Oral presentation | Data science and AI

Data science and AI-III

Thu. Jul 18, 2024 2:00 PM - 4:00 PM Room C

[11-C-03] Resolving weak flow interactions on an array of cylinders with PINN

*Adhika Satyadharma[1], Ming-Jyh Chern[1], Heng-Chuan Kan[2], Wen-Yao Chiang[1] （1. National Taiwan University of Science and Technology, 2. National Center for High-performance Computing, National Applied Research Laboratories）

Keywords: Physics Informed Neural Network, Machine Learning, Data Assimilation, Computational Fluid Dynamics.

**Twelfth International Conference on
Computational Fluid Dynamics (ICCFD12),
Kobe, Japan, July 14-19, 2024**

Resolving weak flow interaction on an array of cylinders with PINN

A. Satyadharma*, M.J. Chern*, H.C. Kan**, and W.Y. Chiang*
Corresponding author: adhika.s33@gmail.com
* National Taiwan University Science and Technology, Taipei, Taiwan.
** National Center for High-performance Computing, National Applied Research Laboratories, Tainan, Taiwan.

Abstract: Simulation of a flow over an array of objects can be difficult to perform due to the high mesh requirement near each object. We intend to reduce this high cost by utilizing both data and physics within a physics informed neural network (PINN). Our study aims to show this potential using a flow past an array of cylinders at Re = 40 as the study case. Using data from a flow past single cylinder, PINN would learn and adapt it using its physics based loss as it interacts with other cylinders in an array. The result shows that PINN is indeed capable of modeling the flow interaction in such flow field, though it performs better at weaker interactions. As this study both utilizes collocation points and data, both of them holds an importance to resolve the flow field. Collocation points are still required to properly resolve the flow around the cylinder, though their amount is significantly minimized compared to not using any data. Collocation points are also required between the cylinders allowing the PINN to model the interaction between them. For the data, the utilization of the main flow variables does lead to the correct flow field and the utilization of its first derivative does lead to an accurate drag coefficient. The adjustments of the weight for every of these variables may be a crucial factor in this PINN as it prevents it from following any incorrect information too much.

Keywords: Computational Fluid Dynamics, Physics Informed Neural Network, Machine Learning, Fluid Structure Interaction, Near Wall Flow Reconstruction.

1 Introduction

Optimization of wind turbine farm layout can be quite a challenging task. Both the terrain [1, 2, 3, 4, 5] and wake [6, 7, 8] effect are predominant, but it can be difficult to model. Currently, simplified analytical approaches are still preferred [8, 9, 10, 11] as computational fluid dynamics (CFD) models are still too expensive [12]. This is more so for blade resolved simulations [13, 14, 15, 16] which requires a fine mesh near each blade surface.

In this research, we would like to lower the load of such simulations by utilizing physics informed neural network (PINN)[17]. By utilizing a physical based loss, PINN is encouraged to give a physical based result, allowing it to simulate many physical systems [18, 19, 20], including fluid flow [21, 22, 23]. While as a pure physics based system PINN still numerous issues [24, 25, 26, 27], this method can also utilize data which not only offers some way to resolve the issues but also offers many unique applications. Some example of this includes flow interpolation [28], solving inverse problems [29], optimization [30] and discovering physical equations [31].

In our study, we focus on utilizing data as a way to lower the computational cost of PINN. By using data, the amount of collocation points (CP) in PINN can be significantly reduced [32], leading to a faster training process [33]. Unlike other data driven neural network, as PINN also utilizes physics, the data requirement is way less [34, 35, 36]. Also, for the same reason, PINN does not require a totally accurate data as the physics can detect this and fix it [32].

For the wind turbine application, the PINN would learn the flow field from a single wind turbine and then adapt it as it interact with each other in the array. However, as this method has not been proven yet, we choose to demonstrate it in a much more simpler case first which is a flow past an array of 3 cylinders at Re = 40. A flow past single cylinder would be initially conducted with CFD and then the PINN would utilize the data around this cylinder to predict the whole array flow field. While one main goal of this research is to demonstrate the capability of such method, we also want to investigate the effect of CP near the object. Both CFD and PINN typically requires a high resolution calculation near the object due to boundary layer. However, with the utilization of data, this requirement may be mitigated. Due to its potentially significant effect, this is going to be the second objective of our research.

2 Methodology

The overall methodology of this research is shown in figure (1). First we conduct a few CFD simulations which includes the flow past singular cylinder to be replicated and the flow past the array of cylinders for verification purposes. Next, some part of the single cylinder simulation is going to be used as a reference for the PINN which tries to resolve the array flow, including the interactions between each cylinders. For the verification, the result of the PINN is going to be directly compared to the CFD data.

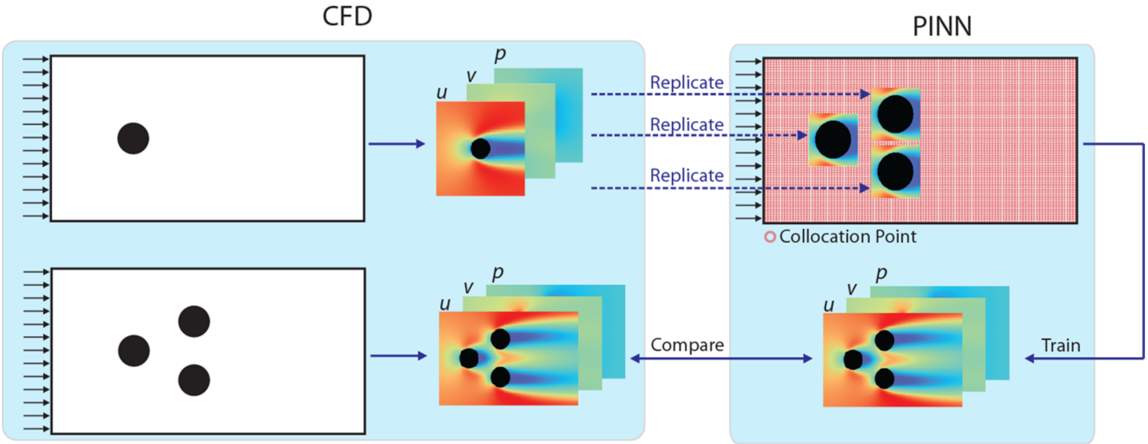


Figure 1: The overall methodology of this research

The details regarding the array, simulation domain and boundary conditions can can be seen in figure (2) with each cylinder diameter set to 1. These conditions also applies for the single cylinder simulation which only utilize the front cylinder. The distance of front and back cylinders (R) is set to $R = \{7.5, 12.5\}$ and the size of the replicated data (s) is set to $s = \{1.2, 2, 6\}$.

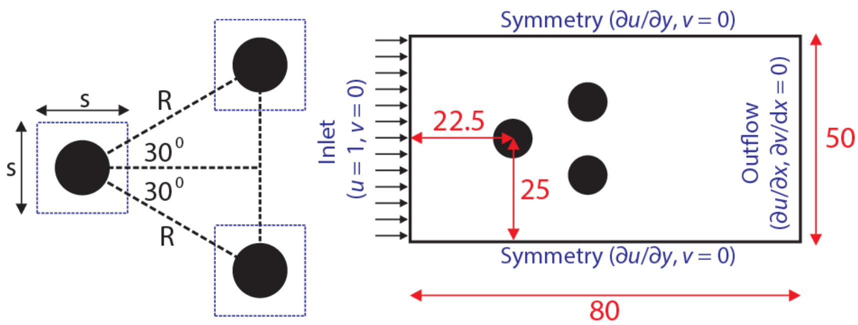


Figure 2: The array configuration and simulation domain.

