

Article

Not peer-reviewed version

---

# Exploring the Capabilities of LLMs from CFD Simulations of Turbulent Flow in a Manifold

---

[Hector Rafael Morano-Okuno](#)<sup>\*</sup>, Armando Rafael San Vicente-Cisneros, Guillermo Sandoval-Benitez

Posted Date: 2 April 2026

doi: 10.20944/preprints202604.0165.v1

Keywords: artificial intelligence; large language model; prompt engineering; CFD simulations; CAD model



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

# Exploring the Capabilities of LLMs from CFD Simulations of Turbulent Flow in a Manifold

Hector Rafael Morano-Okuno \*, Armando Rafael San Vicente-Cisneros  
and Guillermo Sandoval-Benitez

School of Engineering and Sciences, Tecnológico de Monterrey, Estado de México, México

\* Correspondence: hector.morano@tec.mx

## Abstract

Currently, the applications of Large Language Models (LLMs) have expanded to diverse areas, from code generation to the medical diagnosis of various pathologies. This work aims to explore what LLMs can achieve using information from CFD simulations of turbulent flow in a manifold, and to determine whether users or students can employ them as a guide for conducting this type of analysis. Through a case study, it is intended to investigate the following aspects of LLMs: 1) the type of information they handle regarding the behavior of turbulent flow within a manifold, 2) whether they identify the boundary conditions necessary to perform a CFD simulation in a manifold, 3) their capacity to provide recommendations for improving CFD simulations based on the results obtained, 4) whether they can predict the results of CFD simulations based on previous results, and 5) whether users or students can use them as a guiding tool for performing CFD simulations. Among the findings, it was discovered that the LLM used has sufficient information on turbulent flows within a manifold and can make recommendations to improve the results of CFD simulations. It was also identified that LLMs offer a user-friendly environment and that it is possible to predict CFD simulation results by varying the manifold boundary conditions.

**Keywords:** artificial intelligence; large language model; prompt engineering; CFD simulations; CAD model

---

## 1. Introduction

The use of LLMs in CFD is important because it enables the automation and streamlining of complex processes, such as boundary condition selection, identification of the appropriate turbulence model, and selection of solution parameters, although their effectiveness remains to be studied [1]. On the other hand, some LLMs can handle multiple CFD tasks, while smaller models, such as QwQ-32B, have limitations, especially in complex configurations and zero-shot approaches, highlighting stability and computational cost issues. Another advantage of using LLMs is that they reduce dependence on specialized expertise and broaden access, allowing simulation requirements to be described in natural language [2]. Further improvements in the generalizability, robustness, and deployment of these systems are still needed to consolidate their use in complex engineering workflows. Some studies develop LLM-based agents to automate tasks in CFD simulations with OpenFOAM [3]. This streamlines and automates complex configurations, reducing manual effort in tasks such as boundary condition adjustments and code implementation. These working methods achieve high performance using models like GPT-4o and o1 with chain-of-thought reasoning, along with an iterative correction cycle and a recovery augmented generation (RAG) approach. This allows for the efficient resolution of diverse flow scenarios, in terms of both cost and the number of iterations. Future challenges include the need for human oversight, continuous performance monitoring, and system adaptation for critical applications, as well as its extension to solvers beyond OpenFOAM. Research has been conducted to develop non-intrusive reduced-order models (NIROMs) for CFD simulations of turbulent flow around buildings [4]. The use of AI reduces computational costs and

enables faster, more accessible simulations. This approach predicts the flow field using a residual adversarial autoencoder with a Wasserstein generative adversarial network (WGAN) for dimensionality reduction, followed by a parallel bidirectional long short-term memory (BiLSTM) network to model time evolution. Transfer learning is also applied to improve results with limited data. Other studies analyze the use of artificial intelligence as a complement to CFD simulations to predict pressure in nanofluids under turbulent pipe flow [5]. The use of AI reduces computation times and reveals relationships between variables that CFD alone cannot identify. With the above, it has been possible to reproduce the CFD results with high precision using an adaptive network-based fuzzy inference system (ANFIS) model trained with simulated data, incorporating variables such as nanoparticle fraction, temperature, and spatial coordinates, thereby reducing computation time significantly.

AI-based models have been proposed to predict wind fields in offshore environments, using CFD simulation data for trajectory planning in Unmanned Aerial Vehicle (UAV) applications [6]. The incorporation of AI addresses the computational cost limitations of CFD and enables real-time predictions. This enables the reconstruction of highly accurate three-dimensional flow fields by combining convolutional autoencoders with a deep neural network trained on Reynolds-Averaged Navier-Stokes (RANS) data, achieving significant computational acceleration and enabling safe UAV navigation. Further improvements are needed in accuracy near walls and in the model's robustness under more complex conditions. In the area of Unmanned Underwater Vehicles (UUVs), AI and CFD simulations are being integrated to balance emotional and energy-efficiency aspects in product design [7]. The use of AI in CFD allows for the simultaneous optimization of user experience and technical performance, enabling the generation and evaluation of design alternatives through generative AI, the performance of eye-tracking experiments, and the analysis of facial expressions, which are then optimized with CFD, resulting in a significant reduction of the drag coefficient in the selected option. Other studies analyze the use of deep learning models to predict indoor climate dynamics (temperature and velocity) from CFD simulation data [8]. The use of AI enables improved prediction and control of complex systems, such as heating, ventilation, and air conditioning (HVAC), by leveraging information from CFD. Artificial Neural Networks (ANNs) match the performance of Long Short-Term Memory (LSTM) networks at lower computational cost, while also identifying semi-chaotic behaviors that enable reliable predictions, using hybrid physical-informed approaches that combine CFD and data models. In the design and simulation of bio-inspired underwater vehicles that integrate CFD and artificial intelligence to emulate whale locomotion and detect fish in marine environments, AI is used because it enables improvements in both hydrodynamic performance and perception capabilities for exploration and monitoring applications [9]. With these systems, a significant drag reduction is achieved through CFD simulations and a high-precision, advanced visual detection model (Ghost-YOLOv12), using tools such as SolidWorks, 6-DOF kinematic modeling, and neural networks trained on specialized datasets. In the design and evaluation of a new turbojet engine (TTX2T), CFD and AI are being used to optimize efficiency, weight, and performance in aeronautical applications [10]. The integration of AI optimizes engine performance and improves adaptation to changing operating conditions. Improvements in thrust-to-weight ratio, reduced specific fuel consumption, and increased thermal stability were achieved through CFD simulations, experimental analyses, and machine-learning-based predictive models. In the optimization of impellers in turbomachinery, CFD simulations, AI-based surrogate models, and evolutionary algorithms are also being integrated [11]. The use of AI enables efficient optimization of multiple design objectives, reducing computational costs and improving performance. Significant improvements in hydraulic efficiency, mass reduction, and cavitation mitigation are achieved through the combination of methods, including the Resistive Sorting Method (RSM), Box-Behnken Design (BBD), and Non-Dominated Sorting Genetic Algorithm II (NSGA-II). In the future, the plan is to incorporate open-source tools, additive manufacturing, and deep neural networks to expand drivers' capabilities for more advanced industrial applications.

## 2. LLMs and Prompt Engineering

Prompt Engineering enables communication between LLMs through structured templates, semantic example selection, and guided reasoning [12]. Some studies have developed educational labs to teach generative AI, including Prompt Engineering, to information systems students [13]. Prompt Engineering empowers students to interact effectively with generative models and build functional applications without requiring advanced programming knowledge. This type of application has been found to improve students' confidence and understanding through tools such as the OpenAI API, text and image generation exercises, transcription, and web application development. These labs require optimizing pedagogical strategies and evaluating their impact in different educational contexts. The use of Prompt Engineering in generating clinical summaries by LLMs requires comparing structured prompts versus vague prompts [14]. Prompt Engineering enables the creation of structured prompts that significantly improve the quality, clinical appropriateness, and consistency of generated summaries, though it introduces additional risks, such as hallucinations and biases. Structured prompts based on role, task, tone, and format have been shown to outperform vague prompts, as evaluated by human experts across multiple models and clinical scenarios. Prompt Engineering significantly influences LLM performance, and some have been used to detect software vulnerabilities [15]. Advanced models such as GPT-4o and Claude-3.5 Sonnet have been found to significantly outperform GPT-3.5 Turbo, especially when using Step-by-Step prompts and utilizing data from the Software Assurance Reference Dataset (SARD) in multiple programming languages. Prompt Engineering focuses on designing and optimizing prompts to guide LLM behavior, thereby improving performance by enabling more effective user interaction [16]. Further research is needed to determine best practices for prompt formulation and to compile techniques and recommendations for building more effective prompts that maximize model performance across different tasks.

