Proceedings of the International Conference on Industrial Engineering and Operations Management

Publisher: IEOM Society International, USA DOI: 10.46254/AN15.20250535
Published: February 18, 2025

A Cognitive Chatbot for Intelligent Engineering Analysis Decision Support: A Case of Optimizing Computational Fluid Dynamics of Cooling Server Stacks

Sudeep Nair, Rohan Nair, Jagrut Brahmbhatt and Roger Jiao
Georgia Institute of Technology
Atlanta, GA 30332, USA

Abstract

Effective cooling is essential for maintaining the performance, reliability, and energy efficiency of server systems. However, optimizing airflow parameters to achieve efficient thermal management is often constrained by the high computational cost of iterative simulations. This paper presents an intelligent decision-support framework that integrates Computational Fluid Dynamics (CFD) with numerical optimization to dynamically refine airflow properties, ensuring efficient cooling while minimizing energy consumption. The framework employs the L-BFGS-B algorithm to iteratively adjust inlet velocity within predefined constraints, efficiently converging to an optimal cooling configuration without the need for exhaustive CFD simulations. By leveraging optimization-driven adjustments, the system reduces computational overhead while maintaining accuracy, making it a practical approach for real-world applications. The framework operates autonomously, reducing the need for manual tuning and accelerating decision-making in server cooling design. Beyond improving cooling performance, this approach demonstrates the broader potential of intelligent optimization in engineering workflows. The integration of CFD with data-driven decision-making enhances adaptability, streamlines design cycles, and supports real-time adjustments to operational conditions. The results highlight how automated optimization can significantly improve thermal management strategies in data centers and other thermally constrained environments, offering a scalable, computationally efficient solution for intelligent airflow control.

Keywords
Cognitive Intelligence, Chatbot, Optimization, Computational Fluid Dynamics and L-BFGS-B algorithm

1. Introduction
Chatbots are often used as a tool to reduce engineering effort. Python chatbots offer interactive software programs that leverage natural language processing (NLP) as stated by Ortiz-Garces (2024) to facilitate human-computer interactions through text or voice-based communication. These chatbots, built using Python's robust programming ecosystem, have become a key tool in a variety of industries, including customer service, healthcare, and education. However, their utility extends beyond conventional applications into the engineering domain, particularly in assisting with complex computational tasks such as Computational Fluid Dynamics (CFD).

1.1 Artificial Intelligence Chatbots
Python's versatility, paired with its numerous libraries like NLTK, spaCy, and TensorFlow for NLP, allows for the creation of intelligent chatbots capable of understanding technical queries and providing detailed, context-aware responses (Chakraborty 2023). Furthermore, Python's integration with scientific computing libraries such as NumPy, SciPy, and CFD-specific tools like Ansys or Abaqus, enables chatbots to act as an interface between engineers and computational models, streamlining workflows and enhancing efficiency.

In modern engineering design, the integration of intelligent tools such as chatbots can significantly streamline processes, reduce manual effort, and improve overall efficiency. By leveraging the capabilities of Python-based chatbots, engineers can automate various aspects of design and Python programming, making it easier to handle independent design modules that require detailed analysis. These chatbots can interact with users, understand design requirements, and execute complex simulations, acting as a virtual assistant throughout the design process.
Python chatbots can be particularly useful in engineering contexts by providing instant access to Python documentation and best practices, allowing engineers to quickly reference guidelines and improve workflow efficiency. They assist with code debugging and optimization, helping identify errors and enhance performance through automated analysis. Additionally, chatbots offer suggestions for appropriate Python libraries tailored to specific engineering tasks, ensuring that users leverage the most efficient tools available. They also automate repetitive coding tasks and generate boilerplate code, reducing manual effort and increasing productivity. Furthermore, chatbots facilitate data analysis and visualization using Python's scientific libraries, enabling engineers to interpret complex datasets and make informed decisions more effectively. By integrating these capabilities, Python chatbots can significantly enhance productivity and accuracy in engineering workflows, making them valuable tools for modern engineering practices.

1.2 Engineering Analysis of Computational Fluid Dynamics

Computational Fluid Dynamics (CFD) is a computational technique widely used in engineering and physical sciences to simulate and analyze the behavior of fluid flow, heat transfer, and other related phenomena under various conditions. By dividing a complex system into smaller, simpler parts called "finite volumes," CFD allows for the detailed assessment of how each element responds to forces, pressures, temperatures, and other physical factors.

The method is rooted in numerical methods and provides approximate solutions to boundary value problems, which are often difficult to solve analytically. Each finite volume is connected at specific points called nodes, and mathematical equations are formulated to describe the behavior of each volume. These equations are then assembled into a larger system, which can be solved using computer algorithms.

CFD is particularly useful in fields such as fluid dynamics, heat transfer, aerodynamics, and chemical process modeling. Its ability to model complex geometries and fluid behaviors makes it indispensable for product design, testing, and optimization. Additionally, CFD helps engineers assess the efficiency and performance of fluid-based systems, predict potential failure modes, and refine designs before physical prototypes are built, saving time and resources (Figure 1).

