

Imperial College London
Department of Earth Science and Engineering
MSc in Applied Computational Science and Engineering

Independent Research Project
Final Report

A comparison of RANS and URANS turbulence models for flow past a circular cylinder in OpenFOAM

by
Yuyao Jiang

Email: yuyao.jiang22@imperial.ac.uk
GitHub username: [acse-yj522](#)
Repository: <https://github.com/ese-msc-2022/irp-yj522>

Supervisors:
Dr. James Percival
Dr. Stephan Kramer

September 2023

Abstract

Flow around a circular cylinder is focused, a canonical problem in fluid dynamics with implications for industries. Turbulence models, including Reynolds-Averaged Navier-Stokes (RANS) and Unsteady Reynolds-Averaged Navier-Stokes (URANS), are indispensable for simulating such flows. This paper aims to assess the performance of RANS and URANS models in capturing flow features around a circular cylinder using OpenFOAM. Starting by doing some simulations of fundamental cases, the study advances by designing refined meshes and optimizing simulation parameters. Rigorous comparisons with existing literature validate the simulations, followed by iterative adjustments of turbulence parameters. Comprehensive analyses, encompassing force coefficients and Strouhal numbers, reveal the strengths and limitations of RANS and URANS models. The drag lift coefficient solved by the PISO algorithm will tend towards a periodic oscillation, while those by SIMPLE or PIMPLE algorithms will grow to a fixed value after iterations. Transient solver PISO simulates well for unsteady flow at the Reynolds numbers 100 and 3900 in the $2D$ case. Another transient solver, PIMPLE, performs well for unsteady flow at the Reynolds number 100 and $3.5e5$ in the $3D$ case. In contrast, results prove that steady-state solver SIMPLE is unsuitable for handling steady flow at the Reynolds number 100 and $3.5e5$ compared to PIMPLE. This study contributes to understanding turbulence modeling for flow past a circular cylinder, shedding light on the capabilities and adaptability of RANS and URANS models.

1 Introduction

Turbulence is crucial to many critical natural phenomena and has become a topic of great interest and relevance in many fields of science and engineering [1]. Its practical applications across various industrial contexts extend to diverse areas such as automotive, aerospace, and marine engineering, where fluid flow and aerodynamics are crucial in achieving optimal performance and efficiency [2]. The flow past a circular cylinder is an essential benchmark problem in fluid dynamics, extensively studied due to its fundamental relevance and complex flow features [3]. Understanding the flow characteristics around a circular cylinder, such as vortex shedding and drag forces, provides insights into practical applications. One prominent application is in the aerospace industry, where the study of aerodynamics around aircraft components, such as wings and fuselages, helps optimize fuel efficiency and minimize drag during flight [4].

However, challenges continue to spur research efforts in various directions. Turbulence presents a significant challenge to researchers because it is both multiscale and highly nonlinear. Analytical solutions for even the simplest turbulent flows do not exist [5]. The turbulence model is essentially based on the Navier-Stokes equations and is difficult to solve directly. Researchers have recently adopted different turbulence models to approximate and simplify the Navier-Stokes equations to study turbulent flows further using the computational fluid dynamics approach [6].

Great works in this domain provide context for subsequent exploration of turbulence models. Schlichting [7] pioneered the investigation of flow past cylinders, followed by studies like Bearman and Obasaju, showcasing the wake's three-dimensional characteristics [8]. Additionally, the efforts of Norberg underscored the Reynolds number's influence on flow separation and vortex shedding [9]. In the context of turbulence modeling, Wilcox introduced the $k - \omega$ model [10], while Spalart and Allmaras proposed the one-equation turbulence model [11]. These models, widely employed in RANS simulations, form a foundation for turbulence modeling in various applications.

In recent years, there has been growing interest in comparing the performance of various tur-

bulence models and revealing their strengths and weaknesses through simulations. As Shibo Wang said, this study aims to investigate and evaluate the accuracy of three widely used turbulence models, URANS, SAS, and DES, for predicting the flow field around a typical high-speed train (HST) [12]. In addition, M. E. Young and A. Ooi concluded that using a three-dimensional URANS model results in only minor improvement over the two-dimensional URANS model for the flow around a circular cylinder at a Reynolds number of 3900. Significant gains can be made by using an LES model in bulk quantities and the wake structure and behaviour [13].

However, considering RANS and URANS models are representative and widely used, and few papers are comparing them for flow past a cylinder, this paper aims to contribute to understanding their efficacy in capturing flow features around a circular cylinder using OpenFOAM. All the computational simulations were conducted using OpenFOAM 9, a C++-based open-source computational fluid dynamics (CFD) software for the Linux platform. Its modular architecture, extensive libraries, and customizable solvers allow researchers and engineers to tailor the software to their needs and contribute to its ongoing development [14].

The study starts by running the basic tutorials of OpenFOAM, then designs a new mesh and adjusts the parameters like Courant number for convergence and relaxation factor for faster convergence of the simulation results. The simulation results were then compared with other literature to ensure the validity of the simulation. After finding errors in the results, the turbulence parameters like k , ω , ν , etc., continued to be adjusted. After confirming the error being minor, the grid is refined to see if the simulation results converge. Three solvers, steady-state solver SIMPLE for RANS transient solvers PISO and PIMPLE for URANS, are used for comparing the performance of two turbulence models. By conducting a comprehensive comparison and parametric analyses like force coefficients, strengths and limitations of RANS and URANS and the changing pattern of parameters are verified.