The CFD simulations are conducted using the commercial software Fluent that solves equation (1) which is the two dimensional incompressible steady state Navier-Stokes. The mesh distribution, can be seen in figure (3) with the number of cells of 38k, 125k and 114k for the single cylinder, array with $R = 7.5$ and $R = 12.5$ respectively. The simulation was conducted until machine accuracy to ensure a high level precision. Regarding the result, the array simulations shows a symmetrical pattern between the top and bottom cylinder, evident by the drag coefficient (C_D) that is presented in table (1). We also calculated the flow difference inside the $s \times s$ domain (s domain) between the single cylinder and the array which is an information that the PINN requires. This error calculation is conducted using equation (2) and the result is shown in table (2).

Table 1: The drag coefficient (C_D) from the CFD data.

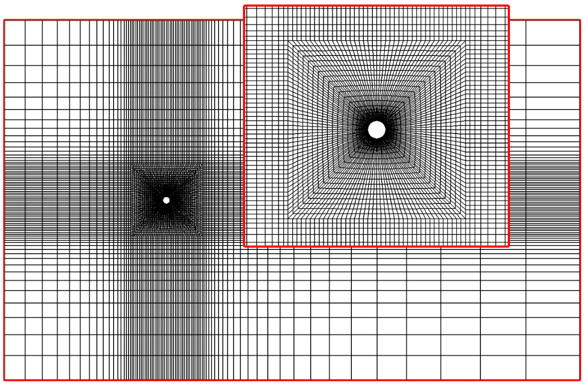
	Front	Back - Top	Back-Bottom
Single Cylinder	1.53		
Array $R = 7.5$	1.49	1.64	1.64
Array $R = 12.5$	1.52	1.63	1.63

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{1a}$$

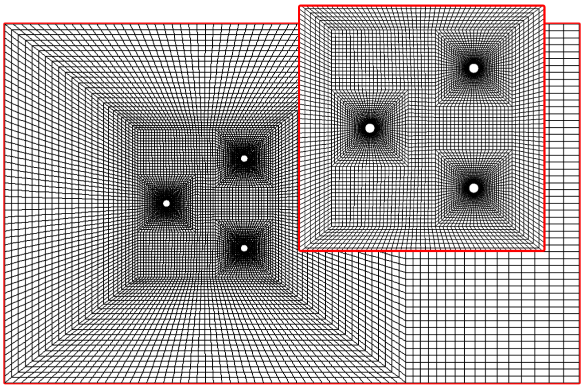
$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x} - \frac{\mu}{\rho}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = 0 \tag{1b}$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial y} - \frac{\mu}{\rho}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) = 0 \tag{1c}$$

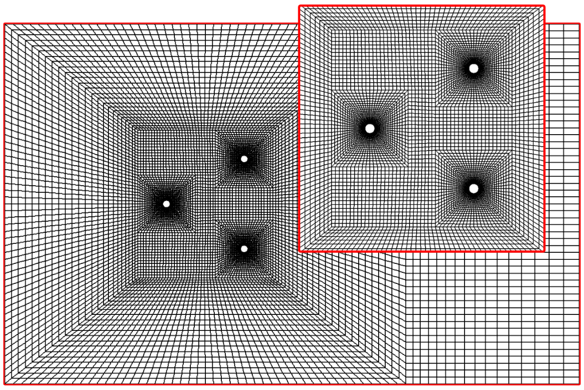
$$\epsilon_u = \frac{\sqrt{\sum (u - u_{\mathrm{Ref}})^2}}{\sqrt{\sum u_{\mathrm{Ref}}^2}} \tag{2}$$



(a) Single cylinder



(b) Array with $R = 7.5$



(c) Array with $R = 12.5$

Figure 3: The CFD simulations mesh

The PINN used to calculate this flow interaction consist of a feed forward neural network with 6 layers each with 64 neurons. The inputs of the network are (x, y) and the outputs are (u, v, p). The loss function mainly consists of 3 terms, the boundary loss ($\mathcal{L}_B$), the governing equation loss ($\mathcal{L}_F$) and a modified data loss ($\mathcal{L}_M$) which is based on our previous research that allows the incorporation of incorrect data into the PINN. Each of these loss are defined in equation (3) - (6), with the $\mathcal{D}, \mathcal{B}, \mathcal{F}$ operators are respectively the data, boundary and governing equation operator. These operators would give either the PINN values ($\mathcal{D}_\mathcal{P}, \mathcal{B}_\mathcal{P}, \mathcal{F}_\mathcal{P}$) or the CFD values (reference values) ($\mathcal{D}_{\mathrm{Ref}}, \mathcal{B}_{\mathrm{Ref}}$) at each respective points, with $\mathcal{F}_\mathcal{P}$ being defined as the residual of the 2 dimensional steady state Navier-Stokes equation. The variable N and ω are the number of points and weight for each respective loss. As for the L_1 norm, these sum each loss for each respective variables such as $\{u, v, p\}$ or over all governing equation.

Table 2: The error comparison of the flow around each cylinder (ϵ^s) towards the single cylinder.

R	s		u	v	p	$\partial u/\partial x$	$\partial u/\partial y$	$\partial v/\partial x$	$\partial v/\partial y$	$\partial p/\partial x$	$\partial p/\partial y$
7.5	1.2	Front	2.24%	2.21%	9.91%	2.93%	2.52%	2.92%	2.91%	4.82%	4.87%
7.5	1.2	Back	6.19%	6.85%	21.19%	7.18%	5.80%	7.71%	7.17%	10.74%	10.32%
7.5	2	Front	2.01%	2.08%	10.86%	2.98%	2.55%	2.77%	2.81%	4.72%	4.67%
7.5	2	Back	5.33%	7.00%	22.77%	7.19%	5.73%	7.32%	7.11%	10.11%	10.53%
7.5	6	Front	2.56%	6.91%	23.03%	4.92%	3.43%	3.77%	4.37%	6.91%	6.23%
7.5	6	Back	6.96%	13.73%	40.78%	9.23%	24.46%	7.57%	8.58%	12.50%	11.70%
12.5	1.2	Front	0.72%	0.74%	3.42%	1.47%	0.99%	1.47%	1.43%	3.23%	3.48%
12.5	1.2	Back	4.87%	4.92%	17.43%	5.35%	4.94%	5.62%	5.34%	8.60%	7.84%
12.5	2	Front	0.61%	0.67%	3.76%	1.64%	1.04%	1.39%	1.35%	3.03%	3.18%
12.5	2	Back	4.38%	4.51%	18.66%	5.29%	4.78%	5.36%	5.20%	8.21%	8.04%
12.5	6	Front	0.69%	1.29%	7.14%	1.87%	1.14%	1.46%	1.43%	3.12%	3.18%
12.5	6	Back	4.02%	5.76%	31.46%	5.33%	4.96%	5.27%	5.18%	8.16%	8.08%

$$\mathcal{L}_B = \left\| \frac{1}{N_B} \sum_{1}^{N_B} \omega_B |\mathcal{B}_{\mathcal{P}} - \mathcal{B}_{\text{Ref}}|^2 \right\|_1 \tag{3}$$

$$\mathcal{L}_F = \left\| \frac{1}{N_F} \sum_{1}^{N_F} \omega_F |\mathcal{F}_{\mathcal{P}}|^2 \right\|_1 \tag{4}$$

$$\begin{aligned}\mathcal{L}_M =& \frac{2\mathcal{L}_{\mathcal{D}}}{1+\exp\left(-4.2\log_{10}(\mathcal{L}_{\mathcal{D}}/\beta_1^2)+2.1\right)}+\\ & \frac{\mathcal{L}_{\mathcal{D}}}{1+\exp(-4.2\log_{10}(\mathcal{L}_F)+4.2\beta_2)}\end{aligned} \tag{5}$$

$$\mathcal{L}_D = \left\| \frac{1}{N_D} \sum_{i=1}^{N_D} \omega_D \Omega_i |\mathcal{D}_{\mathcal{P}} - \mathcal{D}_{\text{Ref}}|^2 \right\|_1 \tag{6}$$