Another application of Prompt Engineering is in intelligent tutoring systems, where it generates recommendations or "tips" for solving mathematical problems [17]. Prompt Engineering automates the generation of educational support using techniques such as Chain of Thought, reducing the manual effort required of teachers. Rules for its development have been generated using models such as ChatGPT and compared to those designed by humans in a Turing test. The results show that, although humans achieve higher scores, AI-generated scores are, in many cases, indistinguishable and useful as support. The development of performance metrics for further refinement is still required. Other studies review the use of generative AI to write educational materials in plain language in the healthcare field, using Prompt Engineering as a technique [18]. Prompt Engineering is crucial because strategies such as specifying reading level, audience, or context improve the clarity and readability of the generated texts. The aforementioned technique has shown that AI models can improve the readability of patient materials across different platforms and prompt configurations. Prompt Engineering improves the accuracy, fairness, and usefulness of LLMs by systematically adapting and evaluating prompts in sensitive contexts, such as healthcare [19], where the aim is to synthesize evidence using a structured methodology (JBI), analyzing the literature in multiple databases to identify strategies, trends, and best practices in prompt design and evaluation. More robust strategies are still needed to optimize the use of LLMs in healthcare education and decision-making. To create effective prompts, studies have proposed a practical guide based on a literature review and real-world examples, including best practices such as using high-quality data, model comparison, validation, and factor analysis for calibration [20]. Among the challenges of Prompt Engineering are risks such as hallucinations, biases, and ethical problems, as well as the need for human validation, transparency, and accountability. It is emphasized that AI does not replace critical thinking or methodological rigor [21].

One classification of prompts is based on the number of examples provided to the LLM [22]: if one example is used, it is called One-shot; if several examples are used, it is called Few-shot; and if no examples are used, it is called Zero-shot. A second classification is by the type of structure [23]: when it uses a single instruction, it is known as Directive; if it includes additional context, it is called

Contextual; if it uses a dialogue, it is Conversational; and if it requires the user to adopt a role, it is Role-Based. A third classification is by the technique used [24]: Chain-of-Thought (CoT) for step-by-step reasoning, Self-consistency for multiple responses and consensus, Tree-of-Thoughts for a tree-like exploration, and Retrieval-Augmented (RAG) for external information. Strategies for creating an appropriate prompt consist of precisely defining what is required [25], avoiding ambiguities (Clarity and Specificity), dividing the desired sequence into steps (Structure), providing key data (Relevant Context), specifying the perspective or role (Role or Perspective), providing examples (Few-shot), defining length, format, tone, and specific variables (Constraints), and adjusting the prompt based on previous results (Iteration). The effectiveness of a prompt depends on its objective [26], but it can be evaluated by determining how accurate its response is (Accuracy), how well it addresses the request (Relevance), whether it produces similar results for the same prompt (Consistency), whether it performs well with small variations (Robustness), whether it requires fewer iterations (Efficiency), and, if possible, employing predictions such as RMSE,  $R^2$ , BLEU, or ROUGE (Quantitative Metrics).

It can be observed that LLMs are being integrated into many areas (education, health, research, programming, etc.) as tools to automate cognitive tasks, support decision-making, and improve efficiency. However, prompts are not entirely reliable on their own (hallucinations, biases, lack of rigor); they function better as assistants than as replacements, and their performance depends largely on their application [27]. Prompt Engineering determines the quality of the output and, depending on its components, improves accuracy, interpretability, contextual relevance, and alignment with specific objectives. Prompt creation is becoming a cross-cutting skill (education, medicine, research, programming, etc.). The true "control" of LLMs lies not in the model, but in the prompt [28]. LLMs do not fail so much because of the model itself, but because of how they are asked questions. The more critical the domain (medicine, statistics, education, research, or CFD), the more important the structure of the prompts becomes, and expert knowledge must be integrated [29].

### 3. Research Context and Computational Resources

The use of LLMs has expanded across various disciplines, and with the help of Prompt Engineering, users can interact with them in a simple, user-friendly way [30]. Every day, LLMs perform increasingly complex tasks, from generating code to recognizing and creating images. The research questions for this work are: To what extent can LLMs help users perform CFD simulations? What information do they handle about the behavior of turbulent flows in different systems? Can they guide the user in defining boundary conditions? With the results of CFD simulations, can LLMs predict any behavior for which further CFD simulations are no longer necessary? To understand the capabilities of LLMs, the following objectives were established: 1) to identify the type of information they possess regarding turbulent flow behavior within a manifold, 2) to determine if they can define the boundary conditions necessary for performing a CFD simulation, 3) to evaluate if they can provide recommendations for improving CFD simulations based on the results obtained, 4) to investigate if they are able to predict CFD simulation results based on previous results, and 5) to determine if novice users or students unfamiliar with this type of CFD simulation can use LLMs as a guide to perform this type of analysis. To answer the above questions, a case study was chosen based on the information available to the LLM. This work does not aim to design the case study through the LLM; the primary objective is to investigate the information the LLM can provide when sharing CFD simulation data.

The software application used to perform the CFD simulations was Fluent in ANSYS Workbench 2025 R2, as this is the tool available at the university and is recognized in this field of simulation.

The LLM used was ChatGPT running on GPT-5.2, which allows for improvements to future versions through feedback in the interface using the like and dislike buttons, as well as reporting incorrect or harmful answers [31]. The free version was used, as it is accessible to any user with access to a computer and an internet connection.

The computers used for this work had an Intel(R) Xeon(R) W-2125 CPU at 4.00 GHz and 32.0 GB of RAM, running a 64-bit operating system.

## 4. Methodology

The methodology consisted of a four-step research process (see Figure 1).

### 4.1. Asking the LLM for Information on Turbulent Flow Manifolds

Initially, ChatGPT was asked what information it knew about manifolds. This would allow us to determine the case study [32] and the LLM's expertise on this topic. For this first question, the prompt used was: "Are you familiar with turbulent flow manifolds?"

Subsequently, the focus shifted to manifolds for cooling towers, since they use water and fluid mixtures are not necessary, and the CAD model would be based on simple geometries, such as cylinders. For this second question, the prompt used was: "What are the applications of manifolds used in water-cooling towers? What types are there?"

To determine if ChatGPT was familiar with the geometries of the different cooling tower manifolds, it was asked to generate an image of each of the manifolds described above. The prompt for this third question was: "Could you generate an image of each of the following manifolds so I can identify them? Header-lateral manifold, Ring (loop) manifold, Grid manifold, By hydraulic mode".

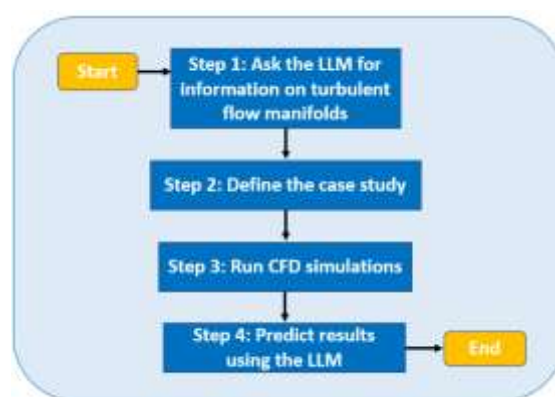


Figure 1. Steps of the methodology used.

To verify LLM's understanding of the variables involved in manifold design, it was asked to describe the equations for a header-lateral manifold. The prompt for this fourth question was: "Can we go into design equations for header-lateral manifolds?"

Finally, it was decided to investigate whether ChatGPT could specify the boundary conditions necessary to perform a CFD simulation of a header-lateral manifold. The prompt for this last question was: "If I wanted to perform a CFD simulation of a header-lateral manifold, what boundary conditions would I need to use?"

### 4.2. Defining the Case Study

When ChatGPT was asked if it was familiar with turbulent flow manifolds, it mentioned that one of its areas of work was CFD simulation to analyze the effects of the curved walls of its geometries. Since this work was the first evaluation of LLMs in this type of geometry and application, it was decided to use a manifold with varying geometries but handling a single-phase fluid, as in the header-lateral manifolds of cooling towers for air conditioning systems. Furthermore, LLM was familiar with the components of this type of system, having generated an image and listed the boundary conditions necessary for its CFD simulation. Therefore, the case study involved a header-lateral manifold feeding a cooling tower for an air conditioning system; this manifold is shown in Figure 2.

The case study consists of a single main header with a diameter of 0.4 m, six laterals with a diameter of 0.1 m, and two nozzles per lateral with a diameter of 0.1 m. Nozzle diameters are normally specified smaller than those of the laterals; in this case, they were left unchanged to

investigate whether ChatGPT would provide any recommendations [33]. On the other hand, the fluid circulating in the manifold will be water.

