# An AI-Accelerated CFD Application on a Benchmark Device: FDA Nozzle

*Ibrahim Basar Aka*[*], Mehmet Iscan[†]
[*]Mechatronics Engineering Department Istanbul Bilgi University, Istanbul, Turkiye
Email: basar.aka@bilgi.edu.tr
[†]Mechatronics Engineering Department Yıldız Technical University
Email: miscan@yildiz.edu.tr

***Abstract*—In this study, we suggest a procedure for speeding CFD (computational fluid dynamics) analysis up by combining a conventional opensource CFD solver with a traditional AI module. The studied case is the FDA benchmark nozzle with various Reynolds numbers. The considered CFD simulations belong to a group of steady-state simulations and utilize the laminar flow solver SimpleFoam in the OpenFOAM toolbox. The proposed module is implemented as a Feed-Forward Neural Network (FFNN) supervised learning procedure. Our method distributes the data by creating a combined AI model for each quantity of the simulated phenomenon for various Reynolds numbers. The model can then be combined after the initial iteration phase to decrease the execution time or to lower memory requirements. We analyze the performance of the proposed method depending on the estimation accuracy of the data of interest, velocity, and pressure. For test data, we achieve time-to-solution discounts of nearly a factor of 10. Comparing simulation results based on the FFNN test results and 3D visualization shows the average accuracy for all the parameters over 99% for the velocity and the pressure.**

***Keywords*—AI acceleration for CFD; Convolutional neural networks; FDA Nozzle; 3D grids; OpenFOAM; CPU computing.**

## I. Introduction

Machine learning and artificial intelligence (AI) methods have grown widespread recently due to various algorithmic improvements and the availability of computational resources[1]–[3]. In computational fluid dynamics (CFD), these methods have been used to replace, accelerate or improve traditional solvers[4]. This work focuses on the Al-based acceleration of a CFD tool used for FDA benchmark nozzle simulations. Computational fluid dynamics (CFD) is ever more used to design blood-wetted medical devices. However, the lack of consistent methods for validating CFD simulations and blood damage predictions limits its use in the reliable computational evaluation of devices[5], [6]. U.S. Food and Drug Administration (FDA) released two benchmark models of typical device flow geometries (a nozzle and a centrifugal pump). These models were examined in numerous laboratories to measure flow-related quantities like velocities, pressures, and blood damage data to provide information for CFD validation efforts[7]–[10]. In addition, computational simulations were performed by more than 20 independent groups to assess current CFD techniques[7]. We propose a domain-specific AI model and a method of integrating them with the simpleFoam solver of OpenFOAM, an open source CFD package.

TCAE (CFD Support, Czechia) software was used for meshing and OpenFOAM v2006 was used for simulation and data generation. Cell positions, wall shear stress, mean pressure, and other parameters were recorded. Aortic flow estimations using neural networks have been performed in the last decade. However, their results have not been validated or compared with a computational benchmark[11].

We aim to provide a two-way interaction between AI and CFD solvers for much faster analysis. This method would help to minimize the computational cost of simulating a cardiovascular device geometry under various flow conditions. Matlab (Mathworks, USA) Feed-Forward Neural Network Toolbox (FFNN) was used to train the model. The scope of this model covers steady-state simulations, which use an iterative method to move forward to convergence. Steady-state models consider a mass and energy balance independent of time. The solver calculates a set of iterations to achieve the convergence state of the simulated case. Therefore, our method aims to predict the convergence state with the AI model based on a few initial iterations generated by the CFD solver. Thus, the rest of the iterations to produce the convergence result are avoided. Consequently, the time required for an accurate solution is considerably reduced. The proposed model makes it viable to study a design under various conditions before moving into the physical experiment stage or making new decisions about the design.

There are three main contributions of this work. Firstly, an AI-based method predicts a simulation's convergence state based on initial iterations generated by the CFD solver. This method is integrated with the CFD solver and significantly reduces the simulation time. Secondly, a method of AI integration with the OpenFOAM toolbox and high-performance computing with a 32-core CPU. An integrated framework is used for two-way data exchange between the CFD solver OpenFOAM and the AI training environment Matlab. Results are visualized after prediction in this framework. Finally, performance and accuracy analysis of the AI-accelerated simulations are also presented.