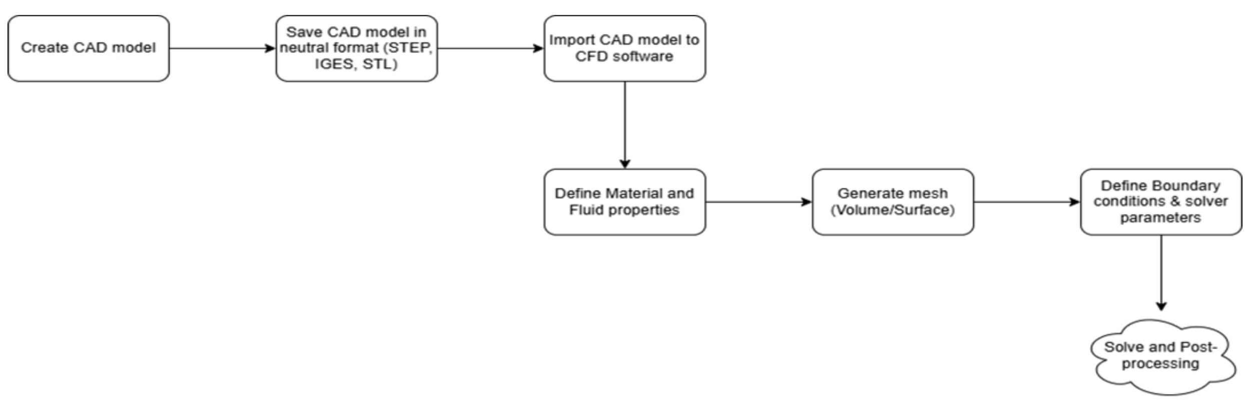


Figure 1. CFD flow for an imported CAD model

Engineering design, particularly in Computational Fluid Dynamics (CFD), has traditionally been a resource-intensive process, requiring meticulous setup, extensive simulations, and iterative refinements. Engineers often spend considerable time adjusting parameters, analyzing results, and fine-tuning models to achieve optimal performance. As the complexity of fluid interactions increases, so does the demand for efficient problem-solving approaches that minimize manual effort without compromising accuracy.

In response to this challenge, intelligent digital assistants are emerging as valuable tools in streamlining CFD workflows. By integrating AI-driven automation, these systems can assist in refining airflow configurations, adjusting boundary conditions, and optimizing design variables with minimal human intervention. More than just automating tasks, they leverage cognitive intelligence to interpret simulation trends, learn from past iterations, and suggest improvements based on real-time data. This evolution in computational assistance not only reduces the cognitive load on engineers but also enhances decision-making, enabling them to explore a wider range of design possibilities in less time. As AI continues to advance, chatbots equipped with cognitive intelligence are becoming indispensable in engineering, bridging the gap between data-driven automation and human expertise to drive more efficient and innovative design solutions.

1.3 Intelligent Decision Support: A Case of Cooling Server Stacks

Optimizing server cooling systems requires balancing thermal performance with computational efficiency. Traditional iterative simulations for airflow management are computationally expensive and time-intensive, limiting their practicality for real-time decision-making. This work aims to develop an intelligent decision-support framework that integrates CFD with numerical optimization to enhance cooling efficiency while minimizing computational overhead.

The primary objective is to refine inlet airflow velocity using the L-BFGS-B algorithm, ensuring effective temperature regulation within predefined constraints. By iteratively adjusting velocity parameters, the framework converges to an optimal cooling configuration without the need for exhaustive CFD simulations. This automation reduces manual intervention, accelerates design iterations, and enables data-driven decision-making for thermal management.

Beyond improving cooling performance, this framework demonstrates the potential of integrating CFD with intelligent optimization techniques to streamline complex engineering workflows. The approach provides a scalable and computationally efficient solution for airflow control, supporting real-time adaptability in data centers and other thermally constrained environments (Figure 2).

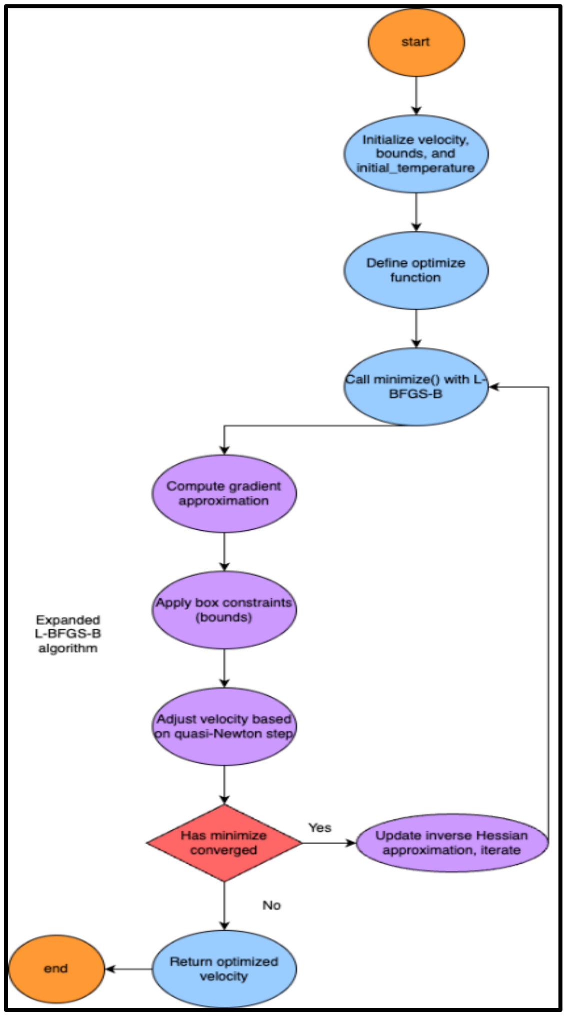


Figure 2. L-BFGS-B flow for intelligent optimization

2. Related Work

2.1 GPT-Powered Design Decision Support

The optimization of structural components through machine learning and advanced computational methods has been a significant area of research. Recent studies have explored the integration of AI-driven models into CFD workflows, focusing on improving efficiency, predictive accuracy, and applicability across diverse scenarios. The following works provide critical insights into the evolution of AI-powered systems for design and analysis (Fei et. al 2023).