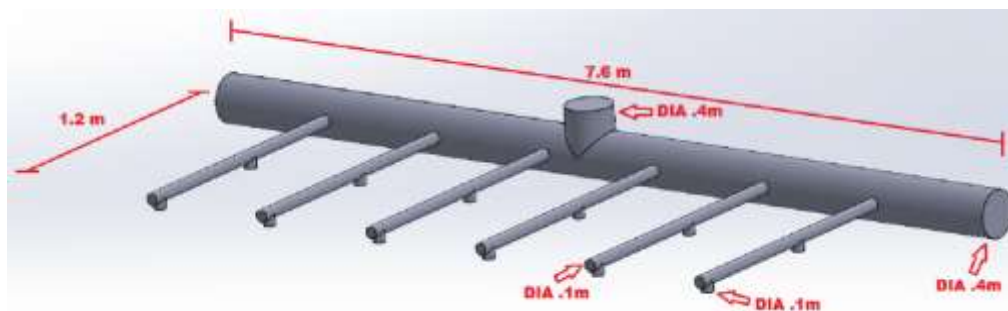


Figure 2. Case study: Header-lateral manifold.

#### 4.3. Running CFD Simulations

A CFD simulation of the case study was performed with the boundary conditions shown in Figure 3. Water enters the manifold at 80°C at a velocity of 0.955 m/s; the manifold walls are at 20°C, and the nozzle outlets are at atmospheric pressure. The boundary conditions are listed in Table 1. The mesh parameters for the case study were left on automatic so that, depending on the results, ChatGPT could be asked for suggestions for improvement. The automatic method uses quad elements with a sweep strategy. Gravity was not activated to investigate whether ChatGPT would recommend any adjustments.

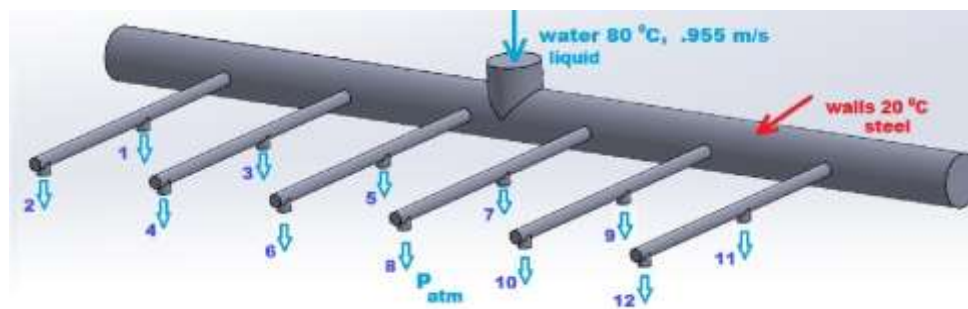


Figure 3. Case study with boundary conditions.

Table 1. Boundary conditions.

Variable	Magnitude
$V_{inlet}$	.955 m/s
Re	924534.55 (turbulent flow)
Temp <sub>water</sub>	80 °C
Temp <sub>walls</sub>	20 °C
P <sub>outlets</sub>	P <sub>atm</sub>
Constant roughness	.00003 m
Manifold Material	Steel
Turbulence model	k- $\omega$ SST

#### 4.4. Predicting Results Using the LLM

The water streamlines within the manifold were obtained using CFD simulations; these are shown in Figure 4. A cross-section of the manifold was created in a horizontal plane to visualize the distributions of velocity (see Figure 5), temperature (see Figure 6), and pressure (see Figure 7) in greater detail. The input variables were parameterized: inlet water velocity (Inlet\_velocity), manifold wall temperature (Temp\_walls), and inlet water temperature (Temp\_Inlet). The output variables

were also parameterized: water velocity at outlet 1 (Outlet\_velocity1), temperature at outlet 1 (Outlet\_temperature1), and the same two variables for the remaining outlets. Twenty-three simulations were performed, varying the values of the three parameterized inputs, and 24 output values were obtained for each. Table 2 shows the three input variables and their 24 output variables for simulations 1, 2, 22, and 23. The reduced number of simulations was selected to investigate whether the LLM would have any observations to inform its prediction model during creation.

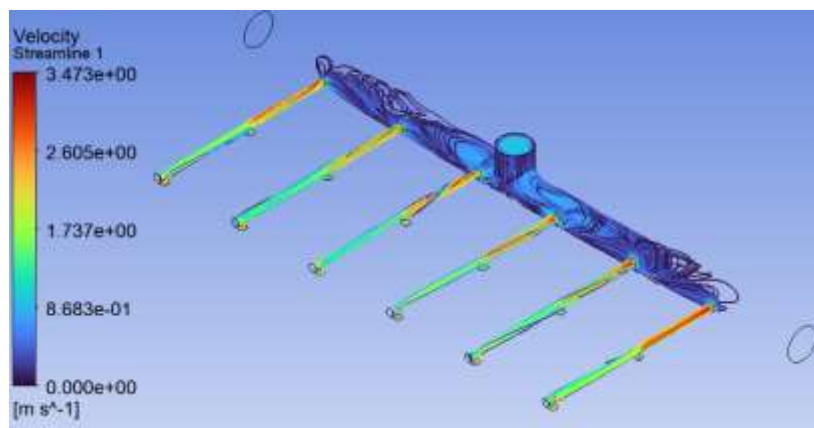


Figure 4. Fluid streamlines within the manifold.

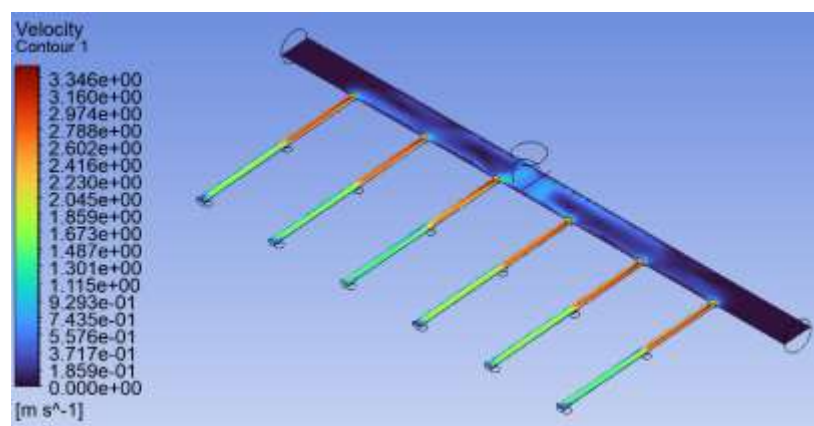


Figure 5. Velocity distribution in the manifold.

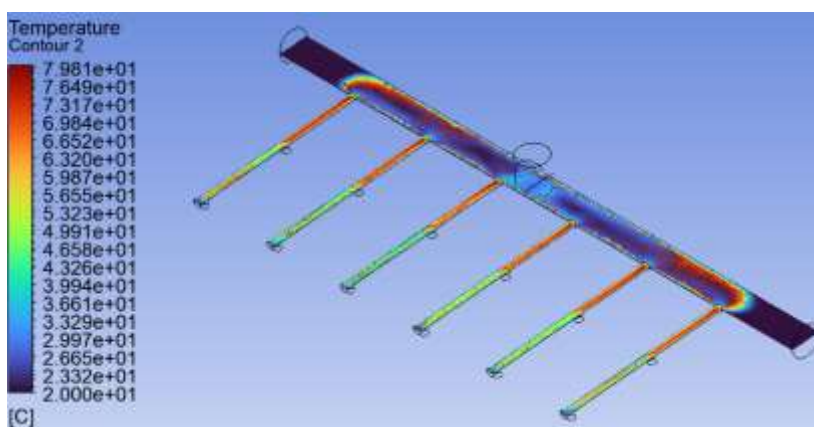


Figure 6. Temperature distribution in the manifold.

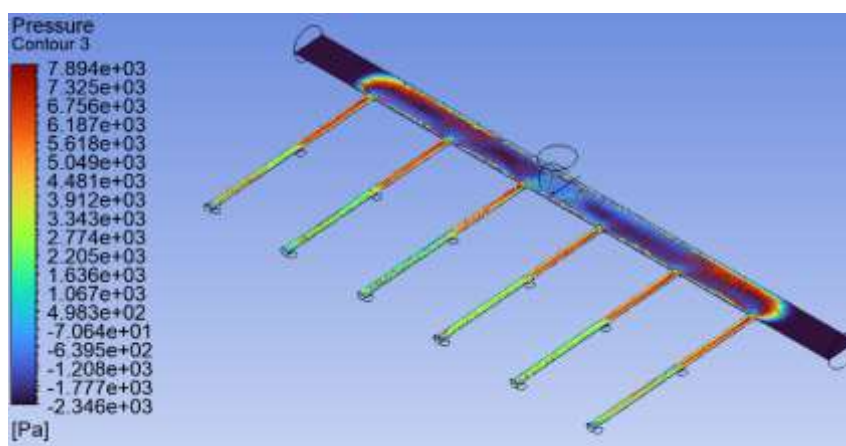


Figure 7. Pressure distribution in the manifold.

Table 2. The results of the 23 simulations were presented as a function of three input variables.