978-1-6654-5432-2/22/$31.00 ©2022 IEEE

2022 Medical Technologies Congress (TIPTEKNO) | 978-1-6654-5432-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/TIPTEKNO56568.2022.9960231

## II. Methods

### A. CFD Method

The FDA Nozzle geometry is created in SolidWorks (Dassault Systems, France) following the dimensions provided in the literature[12]. The dimensions are given in Fig. 1.

TCAE software uses the open source text-based snappyHexMesh algorithm for creating the grid with a user interface. The grid size is 0.25 mm and the inflation layers were added to capture the boundary layer. FDA nozzle geometry is intended to have a sudden expansion for investigating jet breakdowns and shear stresses acting on the blood cells. Therefore the flow conditions were selected with high Reynolds numbers in the nozzle section.5M to 20M elements were used in the literature To capture the flow field completely at this high-velocity flow regime. However, We used 3.5K elements to reduce the computational cost. The grid generation process was controlled regarding orthogonal quality, aspect ratio, and skewness were checked to be in the required range.

CFD calculations were conducted with the simpleFOAM algorithm of OpenFOAM v2006. Three steady-state cases with velocity-inlet pressure-outlet boundary conditions were studied considering Re numbers corresponding to 3500, 5000, and 6500 inside the narrow tube. A convergence criterion of the scaled residuals was $10^{-4}$. The blood was modeled as an incompressible Newtonian fluid with 1050 $kg/m^3$ density and 0.0035 Pa.s dynamic viscosity. Even though the flow conditions were mostly turbulent, a laminar solver was used for the simplicity of the computation for the scope of this work.

The average solution for this setup takes 10 minutes with a 32-core Intel Xeon Gold 5218 2.3 GHz CPU. The setup case files were parallelized for 32 cores using the decomposePar command. Results were recorded per 10 iterations. For the neural network training, the 10th iteration results were assumed as the inlet conditions, and the 250th iteration was assumed to be the convergence result.

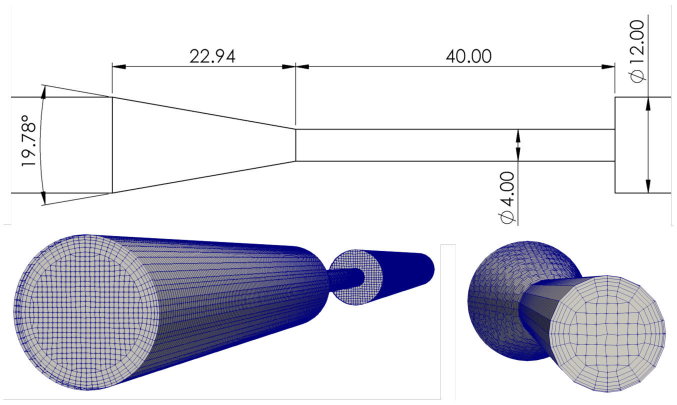

Fig. 1: Top: Dimensions of the FDA nozzle in mm. Bottom Left: The reduced grid of the nozzle at the inlet. Bottom Right: The grid inside the nozzle section.

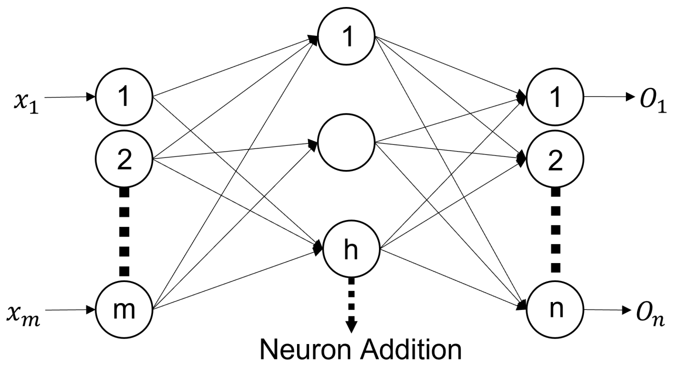

Fig. 2: ANN Scheme.

### B. Feed-Forward Neural Network Method

A Feed-forward Neural Network was trained on Matlab Feed-Forward Neural Network Toolbox. The neural network training scheme is depicted in Fig. 2.