2 Methods

Steady flow, whose state is unchanged in time, and unsteady flow, whose velocity and pressure fields change with time, are two standard fluid states. Given this, steady flow will be studied with the RANS turbulence model and unsteady flow with URANS. The equations for the RANS and URANS models are as follows,

$$RANS : \frac{\partial(u_j u_i)}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial(2\nu S_{ij} + \tau_{ij})}{\partial x_j} \quad (1)$$

$$URANS : \frac{\partial u_i}{\partial t} + \frac{\partial(u_j u_i)}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{\partial(2\nu S_{ij} + \tau_{ij})}{\partial x_j} \quad (2)$$

where u_j or u_i is average velocity of the fluid in the j or i direction, x_j or x_i is spatial coordinate in the j or i direction, p is average pressure, ρ is average density, ν is kinematic viscosity, t is time, $S_{ij} = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$ is mean strain rate tensor, $\tau_{ij} = -\rho \overline{u'_i u'_j}$ is Reynolds stress tensor [15].

The RANS turbulence model is widely used for modeling the time-averaged behavior of turbulent flows in computational fluid dynamics. In this framework, the governing Navier-Stokes equations are averaged over time, so the equations describing the mean turbulent flow behavior and the turbulent fluctuations can be solved numerically. Additional turbulence closure terms are provided for the averaged equations to close the system, which employs the eddy viscosity concept to approximate turbulent stresses [16]. RANS models offer simplicity and computational efficiency but do not capture unsteady turbulence effects.

Limitations of RANS in predicting unsteady flow dynamics that change rapidly in time led to the development of URANS. URANS turbulence models extend the RANS approach by taking time variations into account, which corresponds to the difference between Equation 2 and Equation 1, an additional temporal derivative term $\frac{\partial u_i}{\partial t}$. URANS introduces the time rate of change of velocity (the unsteady term) alongside the spatial derivatives present in the RANS equation. URANS, incorporating unsteady effects, showcase enhanced capabilities in capturing flow unsteadiness, vortex shedding frequencies, and transient behaviors around bluff bodies. At the same time, it still accounts for turbulence's effects using the Reynolds averaging approach.

This paper will implement two turbulence modeling approaches, RANS and URANS, to simulate the flow past a circular cylinder. The comparison of their capabilities to capture both steady and unsteady flow features will provide insights into each model's respective strengths and limitations in predicting the complex flow behavior around the cylinder.

2.1 Computational domains

The two computational domains used in the simulation are demonstrated in Figure 1. In the case of $2D$, the length of the domain is $40D$, the width is $20D$, and the height is $1D$, where D denotes cylinder diameter. The domain extended $10D$ forward, above, and below the cylinder and $30D$ afterward. For the $3D$ case, the length of the domain is $14.5D$, the width is $10D$, and the height is $7D$, where D denotes cylinder diameter. The domain extended $3.5D$ forward, $5D$ above and below, and $11D$ after the cylinder.