This study examines the use of GPT models in smart manufacturing systems to enhance productivity by reducing costs and saving time. The proposed architecture leverages a pre-trained GPT model, enriched with publicly available data, to handle a broad spectrum of open-ended tasks. Notably, a use case demonstrates ChatGPT's ability to autonomously generate Python code for robotic operations, such as object picking, inspection, and movement. The versatility and adaptability of GPT models in automation underscore their potential for efficient and scalable manufacturing processes. These findings highlight how generative AI can streamline operational workflows, providing a basis for its application in complex design optimization frameworks.

2.2 Cognitive Intelligence

Artificial Intelligence (AI) chatbots have become increasingly prevalent in various aspects of our daily lives, from customer service to personal assistants. Their potential to enhance cognitive functions and support mental health has garnered significant attention in recent years. As these AI-driven conversational agents become more sophisticated, researchers are exploring their applications in cognitive training and therapy.

This paper by Fabio et al. (2025) investigates the potential of AI chatbots in reinforcing executive functions. This systematic review aimed to evaluate the use of conversational chatbots in executive functioning training, their impact on executive-cognitive skills, and the duration of any improvements The review included experimental, observational, and mixed-method studies that utilized AI-based chatbots to support executive functions or related conditions such as anxiety, stress, and depression.

2.3 AI applications for CFD

The evolution of numerical techniques in CFD, particularly the integration of turbulence models, multiphase flow simulations, and reactive flow modeling has been remarkable (B. Wang and J. Wang 2021). Traditional Reynolds-Averaged Navier-Stokes (RANS) models, while computationally efficient, often fail to capture complex flow dynamics in industrial settings. Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS) are discussed as higher-fidelity alternatives that offer greater accuracy but require significant computational resources. The authors also investigate the application of machine learning in turbulence modeling, demonstrating its potential to enhance predictive accuracy and computational efficiency.

While CFD has significantly enhanced process modeling and optimization, the study acknowledges persistent challenges, including computational cost, numerical stability, and the need for high-quality experimental validation. The authors suggest that future research should focus on developing reduced-order models, integrating artificial intelligence for real-time simulations, and expanding cloud-based CFD applications to make high-fidelity simulations more accessible. (B. Wang and J. Wang 2021)

2.4 ML Integration with CFD

The study by Vinuesa and Brunton (2022) in *Nature Computational Science* explores the integration of machine learning (ML) with CFD, emphasizing how data-driven techniques can enhance accuracy, efficiency, and physical insight in fluid flow simulations. The authors highlight how ML-based techniques are transforming CFD by providing new ways to model turbulence, improve numerical solvers, and accelerate simulations. Conventional CFD methods rely on turbulence models such as Reynolds-Averaged Navier-Stokes (RANS) and Large Eddy Simulation (LES), which are often limited by empirical assumptions and computational demands. By contrast, ML-driven models offer the ability to learn flow dynamics directly from data, reducing reliance on empirical approximations while improving computational efficiency.

3. System Analysis and Design

Building upon these studies, the current research aims to integrate machine learning optimization algorithm to reduce human intervention towards the CFD process by developing an iterative optimization framework. By combining the predictive power of machine learning with the physical accuracy of CFD, this study aims to minimize computational costs while ensuring optimal thermal performance. The literature not only validates the feasibility of this hybrid approach but also provides key methodological insights that guide the development of the system.

3.1 Requirement Analysis of Intelligent Decision-Making

The use case focuses on the interaction between the engineer and the system, illustrating how user inputs—such as temperature conditions—drive the optimization process. The system adapts dynamically to ensure constraints are met while providing reliable outputs. Additionally, alternative scenarios, such as inconsistencies in CFD results or deviations in optimized velocity values, are considered to ensure a robust and adaptable workflow (Table 1).

Table 1. Fully dressed use case for the optimized decision support module

SYSTEM	**OPTIMIZATION SYSTEM FOR CFD PARAMETERS**
Name	Optimize Airflow Parameters for Server Cooling
Actors	**Primary:** Engineer using the optimization system
Description	The system identifies optimal airflow velocity and geometric parameters to maintain server temperature within predefined thresholds, ensuring efficient cooling.
Trigger	The engineer inputs constraints or design specifications (e.g., allowable temperature range) and initiates the optimization process.
Preconditions	1. The system has a CFD dataset preloaded with airflow properties and geometric parameter combinations. 2. The CFD solver is integrated and ready to run simulations based on input conditions.
Main Success Scenario	1. The engineer specifies design constraints (e.g., allowable temperature threshold). 2. The system retrieves the relevant dataset and initializes the optimization process. 3. The system specifies the inlet air temperature as input and optimizes airflow velocity. 4. If the initial simulation results exceed temperature thresholds, the system iteratively adjusts velocity and geometric parameters. 5. The system performs CFD validation on the optimized parameter set. 6. If CFD results meet the constraints, the optimal values are stored and presented to the engineer. 7. The engineer reviews the results, and the system provides a summary of parameter values and corresponding performance metrics.
Alternative Scenario 1	**Dataset Inconsistency:** If the dataset contains missing or inconsistent entries, the system flags the issue and requests correction before proceeding.
Alternative Scenario 2	**Simulation Convergence Failure:** If the CFD solver fails to converge on a solution, the system re-initiates the simulation with adjusted parameters or refined meshing.
Alternative Scenario 3	**Constraint Violation During CFD Validation:** If CFD validation fails (e.g., temperature exceeds threshold), the system re-initiates optimization with adjusted airflow velocity and geometric parameters, providing feedback to the engineer.
Preconditions	1. Optimal parameter values are saved for future reference and design implementation. 2. The system generates a report summarizing the optimization process and results.