Feed Forward Neural Networks (FFNN) is a neural network model that can structurally increase or decrease neurons for given input and output values in unknown complexity level situations[13]. Instead of backpropagation learning as a training algorithm Levenberg-Marquardt algorithm is used[14]. This algorithm can learn fairly quickly, unlike backpropagation. In this way, the training of the coefficients can be completed instantly, and there is an ability to express situations by changing the parameters optimally. The expansion algorithm was used only when the training was no longer realized. The FFNN model was started with three neurons and a maximum of 10 was allowed to grow up to a neuron. The increment rule was $\Delta$ E<$10^{-7}$. The input vector x is generated with the values of $10^{th}$ iteration and the output vector y with $250^{th}$ iteration, shown in expression (1) and (2).

$$x = [U_x, U_y, U_z, P, Re, C_x, C_y, C_z] \tag{1}$$

$$y = [U_x, U_y, U_z, P] \tag{2}$$

The Matlab code reads the OpenFoam result files, U and p, as the FFNN inputs. The activation function is selected as sigmoid one, which is helpful to generate between 0 to 1 values at the end of each neuron. This function enables FFNN to converge quickly by using Levenberg-Marquardt at each iteration. In the hidden layer, the real neurons are created using this kind of neurons. The output layer of neurons consists of a linear activation function that produces linear outputs, not between 0 and 1.

Normalization is used to prepare the input and output vector at the learning phase for setting the range of all input/output parameters. Each input/output value is scaled with its range's highest and lowest values. Then, the FFNN structure can easily be converted using the Levenberg-Marquardt method since all values are restricted between 0 to 1 values. Test results of the FFNN are also converted to OpenFoam file format for visualization purposes.

## III. Results

The whole training and test procedure were performed in MATLAB software with the aid of Neural Network Toolbox to establish the FFNN structure. There were two phases named with normalization process and training, one to train the proposed method. After that, the proposed method was kept with its trained coefficients and different OpenFOAM inputs were fed to the FFNN structure.

The maximum velocity of the cases is between 1.1x$10^{-7}$ to 1.7x$10^{-7}$ m/s inside the nozzle section. The convergence is achieved between 175 to 240 iterations for the given BCs. The solution takes  10 min. CFD results of the midplane is presented in Fig. 3. FFNN training took 3 hours with a 30 core parallelization in Matlab.

The estimated case Re 5000 is compared with the CFD results on the mid-plane of the model. Estimation was reasonably successful, showing a difference of less than an order of $10^{-20}$ in Fig. 4.

In Table I, four parameters indicate the FFNN structure regarding the number of hidden neurons, test, and training results. In our approach, FFNN was iterated in a two-layer structure, and each hidden layer possesses a different number of neurons.

Table I showed that FFNN performance was improved when increasing neuron numbers at each layer. Training results were always greater than test one since Reynolds number(5000) differs from the trained one. Besides, test results indicated that the best performance was reached at two hidden layers with ten neuron structures in which it was suitable to utilize offline operation. It was noteworthy that the proposed FFNN might be performed for real-time operation while using a hidden layer with six neurons which attained the 8.52x$10^{-7}$ error value.

| Hidden Layers | Neuron Number | Test Results | Training Results |
|---|---|---|---|
| 1 | 3 | 5.52e-5 | 5.32e-5 |
| 1 | 4 | 2.94e-5 | 2.89e-5 |
| 1 | 5 | 2.03e-6 | 2.02e-6 |
| 1 | 6 | 8.52e-7 | 8.74e-7 |
| 1 | 7 | 5.72e-7 | 5.58e-7 |
| 1 | 8 | 7.24e-7 | 7.27e-7 |
| 1 | 9 | 7.43e-7 | 7.35e-7 |
| 1 | 10 | 5.45e-7 | 5.37e-7 |
| 2 | 3 | 2.92e-5 | 2.64e-5 |
| 2 | 4 | 5.48e-5 | 4.74e-5 |
| 2 | 5 | 1.00e-6 | 9.91e-7 |
| 2 | 6 | 5.50e-7 | 5.58e-7 |
| 2 | 7 | 7.11e-7 | 7.08e-7 |
| 2 | 8 | 4.02e-7 | 4.05e-7 |
| 2 | 9 | 3.66e-7 | 3.62e-7 |
| 2 | 10 | 3.63e-7 | 3.56e-7 |

TABLE I: The test and training results of the network.