Simulation	Inlet Vel (m/s)	Temp walls (°C)	Temp Inlet (°C)	Outlet Vel 1 (m/s)	Outlet Vel 2 (m/s)	Outlet Vel 3 (m/s)	Outlet Vel 4 (m/s)	Outlet Vel 5 (m/s)	Outlet Vel 6 (m/s)
1	0.955	20	80	1.25466	1.80999	1.22838	1.8053	1.34525	1.73497
2	1	19	80	1.35044	1.92338	1.32116	1.86866	1.42323	1.81078
...	...	...	...	...	...	...	...	...	...
22	1.75	17	83	2.30962	3.35957	2.29337	3.25306	2.52137	3.08176
23	1.8	23	87	2.37434	3.45639	2.36166	3.34216	2.59446	3.166

Simulation	Outlet Vel 7 (m/s)	Outlet Vel 8 (m/s)	Outlet Vel 9 (m/s)	Outlet Vel 10 (m/s)	Outlet Vel 11 (m/s)	Outlet Vel 12 (m/s)	Outlet Temp 1 (°C)	Outlet Temp 2 (°C)	Outlet Temp 3 (°C)
1	1.23728	1.79261	1.19051	1.88414	1.34173	1.79716	43.477	54.131	47.002
2	1.31546	1.86497	1.30703	1.86598	1.34956	1.91842	45.855	51.309	47.351
...	...	...	...	...	...	...	...	...	...
22	2.22773	3.29674	2.12367	3.52548	2.25611	3.44066	46.806	52.147	48.428
23	2.28121	3.39978	2.18247	3.61619	2.32226	3.54225	48.653	54.177	50.176

Simulation	Outlet Temp 4 (°C)	Outlet Temp 5 (°C)	Outlet Temp 6 (°C)	Outlet Temp 7 (°C)	Outlet Temp 8 (°C)	Outlet Temp 9 (°C)	Outlet Temp 10 (°C)	Outlet Temp 11 (°C)	Outlet Temp 12 (°C)
1	55.391	49.801	56.714	48.174	56.131	45.445	57.712	45.102	47.56
2	54.608	49.795	57.566	48.222	55.971	45.657	56.266	45.018	50.745
...	...	...	...	...	...	...	...	...	...
22	55.92	51.498	58.54	48.834	57.737	46.135	59.692	45.05	52.248
23	58.228	53.501	61.163	50.583	60.364	47.651	62.322	46.473	54.041

Once the outlet velocities and temperatures were obtained, it was investigated whether the LLM could predict these 24 outcomes from the three input variables. ChatGPT was then asked to generate a predictive model from the results of the first 21 simulations. Subsequently, to evaluate this model, three input data points were provided, and the 24 outputs were predicted.

To evaluate the prediction results, the values from simulations 22 and 23 were used; these tests were referred to as the first and second predictions, respectively

## 5. Results Analysis

### 5.1. Asking the LLM for information on turbulent flow in manifolds

When ChatGPT was asked about its knowledge of manifolds, it categorized them into applications for Physics, Robotics, CFD/Mechanics, Machine Learning, and Differential Equations. Table 3 summarizes the LLM's classification. This indicates that it has sufficient information about manifold applications and is familiar with CFD.

When asked about the applications of manifolds used in water-cooling towers, ChatGPT categorized them by purpose, function, and geometry (see Table 4), suggesting a case study on manifold geometry for CFD simulations.

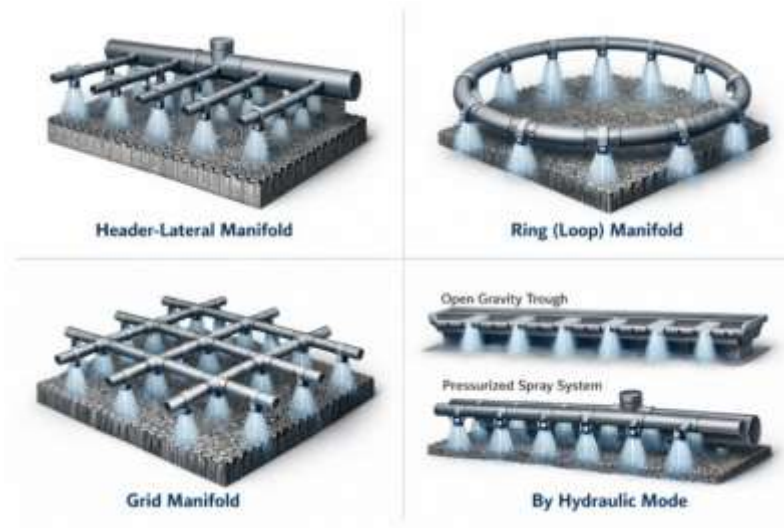
**Table 3.** Classification of manifolds according to the LLM used.

Manifold application	Uses
Physics	spacetime, general relativity
Robotics	configuration spaces
CFD / mechanics	curved surfaces, shells
Machine learning	data lies on a low-dimensional manifold
Differential equations	solution spaces

**Table 4.** Applications of manifolds used in water-cooling towers according to the LLM used.

Classification	Function	Uses
By purpose	Hot water distribution (spray manifold)	Takes hot water from the condenser loop Spreads it evenly over the fill media Maximizes surface area for evaporation
	Cold water collection manifolds	Collect cooled water from multiple outlets Feed it uniformly into the basin or the pump suction
	Make-up water manifolds	Add fresh water to replace evaporation + blowdown Distribute it gently to avoid disturbing the airflow or fill
By function	Spray distribution manifolds	Main header + lateral pipes Nozzles or orifices along laterals
	Collection manifolds	Combine the flow from multiple outlets Used in closed circuits or hybrid towers
By geometry	Header-lateral manifold	Crossflow towers Counterflow towers
	Ring (loop) manifold	Circular or rectangular loop Helps equalize pressure Very uniform flow
	Grid manifold	Network of pipes in X-Y pattern Very uniform distribution

To demonstrate that ChatGPT understood the difference between the manifold geometries—Header-lateral manifold, Ring (loop) manifold, Grid manifold, and hydraulic mode—it was asked to generate an image of each, shown in Figure 8. It can be observed that each configuration includes the main details, confirming that ChatGPT distinguishes between them.



**Figure 8.** Images generated by ChatGPT of manifolds classified by geometry.

Subsequently, ChatGPT was asked to describe the equations for the header-lateral manifold. The equations provided by ChatGPT are listed below, along with the sequence of use and recommended considerations:

Design goal for a header with laterals and nozzles (each nozzle gets the same flow):

$$Q_1 = Q_2 = Q_3 = \dots = Q_N \quad (1)$$

$$P_{\text{nozzle},1} \approx P_{\text{nozzle},2} \approx \dots \quad (2)$$

5.1.1. Flow from each nozzle (orifice equation):

$$Q_n = C_d A_n \sqrt{\frac{2(P_h - P_{\text{atm}})}{\rho}} \quad (3)$$

Where:

$Q_n$  = flow per nozzle

$C_d$  = discharge coefficient ( $\approx 0.6$ – $0.9$ )

$A_n$  = nozzle area

$P_h$  = pressure in lateral at nozzle

$\rho$  = water density

Uniform flow  $\Rightarrow$  uniform  $P_h$ .

5.1.2. Pressure Drops Along a Lateral (Darcy–Weisbach), for Each Lateral:

$$\Delta P_f = f \frac{L}{D} \frac{\rho V^2}{2} \quad (4)$$

Velocity decreases as water exits through nozzles, so more properly:

$$\Delta P_{\text{lateral}} = \sum_i f \frac{\Delta x}{D} \frac{\rho V_i^2}{2} \quad (5)$$

Engineering approximation (common in cooling tower design):

$$\Delta P_{\text{lateral}} \leq 0.2 \Delta P_{\text{nozzle}} \quad (6)$$

## 5.1.3. Header Pressure Drops, Same Idea for the Main Header.

$$\Delta P_{header} = f \frac{L_h \rho V_h^2}{D_h 2} \quad (7)$$

$$\Delta P_{header} \leq 0.1 \Delta P_{nozzle} \quad (8)$$

$$\Delta P_{pipes} \leq 0.3 \Delta P_{nozzle} \quad (9)$$

The equations above ensure pressure at all laterals is nearly the same.

