International Journal of Sustainable Social Culture, Science Technology, Management, and Law Humanities, 3(1), (2026)

Parametric Analysis of Reynolds Number Effects on Wake Region Characteristics in LBM-Based Fluid Flow Simulations

Aulia Rahman[1], Putra Dwiantoko[1], Novan Habiburrahman[1], Djarot Winoto[1], Ahmad Robittah[2], Reinaldo Evan Audrey[3]

[1]Teknik Mesin, Universtias Kristen Cipta Wacana, Malang, Indonesia
[2]Teknik Elektromedik, Politeknik Unggulan Kalimantan, Banjarmasin, Indonesia
[3] Teknik Mesin, Universitas Katolik Widya Karya, Malang, Indonesia

Article history
Received: 7 May 2026
Revised: 21 June 2026
Accepted: 10 July 2026

Keywords
Lattice Boltzmann Method
CFD
Python
Laminar Flow
Reynolds Number

Abstract
This study investigates the fundamental fluid dynamics of steady-state laminar flow past a circular cylinder using the Lattice Boltzmann Method (LBM) with a D2Q9 lattice arrangement. Developed using a vectorized Pythonic approach, the numerical solver employs the Bhatnagar-Gross-Krook (BGK) collision operator to resolve the mesoscopic particle distribution functions. The research focuses on the flow characteristics at low Reynolds numbers (*Re*) ranging from 20 to 70 to ensure stable convergence toward steady-state conditions. Results demonstrate that the LBM effectively captures critical hydrodynamic phenomena, including stagnation points, flow acceleration due to blockage effects, and the development of symmetric wake regions. Analysis reveals a linear correlation between increasing Reynolds numbers and the longitudinal elongation of the recirculation zone, confirming the solver's ability to maintain numerical stability and momentum conservation without the computational expense of conventional pressure-velocity coupling. These findings validate the use of open-source computational tools for rigorous fluid flow analysis and provide a robust dataset for future integration with multi-scale modeling and machine learning frameworks, such as Physics-Informed Neural Networks (PINNs).

1. Introduction

The development of Computational Fluid Dynamics (CFD) in the last two decades has undergone a paradigm shift from macroscopic solutions toward mesoscopic approaches (Mansouri et al., 2024). Conventional methods based on the Navier-Stokes equations often face efficiency hurdles, particularly in solving the Poisson equation for pressure, which requires high computational costs in incompressible systems. As a promising alternative, the Lattice Boltzmann Method (LBM) has emerged, offering flexibility through simplified gas kinetic theory (Ataei & Salehipour, 2024). LBM stands out for its capability to handle complex boundary conditions locally and its algorithmic efficiency, which is highly adaptive to modern parallel computing architectures(Mansouri et al., 2024). The study of laminar flow over a stationary cylinder obstacle remains a gold standard benchmark in fundamental hydrodynamics. Although appearing simple, flow dynamics at low Reynolds numbers (Re) represent a crucial physical representation for understanding flow separation mechanisms and the formation of recirculation zones (Ataei & Salehipour, 2024). In an engineering context, a deep understanding of steady-state flow structures around such obstacles is a vital prerequisite in the design of heat exchanger systems (Ha et al., 2026), the optimization of offshore structures, and the development of microfluidic devices. Precision in capturing hydrodynamic phenomena at this scale significantly determines the validity of analysis at broader system levels (Fox et al., 2024).

Besides methodological advantages, the choice of computational tools plays a strategic role in research transparency (da Silva, 2026). Utilizing the Python programming language in LBM implementations offers advantages over commercial software that is often "black-box" in nature (Banh et al., 2025). Through a Pythonic approach leveraging vectorization techniques, researchers can precisely control every numerical aspect (Pasa et al., 2026), from collision schemes to the handling of non-periodic boundary conditions (Ataei & Salehipour, 2024). Furthermore, Python integration facilitates a bridge between classical numerical simulations and the development of new methodologies like Physics-Informed

*Aulia Rahman, email: auliarahman.fzn@gmail.com
doi: https://doi.org/10.71131/9f1ged35

Neural Networks (PINNs) (Zhai et al., 2026), which are currently a major trend in multiscale computational research (Ghafori, 2026).