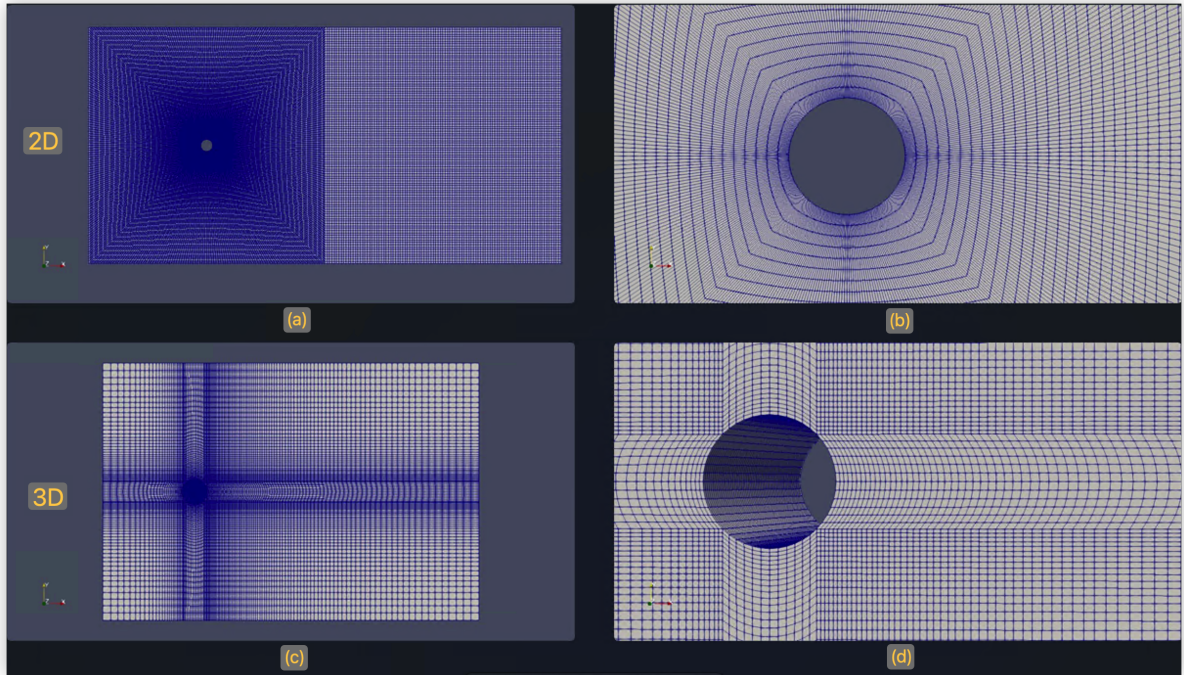


Figure 1: (a) Mesh of $2D$ computational domain; (b) Detail of mesh around cylinder in $2D$ computational domain; (c) Mesh of $3D$ computational domain; (d) Detail of mesh around cylinder in $3D$ computational domain.

2.2 Model and solvers selection

The $k - \epsilon$ and $k - \omega$ models are two turbulence models commonly used to simulate turbulent flow behaviors. The $k - \epsilon$ model focuses on predicting turbulence kinetic energy k and its

dissipation rate ϵ , emphasizing capturing turbulent mixing and diffusion processes accurately. On the other hand, the $k - \omega$ model emphasizes the near-wall behavior and turbulent boundary layers, allowing it to capture complex flow separation phenomena better.

This study uses the SST $k - \omega$ (Shear Stress Transport $k - \omega$) model, which combines the strengths of the $k - \epsilon$ model and $k - \omega$ model. It employs the $k - \omega$ formulation in the near-wall region to accurately predict near-wall turbulent behaviors and the $k - \epsilon$ formulation in the outer region of the boundary layer, capturing the turbulent behavior away from the wall [17]. The integrated approach delivers more accurate flow simulation than the other two models.

Most fluid dynamics solver applications in OpenFOAM use the pressure-implicit split-operator (PISO) or semi-implicit method for pressure-linked equations (SIMPLE) algorithms [18]. They solve equations of velocity and pressure through iterative procedures for incompressible fluids. PISO is a transient solver running with timesteps so that the turbulent fluctuations will be captured at each timestep. While SIMPLE is a steady-state solver, it aims to reach a converged solution in a set number of iterations, which is irrelevant with time. Due to the transient solver PISO needing to make a solution for each timestep, it takes much time to compute compared to SIMPLE, which makes only one solution.

PIMPLE combines the strengths of PISO and SIMPLE, which inherits the PISO's effectiveness for incompressible flows and the robustness of the SIMPLE, resulting in a versatile solver capable of simulating fluid flows across a spectrum of conditions. This study employs PISO for the 2D URANS turbulent model, SIMPLE for 3D RANS, and PIMPLE for 3D URANS.

2.3 Parameters calculation

2.3.1 Turbulence properties

In the SST $k - \omega$ model, the relevant turbulence parameters are Reynolds number Re , turbulent kinetic energy k , turbulent dissipation rate ω and their initial values need to be set using the following equations, where ρ is fluid density, U is flow velocity, D is cylinder diameter, μ is dynamic viscosity, $I = 8\%$ is turbulent intensity, l is turbulent length scale and $C_\mu = 0.09$ is turbulence eddy viscosity coefficient.

$$Re = \frac{\rho U D}{\mu} \quad (3)$$

$$k = \frac{3}{2}(UI)^2 \quad (4)$$

$$l = \frac{0.07D}{C_\mu^{3/4}} \quad (5)$$

$$\omega = \frac{k^{1/2}}{C_\mu l} \quad (6)$$

2.3.2 Convergence control

Courant number C is dimensionless, U is flow velocity, Δt is the time step, and Δx is the minimum grid length. Courant number affects the stability and accuracy of the simulation results, which typically requires less than 1 to avoid numerical instability, such as oscillations or even divergence of the solution. To get reliable results, adjusting the time step and grid resolution to ensure the Courant number remains below 1 is crucial.

$$C = \frac{U\Delta t}{\Delta x} \quad (7)$$

2.3.3 Validation

Drag coefficient C_d , lift coefficient C_l , Strouhal number St , and pressure coefficient C_p are essential for validating simulation results. Matching these parameters with empirical data indicates that the simulation accurately reproduces the complex flow behavior around the circular cylinder and captures the flow physics faithfully.

In the equations below, F_d and F_l are unit drag forces and unit lift forces, f is vortex shedding frequency, p is static pressure, and p_∞ , ρ_∞ , U_∞ are static pressure, density, velocity of free-stream.

$$C_d = \frac{F_d}{0.5\rho U^2 D} \quad (8)$$

$$C_l = \frac{\sqrt{\frac{1}{n}(F_{l_1}^2 + F_{l_2}^2 + \dots + F_{l_n}^2)}}{0.5\rho U^2 D} \quad (9)$$

$$St = \frac{fD}{U} \quad (10)$$

$$C_p = \frac{p - p_\infty}{0.5\rho_\infty U_\infty^2} \quad (11)$$

2.4 Grid refinement

Grid refinement aims to enhance the accuracy and reliability of numerical simulations. By refining the computational grid, finer spatial resolution is achieved, and this refined grid can represent the flow physics and prediction of flow quantities better and more accurately, such as pressure distribution, drag, and lift coefficients. The grid schemes used in this study are shown in Table 1. N_r denotes number of radial nodes, N_w denotes number of nodes in the wake, N_a denotes number of azimuthal nodes and N_z denotes the number of spanwise elements.