## 5.1.4. Nozzle Spacing and Count:

Total flow =  $Q_T$

Number of nozzles =  $N$

$$Q_n = \frac{Q_T}{N} \quad (10)$$

Select nozzle size from:

$$A_n = \frac{Q_n}{C_d \sqrt{2 \Delta P_n / \rho}} \quad (11)$$

Spacing  $s$  usually:

$$s \approx 0.3 \text{ to } 0.6 \text{ m} \quad (12)$$

## 5.1.5. Lateral Diameter Sizing.

At the inlet of the lateral:

$$Q_{lat} = n Q_n \quad (13)$$

Choose velocity:

$$V_{lat,in} = 1\text{--}2 \text{ m/s} \quad (14)$$

$$D_{lat} = \sqrt{\frac{4 Q_{lat}}{\pi V_{lat}}} \quad (15)$$

Check friction loss:

$$\Delta P_{lateral} \leq 0.2 \Delta P_{nozzle} \quad (16)$$

## 5.1.6. Header Diameter Sizing.

Total flow:

$$Q_h = Q_T \quad (17)$$

Choose velocity:

$$V_h = 1.5\text{--}3 \text{ m/s} \quad (18)$$

$$D_h = \sqrt{\frac{4 Q_h}{\pi V_h}} \quad (19)$$

Check:

$$\Delta P_{header} \leq 0.1\Delta P_{nozzle} \quad (20)$$

### 5.1.7. Uniformity Criterion.

Uniformity error:

$$\epsilon = \frac{Q_{max} - Q_{min}}{Q_{avg}} \quad (21)$$

Good design:

$$\epsilon \leq 5\% \quad (22)$$

Achieved if:

$$\frac{\Delta P_{pipes}}{\Delta P_{nozzle}} \leq 0.3 \quad (23)$$

Typical numbers (cooling tower spray system).

Nozzle  $\Delta P$ : 20–50 kPa

Lateral friction:  $\leq 5$ –10 kPa

Header friction:  $\leq 3$ –5 kPa

Nozzle spacing: 300–600 mm

Cd: 0.7–0.85

From the above, it can be observed that ChatGPT identifies the main variables involved in the design of a header-lateral manifold. While some equations lacked definitions for certain variables, it can generally be stated that they distinguish the essential variables for the design.

When ChatGPT was asked about the boundary conditions required to perform a CFD simulation of a header-lateral manifold, it specified the sequence of steps illustrated in Figure 9.

Initially, the boundary conditions of the main inlet (header) must be defined; specifically, whether to register the mass flow inlet or the velocity inlet. Subsequently, the boundary conditions for the outlets are specified, either by defining a mass flow outlet or a pressure outlet. Following this, it is specified that the walls have no movement or displacement, that gravity is activated, that turbulent flow is specified, and, finally, that the fluid type is indicated, specifying whether it is single-phase or multi-phase. It appears that ChatGPT has a general understanding of what needs to be defined to perform a CFD simulation. Furthermore, ChatGPT mentioned that the expected results would be the flow rate per nozzle, the pressure distribution along the manifold, the velocity field, and the streamlines.

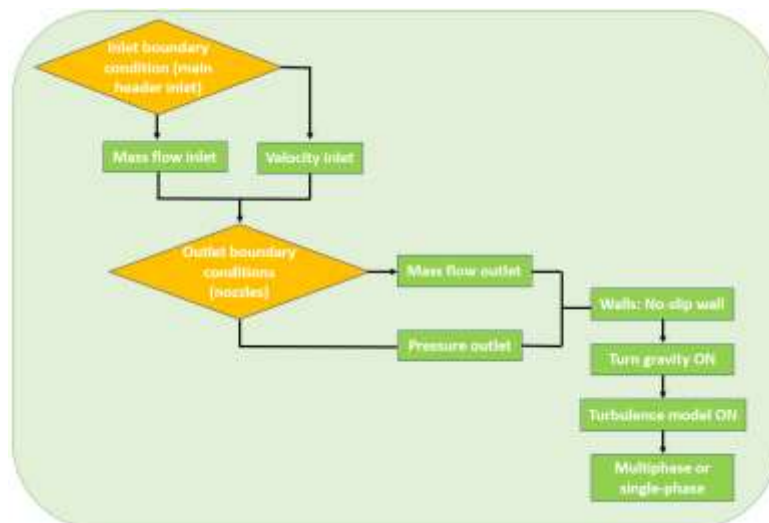


Figure 9. Definition of boundary conditions by the employed LLM.

## 5.2. Running CFD Simulations

Using the case study in Figure 2 and the boundary conditions in Table 1, a CFD simulation was performed. The mesh was generated using the automatic method; the resulting mesh is shown in Figure 10. The general mesh parameters are shown in Figure 11, and the mesh inflation and sizing parameters are shown in Figure 12.

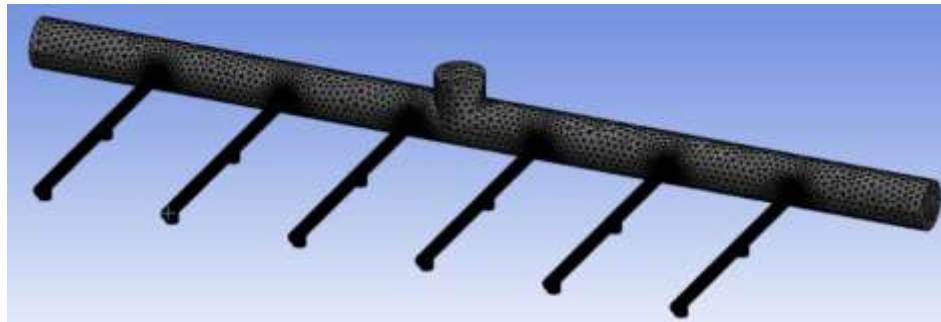


Figure 10. Mesh of the case study.

Details of "Mesh"		Advanced	
Physics Preference	CFD	Number of CPUs for Parallel...	Program Controlled
Solver Preference	Fluent	Straight Sided Elements	
Element Size	Default (0.40438 m)	Rigid Body Behavior	Dimensionality Reduced
Export Format	Standard	Triangle Surface Mesher	Program Controlled
Export Preview Surface Mesh	No	Topology Checking	Yes
		Pinch Tolerance	Default (3.6394e-003 m)
		Generate Pinch on Refresh	No
		Automatic Methods	
		Sheet Body Method	Quad Dominant
		Sweepable Body Method	Sweep
		Statistics	
		Nodes	63207
		Elements	296886
		Show Detailed Statistics	No

Figure 11. General mesh parameters.

Inflation		Sizing	
Use Automatic Inflation	None	Use Adaptive Sizing	No
Inflation Option	Smooth Transition	Growth Rate	Default (1.2)
Transition Ratio	0.272	Max Size	Default (0.80876 m)
Maximum Layers	5	Mesh Defeatureing	Yes
Growth Rate	1.2	Defeature Size	Default (2.0219e-003 m)
Inflation Algorithm	Pre	Capture Curvature	Yes
View Advanced Options	No	Curvature Min Size	Default (4.0438e-003 m)
		Curvature Normal Angle	Default (18.0°)
		Capture Proximity	No
		Bounding Box Diagonal	8.0876 m
		Average Surface Area	0.19783 m <sup>2</sup>
		Minimum Edge Length	5.e-002 m

Figure 12. Mesh inflation and sizing parameters.

A few iterations (100) were performed to investigate whether ChatGPT suggested any recommended values. On the other hand, the mass balance between the inlet and the 12 outlets was verified, as shown in Figure 13. The difference between the mass of the inlet and the mass of the 12 outlets was 0.0958004 kg/s. To improve this result, ChatGPT was provided with the case study and mesh parameters. LLM was then asked what should be corrected or done to reduce the difference between the inlet mass and the mass of the 12 outlets.

Calculation complete.

Mass Flow Rate	[kg/s]
inlet	117.83202
outlet-1	-7.7980499
outlet-10	-11.762518
outlet-11	-8.4094057
outlet-12	-11.113256
outlet-2	-12.059905
outlet-3	-7.9658813
outlet-4	-11.812842
outlet-5	-8.4044237
outlet-6	-10.882451
outlet-7	-7.1444826
outlet-8	-12.366513
outlet-9	-8.0164862
Net	0.0958004

Figure 13. Mass balance of the inlet and the 12 outlets.

From the images provided to ChatGPT, it identified the dimensions of the main header, laterals, and nozzles. The first observation was that the nozzles should have a smaller diameter than the laterals, since they should generate the greatest pressure loss to ensure flow uniformity. Otherwise, the flow could become unbalanced: the first nozzles might discharge more water, or the last ones might receive less. It also identified the boundary conditions and calculated the Reynolds number, which matched the value presented in Table 1. Regarding the temperatures on the walls and in the water, it was concluded that when simulating the cooling of the water inside the manifold, intense cooling occurs in the cooling tower, and that, in the manifold, little heat is normally lost. It mentioned that, in the results, a maximum pressure should be observed near the inlet and a progressive decrease along the 7.6 m. The flow should enter strongly in the first few laterals and decrease gradually. The hot water should be near the inlet, and we would expect gradual cooling near the walls, with thermal mixing in the header. However, the effect would be small because the residence time is short. It also commented that it recommended activating gravity at  $9.81 \text{ m/s}^2$ , since the fluid was water. Regarding the mesh parameters, the first observation was the element size of  $0.404 \text{ m}$ , since the main header diameter was  $0.4 \text{ m}$ . This would not be represented or could cause errors, as it exceeded the diameter of the sides and nozzles. It indicated that, in CFD, it typically aims for 10-20 cells across the tube diameter.