For the total loss, this is defined in equation (7) which consist of 6 components. There are 2 types of boundary loss, the Dirichlet boundary ($\mathcal{L}_B^D$) and Neumann boundary ($\mathcal{L}_B^N$) with a total $N_B = 2380$. There are also 2 types of modified data loss, the main flow data ($\mathcal{L}_M^u$) and the first derivative data ($\mathcal{L}_M^{\partial u/\partial x}$) with the later being used for a better approximation of the C_D. The total N_D is $N_D = 3.9$K, 7.5K, 14.1K for $R = 1.2$, 2, 6 respectively. The governing equation is also split into 2 parts, the inner and outer loss which are based on the zones in figure (4). The outer loss is only based on the freestream zone which has a CP distribution that becomes more dense near the cylinder. As for the inner loss, this is based on all of the other zones each having a different distribution. The interaction zone has a uniform distribution but when it is near the cylinder it would transition to become denser. The near cylinder zone would also have a distribution that is dense towards the cylinder, but unlike the other zones that has fixed amount of CP, this zone would have a varying amount of CP to test the effect of reducing the CP near the cylinder. An example of this CP distribution can be seen in figure (5) with the detailed amount provided in table (3). However, as each s has a different near cylinder area, the CP would later be quantified with ϕ_F which is the CP density per unit area.

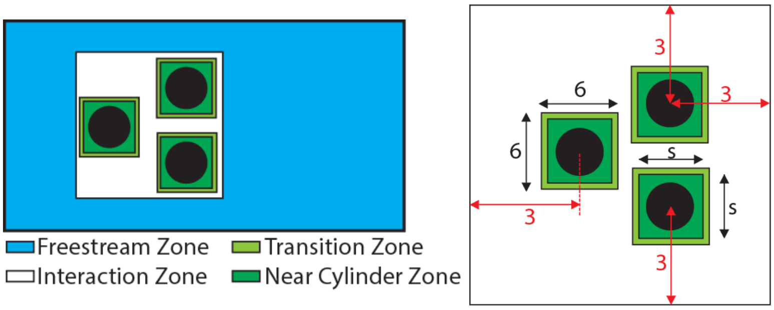


Figure 4: The collocation points zone separation.

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_B^D + \mathcal{L}_B^N + \mathcal{L}_M^u + \mathcal{L}_M^{\partial u/\partial x} + \mathcal{L}_F^{\text{out}} + \mathcal{L}_F^{\text{in}} \tag{7}$$

There are three special settings in our PINN. First is the governing equation weight which is defined in equation (8) with r being the minimum distance to any of the three cylinders. This equation would focus the residual near the cylinders, ensuring the propagation of the boundary information towards the cylinders. The second one is β_1 which is the tolerance for the incorrect data. This value is set following equation (9) with $\bar{\epsilon}^s$ is the average error across u, v, p from the flow field near all of the cylinders. Both formula in equation (9) are empirical formula based on our initial test. Regarding β_2, we set it at a constant value of 3. The third special setting is the weight for each variable in equation (6). This weight is set to normalize every component for the result in table (2). The formulation of this weight is shown in equation (10) with the L_1 norm being the summation of all the main flow variables or its first derivative and this value is being summed over all cylinders.

Table 3: The amount of CP on each zone

		N_F			
R	s	Freestream	Interaction	Transition	Near Cylinder
7.5	1.2	8532	950	3372	$0 \leq N_F \leq 372$
7.5	2	8532	950	2772	$0 \leq N_F \leq 972$
7.5	6	8532	950	144	$0 \leq N_F \leq 3600$
12.5	1.2	7538	3300	3372	$0 \leq N_F \leq 372$
12.5	2	7538	3300	2722	$0 \leq N_F \leq 972$
12.5	6	7538	3300	144	$0 \leq N_F \leq 3600$

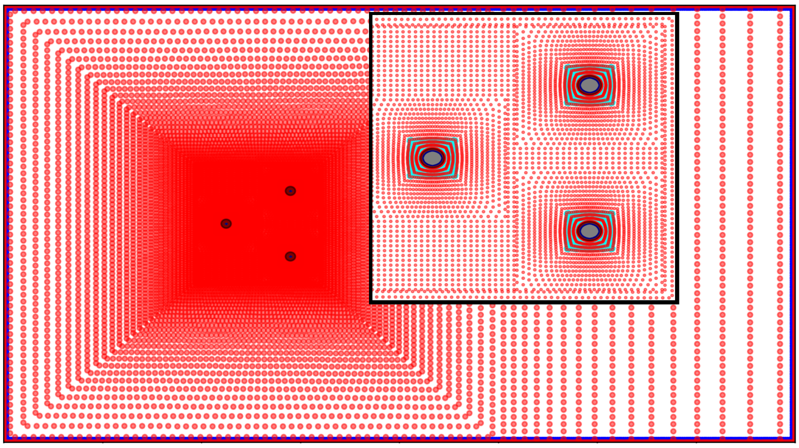


Figure 5: The collocation points distribution on $R = 7.5$, $s = 2$. Dark blue points are the boundary points, light blue points are the data points and red are the collocation points.

$$\omega_F = \frac{200r}{1 + \exp(25r - 15)} \tag{8}$$

$$\beta_1^u = \exp\left(1.6\exp(\bar{\epsilon}^s) - 1.63\right) - 1 \tag{9a}$$

$$\beta_1^{\partial u/\partial x} = 0.6\beta_1^u \tag{9b}$$

$$\Omega_u = \frac{1}{9}\frac{\sum_{\text{Cyl}=1}^{3} \|\epsilon\|_1}{\epsilon_u} \tag{10a}$$

$$\Omega_{\partial u/\partial x} = \frac{1}{18}\frac{\sum_{\text{Cyl}=1}^{3} \|\epsilon\|_1}{\epsilon_{\partial u/\partial x}} \tag{10b}$$

The training is conducted 2 times each using the Adam method for the first 10^4 iterations and the L-BFGS-B method afterwards. The first training only utilize the replicated data and the collocation points at the freestream zone which is then refined using the collocation points from the whole domain for the second training. This 2 time training process was done to ensure a more stable and faster training, especially due to the variation of collocation points near the cylinder. Some of the weights for both training's are set the same such as $\omega_B = 1$ for both boundary loss, $\omega_D^u = 1$ and $\omega_D^{\partial u/\partial x} = 0.01$. As for the different settings, the first training utilize $\omega_F = 100$ and $\mathcal{L}_F^{\text{out}}$ for the $\mathcal{L}_F$ in equation (5). As for the second training, $\omega_F = 1000$ and it uses $\mathcal{L}_F^{\text{in}}$ instead.