This research aims to perform a numerical investigation of steady laminar flow characteristics over a single cylinder by applying a Python-based D2Q9 LBM algorithm (Ataei & Salehipour, 2024). The primary focus is directed toward evaluating flow stability and convergence in the low Re regime, as well as how bounce-back schemes on a fixed lattice are able to accurately represent velocity profiles (Oubnaki et al., 2026). Through the development of this custom solver, it is expected that this study will not only contribute to the fundamental understanding of vorticity dynamics but also demonstrate the effectiveness of open-source tools in producing high-quality, reproducible simulation data for further academic research needs (Santana et al., 2026).

2. Method

This research implements the Lattice Boltzmann Method using the D2Q9 discrete velocity model, representing two dimensions with nine velocity directions, to simulate steady laminar flow over a cylindrical obstacle (Ataei & Salehipour, 2024). In contrast to conventional Navier-Stokes solvers that utilize macroscopic approximations of derivatives, the Lattice Boltzmann Method operates at the mesoscopic level by evolving particle distribution functions (Kummerländer et al., 2025). These distribution functions are updated through a fundamental mechanism of collision and streaming on a structured lattice grid(Wen et al., 2017). From these mesoscopic distributions, macroscopic state variables such as fluid density and velocity are directly computed through summation, providing a robust framework for benchmarking fundamental hydrodynamic phenomena. Utilizing Python for this implementation offers strategic advantages in terms of code readability and ease of use while providing the necessary high-level interfaces for conducting complex mathematical operations (Ataei & Salehipour, 2024).

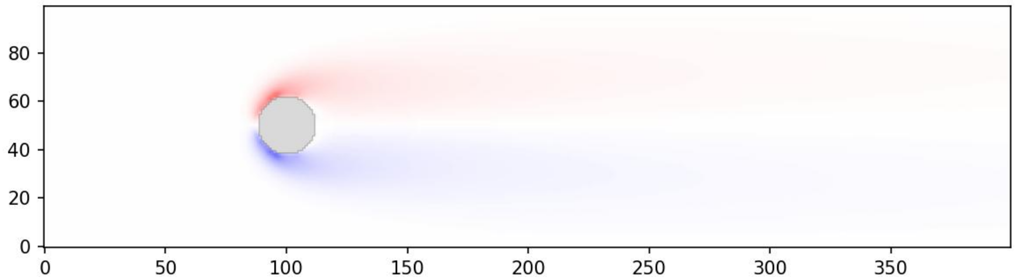

Fig. 1. Numerical visualization of the vorticity field for laminar flow past a circular cylinder, demonstrating the anti-symmetric distribution of vortex structures generated through the D2Q9 Lattice Boltzmann Method.

2.1. Model Kisi D2Q9

The computational domain is discretized into a structured Cartesian grid where every lattice point is defined by nine discrete velocity vectors e_i , which characterize the spatial and temporal dynamics of the fluid.

$$e_i = \begin{cases} (0,0) & i = 0 \\ (1,0), (0,1), (-1,0), (0,-1) & i = 1,2,3,4 \\ (1,1), (-1,1), (-1,-1), (1,-1) & i = 5,6,7,8 \end{cases} \quad \text{......(1)}$$

These discrete velocities are categorized based on their orientation as $e_i = (0,0)\ for\ i = 0$ representing stationary particles, axial directions, $e_i = (1,0), (0,1), (-1,0), (0,-1)\ for\ i = 1,2,3,4$ and $e_i = (1,1), (-1,1), (-1,-1), (1,-1)\ for\ i = 5,6,7,8$ for diagonal directions (Ataei & Salehipour, 2024). To guarantee the recovery of isotropy at the macroscopic level, each velocity direction is assigned a specific weight factor w_i, which is essential for the derivation of macroscopic state variables (Kummerländer et al., 2026). For this D2Q9 model, the weights are specifically defined as $w_0 = {}^4/_9$ for the stationary component, $w_{1-4} = {}^1/_9$for the axial vectors, and $w_{5-8} = {}^1/_{36}$ for the diagonal vectors. This mathematical framework allows the discrete populations to accurately represent the underlying physical properties of the fluid, such as density and velocity, through simple summation operations (Ataei & Salehipour, 2024).