3.2. System Architecture and Workflow

While ML-enhanced CFD has demonstrated significant promise, the study also acknowledges key challenges. One major concern is the need for high-quality training data, as ML models rely on accurate flow simulations or experimental data to make reliable predictions. Additionally, generalizability remains a challenge, as ML models trained on specific cases may struggle to extrapolate to different flow conditions, limiting their applicability in dynamic environments. Another critical issue is the interpretability of ML models; many operate as black-box approaches, which can lack physical intuition and reliability in critical engineering applications where explainability is essential for validation and decision-making.

The optimization system for Computational Fluid Dynamics (CFD) parameters is designed to enhance the efficiency and accuracy of airflow optimization in server cooling applications. This system utilizes CFD simulations to iteratively refine airflow velocity until the server temperature meets predefined thresholds.

The system aims to streamline the traditionally time-consuming process of CFD-based parameter optimization. This approach directly specifies the inlet air temperature as input and optimizes airflow velocity to achieve the desired thermal performance. The system consists of three main scripts: processing.py, optimize.py, and chatbot.py, each handling a specific aspect of simulation, optimization, and user interaction (Figure 3).

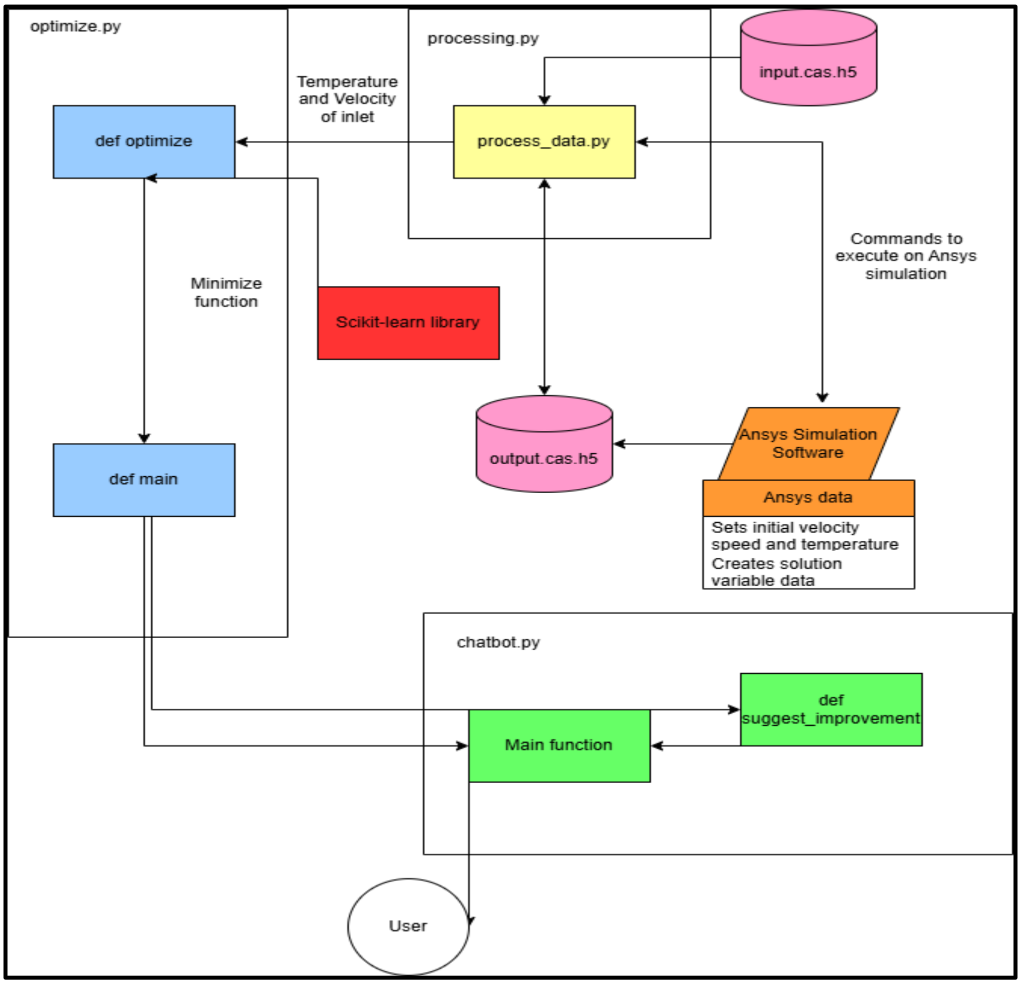


Figure 3. Software flow outlining all the functions within the chatbot

(1) processing.py – Running the CFD Simulation & Extracting Data

This script is responsible for interfacing with Ansys Fluent to execute CFD simulations and extract relevant temperature data. The key function, *process_data()*, begins by establishing a connection with Ansys Fluent and loading the pre-processed case file, input.cas.h5, which contains the mesh and boundary conditions. It then sets the initial inlet velocity and inlet temperature for the simulation before declaring the necessary Fluent solution variables that will be extracted later.