The suggested FFNN structure was trained with the Levenberg-Marquardt algorithm, which paved the way for the fastest convergence speed in a large dataset. However, the standard backpropagation algorithm might be reached the highest rate in test results.

On the other hand, Fig. 4 clearly stated that each velocity and pressure difference which were at the rate of $10^{-20}$

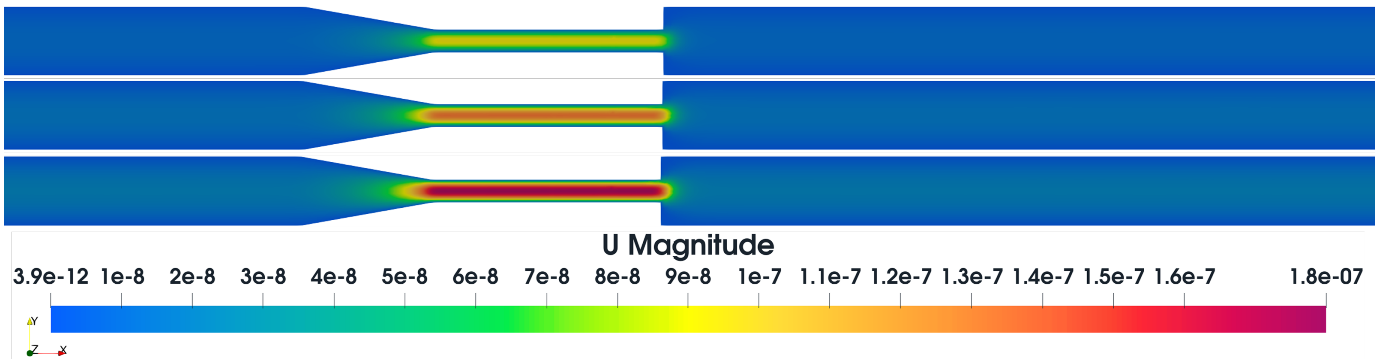

Fig. 3: Velocity profile of the cases Re 3500, Re 5000, Re 6500, from top to bottom.

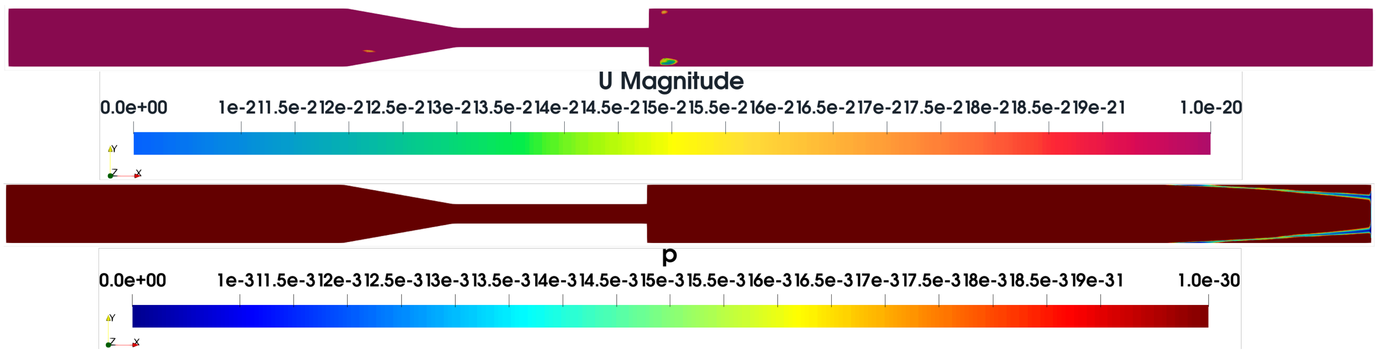

Fig. 4: Difference between the CFD result vs. Estimation for the Re 5000 case.Top: $\Delta U$, Bottom:$\Delta P$.

and $10^{-30}$ were converged to stability when the 3D results were evaluated. The regional patterns were captured with the different areas of nozzle structure, which means that physical parameters such as pressure and velocities could be imitated using the proposed method.

## IV. Conclusion

This proof-of-concept study investigates machine-learning supported CFD calculations. The results show good agreement with the traditional CFD calculations. Even though the scope of this study is limited to a single geometry and three cases, this method applies to studies like blood pumps or aerial vehicle design, where a vast number of simulations are required. There are several limitations to this study. First, mesh size is insufficient to capture the boundary layer or all flow features. Also, the number of cases is limited to three. Finally, a laminar solver is used for simplicity where the flow is mainly turbulent.