2.2. Evolusi Boltzmann (LBGK) Equation

The fundamental equations governing the evolution of the distribution functions $f_i(x,t)$are numerically solved using the Bhatnagar-Gross-Krook (BGK) single-relaxation-time collision operator approach (Kummerländer et al., 2025). This operator characterizes the physical process where particle populations relax toward a local equilibrium state,f_i^{eq} , over a specific timescale (Ataei & Salehipour, 2024). The overall Lattice Boltzmann Method (LBM) algorithm is systematically divided into two primary sequential steps which are the collision step and the streaming step (Kummerländer et al., 2025).

During the collision phase, the distribution functions are modified based on their deviation from equilibrium, with the rate of relaxation determined by a parameter that is directly linked to the fluid's kinematic viscosity (Kummerländer et al., 2025). Subsequently, the streaming phase manages the spatial propagation of the fluid information by shifting the post-collision distribution functions to adjacent lattice nodes according to their respective discrete velocity vectors (Ataei & Salehipour, 2024). This fundamental "collide-and-stream" mechanism on a Cartesian grid allows the method to efficiently capture complex fluid dynamics at the mesoscopic level (Mansouri et al., 2024).

2.2.1. Collision Step

The particle distribution functions undergo a relaxation process toward a local equilibrium state, f_i^{eq} , at a rate determined by the single relaxation time parameter, τ (Kummerländer et al., 2025). This numerical process is governed by the Bhatnagar-Gross-Krook (BGK) collision operator, which is expressed as:

$$f_i^*(x,t) = f_i(x,t) - \frac{\Delta t}{\tau}[f_i(x,t) - f_i^{eq}(x,t)] \quad (2)$$

In this lattice-based framework, $f_i(x,t)$ represents the state of the particle distribution immediately before the collision phase, while $f_i^{eq}(x,t)$ defines the ideal equilibrium condition the fluid system theoretically aims to achieve (Kummerländer et al., 2025). Following the particle interactions, the final distribution is expressed as $f_i^*(x,t)$, reflecting the post-collision result (Ataei & Salehipour, 2024). This mechanism emphasizes that the fictitious fluid particles do not transform instantaneously into an equilibrium state; instead, they undergo a gradual "relaxation" or shift regulated by the parameter τ (Mansouri et al., 2024). Physically, this relaxation time represents the fluid's internal resistance or kinematic viscosity in response to flow field disturbances, and is directly related to the viscosity v through the relationship (Kummerländer et al., 2025):

$$v_{lbm} = c_s^2(\tau - 0.5)\Delta t \quad (3)$$

The lattice speed of sound parameter (c_s^2 = 1/3) is a fundamental constant in the D2Q9 model representing the mesoscopic thermodynamic characteristics of the fluid (Kummerländer et al., 2026). This value is a logical consequence of the integration of the Maxwell-Boltzmann distribution function, ensuring the accurate recovery of the Navier-Stokes equations within the incompressible limit (Kummerländer et al., 2025). Functionally, c_s^2 acts as a bridge in the equation of state, enabling the macroscopic pressure to be calculated locally through a linear relationship with density ($p = c_s^2\rho$), which effectively eliminates the requirement for complex pressure solvers (Ataei & Salehipour, 2024).

Furthermore, this constant serves as a critical parameter for controlling stability via the Mach number ($Ma = u/c_s$), which is maintained at low values ($Ma \ll 1$) to minimize compressibility errors (Ataei & Salehipour, 2024). It also functions as a multiplier in determining kinematic viscosity, where $v = c_s^2 (\tau - 0.5)$ (Kummerländer et al., 2025). Consequently, the constant c_s^2 = 1/3 rigorously integrates kinetic-level interactions with the macroscopic physical and mechanical properties of the fluid in a systematic manner (Ataei & Salehipour, 2024).