Once the parameters are set, the script runs the simulation using the *iter()* function, generating a new case file, *output.cas.h5*, that contains the updated solution data. The script then reads this file to extract temperature arrays for Server 1, Server 2, and Server 3. The final step involves calculating the maximum temperature across all three servers, which serves as the key output for the optimization step. This script ensures that accurate thermal data is obtained for given airflow conditions, forming the foundation for the optimization process.

(2) optimize.py – Velocity Optimization
This script automates the adjustment of inlet velocity using numerical optimization to ensure efficient cooling performance. The optimize () function calls *process_data()* to retrieve the maximum server temperature corresponding to a given inlet velocity. It then returns this temperature value as the objective function output for the optimization process.

The *main()* function leverages Scikit-learn's minimize function to iteratively adjust the inlet velocity. Within this optimization loop, *optimize()* is repeatedly called to evaluate different velocity values, guiding the system toward an optimal solution. The process ensures that the system converges to an inlet velocity that maintains server temperatures within safe operational limits. By automating the tuning of airflow velocity, this script significantly reduces manual intervention while enhancing cooling efficiency and computational performance.

(3) chatbot.py – Intelligent Decision Support
This script provides user interaction and decision support, helping engineers refine cooling strategies by analyzing optimization results and suggesting necessary adjustments. The *suggest_improvement()* function evaluates system performance by checking whether the final temperature output falls within acceptable limits. If the temperature exceeds the threshold, it provides corrective recommendations, such as adjusting the inlet velocity or modifying airflow conditions. When the system is already optimized, it confirms stability and suggests potential refinements to improve efficiency further. By acting as an intelligent assistant, this chatbot component streamlines human decision-making, enabling users to interactively enhance system performance without requiring extensive CFD expertise.

4. CFD Model Setup and Data Acquisition

The design problem revolves around mitigating excessive heat buildup in server stacks, particularly in environments that exacerbate heat retention within ruggedized server boxes. This issue leads to emergency shutdowns when internal temperatures exceed safe operating limits.

Several technical constraints must be considered. The system must accommodate CISCO brand servers that have an operating temperature range of 0 – 100 °F. The ruggedized server boxes are designed to open while in use but have no available space for additional components. The operational environment is assumed to exceed 100 °F with 0% humidity, presenting extreme cooling challenges. Additionally, modifications to the server box and its wiring layout are not allowed, restricting potential design solutions.

The primary design objective is to develop a hybrid cooling system that incorporates both passive and active cooling strategies. Passive cooling focuses on optimizing airflow through inlets and outlets, while active cooling involves conditioning the air around the servers to maintain temperatures below 100 °F. Computational Fluid Dynamics (CFD) will be utilized to analyze airflow dynamics over the server surfaces, while thermodynamic principles will guide the heat exchange process, ensuring hot air is efficiently expelled and replaced with cooler air.

To create an accurate CFD model, design specifications of the **Cisco UCS C220 M6** server were utilized, including external dimensions, material properties, internal heat generation rate, and power consumption. These specifications will be integrated into a CAD model for CFD simulations, ensuring a close representation of real-world conditions.

To ensure an accurate representation of the cooling system, ducting parameters were obtained from existing CAD models for connectors, providing a realistic basis for airflow optimization. A solid model was designed with these parameters in mind, ensuring compatibility with the physical constraints of the system (Figure 4).

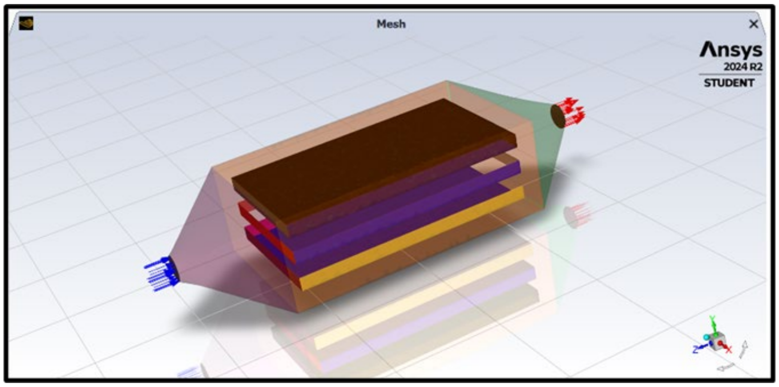


Figure 4. Meshed geometry with a transparent fluid zone for airflow representation

Before integrating the chatbot-driven optimization framework, CFD simulations were conducted to evaluate the feasibility of the cooling strategy. These preliminary simulations helped validate airflow behavior, assess temperature distributions, and refine initial design assumptions, ensuring that the optimization procedure would be both effective and computationally viable (Figure 5).