## Acknowledgment

This study is supported by Istanbul Bilgi University Scientific Research Fund (BAP).

## References

[1] R. Archibald, E. Chow, E. D'Azevedo, J. Dongarra, M. Eisenbach, R. Febbo, F. Lopez, D. Nichols, S. Tomov, and K. Wong, "Integrating deep learning in domain sciences at exascale," in *Smoky Mountains Computational Sciences and Engineering Conference*. Springer, Conference Proceedings, pp. 35–50.

[2] D. Bhatt, B. W. Zhang, and D. M. Zuckerman, "Steady-state simulations using weighted ensemble path sampling," *The Journal of chemical physics*, vol. 133, no. 1, p. 014110, 2010.

[3] N. P. Jouppi, C. Young, N. Patil, and D. Patterson, "A domain-specific architecture for deep neural networks," *Communications of the ACM*, vol. 61, no. 9, pp. 50–59, 2018.

[4] O. Obiols-Sales, A. Vishnu, N. Malaya, and A. Chandramowliswharan, "Cfdnet: A deep learning-based accelerator for fluid simulations," in *Proceedings of the 34th ACM international conference on supercomputing*, Conference Proceedings, pp. 1–12.

[5] I. B. Aka, C. Ozturk, and I. Lazoglu, "Numerical investigation of volute tongue design on hemodynamic characteristics and hemolysis of the centrifugal blood pump," *SN Applied Sciences*, vol. 3, no. 1, pp. 1–9, 2021.

[6] C. Ozturk, I. B. Aka, and I. Lazoglu, "Effect of blade curvature on the hemolytic and hydraulic characteristics of a centrifugal blood pump," *The International journal of artificial organs*, vol. 41, no. 11, pp. 730–737, 2018.

[7] K. Jain, "Efficacy of the fda nozzle benchmark and the lattice boltzmann method for the analysis of biomedical flows in transitional regime," *Medical Biological Engineering Computing*, vol. 58, no. 8, pp. 1817–1830, 2020.

[8] A. W. Bergersen, M. Mortensen, and K. Valen-Sendstad, "The fda nozzle benchmark:"in theory there is no difference between theory and practice, but in practice there is"," *International journal for numerical methods in biomedical engineering*, vol. 35, no. 1, p. e3150, 2019.

[9] L. H. Herbertson, S. E. Olia, A. Daly, C. P. Noatch, W. A. Smith, M. V. Kameneva, and R. A. Malinauskas, "Multilaboratory study of flow-induced hemolysis using the fda benchmark nozzle model," *Artificial organs*, vol. 39, no. 3, pp. 237–248, 2015.

[10] P. Hariharan, M. Giarra, V. Reddy, S. W. Day, K. B. Manning, S. Deutsch, S. F. Stewart, M. R. Myers, M. R. Berman, and G. W. Burgreen, "Multilaboratory particle image velocimetry analysis of the fda benchmark nozzle model to support validation of computational fluid dynamics simulations," *Journal of biomechanical engineering*, vol. 133, no. 4, 2011.

[11] B. Feiger, J. Gounley, D. Adler, J. A. Leopold, E. W. Draeger, R. Chaudhury, J. Ryan, G. Pathangey, K. Winarta, and D. Frakes, "Accelerating massively parallel hemodynamic models of coarctation of the aorta using neural networks," *Scientific reports*, vol. 10, no. 1, pp. 1–13, 2020.

[12] R. A. Malinauskas, P. Hariharan, S. W. Day, L. H. Herbertson, M. Buesen, U. Steinseifer, K. I. Aycock, B. C. Good, S. Deutsch, and K. B. Manning, "Fda benchmark medical device flow models for cfd validation," *Asaio Journal*, vol. 63, no. 2, pp. 150–160, 2017.

[13] G.-B. Huang, P. Saratchandran, and N. Sundararajan, "A generalized growing and pruning rbf (ggap-rbf) neural network for function approximation," *IEEE transactions on neural networks*, vol. 16, no. 1, pp. 57–67, 2005.

[14] M. T. Hagan and M. B. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.