2.2.2. Streaming Step

The methodology chapter provides a systematic overview of the research design, including the identification of subjects, the configuration of simulation instruments, and the precise procedures for data acquisition and subsequent analysis (Santana et al., 2026). Following the collision phase in the Lattice Boltzmann algorithm, the streaming step facilitates the non-local propagation of particle distribution functions through the computational domain (Kummerländer et al., 2025). This process is governed by the discrete evolution equation:

$$f_i(x + e_i\Delta t, t + \Delta t) = f_i^*(x, t) \quad (4)$$

In this numerical stage, the post-collision populations are shifted from their current lattice positions to adjacent nodes according to their respective discrete velocity vectors (Ataei & Salehipour, 2024). This fundamental mechanism ensures the temporal and spatial evolution of the fluid, where the streaming operation effectively mimics the free-flight of fictitious particles between grid points over a single time step (Kummerländer et al., 2025). By decoupling the local interactions from the spatial transport, the algorithm achieves high computational efficiency and remains highly adaptable for execution on modern massively parallel hardware architectures(Bahiraei et al., 2022). Furthermore, this step is essential for the application of boundary conditions, as certain schemes are imposed specifically after the streaming operation is completed to ensure the physical consistency of the simulated system (Ataei & Salehipour, 2024).

2.3. Equilibrium Distribution Function

To ensure the recovery of the incompressible Navier-Stokes equations, the equilibrium distribution function, f_i^{eq}, is mathematically defined through a quadratic expansion of the Maxwell-Boltzmann distribution (Ataei & Salehipour, 2024). In this mesoscopic framework, f_i^{eq} characterizes the ideal state toward which particle populations relax during the collision process (Kummerländer et al., 2025). The specific formulation for this model is given by:

$$f_i^{eq} = w_i\rho\left[1 + \frac{3(e_i \cdot \boldsymbol{u})}{c^2} + \frac{9(e_i \cdot \boldsymbol{u})^2}{2c^4} - \frac{3\boldsymbol{u}^2}{2c^2}\right] \quad (5)$$

Here, the macroscopic state variables, specifically fluid density (ρ) and velocity (**u**), are formally derived from the moments of the discrete distribution functions (Ataei & Salehipour, 2024). This allows the local particle populations to represent physical fluid properties through simple summation operations:

$$\rho = \sum_{i=0}^{8} f_i\,, \quad \rho\boldsymbol{u} = \sum_{i=0}^{8} f_i\, e_i \quad (6)$$

This derivation ensures that the algorithm remains consistent with the fundamental laws of mass and momentum conservation, effectively bridging the gap between kinetic theory and macroscopic fluid mechanics (Ataei & Salehipour, 2024). Through these moments, state variables such as pressure can also be locally computed, eliminating the need for iterative pressure-linked solvers (Ataei & Salehipour, 2024).

2.4. Boundary Conditions and Numerical Implementation

Boundary conditions are a crucial aspect in ensuring the overall accuracy and physical validity of the simulation (Ataei & Salehipour, 2024). On the cylinder surface, a no-slip bounce-back scheme is applied (Q. W. Wang et al., 2010), where the particle distribution functions that strike the wall are returned toward their nodes of origin (Kummerländer et al., 2026). For the inlet condition, a uniform velocity profile is implemented by enforcing the equilibrium distribution function on the left boundary of the computational domain (Ataei & Salehipour, 2024). At the outlet, an open boundary condition is utilized by applying a zero-gradient to the distribution functions, which effectively prevents numerical wave reflections from returning into the domain and destabilizing the solution (Kummerländer et al., 2026).

The entire algorithm is implemented using the Python programming language, specifically leveraging the NumPy library for advanced vectorization techniques (Najafi Khaboshan et al., 2024). The use of array-slicing operations to replace conventional nested loops significantly enhances computational efficiency and minimizes procedural errors during the transient iterations required to reach a steady-state condition (Ataei & Salehipour, 2024). This Pythonic approach allows the solver to maintain high performance and accessibility while ensuring that the numerical implementation is robust enough for benchmarking complex hydrodynamic phenomena (Ataei & Salehipour, 2024).