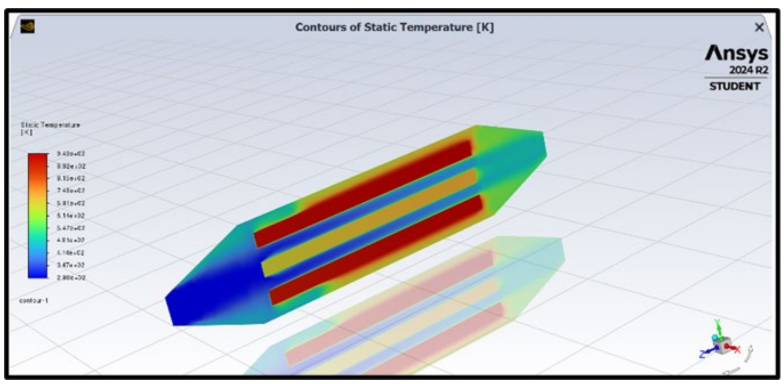


Figure 5. CFD contour for heat exchange in the YZ plane

By leveraging these specifications and constraints, the project aims to develop an intelligent, data-driven cooling solution that ensures long-term operational stability for military server deployments in extreme climates.

5. System Implementation and Validation

The effectiveness of the proposed optimization framework is evaluated through numerical simulations and chatbot-guided decision-making. This section presents the mesh quality metrics, CFD simulation results, and the chatbot's performance in optimizing airflow conditions. The numerical results highlight the convergence behavior of the optimization process, while the chatbot's recommendations demonstrate its capability to iteratively refine cooling parameters. Additionally, the discussion explores the impact of inlet velocity and air temperature adjustments on maintaining server temperatures within safe operating limits. The findings validate the efficiency of integrating CFD, numerical optimization, and AI-driven decision support, providing a scalable and computationally efficient approach for real-time thermal management.

5.1 Mesh Quality and Computational Setup

The computational domain was discretized using a high-quality mesh to ensure accurate and stable CFD simulations. The key mesh quality metrics are as follows:

- Orthogonal quality = 0.916 (indicating well-structured cells with minimal numerical diffusion).
- Average skewness = 0.079 (ensuring near-equilateral cells for stable computations).
- Average aspect ratio = 5.683 (maintaining a balanced cell distribution for capturing flow details).
- Total cell count = 99,758 cells (ensuring a refined resolution for temperature and velocity gradients).

5.2 Chatbot-Guided Optimization

(1) User Input and Optimization Initialization (Fig 6): The process begins with the chatbot prompting the user to input an initial temperature. This serves as the starting condition for the optimization routine. Once the user provides a temperature value, the chatbot initiates the minimization loop using the L-BFGS-B algorithm**,** a gradient-based optimization technique suited for problems with bound constraints. This algorithm iteratively searches for the optimal inlet velocity that effectively minimizes the server's internal temperature. While this process is running, the chatbot displays a status message such as "Running optimization"**,** signaling that the computations are actively in progress (Figure 6).

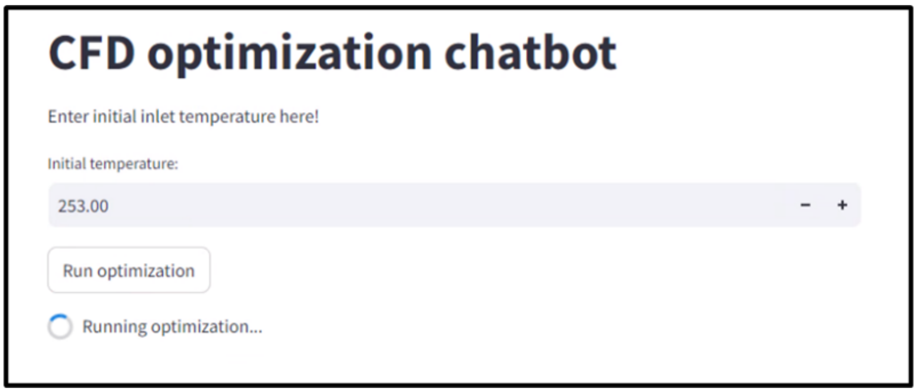
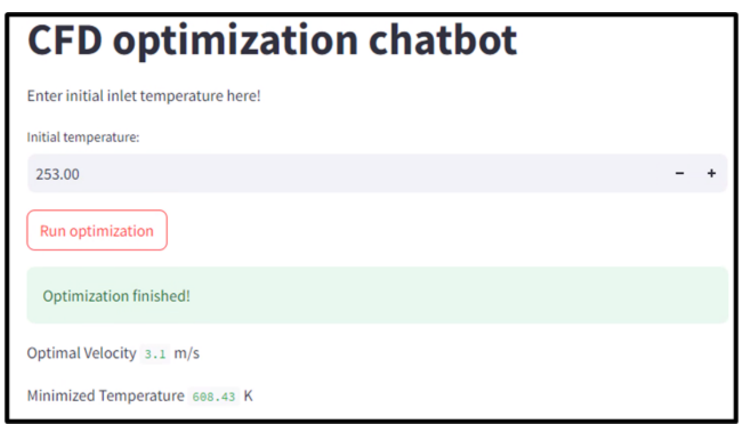


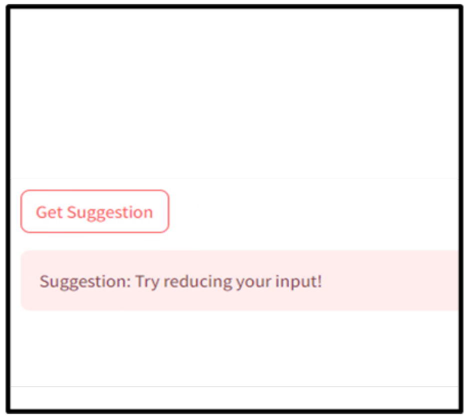
Figure 6. Cognitive chatbot running the L-BFGS-B optimization loop

(2) Optimization Results Output (Figure 7a): After completing the minimization process, the chatbot presents the results, which include the optimal velocity that minimizes the server temperature as well as the corresponding minimized temperature**.** These values provide the user with precise numerical insights into the most efficient cooling parameters, ensuring that the server operates at an optimal thermal level.

