 25-27.

Pope, S. B. (1997). Computationally efficient implementation of combustion chemistry using in situ adaptive tabulation.

Tian, W., Sevilla, T. A., Li, D., Zuo, W., & Wetter, M. (2018). Fast and self-learning indoor airflow simulation based on in situ adaptive tabulation. Journal of Building Performance Simulation, 11(1), 99-112.

Tian, W., Han, X., Zuo, W., Wang, Q., Fu, Y., & Jin, M. (2019). An optimization platform based on coupled indoor environment and HVAC simulation and its application in optimal thermostat placement. Energy and Buildings, 199, 342-351.

Zuo, W., & Chen, Q. (2009). Real-time or faster-than-real-time simulation of airflow in buildings. Indoor air, 19(1), 33.

Zhou, Q., & Ooka, R. (2020, December). Comparison of different deep neural network architectures for isothermal indoor airflow prediction. In Building Simulation (Vol. 13, No. 6, pp. 1409-1423). Tsinghua University Press.