3. Result and Discussion

The simulation results (fig.2) demonstrate the steady-state velocity profile of the fluid as it flows past a single cylindrical obstacle at a Reynolds number (*Re*) of 10. The velocity contour visualizations confirm the successful implementation of the D2Q9 LBM algorithm in capturing fundamental hydrodynamic phenomena, where the velocity distribution around the cylinder exhibits stable laminar flow characteristics (Ho et al., 2026).

A prominent stagnation point is observed at the leading edge of the cylinder, characterized by a reduction in velocity magnitude until it reaches zero at the obstacle surface (S. Wang et al., 2014), which is consistent with the applied no-slip boundary condition (Ghafori, 2026). In contrast, the fluid undergoes significant acceleration along the upper and lower surfaces of the cylinder due to the blockage effect and flow area constriction, as indicated by the peak velocity magnitudes in the contour map (Dadda et al., 2025).

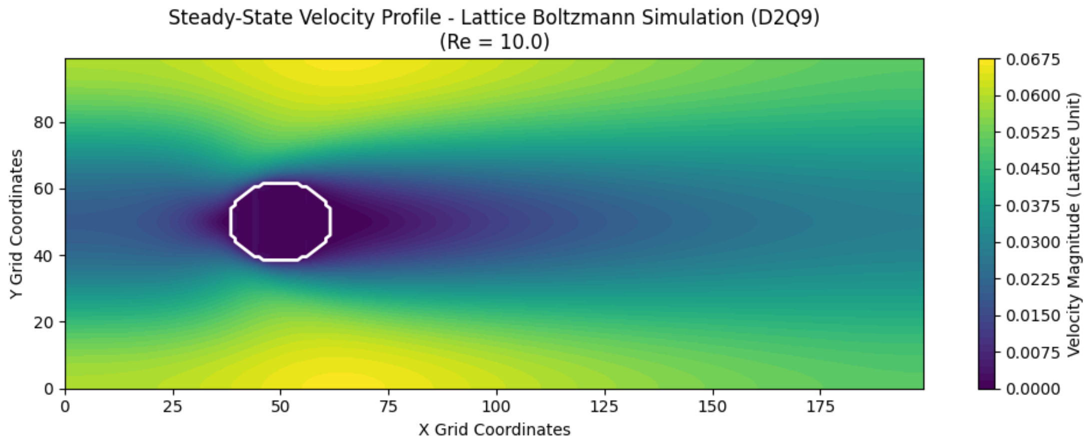

Fig.2. Steady-state velocity magnitude contour of laminar flow past a circular

Downstream of the cylinder, the simulation clearly resolves the formation of a symmetrical wake region. At , this zone is defined by a pair of stationary vortices that extend horizontally without the onset of vortex shedding, aligning with the physical expectations for steady laminar flow regimes (Ataei & Salehipour, 2024). The resulting velocity deficit behind the cylinder gradually decays as the distance from the obstacle increases, allowing the profile to trend back toward a developed flow state (Benemerito et al., 2025). These results prove that the mesoscopic LBM approach, utilizing an appropriately calibrated relaxation time (τ), accurately represents the equilibrium between inertial and viscous forces while generating convergent numerical data suitable for systematic parametric investigations (Kummerländer et al., 2025).

Further parametric analysis across Reynolds numbers (*Re*) ranging from 20 to 70 reveals a significant structural transformation within the flow's wake region. As the Re value increases, the growing dominance of inertial forces over viscous effects causes a longitudinal extension of the recirculation zone (Dadda et al., 2025). This phenomenon indicates that the flow separation point on the cylinder surface shifts forward, which is followed by the formation of a sharper and more stable velocity deficit tail along the horizontal axis. Such detailed capturing of flow dynamics is essential for standard benchmarks involving open flow over 2D obstacles (Ataei & Salehipour, 2024).