$$\Delta x \approx \frac{D}{15} ; \quad \Delta x \approx \frac{0.4}{15} \approx 0.027 \text{ m} \quad (24)$$

Regarding the  $k-\omega$  SST turbulence model, it was noted that it is an excellent choice, as it is well-suited for internal flows, flow separations, adverse pressure gradients, and T-shaped junctions (such as those found on laterals). It added that many industrial studies use this model for manifolds. It recommended a skewness  $< 0.85$ ; if possible, a skewness  $< 0.5$  would be excellent. It suggested refining the mesh at the header-lateral connection, where flow separation and pressure redistribution occur. Refining the mesh at the header inlet is also recommended, as this is where flow development and the velocity gradient are recorded. Furthermore, it emphasized the need to refine the nozzle mesh due to the large accelerations and pressure drops. If the element size is reduced to  $0.03 \text{ m}$ , the mesh would likely increase to 2–4 million elements, which would be typical in a CFD case like this. For inflation parameters, he recommends 10–15 layers and a growth rate of approximately 1.2. Regarding the mass balance, it identified a mass of  $117.832 \text{ kg/s}$  at the inlet and approximately  $117.736 \text{ kg/s}$  at the outlets, resulting in a difference of  $0.0958 \text{ kg/s}$ . It mentioned that the simulation is already at approximately  $0.08\%$ , which is a very good balance, but it can be improved.

$$\text{Error} = \frac{0.0958}{117.832} = 8.13 \times 10^{-4} ; \quad \text{Error} = 0.081\% \quad (25)$$

To improve results, ChatGPT suggested increasing the number of iterations; for turbulent flow simulations with energy, a range of 1000-3000 iterations is recommended. The residuals for the continuity equation should be in the range of  $1e-4$ , for the momentum equation in the range of  $1e-5$ ,

for turbulence in the range of  $1e-5$ , and for the energy equation in the range of  $1e-6$ . For discretization, a second-order scheme is recommended for pressure, momentum, turbulence, and energy.

ChatGPT identified areas for improvement in the CFD simulation results. For a novice user or student, ChatGPT would indeed help them obtain better results in their simulations. It is also observed that ChatGPT can read images that contain values; for example, the mesh parameter tables were images, and it extracted numerical values.

### 5.3. Predict Results Using the LLM

To predict the 24 outputs (velocity and temperature), given three inputs (inlet velocity, manifold wall temperature, and water inlet temperature), ChatGPT was provided with an Excel spreadsheet containing the results of 23 CFD simulations. The following prompt asked them to describe the variables in the file: "Could you describe the Excel file? What data is in it? How many variables are there, and what are their units?" Table 5 summarizes the structure of the file identified by ChatGPT.

**Table 5.** Results of the file analysis performed by ChatGPT.

Structure	Description
File Size	Total Rows: 24
	Total Columns: 28
	Total Rows: 24
Row Structure	Row 1: Units of the variables.
	Rows 2–24: Experimental data or simulations (23 records named Data 1, Data 2, ...).
Summary of input/control variables	Inlet velocity (m/s) Temp_walls (°C) Temp_Inlet (°C)
Summary of output variables	12 outlet velocities 12 outlet temperatures
Total variables:	3 input variables 24 output variables
Inlet velocity ranges	0.955 – 2.0 m/s
Outlet temperature ranges	47.6 – 54.6 °C
High-flow outlets	Outlets 2, 4, 8, 10, and 12 Velocity $\approx$ 2.7 – 2.9 m/s
Low-flow outlets	Outlets 1, 3, 7, 9, and 11 Velocity $\approx$ 1.8 – 1.9 m/s

ChatGPT was subsequently asked to create a predictive model using data from the Excel file and to predict 24 output variables given three input values (Inlet Velocity, Wall Temperature, and Inlet Temperature). Data 22 was used as data input for the first prediction. The prompt used to make the first prediction is shown in Figure 14.

> Please disregard the data in Data 22 and Data 23.

Using the remaining information, can you create a predictive model? Given three input data points (Inlet Velocity, Wall Temperature, and Inlet Temperature), determine the 24 output variables.

For example, what would the 24 output variables be for Inlet Velocity = 1.75 m/s, Wall Temperature = 17 °C, and Inlet Temperature = 83 °C?

**Figure 14.** Prompt used to make the first prediction.

ChatGPT mentioned that it built a simple multivariable linear regression model using the CFD data, excluding Data 22 and Data 23, as requested. Linear regression was considered given the small range of input speed variation. The summary of the prediction model variables is shown in Figure 15. The model is a multi-output multiple linear regression model, where all outputs are predicted simultaneously and is mathematically represented as:

$$Y = B_0 + B_1V_{in} + B_2T_{wall} + B_3T_{in} \quad (26)$$

where:

- $Y$ = vector of 24 output variables
- $V_{in}$ = inlet velocity
- $T_{wall}$ = wall temperature
- $T_{in}$ = inlet temperature

In Machine Learning, linear regression works well with small datasets:

$$N_{data} \geq 10 \times N_{parameters} \quad (27)$$

In many CFD problems near an operating point, velocity, pressure, and temperature respond approximately linearly to small changes in boundary conditions, especially if the flow regime or fluid properties remain unchanged. The model uses supervised learning with known inputs and outputs. The prediction type is multi-output regression, predicting 12 speeds and 12 temperatures simultaneously. The model assumes a linear relationship and independent variables. The system contains a 24-by-4 coefficient matrix B, and the model is represented by:

$$\begin{bmatrix} V_1 \\ V_2 \\ \dots \\ V_{12} \\ T_1 \\ T_2 \\ \dots \\ T_{12} \end{bmatrix} = \mathbf{B} \begin{bmatrix} 1 \\ V_{in} \\ T_{wall} \\ T_{in} \end{bmatrix} \quad (28)$$

ChatGPT recommended other models if more data were available. For example, with 40–100 simulations, the best models would be Random Forest Regression for small datasets, Gaussian Process Regression for CFD optimizations, Neural Networks for complex simulations (which require more data), and Polynomial Regression to capture nonlinearities.

```
> Model used:
  Inputs (3):
  Inlet velocity  $V_{in}$ 
  Wall temperature  $T_{wall}$ 
  Inlet temperature  $T_{in}$ 

  Outputs (24):
  12 outlet velocities
  12 outlet temperatures
  The regression was fitted with 21 CFD simulations.

Requested prediction
Model input
  Inlet velocity = 1.35 m/s
  Wall temperature = 18 °C
  Inlet temperature = 76 °C
```

Figure 15. Summary of the prediction model variables.

The prediction results are presented in Table 6, which also shows the values compared to those obtained in the CFD simulation. Subsequently, a second prediction was performed using the Data 23 data; the prompt used is shown in Figure 16, and the prediction results are presented in Table 7.

**Table 6.** Results of the first prediction.

Outlet	Predicted values		CFD simulation values	
	Vel (m/s)	Temp (°C)	Vel (m/s)	Temp (°C)
1	2.332	47.12	2.30962	46.806
2	3.326	52.31	3.35957	52.147
3	2.305	48.77	2.29337	48.428
4	3.238	56.08	3.25306	55.92
5	2.5	51.36	2.52137	51.498
6	3.07	58.46	3.08176	58.54
7	2.213	49.05	2.22773	48.834
8	3.281	57.71	3.29674	57.737
9	2.108	46.42	2.12367	46.135
10	3.46	59.88	3.52548	59.692
11	2.281	45.92	2.25611	45.05
12	3.397	52.55	3.44066	52.248

> What would the 24 output variables be for Inlet Velocity = 1.8 m/s, Wall Temperature = 23 °C, and Inlet Temperature = 87 °C?

**Figure 16.** Prompt used to make the second prediction.

**Table 7.** Results of the second prediction.

Outlet	Predicted values		CFD simulation values	
	Vel (m/s)	Temp (°C)	Vel (m/s)	Temp (°C)
1	2.395	48.21	2.37434	48.653
2	3.42	53.64	3.45639	54.177
3	2.366	49.88	2.36166	50.176
4	3.329	57.32	3.34216	58.228
5	2.571	52.63	2.59446	53.501
6	3.161	59.84	3.166	61.163
7	2.267	50.11	2.28121	50.583
8	3.365	59.02	3.39978	60.364
9	2.154	47.51	2.18247	47.651
10	3.548	61.24	3.61619	62.322
11	2.338	46.92	2.32226	46.473
12	3.486	53.88	3.54225	54.041

For the first and second predictions, the root mean square error (RMSE) and the coefficient of determination ( $R^2$ ) were calculated; the results are shown in Table 8. Both the RMSEs for velocity and temperature are acceptable, as is the  $R^2$ . A parity plot (Predicted vs. CFD) was also generated for both velocity and temperature predictions. Figure 17 shows the parity plot for velocity and temperature from the first prediction, and Figure 18 shows the parity plot for the second prediction.