To quantify the error of the PINN, we use three different parameters. The first one is $\overline{\epsilon}^G$ which is the error in the whole domain with the $\overline{\epsilon}$ described in equation (11). The second one is $\overline{\epsilon}^I$ which is the error in the inner zones which includes the interaction, transition and near cylinder zone. Both of these calculations would be done on every data point in respective area. The last parameter is the E_D which is the average drag error from all cylinders. This is mathematically described in equations (12) & (13) with $u_\infty = 1$, $A = 1$, n being the normal direction and S is the cylinder surface.

$$\overline{\epsilon} = (\epsilon_u + \epsilon_v + \epsilon_p)/3 \tag{11}$$

$$E_\text{D} = \frac{1}{3}\sum_{\text{Cyl}=1}^{3} \frac{|C_{D,\mathcal{P}} - C_{D,\text{Ref}}|}{C_{D,\text{Ref}}} \tag{12}$$

$$C_D = \frac{1}{0.5\rho u_\infty^2 A}\int \left[\mu\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)n_x + \mu\left(\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x}\right)n_y - pn_x\right] dS \tag{13}$$

3 Results & Discussion

While the PINN did reference the flow past a single cylinder for all three cylinders, as seen in the examples of figures (6) and (7), the PINN managed accurately fix this. This capability is due to the modified data loss. As shown in figure (8), the modified data loss make the effect of the data less significant than the governing equation loss, allowing the PINN to prioritize the governing equation instead.

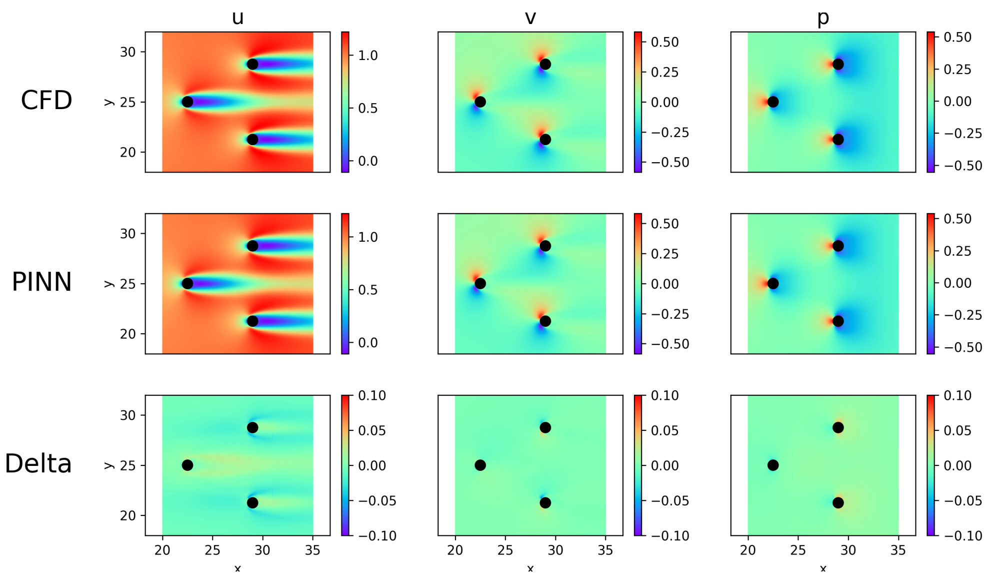


Figure 6: The flow comparison of $R = 7.5$ with $s = 1.2$ and $\phi_F^s = 36.66$

The importance of the data is shown more clearly in table (4). By using data, a PINN with $\phi_F^s = 0$ is not only more accurate than $\phi_F^s = 34$ but also simulates faster which is 4.3×10^4s compared to 6.1×10^4s. The higher accuracy can be attributed due to the constraining effect of the data loss. While the PINN does not exactly follow the data, it does frequently prefers a highly deviating solution. With the presence of the data loss, such tendency can be prevented, allowing the PINN to favor a more accurate solution. As for the faster simulation time, this is caused by two reasons. The first one is due the data loss that does not require additional calculation time from the auto differentiation. The second reason is the amount of iterations which is only 20.7K for $\phi_F^s = 0$ and 43K for $\phi_F^s = 34$. This difference is due to stabilization of the flow near the cylinder which typically requires more iteration with higher CP.

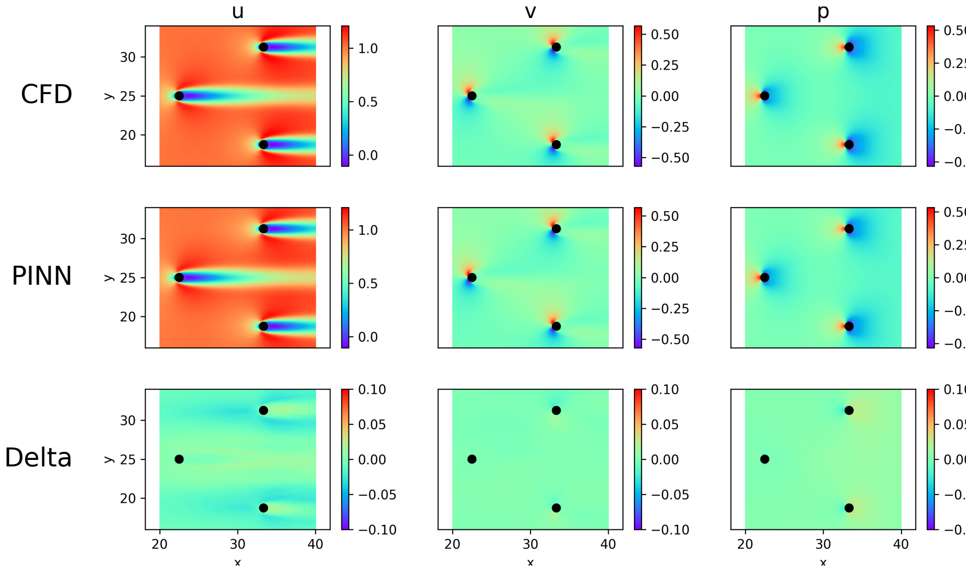


Figure 7: The flow comparison of $R = 12.5$ with $s = 6$ and $\phi_F^s = 34$

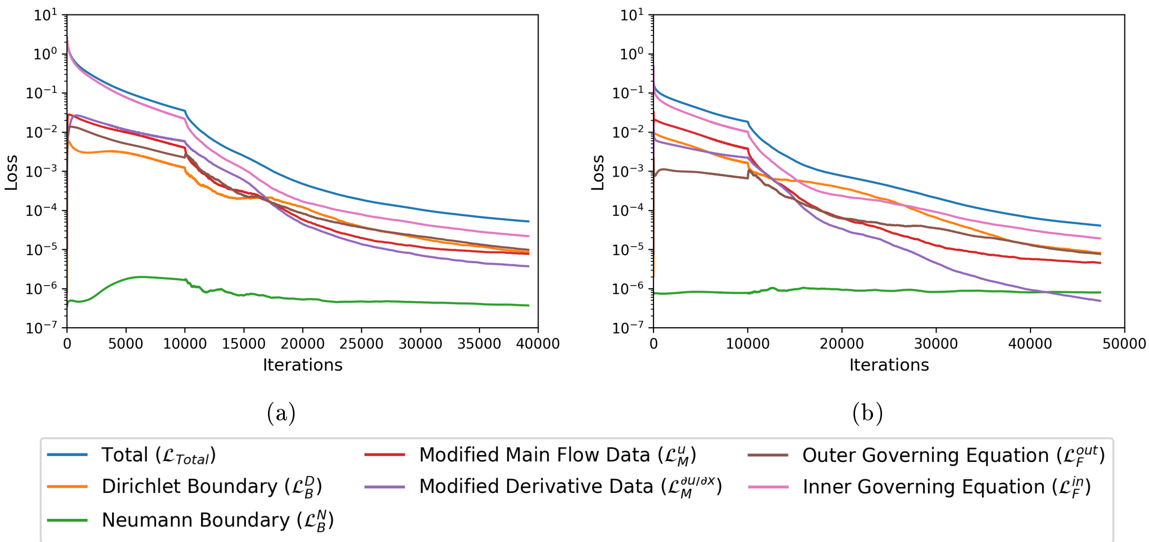


Figure 8: The loss curve for (a) figure (6) & (b) figure (7)