Mesh	N_{total}	N_r	N_w	N_a	N_z	CPUS	Total CPU Time (hrs)
2D URANS coarse	18000	60	120	240	-	1	2
2D URANS fine	50000	100	200	400	-	1	6
3D RANS coarse	679800	27	90	40	60	1	10
3D RANS fine	798000	27	90	80	60	1	15
3D URANS coarse	679800	27	90	40	60	1	27
3D URANS fine	798000	27	90	80	60	1	39

Table 1: Detail of the computational grids and computational time

3 Results

3.1 Drag and Lift coefficient

The drag and lift coefficients are dimensionless parameters used to quantify the resistance and lifting forces, respectively. The drag coefficient represents the resistance force opposing the motion of the cylinder through the fluid. On the other hand, the lift coefficient measures the upward force generated due to the pressure differences between the upper and lower surfaces of the object.

3.1.1 2D PISO (Re = 100, 3900)

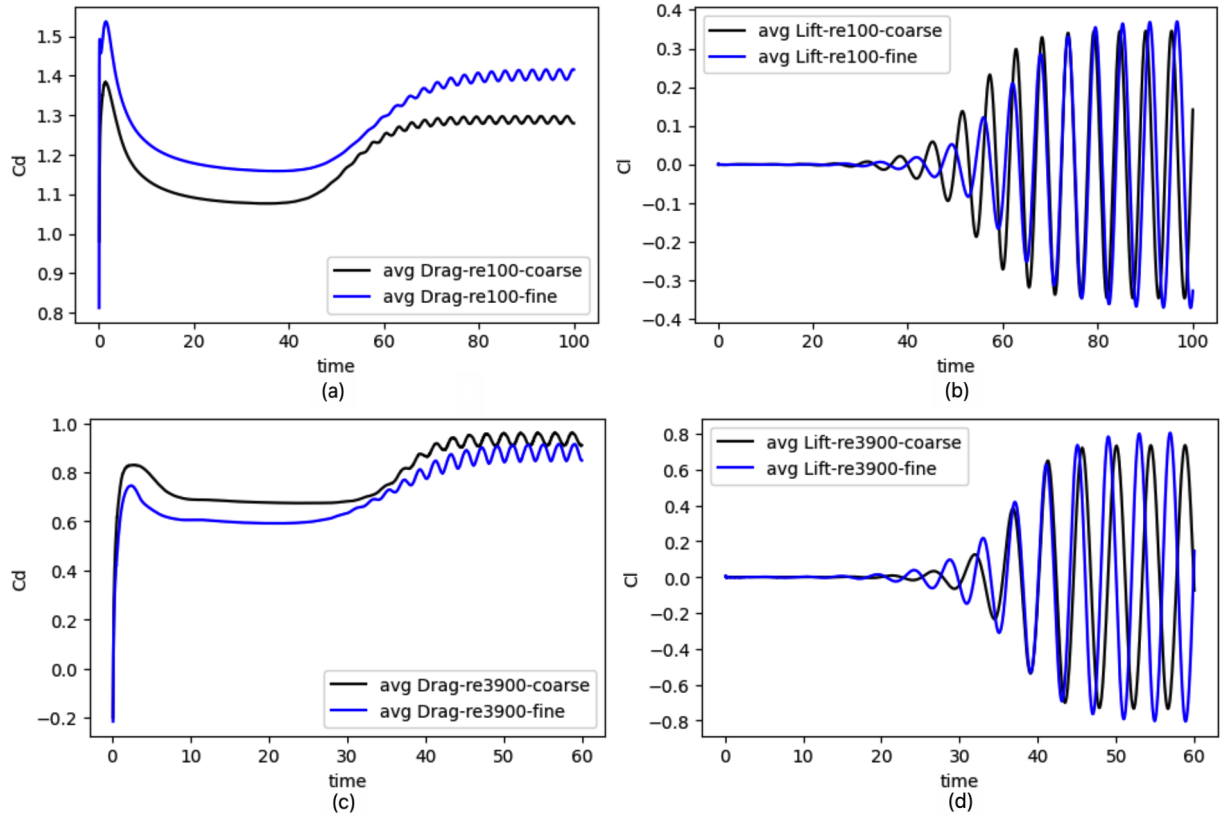


Figure 2: Comparison of drag and lift coefficients of cylinder solved by PISO on coarse and fine mesh (a) Drag coefficient at Re = 100; (b) Lift coefficient at Re = 100; (c) Drag coefficient at Re = 3900; (d) Lift coefficient at Re = 3900.

The results shown in Table 2 below are relatively valid because the error between it and those from C Liu, X Zheng, and CH Sung [19] is very small, less than 0.05.