(3) Performance Recommendations (Figure 7b): Beyond numerical optimization, the chatbot also serves as an advisory tool by offering actionable insights for further server efficiency improvements. When prompted, it analyzes the scenario and generates strategic recommendations. In this case, the chatbot determines that lowering the input temperature is a key factor in enhancing server efficiency. By reducing the inlet air temperature, the system can achieve more effective cooling**,** leading to better performance and longevity.



a



b

Figure 7a. Optimized outputs with optimal velocity suggestion and corresponding minimum temperature
7b: Chatbot suggestion for reducing cold air temperature input to get an optimized output in the operating range of CISCO servers

This structured approach ensures that the chatbot not only finds the optimal velocity for cooling but also provides data-driven insights to help users make informed decisions about improving server operations.

5.3 Intelligent Computational Bottleneck Solutions

While the current framework effectively optimizes airflow parameters using numerical techniques within CFD simulations, the reliance on Fluent simulations remains a computational bottleneck. To further enhance efficiency, we propose integrating a regression-based predictive model to replace the need for direct Fluent simulations, significantly reducing computational overhead and enabling real-time optimization.

By training a regression model on an extensive dataset of precomputed CFD simulations, the system can predict server temperature based on airflow parameters without requiring new simulations for each iteration. This eliminates the need for costly Fluent computations, allowing for near-instantaneous decision-making in optimizing cooling configurations. However, the success of this approach depends on selecting and implementing an appropriate regression model that accurately captures complex airflow-thermal interactions.

To ensure reliability, the regression model must be rigorously validated against Fluent simulation results. This involves:

i) Model Selection and Training: Exploring multiple regression techniques, such as Gaussian Process Regression (GPR), Random Forest Regression (RFR), or Neural Networks, to determine the most accurate and computationally efficient approach.
ii) Error Analysis and Refinement: Identifying prediction deviations and refining the model architecture or training dataset to improve performance.
iii) Adaptive Correction Mechanism: Implementing a threshold-based validation system where the regression model continuously improves based on new CFD simulation data if significant discrepancies arise.

By replacing direct Fluent simulations with a robust regression model, this improvement not only accelerates optimization but also ensures adaptability to different cooling configurations. Implementing a validated, high-accuracy regression model will transform the framework into a scalable, real-time decision-support system for server thermal management.

5.4 Testing Result and Analysis

To validate the optimized airflow parameters, the results obtained through the chatbot-driven optimization were tested using CFD simulations. The optimized inlet velocity and air temperature values were reintroduced into the simulation environment to assess whether the predicted cooling performance was achieved. The validation process confirmed that the optimized parameters successfully maintained server temperatures within the acceptable range, aligning closely with the chatbot's recommendations. The consistency between the optimization results and the CFD validation highlights the accuracy and reliability of the framework, demonstrating that the chatbot-driven approach can effectively replace iterative simulation-based tuning while maintaining precision in thermal management (Figure 8).

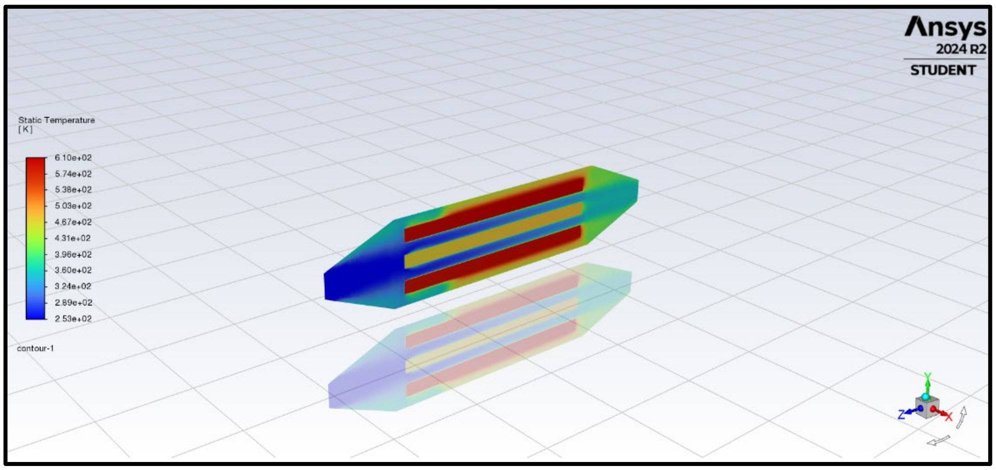


Figure 8. CFD validation for optimized chatbot parameters

To validate the optimized airflow parameters, the final results obtained through the chatbot-driven optimization were tested against CFD simulations. The chatbot predicted an optimized server temperature of 608.43K, while the corresponding CFD simulation result was 610K. The error between the two results was approximately 0.26%, demonstrating a strong agreement between the predicted and simulated values. This minimal deviation highlights the accuracy and reliability of the chatbot-driven approach in optimizing airflow conditions. The close correlation between the two methods confirms that the optimization framework can effectively reduce reliance on iterative CFD simulations while maintaining precision in thermal management.