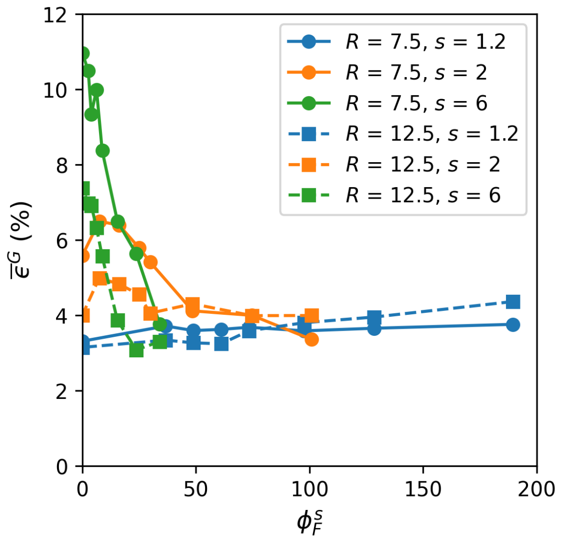
The usage of both main flow variables and first derivative data is also important, especially at lower CP. The main flow variables are responsible to make the flow field more accurate. As for the first derivatives, these variables helps correct the C_D which is primarily calculated using derivative values. In lower CP, with the lack of CP, they are highly dependent on the data to get an accurate result.

To fix the data, the PINN does require CP. There are two main functions of CP in this study. The first one is to transfer the information from the boundaries, especially the outer boundaries. As for the second one is to detect any local mistake in the near cylinder zone where the data is located.

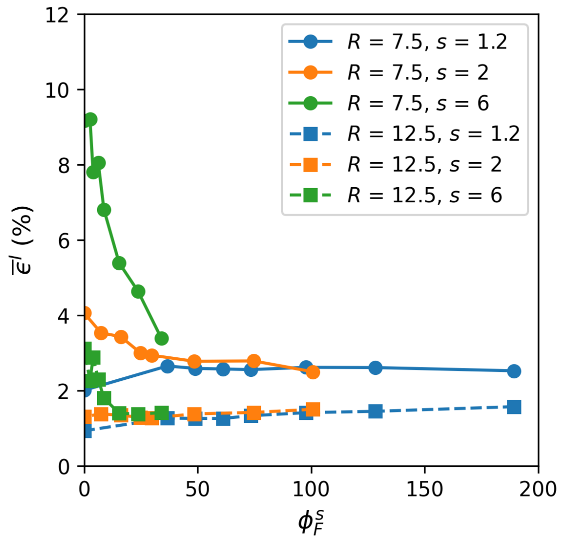
Within this study, a special weight function (Eq. (8)) was used to ensure that the information from the outer boundaries is transmitted well to the cylinders. This is done by prioritizing the residuals near the boundary which resulted in an almost no error transmission between the boundary and cylinders, though there may still be some error in between or around each cylinder. Compared to that, the usage of no weight function ($\omega_F = 1$) usually leads to a significantly worse result which is clearly seen in table (4). In such condition, the error are spread out in the domain, allowing a high error between the boundaries and cylinders. This would make the cylinders experience the wrong flow conditions, thus a high error.

Table 4: A comparison of using various settings at $R = 12.5, s = 6$ with $\phi_F^s = 0$ and $\phi_F^s = 34$

	$\phi_F^s = 0$ ($N_F = 0$)			$\phi_F^s = 34$ ($N_F = 3600$)		
Settings	$\overline{\epsilon}^G$	$\overline{\epsilon}^I$	E_D	$\overline{\epsilon}^G$	$\overline{\epsilon}^I$	E_D
Current Setting	7.39%	3.12%	1.50%	3.31%	1.42%	3.82%
Data = u, v, p	7.23%	2.94%	7.86%	3.67%	1.57%	4.53%
Data = $\partial u/\partial x, ..., \partial p/\partial y$	17.3%	19.2%	15.7%	14.1%	10.6%	3.06%
No Data	50.2%	64.8%	38.9%	11.3%	9.57%	2.98%
$\omega_F = 1$	7.45%	3.11%	2.13%	7.24%	3.51%	5.53%
No Data, $\omega_F = 1$	52.3%	66.6%	39.1%	42.5%	55.6%	38.8%



(a)



(b)

Figure 9: (a) The whole domain flow error ($\overline{\epsilon}^G$), (b) the flow error inside the interaction zone ($\overline{\epsilon}^I$)

For the second effect of CP, this is quantitatively shown in figure (9). In general, the PINN resolve $R = 12.5$ better than $R = 7.5$. The lower error seems to be caused by the weaker flow interaction in $R = 12.5$ which makes it easier for the PINN to solve.

There are several observable trends in figure (9). First is that a higher ϕ_F^s does tends to give a more accurate flow field. This trend occurs at $s = 2$ and $s = 6$ in which with more CP, the flow is being corrected at more places, thus more accurate.

For the second trend, an increase in ϕ_F^s does not changes the error. This occurs for $s = 1.2$, which we think is due to the lack of CP. For $s = 1.2$ the maximum CP in the near cylinder zone compared the whole inner zone (interaction, transition and near cylinder zone) is only about 8%. This is in contrast to $s = 2$ which has a maximum percentage of 21% and $s = 6$ which has 77%. The lack of CP means that it is going to have less effect, thus the error does not change.

Another potential reason that the error does not change is that increasing the CP in the near cylinder zone is not enough to make the flow field better. This is visualized more clearly in figure (10) where the residual plot is presented. At a lower $\phi_F^s = 0$, the residual is totally concentrated in the near cylinder zone where there are no CP to correct the flow. However at a higher ϕ_F^s, the high residual spreads out to both the transition and interaction zone. This would hamper the interaction modeling of the PINN leading to a not so accurate result. With the CP at both of these zones being unchangeable in this study, this may limit how accurate the PINN can be, indicated by the constant, non-decreasing error.

The third trend is that at $\phi_F^s = 0$, especially at lower s, the value of ϵ can decrease. This condition can happen as the usage of some, but insufficient CP, tends to create a more confused state for the PINN. In such condition, as seen in figure (10), the residual fluctuates more as it only being minimized at a very few locations. In contrast, without any CP, there is less confusion, thus a more accurate result. This confused state is more prominent at lower s as the distance between the CP at the transition zone towards the near cylinder zone is closer, leading to the fluctuating residual trend. On the other hand, at a larger s, with the larger distance between CP, the residual is going to be smoother.