Mesh	Re	N_{total}	C_d	C_l	St
2D URANS coarse	100	18000	1.2890.009	0.346	0.174
2D URANS fine	100	50000	1.4030.013	0.371	0.166
2D URANS coarse	3900	18000	0.9350.026	0.736	0.228
2D URANS fine	3900	50000	0.8800.034	0.805	0.217

Table 2: Force coefficients and Strouhal numbers in 2D cases

3.1.2 3D SIMPLE (Re = 100, 3.5e5)

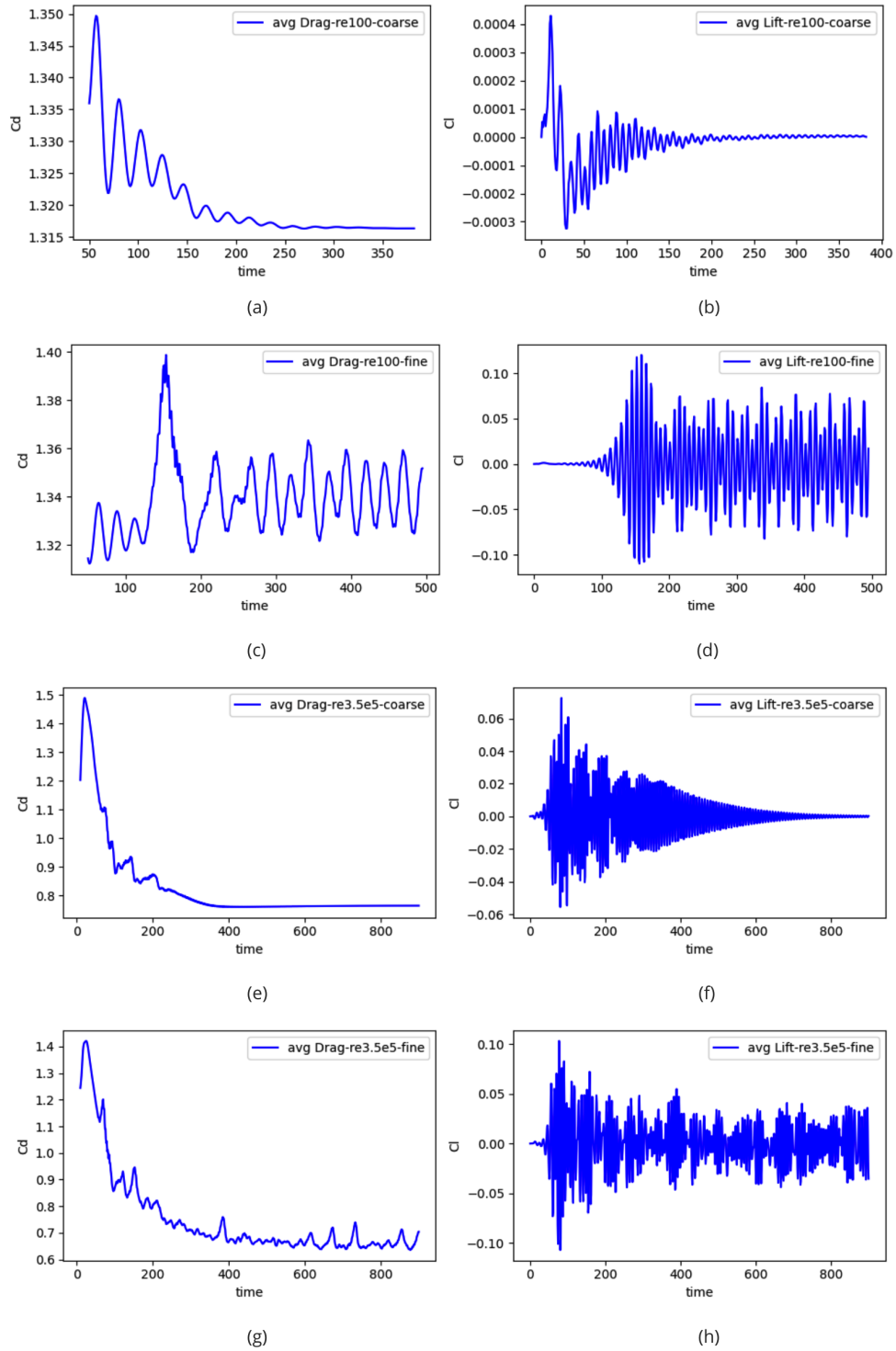


Figure 3: Drag and lift coefficients of cylinder solved by SIMPLE on coarse and fine mesh (a) Drag coefficient on coarse mesh at Re = 100; (b) Lift coefficient on coarse mesh at Re = 100; (c) Drag coefficient on fine mesh at Re = 100; (d) Lift coefficient on fine mesh at Re = 100; (e) Drag coefficient on coarse mesh at Re = 3.5e5; (f) Lift coefficient on coarse mesh at Re = 3.5e5; (g) Drag coefficient on fine mesh at Re = 3.5e5; (h) Lift coefficient on fine mesh at Re = 3.5e5.