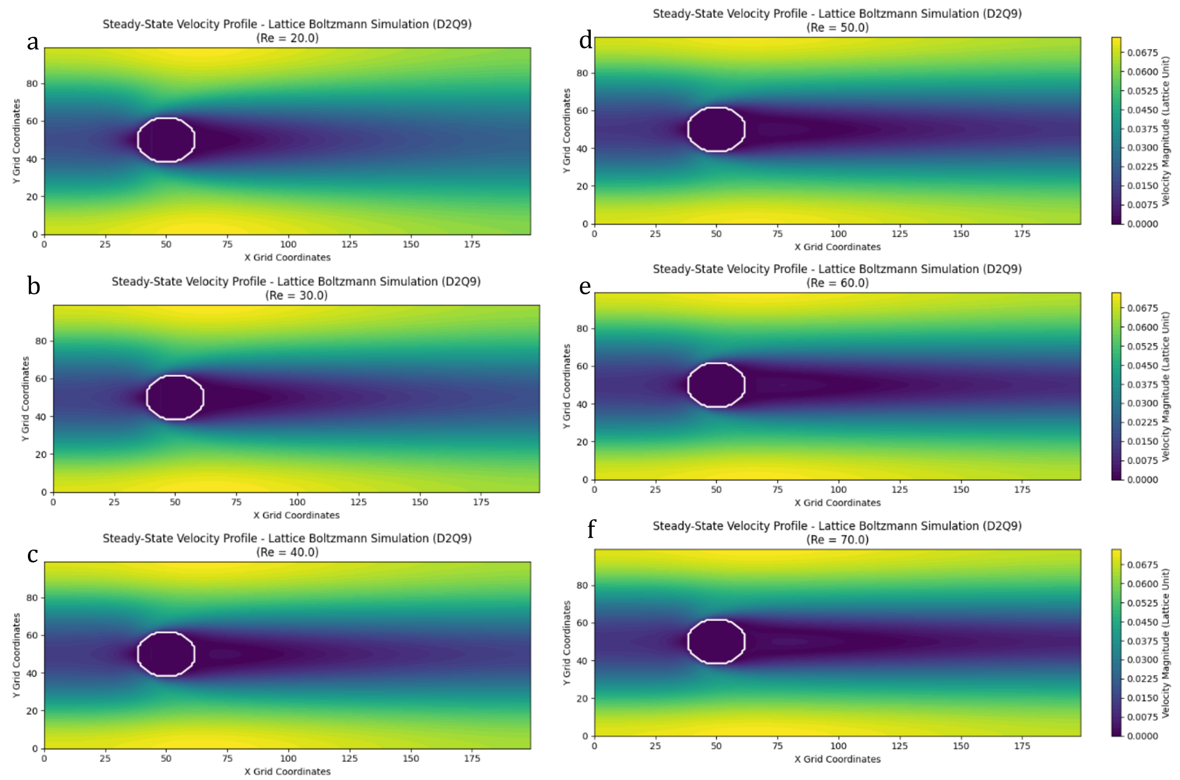

Fig. 3. Comparison of steady-state velocity magnitude profiles for laminar flow past a circular cylinder at various Reynolds numbers: (a) Re=20, (b) Re=30, (c) Re=40, (d) Re=50, (e) Re=60, and (f) Re=70. The simulations, performed using the D2Q9 Lattice Boltzmann Method, illustrate the elongation of the wake region as the Reynolds number increases.

The stability of the velocity profiles produced within this range confirms that the Lattice Boltzmann Method (LBM) utilizing the D2Q9 model effectively maintains momentum conservation without encountering significant numerical instability. Despite the elongation of the vortex zones, the flow remains symmetrical, suggesting that the critical threshold for the transition into an unsteady regime has not been reached under these simulation conditions. These results provide a robust foundation for using simulation data in structural design optimization or as a high-fidelity training dataset for Artificial Intelligence (AI) and machine learning models (Ataei & Salehipour, 2024). Integrating validated CFD data with machine learning allows for the development of accurate surrogate models capable of predicting complex hydrodynamic behaviors in broader engineering systems (Banh et al., 2025).