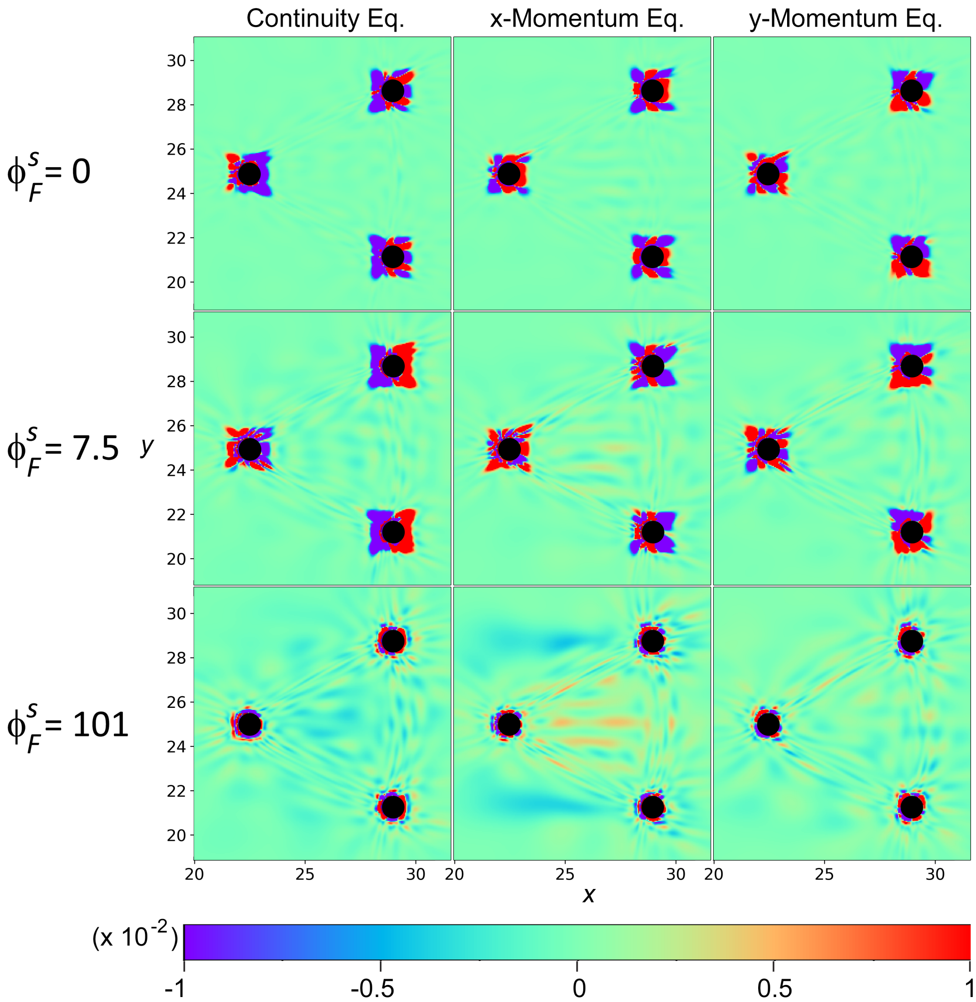
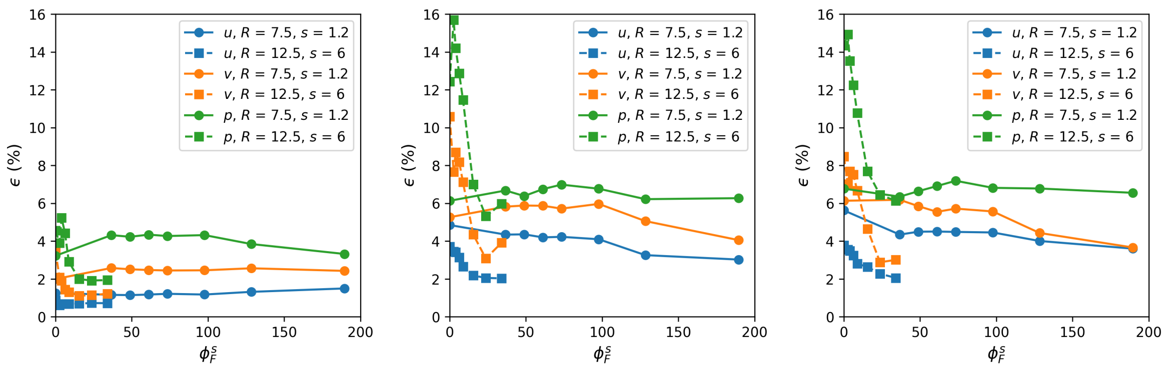


Figure 10: The residuals on $R = 7.5$ and $s = 1.2$

Regarding the accuracy at $\phi_F^s = 0$, even with no CP near the cylinder the result can still be accurate enough if s is quite small. The boundary loss does ensure that the flow on the surface of each cylinder is correct. This information is then propagated through the CP, in which due to the closer CP at lower s, it propagates better in such condition. A such, even when there are no CP in the near cylinder zone, the interaction between object is still being modeled by the PINN leading to the accurate result.



(a) Front cylinder (b) Back - top cylinder (c) Back - bottom cylinder

Figure 11: The error of each variable on the near cylinder zone of each cylinder for $R = 7.5, s = 1.2$ and $R = 12.5, s = 6$.

A more detailed investigation about ϵ^s reveals that the PINN seems to have more difficulty in solving the back cylinders and also the pressure. This can be seen in more detail within figure (11) where the error of each variable near each cylinder are presented. This difficulty can be attributed to the difference between the reference data and correct flow which is already presented in table (2). With a higher difference, the PINN cant just follow the reference data and instead it has to put more effort to resolve the correct flow field.

The value of Ω seems very important to solve this difficulty. This value can be understood as how much the PINN should follow the reference data. With a higher Ω the PINN would trust that variable more, making it more similar to the reference data. Due to the error values in table (2), a uniform Ω ($\Omega = 1$), would lead to the PINN to trust the pressure data more. This would resulted in a higher ϵ_p^s which could be twice as high compared to the our current normalized Ω, as shown in table (5).

Table 5: A comparison of the Ω effect on $R = 7.5, s = 1.2$ with $\phi_F^s = 189$

	Current Ω			Uniform Ω ($\Omega = 1$)		
Location	ϵ_u^s	ϵ_v^s	ϵ_p^s	ϵ_u^s	ϵ_v^s	ϵ_p^s
Front	1.50%	2.43%	3.32%	1.23%	1.79%	8.16% ($\times$ 2.46)
Back - top	3.03%	4.06%	6.27%	4.82%	5.24%	11.49% ($\times$ 1.83)
Back - bottom	3.61%	3.67%	6.56%	5.78%	6.40%	12.32% ($\times$ 1.88)

There are two current issues with Ω in which the first one is its formulation. As ϵ_p^s seems to be still the dominant error, the current formulation of Ω seems to fail to completely normalize the error from each variable. The second issues is that currently, a prior knowledge of the correct flow field is required to set the value of Ω. While the PINN does not require an exact value for these weights, approximation or educated guess regarding these values still requires a knowledge or expectation of the correct flow field. Without this, in order to get an accurate result, the value of β_1 should be set higher, disregarding the data as a whole even more. However this would potentially increase the required CP to get an accurate result which also increases the computational cost.