3.1.3 3D PIMPLE (Re = 100, 3.5e5)

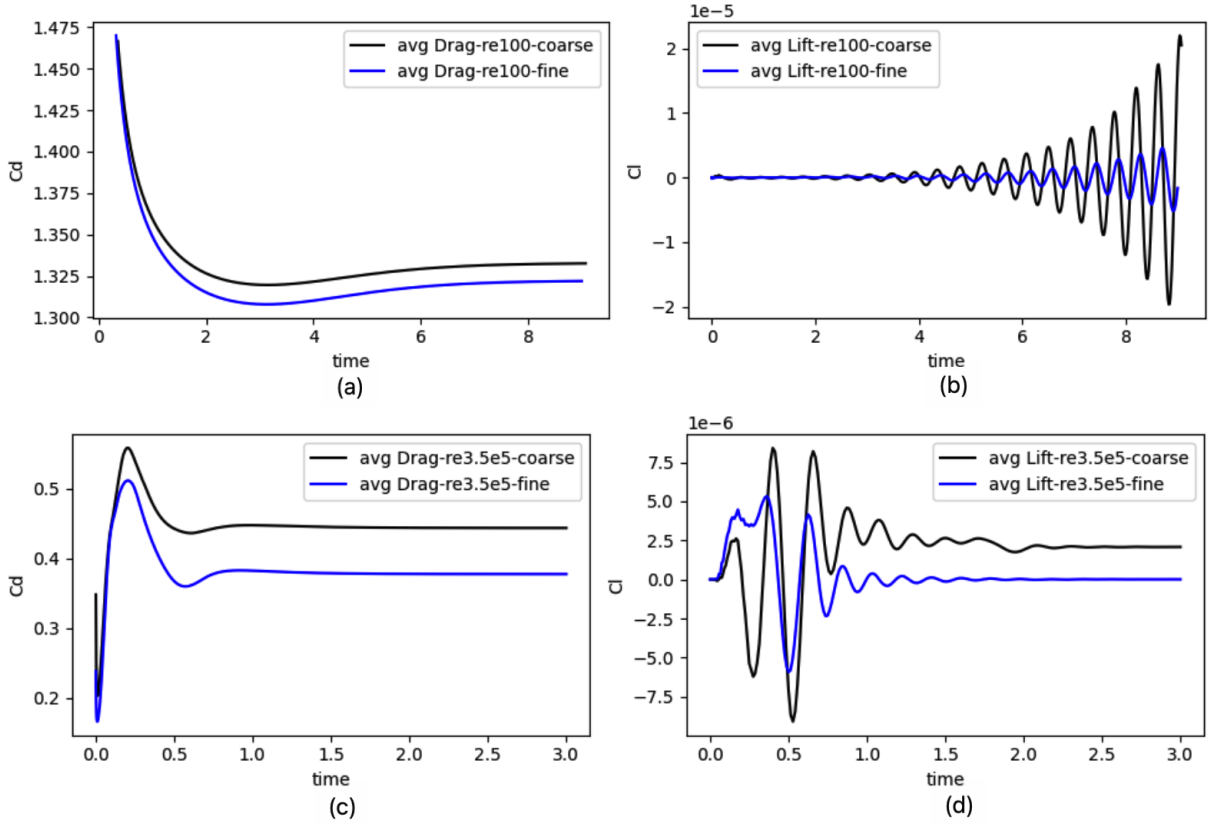


Figure 4: Comparison of drag and lift coefficients of cylinder solved by PIMPLE on coarse and fine mesh (a) Drag coefficient at $Re = 100$; (b) Lift coefficient at $Re = 100$; (c) Drag coefficient at $Re = 3.5e5$; (d) Lift coefficient at $Re = 3.5e5$.

In the $2D$ case using PISO at $Re = 100$ and 3900 , shown in Figure 2, throughout the iteration, both the drag coefficient and lift coefficient have entered a state of periodic oscillation, and its amplitude remains almost the same. In the $3D$ case using PIMPLE at $Re = 100$ and $3.5e5$, which is shown in Figure 4, after a period of iteration, both the drag coefficient and lift coefficient will gradually converge to a fixed value with the lift coefficient ends up being close to zero. Although the amplitude of the oscillation of the lift coefficient is increasing in Figure 4(b), it has a correct trend that will eventually converge to a fixed value after more iterations. When the grid refinement is performed, the drag and lift coefficients converge in both cases. However, it does not apply to the $3D$ case using SIMPLE at $Re = 100$ and 3900 shown in Figure 3. The drag and lift coefficients can converge on a coarse mesh, but they will continue to oscillate on a fine mesh.

Regarding the reason for that, SIMPLE is often used to solve for the steady flow whose Reynolds number is lower than 50. Although in a coarse mesh, SIMPLE converges to a fixed value. Once the mesh is sufficiently fine, SIMPLE can not handle the case at a high Reynolds number, where PIMPLE is effective. In addition, both SIMPLE and PIMPLE converge to some fixed value rather than oscillating periodically due to their respective algorithms comprising two steps. Initially, a PISO-like algorithm is employed to iteratively address the pressure correction equations and attain a preliminary pressure field. Subsequently, a SIMPLE-like algorithm is adopted, and iterations are performed to simultaneously rectify the velocity and pressure fields to achieve satisfaction of the velocity and continuity equations.

From Figures 2 and 4, as the Reynolds number increases, the inertial effects of the fluid become more prominent relative to its viscosity, leading to a greater tendency for the fluid to exhibit regional flow separation behind the cylinder. Additionally, the non-steady effects may accelerate the formation and evolution of vortices around the cylinder. In addition, cases at high Reynolds numbers need less time to reach the state of periodic oscillation or a fixed value.