4. Conclusion

This research successfully demonstrates the effectiveness of the Lattice Boltzmann Method (LBM) utilizing the D2Q9 lattice model and the Bhatnagar-Gross-Krook (BGK) collision operator for simulating steady laminar flow characteristics over a single cylindrical obstacle. Through a Python-based implementation leveraging advanced vectorization techniques, the simulation accurately captures fundamental hydrodynamic phenomena, including the formation of stagnation points, flow acceleration due to blockage effects, and boundary layer development. The consistent velocity profiles observed across the *Re* range of 20 to 70 confirm that this mesoscopic approach effectively maintains numerical stability and momentum conservation without the requirement for computationally intensive pressure-linked solvers. These results underscore how transparently managed open-source computational instruments can produce rigorous research data that meets established scientific validity standards.

Analytically, observations across varying Reynolds numbers reveal a correlation between increased inertial forces and the longitudinal elongation of the recirculation zone downstream of the obstacle. The structural transformation of the wake region, while maintaining flow symmetry, indicates that the stability limits of steady laminar flow can be precisely mapped using validated relaxation time (τ) parameters. These findings have strategic implications for developing more complex computational models, as the generated high-fidelity datasets can serve as a robust foundation for training Artificial Intelligence (AI) models such as Physics-Informed Neural Networks (PINNs). Future research is encouraged to explore the influence of asymmetric cylinder positioning within channels to optimize heat transfer rates within advanced multiscale modeling frameworks.

Author Contributions

Aulia Rahman: Conceptualization, Methodology, Software. **Putra Dwiantoko**: Data curation, Writing-Original draft preparation. **Novan Habiburrahman**: Investigation, **Agung Hariadi:** Formal analysis. **Ahmad Robittah:** Visualization. **Reinaldo Evan Audrey:** Review & Editing.

Declaration of Conflicting Interests

The authors declare that there is no conflict of interest regarding the publication of this article. No financial or personal relationships with other people or organizations have inappropriately influenced the research, data collection, or the development of the computational model presented in this study.

References

Ataei, M., & Salehipour, H. (2024). XLB: A differentiable massively parallel lattice Boltzmann library in Python. *Computer Physics Communications*, *300*. https://doi.org/10.1016/j.cpc.2024.109187

Bahiraei, M., Naseri, M., & Monavari, A. (2022). Thermal-hydraulic performance of a nanofluid in a shell-and-tube heat exchanger equipped with new trapezoidal inclined baffles: Nanoparticle shape effect. *Powder Technology*, *395*, 348–359. https://doi.org/10.1016/j.powtec.2021.09.009

Banh, Q. N., Truong, P. D., & Ho, M. T. (2025). Multi-objective optimization of water-assisted injection molding using machine learning and CFD simulations. *Results in Engineering*, *28*. https://doi.org/10.1016/j.rineng.2025.107981

Benemerito, I., McCullough, J., Narracott, A., Coveney, P. V., & Marzo, A. (2025). Comparison of Navier-Stokes and lattice Boltzmann solvers for subject-specific modelling of intracranial aneurysms. *Computers in Biology and Medicine*, *197*. https://doi.org/10.1016/j.compbiomed.2025.111050

da Silva, M. V. S. (2026). Performance evaluation of commercial pumps in turbine mode: A hybrid experimental and numerical approach in python. *Flow Measurement and Instrumentation*, *107*. https://doi.org/10.1016/j.flowmeasinst.2025.103098

Dadda, A., Boujoudar, M., Houran, N., Messaoudi, H., Asbik, M., Haddou, A., & Arshad, A. (2025). CFD-ANN-based model for parametric analysis of segmented plate-fin heat sinks: Exploring heat transfer and pressure drop trade-offs. *International Journal of Thermal Sciences*, *218*. https://doi.org/10.1016/j.ijthermalsci.2025.110109

Fox, B., Wu, K. C. H., Ma, S., & Wan, S. Y. M. (2024). A CFD simulation platform for surface finishing processes in advanced manufacturing. *Advances in Engineering Software*, *196*. https://doi.org/10.1016/j.advengsoft.2024.103716

Ghafori, H. (2026). Hybrid ANN-CFD modeling of powder sedimentation in pneumatic conveying systems. *Powder Technology*, *471*. https://doi.org/10.1016/j.powtec.2025.122054