6. Concluding Remarks

This study introduced a cognitive chatbot framework that enhances decision-making in Computational Fluid Dynamics (CFD) optimization for cooling server stacks. The primary objective was to automate and streamline the airflow optimization process by refining inlet velocity, temperature, and angle to ensure optimal cooling performance while maintaining computational efficiency. By integrating PyFluent into the system, the chatbot effectively replaces time-consuming manual CFD iterations with an intelligent, automated optimization workflow.

The study successfully demonstrated that numerical optimization techniques, specifically L-BFGS-B, can iteratively adjust airflow parameters within predefined constraints to maintain server temperatures below critical thresholds. The chatbot interface simplifies interaction with the optimization system, allowing users to define constraints and receive optimized airflow conditions in real time. This approach minimizes the need for manual interventions, significantly reduces design cycles, and enhances energy efficiency in server cooling systems.

A key takeaway from this work is the potential to eliminate direct CFD simulations by implementing robust regression models trained on high-fidelity CFD data. Such an approach would further reduce computational costs by enabling real-time predictions of airflow performance without running full-scale simulations. However, ensuring the accuracy of regression models remains a critical challenge, requiring careful validation against actual CFD results to maintain reliability.

The implications of this research extend beyond server cooling applications. The integration of CFD with AI-driven optimization has the potential to improve thermal management across various industries, including data centers, aerospace, and industrial HVAC systems. Future work will focus on refining regression-based predictive models, expanding the chatbot's adaptability to different cooling environments, and exploring reinforcement learning techniques to enhance optimization robustness under dynamic conditions. By combining intelligent decision support with CFD-driven optimization, this study contributes to the development of smarter, more energy-efficient cooling strategies for thermally constrained environments.

References

Ortiz-Garces, I., Govea, J., Andrade, R. O., and Villegas-Ch, W., Optimizing chatbot effectiveness through advanced syntactic analysis: A comprehensive study in natural language processing. *Applied Sciences*, 14(5), 1737. 2024. doi: 10.3390/app14051737.

Chakraborty, C., Pal, S., Bhattacharya, M., Dash, S., and Lee, S. S., Overview of chatbots with special emphasis on artificial intelligence-enabled ChatGPT in medical science. *Frontiers in Artificial Intelligence*, 6, 1237704, 2023. doi: 10.3389/frai.2023.1237704.

Fei, Y., Wang, S., Song, M., and Jiao, J., Design and operation of a GPT-powered smart manufacturing system. *Proceedings of the Second Australian International Conference on Industrial Engineering and Operations Management*. 2023. doi: 10.46254/AU02.20230273.

Carmelo Fabio Longo, Paolo Marco Riela, Daniele Francesco Santamaria, Corrado Santoro, Antonio Lieto, A framework for cognitive chatbots based on abductive deductive inference, *Cognitive Systems Research*, Volume 81, , Pages 64-79, 2023, ISSN 1389-0417, https://doi.org/10.1016/j.cogsys.2023.05.002.

Bo Wang and Jingtao Wang, Application of Artificial Intelligence in Computational Fluid Dynamics, *Industrial & Engineering Chemistry Research 60* (7), 2772-2790. 2021. DOI: 10.1021/acs.iecr.0c05045

Vinuesa, R., and Brunton, S. L., Enhancing computational fluid dynamics with machine learning. *Nature Computational Science*, 2, 358–366. 2022. doi: 10.1038/s43588-022-00264-7.

Biographies

Sudeep Nair is a research student in the School of Mechanical Engineering at the Georgia Institute of Technology, USA. His research interests include Finite element analysis, solid modeling, design optimization, operation planning, decision making, and data analysis.

Rohan Nair is an undergraduate student in the School of Computer Science at the Georgia Institute of Technology, USA. His research interests include large language models for industrial applications, machine learning techniques and reinforcement learning.

Jagrut Brahmbhatt is a research student in the School of Mechanical Engineering at the Georgia Institute of Technology, USA. His research interests include computer aided design, computational fluid dynamics and finite element modeling.

Roger Jiao is the editor-in-chief of Journal of Engineering Design and an associate professor of mechanical and industrial engineering at Georgia Tech, USA. Prior to joining the School of Mechanical Engineering at Georgia Tech in December 2008, he was an Assistant Professor and then Associate Professor in the School of Mechanical and Aerospace Engineering at Nanyang Technological University, Singapore. Before his career in Singapore, he was a Visiting Scholar in the Department of Industrial Engineering and Engineering Management at Hong Kong University of Science and Technology from 1998 to 1999. From 1993 to 1994, he was a Lecturer of Industrial Engineering in the School of Management at Tianjin University, China. From 1988 to 1990, he worked as an Associate Lecturer in the Department of Industrial Design at Tianjin University of Science and Technology, China. More info about his research: https://scholar.google.com/citations?user=9yikEHAAAAAJ&hl=en&oi=ao.