Figures 2 and 3 illustrate the difference between PISO and SIMPLE. They both reach the same result on average, but SIMPLE as a steady-state solver cannot show the dynamic formation and evolution of eddies, which can only yield an averaged solution. SIMPLE can find out the drag, lift, and pressure coefficient. Nevertheless, a transient solver like PISO is needed if what is happening in the turbulent flow wants to be seen.

3.2 Pressure coefficient

The pressure coefficient is a dimensionless index of the pressure of a fluid at a point relative to the static pressure of the free fluid, and the distribution of C_p can reveal the pressure change of the fluid at different locations on the cylinder.

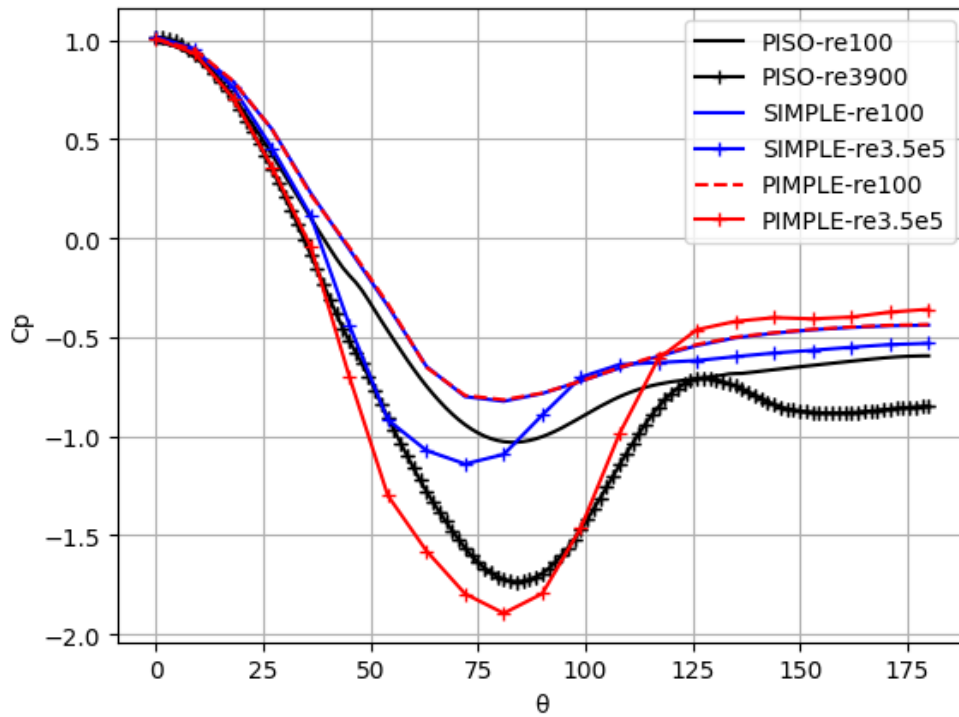


Figure 5: Comparison of the distribution of pressure coefficient around the cylinder

From Figure 5, it can be found that SIMPLE and PIMPLE have the same pressure distribution at $Re = 100$. At high Reynolds number $3.5e5$, the simulation results of SIMPLE are worse than those of PIMPLE, with significant errors occurring. The initial value of C_p is usually 1, representing the static pressure in the direction of the flow directly in front of the cylinder. As the fluid travels around the cylinder, C_p exhibits a small negative value in front of it. It then rapidly decreases and reaches a minimum value behind the cylinder represented by the lowest point of C_p . Due to fluid separation on the backside, the distribution of C_p results in a negative pressure region behind the cylinder.

4 Discussion and conclusions

This paper explores the difference between RANS and URANS during the flow simulation past the circular cylinder. The critical difference between RANS and URANS lies in their treatment of flow unsteadiness. RANS models focus on time-averaged behavior and may need to capture transient or unsteady phenomena better. URANS, on the other hand, incorporates the temporal derivative term, enabling it to account for both time-averaged and unsteady flow characteristics, making it more suitable for simulations involving unsteady turbulence effects.

Then, different computational domains and grid refinement are set up for the $2D$ and $3D$ cases, taking into account that in the $2D$ domain, the fluid needs a longer distance ($10D$) before reaching the cylinder, and distance ($3.5D$) in the $3D$ domain results in a larger calculated drag coefficient. SST $k - \omega$ model is used for fluid simulation in $2D$ PISO, $3D$ SIMPLE, and $3D$ PIMPLE scenarios. By comparing the metrics like drag coefficient, lift coefficient, Strouhal number, and pressure coefficient, the validity of the simulation results is verified, while some conclusions are obtained. The drag and lift coefficient solved by PISO will tend towards a periodic oscillation. In contrast, those solved by SIMPLE or PIMPLE will tend to a fixed value at some appropriate Reynolds number. Re-simulation with a finer mesh is needed to verify that results converge. A high Reynolds number means rapid evolution of vortices and less time to reach the state of periodic oscillation or some fixed value.