Ha, T. J., Kang, D. K., Min, J. H., Park, J. W., & Shin, J.-H. (2026). Optimization of heater configurations in single-phase immersion cooling systems using CFD-validated artificial neural networks. *Case Studies in Thermal Engineering*, *82*, 108122. https://doi.org/10.1016/j.csite.2026.108122

Ho, M. T., Dang, M. S., Truong, P. D., Vo, T. N. A., Tran, A. S., & Banh, Q. N. (2026). CFD-based and machine learning-assisted optimization of screw geometry in food extrusion processes. *Results in Engineering*, *30*. https://doi.org/10.1016/j.rineng.2026.109975

Kummerländer, A., Bukreev, F., Teutscher, D., Dorn, M., & Krause, M. J. (2025). Optimization of single node load balancing for lattice Boltzmann method on heterogeneous high performance computers. *Journal of Parallel and Distributed Computing*, *206*. https://doi.org/10.1016/j.jpdc.2025.105169

Kummerländer, A., Tur, B., Haase, M., Bukreev, F., Döllinger, M., Krause, M. J., & Kniesburges, S. (2026). Efficient fluid structure interaction simulation of vocal fold oscillations using a homogenized Lattice Boltzmann Method. *Computer Methods in Applied Mechanics and Engineering*, *457*, 119009. https://doi.org/10.1016/j.cma.2026.119009

Mansouri, K., Zobiri, O., Atia, A., & Arıcı, M. (2024). A python implementation based lattice Boltzmann method for thermal behavior analysis in silicon carbide MOSFET. *Micro and Nanostructures*, *187*. https://doi.org/10.1016/j.micrna.2024.207769

Najafi Khaboshan, H., Jaliliantabar, F., Abdullah, A. A., Panchal, S., & Azarinia, A. (2024). Parametric investigation of battery thermal management system with phase change material, metal foam, and fins; utilizing CFD and ANN models. *Applied Thermal Engineering*, *247*. https://doi.org/10.1016/j.applthermaleng.2024.123080

Oubnaki, H., Mounir, I., Mounir, B., Farchi, A., & Sak, A. (2026). Integration of CFD and machine learning for vehicle cabin thermal management using innovative materials. *Results in Engineering*, *30*. https://doi.org/10.1016/j.rineng.2026.110214

Pasa, D., Ranjan, K., Rout, S. K., Biswal, G., & Meshram, G. S. (2026). CFD-driven machine learning models for predicting flow and thermal characteristics in brazed plate heat exchanger. *International Communications in Heat and Mass Transfer*, *175*(P3). https://doi.org/10.1016/j.icheatmasstransfer.2026.111252

Santana, H. M., Vaz, J. R. P., & Lins, E. F. (2026). Python-based GUI for automated geometry generation and CFD simulation setup of Wageningen B-series propellers. *Ocean Engineering*, *356*(P1). https://doi.org/10.1016/j.oceaneng.2026.125217

Wang, Q. W., Chen, G. D., Xu, J., & Ji, Y. P. (2010). Second-law thermodynamic comparison and maximal velocity ratio design of shell-and-tube heat exchangers with continuous helical baffles. *Journal of Heat Transfer*, *132*(10), 101801. https://doi.org/10.1115/1.4001755

Wang, S., Wen, J., Yang, H., Xue, Y., & Tuo, H. (2014). Experimental investigation on heat transfer enhancement of a heat exchanger with helical baffles through blockage of triangle leakage zones. *Applied Thermal Engineering*, *67*(1–2), 122–130. https://doi.org/10.1016/j.applthermaleng.2014.03.017

Wen, J., Gu, X., Wang, M., Wang, S., & Tu, J. (2017). Numerical investigation on the multi-objective optimization of a shell-and-tube heat exchanger with helical baffles. *International Communications in Heat and Mass Transfer*, *89*, 91–97. https://doi.org/10.1016/j.icheatmasstransfer.2017.09.014

Zhai, Y., Yang, X., & Chen, S. (2026). A CFD-informed machine learning framework for generalized ship resistance prediction under wind and wave conditions. *Ocean Engineering*, *353*. https://doi.org/10.1016/j.oceaneng.2026.124759