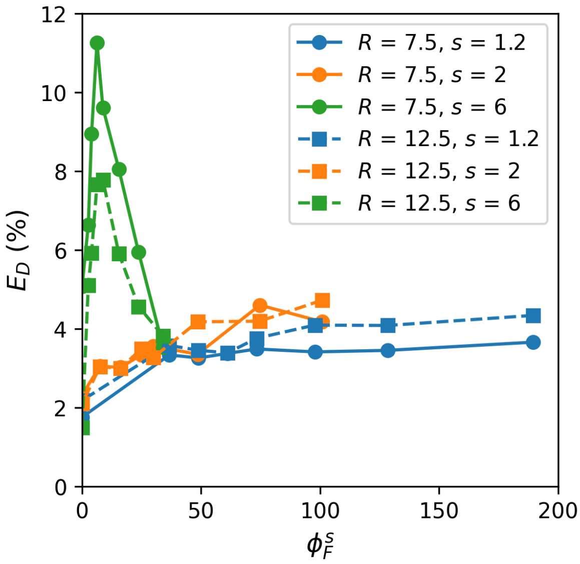
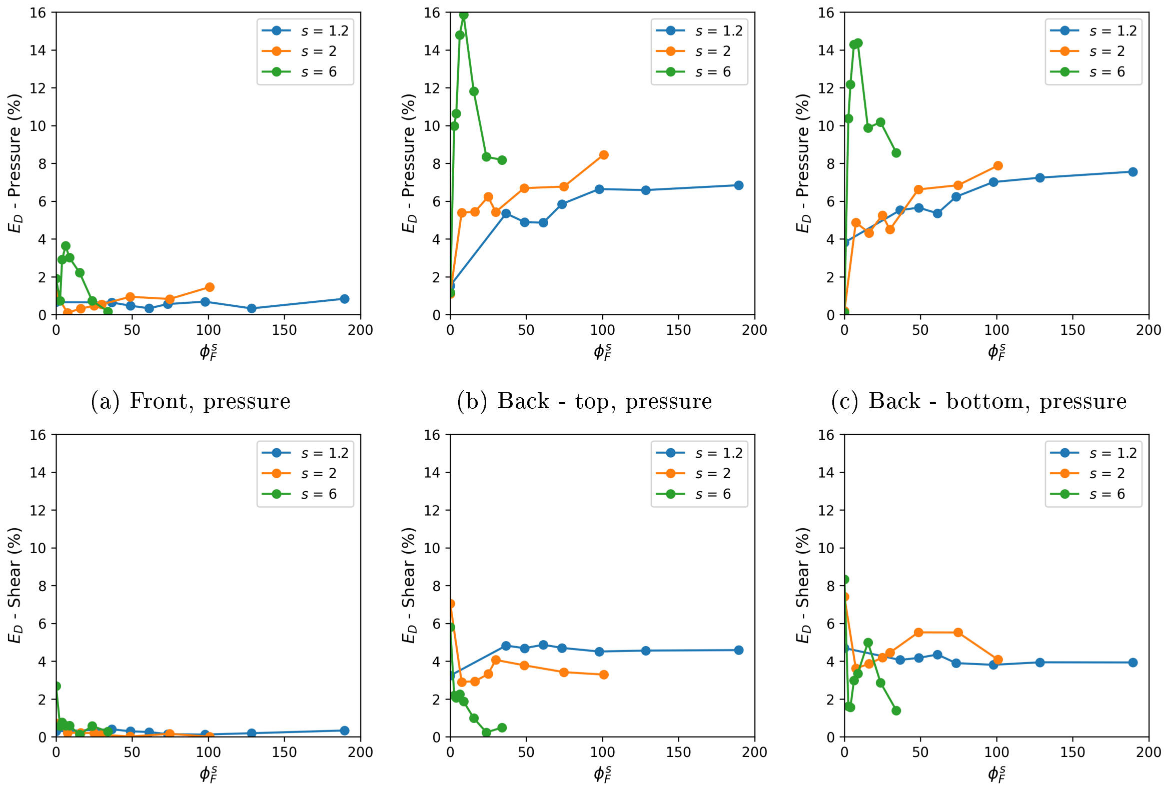


Figure 12: The C_D error

For E_D, this result is presented in figure (12). For both $R = 7.5$ and $R = 12.5$, E_D have the same trend for every s. While at lower s the E_D for $R = 12.5$ seems slightly higher, this trend would flip at higher s. In order to better understand these trends, a more detailed investigation of E_D was conducted by separating the drag force into its pressure and shear components. This result is presented in figure (13) where it could clearly be seen that the pressure drag error is always be the more dominant one, especially at higher s and at the back cylinders.

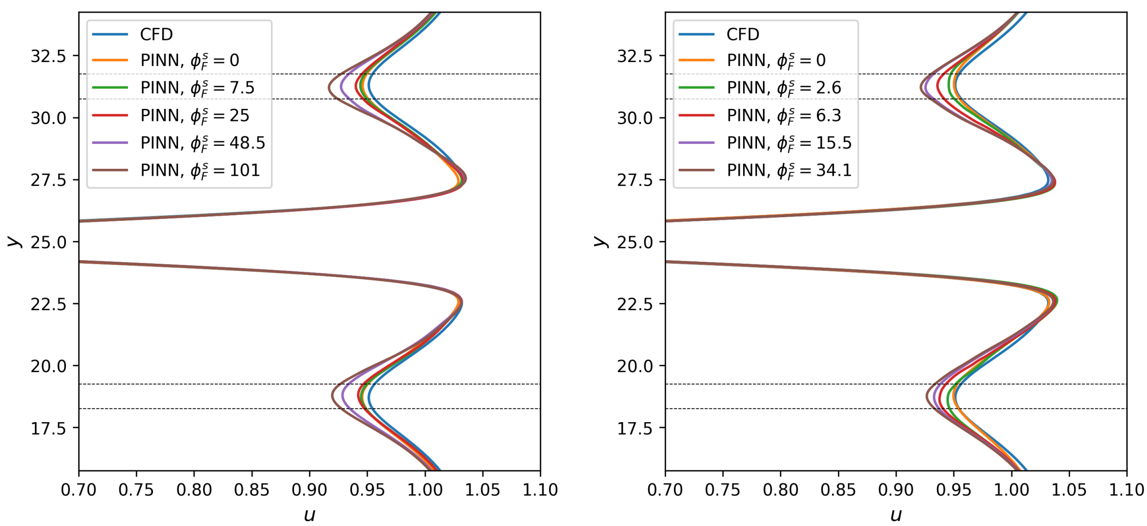


(a) Front, pressure (b) Back - top, pressure (c) Back - bottom, pressure

(d) Front, shear (e) Back - top, shear (f) Back - bottom, shear

Figure 13: The pressure drag and shear drag error on $R = 12.5$

For $s = 1.2$ and $s = 2$, our current best explanation of why the pressure drag error kept increasing is due to the residual distribution. As seen in figure (10), at higher ϕ_F^s there is going to be a high residual in front of the cylinder. This residual would affect the strength of the flow experienced by the back cylinder. As further proof, figure (14) shows the u-velocity at $x = 30.3$ which is 3 units in front of the back cylinders. A significant velocity deviation can be seen, especially at higher ϕ_F^s which would then affect the pressure distribution on the back cylinders.



(a) $s = 2$ (b) $s = 6$

Figure 14: The u-velocity profile on $R = 12.5$, located at 3 units in front of the back cylinders ($x = 30.3$). The dashed line indicates the location of the back cylinders.

Regarding $\phi_F^s = 0$ which gives the more accurate result, this is due to C_D being an integral quantity and there is no information propagation error anywhere else in the domain except near the cylinder. As presented in figure (15), the pressure distribution along the back - top cylinder shows that at $\phi_F^s = 0$, the pressure is not the most accurate. Due to being an integral quantity, the positive and negative pressure does balance out and this act of balancing is better at lower ϕ_F^s, due to the more accurate velocity.