The model accurately captures physical trends. The prediction model successfully reproduces the outlet velocity hierarchy, the temperature distribution pattern, and the response to inlet velocity

and temperature changes. This means the dataset has a strong predictive structure, which is ideal for surrogate modeling.

Table 8. RMSE and R<sup>2</sup> results.

Prediction	Values Obtained
First prediction	RMSE_velocity = 0.0290 m/s RMSE_temperature = 0.330 °C R <sup>2</sup> velocity = 0.999 R <sup>2</sup> temperature = 0.998
Second prediction	RMSE_velocity = 0.033 m/s RMSE_temperature = 0.783 °C R <sup>2</sup> velocity = 0.999 R <sup>2</sup> temperature = 0.997

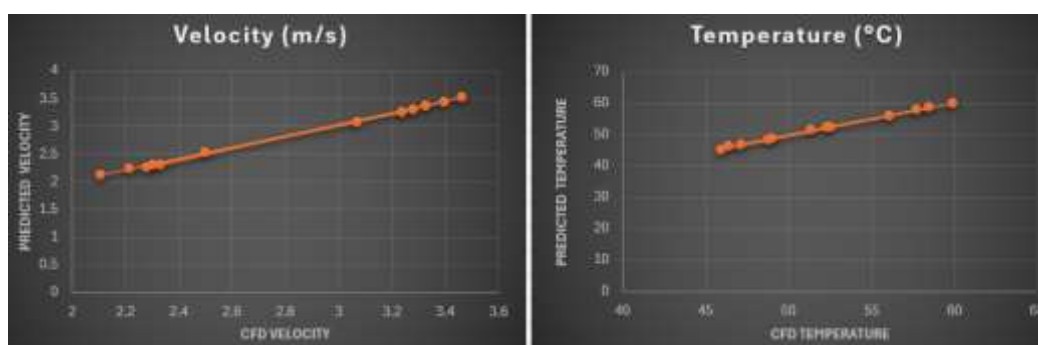


Figure 17. Parity plot of velocity and temperature for the first prediction.

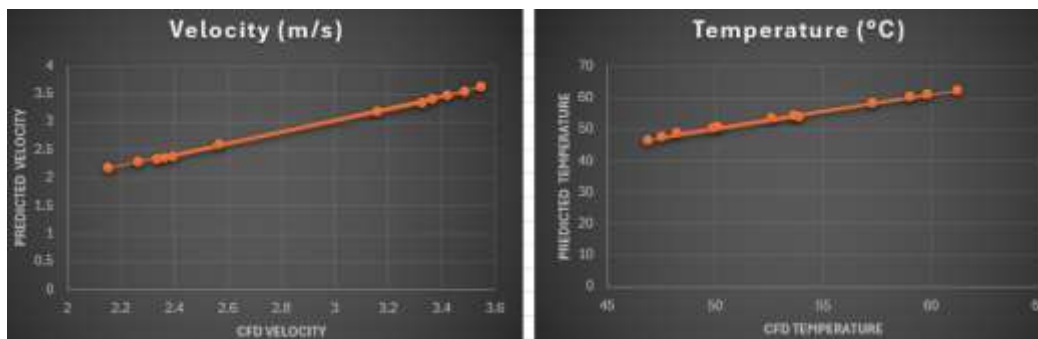


Figure 18. Parity plot of velocity and temperature for the second prediction.

Finally, the LLM was asked what it could infer from the nozzle temperature distributions. The prompt used was: "In the Excel file, regarding the temperature distributions in the nozzles, what do you infer?" In the answer, it was mentioned that the outlet temperatures range from  $\approx 45$  °C (coolest) to  $\approx 59$  °C (warmest), so the manifold shows a temperature spread of about 14 °C, indicating a significant thermal imbalance among the nozzles. The highest temperatures occur at outlets 10 ( $\approx 59$  °C), 6 ( $\approx 58$  °C), 8 ( $\approx 57$  °C), and 4 ( $\approx 56$  °C); these outlets likely correspond to the main high-momentum flow paths within the manifold. One interpretation is that fluid reaching these outlets probably travels more directly from the inlet and therefore spends less time near the cooled walls, thereby losing less heat. The lowest temperatures occur at outlets 11 ( $\approx 45$  °C), 9 ( $\approx 45.9$  °C), and 1 ( $\approx 46.4$  °C); these branches likely experience longer residence times. The fluid interacts more with the cooler walls, leading to greater heat transfer and a lower outlet temperature. There is a strong correlation with flow distribution: high-flow outlets  $\rightarrow$  higher temperatures and low-flow outlets  $\rightarrow$  lower temperatures. This happens because higher velocity  $\rightarrow$  shorter residence time  $\rightarrow$  less cooling and

lower velocity → longer residence time → more heat loss, so the hydraulic imbalance of the manifold mainly drives the thermal imbalance.

CFD results indicate that the manifold exhibits flow maldistribution, resulting in temperature maldistribution. This is common in manifolds when outlet spacing is uniform, but pressure losses along the header are not compensated. What this means for the design is that the temperature pattern suggests the manifold might benefit from progressively smaller diameters, flow-restrictor or orifice balancing, optimized outlet spacing, or a tapered manifold. These strategies help equalize the mass flow rate and residence time, thereby reducing temperature variation between nozzles.

## 6. Conclusions and Future Work

The LLM used contains extensive information on turbulent flow in manifolds and can offer recommendations to improve the results of CFD simulations in this type of system.

The LLM can interpret CAD images of the manifold model and images of CFD simulation results. It was also able to interpret the variables in an Excel file, as well as the behavior of velocities and pressures.

LLM was also found to be capable of predicting turbulent flow behavior in manifolds based on its CFD simulations.

One advantage of using LLMs for predictions is that it is not necessary to program Machine Learning code; only Prompt Engineering is used to communicate with them.

To interact with the LLM, the user needs to be familiar with CFD simulations and the variables involved in turbulent flows, as the LLM requires clarification of any ambiguities identified to provide accurate results. It is suggested to use case studies with CAD models that allow their dimensions to be modified to evaluate the scope of LLMs further.

On the other hand, it is recommended to use a fluid mixture to study the behavior and interpretation of variables in turbulent flows within manifolds using LLMs.

**Author Contributions:** Conceptualization, H.R.M.-O. and A.R.S.V.-C.; methodology, H.R.M.-O., A.R.S.V.-C. and G.S.-B; software, H.R.M.-O.; validation, H.R.M.-O., A.R.S.V.-C. and G.S.-B; formal analysis, A.R.S.V.-C. and G.S.-B; investigation, H.R.M.-O., A.R.S.V.-C. and G.S.-B; resources, H.R.M.-O., A.R.S.V.-C. and G.S.-B; data curation, G.S.-B; writing—original draft preparation, H.R.M.-O.; writing—review and editing, H.R.M.-O., A.R.S.V.-C. and G.S.-B; visualization, H.R.M.-O., A.R.S.V.-C. and G.S.-B; supervision, A.R.S.V.-C. and G.S.-B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author. Further inquiries can be directed to the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

CFD	Computational Fluid Dynamics
LLMs	Large Language Models
CAD	Computer-Aided Design
QwQ-32B	Qwen with Question/thought
GPT	Generative Pre-trained Transformer
RAG	Recovery augmented generation
NIROMs	Non-intrusive reduced-order models

WGAN	Wasserstein generative adversarial network
BiLSTM	Bidirectional long short-term memory
UAV	Unmanned Aerial Vehicle
RANS	Reynolds-Averaged Navier-Stokes
UUVs	Unmanned Underwater Vehicles
HVAC	Heating, ventilation, and air conditioning
LSTM	Long Short-Term Memory
ANNs	Artificial Neural Networks
TTX2T	Twin Turbocharged Cross-Dual Tubular
DOF	Degrees of freedom
RSM	Resistive Sorting Method
BBD	Box-Behnken Design
NSGA-II	Non-Dominated Sorting Genetic Algorithm II
API	Application Programming Interface
SARD	Software Assurance Reference Dataset
RMSE	Root Mean Square Error
R <sup>2</sup>	R-squared, the coefficient of determination
BLEU	Bilingual Evaluation Understudy
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
$Q_n$	Flow per nozzle
$C_d$	Discharge coefficient
$A_n$	Nozzle area
$P_h$	Pressure in the lateral at the nozzle
$\rho$	Water density
$\Delta P_{header}$	Header pressure drops
$Q_T$	Total flow
$N$	Number of nozzles
$\epsilon$	Uniformity error
$D$	Tube diameter