There exist some difficulties in completing the implementation. For example, waiting for parameters to converge takes long, like the lift coefficient C_l in Figure 4(b). In addition, trying to analyse a model at different Reynolds numbers means that many attempts are needed, like adjusting parameters k , ω , ν to see the simulation performance in OpenFOAM. Then, the simulation results show the strengths and limitations of different solvers. As a transient solver, PISO simulates well for unsteady flow at the Reynolds numbers 100 and 3900 in the $2D$ case. Another transient solver, PIMPLE, performs well for unsteady flow at the Reynolds number 100 and $3.5e5$ in the $3D$ case. It adjusts the time step size automatically based on the local flow conditions, so there is no need for manual tuning. However, its implicit-explicit coupling and adaptive time stepping increase the computational cost [20], especially for large-scale simulations in Figure 4, which still needs to perform more iterations to see if it eventually converges. Simulation results prove that the steady-state solver SIMPLE is unsuitable for handling steady flow at the Reynolds number 100 and $3.5e5$ in the $3D$ case compared to PIMPLE.

In the future, the $3D$ URANS turbulence model will be simulated with the PISO algorithm, which can not be implemented currently because of insufficient computing resources. The simulation results of $3D$ PISO will be compared with those of $2D$ PISO presented above, and observe whether the curves for various force coefficients have a trend similar to the $2D$ case at the corresponding Reynolds number. Moreover, comparisons will be made between $3D$ PISO simulations and $3D$ PIMPLE simulations above to see which algorithm takes longer to reach convergence, what aspects of PIMPLE differ from PISO when PIMPLE combines SIMPLE based on PISO, as well as explaining these differences from the perspective of the SIMPLE algorithm principle. Because these three algorithms are related and different, comparing and analysing the simulation phenomena from principles makes the exploration process very interesting.

Acknowledgements

The author would like to thank Drs. Stephan Kramer and Rhodri Nelson from Imperial College London, for listening to his presentation and offering valuable advice. Finally, the author would like to thank Dr. James Percival from Imperial College London for patient guidance and helpful discussion at weekly meetings.

References

- [1] Wolfgang Rodi. *Turbulence models and their application in hydraulics*. Routledge, 2017.
- [2] Momchilo M Zdravkovich. *Flow around circular cylinders: Volume 2: Applications*, volume 2. Oxford university press, 1997.
- [3] A Grucelski and J Pozorski. Lattice boltzmann simulations of flow past a circular cylinder and in simple porous media. *Computers & Fluids*, 71:406–416, 2013.
- [4] Adel Abbas-Bayoumi and Klaus Becker. An industrial view on numerical simulation for aircraft aerodynamic design. *Journal of Mathematics in Industry*, 1:1–14, 2011.
- [5] Parviz Moin and Krishnan Mahesh. Direct numerical simulation: a tool in turbulence research. *Annual review of fluid mechanics*, 30(1):539–578, 1998.
- [6] Desmond C Lim, Hussain H Al-Kayiem, and Jundika C Kurnia. Comparison of different turbulence models in pipe flow of various reynolds numbers. In *AIP Conference Proceedings*, volume 2035. AIP Publishing, 2018.
- [7] H Schlichting. Hauptaufsätze. turbulenz bei wärmeschichtung. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 15(6):313–338, 1935.
- [8] PW Bearman and ED Obasaju. An experimental study of pressure fluctuations on fixed and oscillating square-section cylinders. *Journal of Fluid Mechanics*, 119:297–321, 1982.
- [9] Christoffer Norberg. Fluctuating lift on a circular cylinder: review and new measurements. *Journal of Fluids and Structures*, 17(1):57–96, 2003.
- [10] David C Wilcox. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA journal*, 26(11):1299–1310, 1988.
- [11] Philippe Spalart and Steven Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th aerospace sciences meeting and exhibit*, page 439, 1992.
- [12] Shibo Wang, James R Bell, David Burton, Astrid H Herbst, John Sheridan, and Mark C Thompson. The performance of different turbulence models (urans, sas and des) for predicting high-speed train slipstream. *Journal of Wind Engineering and Industrial Aerodynamics*, 165:46–57, 2017.
- [13] ME Young and A Ooi. Comparative assessment of les and urans for flow over a cylinder at a reynolds number of 3900. 2007.
- [14] Hrvoje Jasak, Aleksandar Jemcov, Zeljko Tukovic, et al. Openfoam: A c++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics*, volume 1000, pages 1–20, 2007.
- [15] Daniel M Israel. The myth of urans. *Journal of Turbulence*, page 2225140, 2023.
- [16] Haralambos Marmanis. Analogy between the navier–stokes equations and maxwell’s equations: Application to turbulence. *Physics of fluids*, 10(6):1428–1437, 1998.
- [17] Florian R Menter. Improved two-equation k-omega turbulence models for aerodynamic flows. Technical report, 1992.
- [18] Taeyoung Han. Computational analysis of three-dimensional turbulent flow around a bluff body in ground proximity. *AIAA journal*, 27(9):1213–1219, 1989.

- [19] C Liu, X Zheng, and CH Sung. Preconditioned multigrid methods for unsteady incompressible flows. *Journal of Computational physics*, 139(1):35–57, 1998.
- [20] Mathieu Olivier and Olivier Paré-Lambert. Strong fluid–solid interactions with segregated cfd solvers. *International Journal of Numerical Methods for Heat & Fluid Flow*, 29(7):2237–2252, 2019.