## References

1. Wang, W.; Xu, R.; Feng, J.; Zhang, Q.; Pandey, S.; Chu, X. A status quo investigation of large-language models for cost-effective computational fluid dynamics automation with OpenFOAMGPT. TAML. 2025, 15, 100623. <https://doi.org/10.1016/j.taml.2025.100623>
2. Dong, Z.; Lu, Z.; Yang, Y. Fine-tuning a large language model for automating computational fluid dynamics simulations. TAML. 2025, 15, 100594. <https://doi.org/10.1016/j.taml.2025.100594>
3. Pandey, S.; Xu, R.; Wang, W.; Chu, X. OpenFOAMGPT: A retrieval-augmented large language model (LLM) agent for OpenFOAM-based computational fluid dynamics. Physics of Fluids. 2025, 37, 035120. <https://doi.org/10.1063/5.0257555>
4. Masoumi-Verki, S.; Haghighat, F.; Bouguila, N.; Eicker, U. The use of GANs and transfer learning in model-order reduction of turbulent wake of an isolated high-rise Building. BUILDENV. 2023, 246, 110948. <https://doi.org/10.1016/j.buildenv.2023.110948>
5. Babanezhad, M.; Behroyan, I.; Nakhjiri, A.T.; Marjani, A.; Shirazian, S. Performance and application analysis of ANFIS artificial intelligence for pressure prediction of nanofluid convective flow in a heated pipe. Scientific Reports. 2021, 11, 902. <https://doi.org/10.1038/s41598-020-79628-w>
6. Veiga-Piñeiro, G.; Domínguez-Estévez, P.; Aldao, E.; González-Jorge, H.; Martín, E. Hybrid AI-reduced-order model for real-time wind flow prediction around offshore wind farms. EUROMECHFLU. 2026, 118, 204491. <https://doi.org/10.1016/j.euromechflu.2026.204491>
7. Yang, Ch.; Zhao, X.; Ye, J. A data-driven UUVs bionic design method toward emotional and energetic sustainability using AI-based morphological synthesis. ESWA. 2026, 300, 130499. <https://doi.org/10.1016/j.eswa.2025.130499>
8. Christakis, N.; Tirchas, P.; Kokkinakis, I.W.; Drikakis, D. Forecasting chaos: AI-enhanced prediction of indoor climate dynamics. ESWA. 2026, 300, 130186. <https://doi.org/10.1016/j.eswa.2025.130186>
9. Sameh, A.; Elhenidy, A.M. Bio-inspired underwater robotic vehicle for marine exploration and AI-powered fish detection. Discover Applied Sciences. 2025, 8, 117. <https://doi.org/10.1007/s42452-025-07991-2>

10. Hasni, A.; Qureshi, M.H.; Ejaz, M.; Khan, A.S.; Mubashir, H.S. Experimental Quantification and Computational Synergy of Dual Cross Tubular Combustion in Twin Turbojet Engines: A Mach-Regime Thermochemical Disruption. *IJASS*. 2026, 27, 142-156. <https://doi.org/10.1007/s42405-025-00975-x>
11. Remache, A.; Pérez-Sánchez, M.; Hidalgo, V.H.; Ramos, H.M. Hybrid Optimization Approaches for Impeller Design in Turbomachinery: Methods, Metrics, and Design Strategies. *Water*. 2025, 17, 1976. <https://doi.org/10.3390/w17131976>
12. Chen, W.-C.; Chen, S.-Y. TIPS: A Prompt Engineering Framework for Code Classification and Generation on Resource-Constrained Systems. *IEEE Access*. 2026, 14, 15723-15735. <https://doi.org/10.1109/ACCESS.2026.3658502>
13. Chan, Y.K.; Lai, E.; Chen, Y. Teaching Tip: Teaching Undergraduate IS Students Hands-on Generative AI Development Skills. *JISE*. 2026, 37, 1-17. <https://doi.org/10.62273/HPDA8033>
14. Thota, D.; Alt, D.; Cole, J.; Tring, V.S. Prompting Pro Tips! Best Practices for Generating Clinical Narrative Summaries. *Mil Med*. 2026, 191, e445-e448. <https://doi.org/10.1093/milmed/usaf247>
15. Bae, J.; Kwon, S.; Myeong, S. Enhancing Software Code Vulnerability Detection Using GPT-4o and Claude-3.5 Sonnet: A Study on Prompt Engineering Techniques. *Electronics*. 2024, 13, <https://doi.org/10.3390/electronics13132657>
16. Nema, B.M.; Nezer, A.I.; Makki, S.W. Prompt Engineering: Unleashing the Power of Large Language Models to Defend Against Social Engineering Attacks. *IJCSM*. 2024, 5, 45. <https://doi.org/10.52866/ijcsm.2024.05.03.024>
17. Jia, J.; Wang, T.; Zhang, Y.; Wang, G. The comparison of general tips for mathematical problem solving generated by generative AI with those generated by human teachers. *APJE*. 2024, 44, 8-28. <https://doi.org/10.1080/02188791.2023.2286920>
18. Ugas, M.; Huynh, J.; Lenarcik-Packham, A.; Kukan, A.; Deol, A.; Fazelzad, R.; Fernando, S.; Giannopoulos, E.; Giuliani, M.E.; Lawrie, K.; Quartey, N.K.; Ralliarum, R.; Thankarajah, A.; Papadacos, J. The utility of artificial intelligence in plain language writing: A scoping review. *PEC*. 2026, 145, 109483. <https://doi.org/10.1016/j.pec.2026.109483>
19. Ndukwe, H.; Oloche-Otukpa, E. Protocol for a scoping review examining the application of large language models in healthcare education and public health learning spaces. *PLOS*. 2026, 21, e0339594. <https://doi.org/10.1371/journal.pone.0339594>
20. Suárez-Álvarez, J.; He, Q.; Guenole, N.; D'Urso, D. Using Artificial Intelligence in Test Construction: A Practical Guide. *Psicothema*. 2026, 38, 1-12. <https://doi.org/10.70478/psicothema.2026.38.01>
21. Ng, J.Y. Prompt engineering for generative artificial intelligence chatbots in health research: A practical guide for traditional, complementary, and integrative medicine researchers. *IMR*. 2025, 14, 101222. <https://doi.org/10.1016/j.imr.2025.101222>
22. Shen, M.; Shen, Y.; Liu, F.; Jin, J. Prompts, privacy, and personalized learning: integrating AI into nursing education—a qualitative study. *BMC Nursing*. 2025, 24, 470. <https://doi.org/10.1186/s12912-025-03115-8>
23. O'Connor, S.K.; Miller, E.E.; Zweifel, A.R.; Schievelbein, D.M.; Parmar, A.R.; Amell, J.W. Use of artificial intelligence processing tools to evaluate qualitative data: Student researchers compared to faculty researchers. *CPTL*. 2025, 17, 102418. <https://doi.org/10.1016/j.cptl.2025.102418>
24. Dietrich, N.; Bradbury, N.C.; Loh, Ch. Prompt Engineering for Large Language Models in Interventional Radiology. *American Journal of Roentgenology*. 2025, 225, e2532956. <https://doi.org/10.2214/AJR.25.32956>
25. Mburu, T.K.; Rong, K.; McColley, C.J.; Werth, A. Methodological foundations for artificial intelligence-driven survey question generation. *JEE*. 2025, 114, e70012. <https://doi.org/10.1002/jee.70012>
26. Reason, T.; Klijn, S.; Rawlinson, W.; Benbow, E.; Langham, J.; Teitsson, S.; Johannesen, K.; Malcolm, B. Using Generative Artificial Intelligence in Health Economics and Outcomes Research: A Primer on Techniques and Breakthroughs. *PharmacoEconomics-Open*. 2025, 9, 501-517. <https://doi.org/10.1007/s41669-025-00580-4>
27. Taneski, V.; Karakatič, S.; Rek, P.; Jošt, G. Impact of Developer Queries on the Effectiveness of Conversational Large Language Models in Programming. *Appl. Sci*. 2025, 15, 6836. <https://doi.org/10.3390/app15126836>

28. Archambault, S.G.; Murph, N.L.; Ramachandran, S. Fostering AI Literacy in Undergraduates: A ChatGPT Workshop Case Study. *Library Trends*. 2025, 73, 443-475. <https://muse.jhu.edu/pub/1/article/968491>
29. Vilakati S. Prompt engineering for accurate statistical reasoning with large language models in medical research. *Front. Artif. Intell.* 2025, 8, 1658316. <https://doi.org/10.3389/frai.2025.1658316>
30. Stergiou, D. Using ChatGPT to support human resource management in the hotel industry. *WHATT*. 2025, 17, 602-615. <https://doi.org/10.1108/WHATT-03-2025-0088>
31. Patil, R.; Heston, T.F.; Bhuse, V. Prompt Engineering in Healthcare. *Electronics*. 2024, 13, 2961. <https://doi.org/10.3390/electronics13152961>
32. Cain, W. Prompting Change: Exploring Prompt Engineering in Large Language Model AI and Its Potential to Transform Education. *TechTrends*. 2024, 68, 47-57. <https://doi.org/10.1007/s11528-023-00896-0>
33. Maaz, S.; Palaganas, J.C; Palaganas, G.; Bajwa, M. A guide to prompt design: foundations and applications for healthcare simulationists. *FMED*. 2024, 11, 1504532. <https://doi.org/10.3389/fmed.2024.1504532>

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.