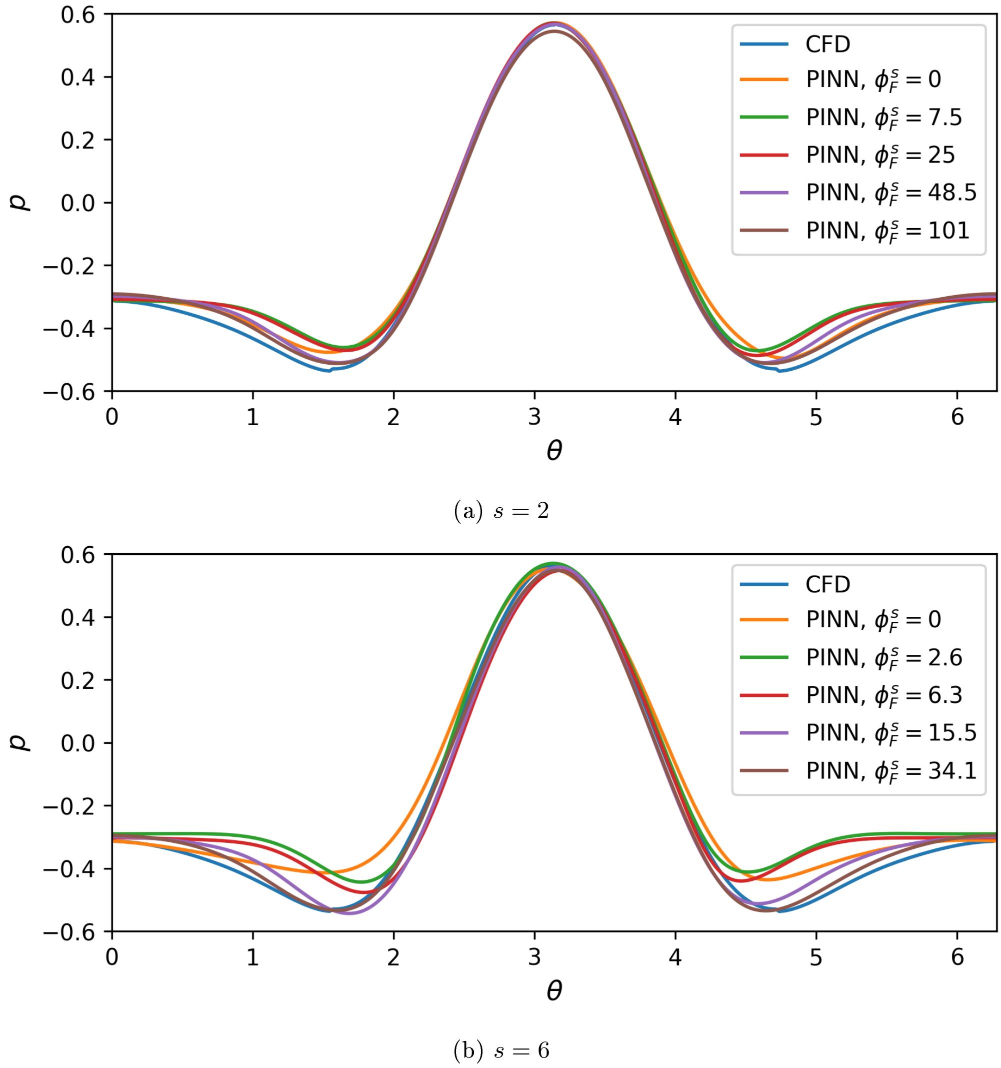


Figure 15: The pressure profile of $R = 12.5$, on the back - top cylinder.

For $s = 6$, all of the same problems as the lower s occurs as well. The differences is that the CP inside the near cylinder zone at $s = 6$ has a more prominent role, unlike the lower s which is still affected by the CP outside this zone. Once again, insufficient CP would only make the PINN confused, thus the increase in error. At a certain point ($\phi_F^s \geq 6.3$), this would then instead improve the flow field, until it later on would not have any effect. It should be noted that while an increase in ϕ_F^s does mean that it could solve the flow around the cylinder better, it still does not solve the high residual in front of these back cylinders. For that, more CP should be added in the interaction zone instead.

Overall, even as the PINN only reference the flow over a single cylinder, the PINN did manage to resolve the correct flow field. The utilization of the data allows the PINN to evade the high collocation point requirement at high gradients [37] which typically occurs due to boundary layers. This makes the PINN capable of running with significantly less collocation points and even improving its accuracy compared to not using any data. While there is still some imperfections especially on the pressure of the back cylinders, the average errors are still within 5%. Obtaining this values though requires a lot of setting in the PINN, especially on the weights. Without it, the PINN may end up with a significant increase in its errors.

4 Conclusion & Future Works

This study demonstrated the use of PINN to solve a flow over an array based on the data of the flow past a single object. The result shows that using such data, PINN is indeed capable of resolving such flow field with good enough accuracy ($\bar{\epsilon}^G, \bar{\epsilon}^I, E_D \leq 5\%$), more so at weaker flow interaction. While CP are still required in the area of the data, the amount of points are significantly less compared to the usage of no data. Sufficient CP are also required where there is some interaction between the objects. Neglecting this could lead to some error in the flow propagation which may have more effect on the drag force error.

Our study also found that for our simulations there are a lot of technical settings that has to be done in order to ensure a good result. These settings includes assigning an individual weight for all of types of data, utilizing both the main flow variable and its derivative, approximating the tolerance towards the incorrect data and using a special weight function for the collocation points. While our study may achieve a good enough result, these settings may not be the best setting and it could be different for different cases. More study is still required to determine the best settings for all of these parameters.

Considering that this model is developed for wind turbine simulations, there are several other things that needs to be upgraded and tested from the current model. One of them is about turbulence. While it is indeed possible to add the RANS (Reynolds Averaged Navier-Stokes) equation, the issue lies on the thin boundary layer which means that sufficient collocation points in a thin area is still required. Another issue lies on modeling the wind turbine blade motion. We prefer to use the multiple reference frame model, but we have not tested of utilizing such model in PINN. Lastly, the challenge is to find an efficient way to simulate such cases while increasing the dimension, both from two dimension to three dimensions and from steady to unsteady.

Acknowledgments

The authors gratefully acknowledge funding support from the National Science and Technology Council (grant no. NSTC 112-2221-E-011-075-MY3). We would also like to thank National Center for High-Performance Computing (NCHC), Taiwan, for providing computing resources and storage.

References

[1] Francesco Castellani, Davide Astolfi, Massimiliano Burlando, and Ludovico Terzi. Numerical modelling for wind farm operational assessment in complex terrain. *Journal of Wind Engineering and Industrial Aerodynamics*, 147:320–329, December 2015.

[2] Francesco Castellani, Davide Astolfi, Matteo Mana, Emanuele Piccioni, Matteo Becchetti, and Ludovico Terzi. Investigation of terrain and wake effects on the performance of wind farms in complex terrain using numerical and experimental data. *Wind Energy*, 20(7):1277–1289, July 2017.

[3] Davide Astolfi, Francesco Castellani, and Ludovico Terzi. A Study of Wind Turbine Wakes in Complex Terrain Through RANS Simulation and SCADA Data. *Journal of Solar Energy Engineering*, 140(3):031001, June 2018.

[4] Avinaash Murali and R.G. Rajagopalan. Numerical simulation of multiple interacting wind turbines on a complex terrain. *Journal of Wind Engineering and Industrial Aerodynamics*, 162:57–72, March 2017.

[5] Alexandros Makridis and John Chick. Validation of a CFD model of wind turbine wakes with terrain effects. *Journal of Wind Engineering and Industrial Aerodynamics*, 123:12–29, December 2013.

[6] R. J. Barthelmie, K. Hansen, S. T. Frandsen, O. Rathmann, J. G. Schepers, W. Schlez, J. Phillips, K. Rados, A. Zervos, E. S. Politis, and P. K. Chaviaropoulos. Modelling and measuring flow and wind turbine wakes in large wind farms offshore. *Wind Energy*, 12(5):431–444, July 2009.

[7] N. Simisiroglou, S.-P. Breton, G. Crasto, K.S. Hansen, and S. Ivanell. Numerical CFD Comparison of Lillgrund Employing RANS. *Energy Procedia*, 53:342–351, 2014.

[8] F. Azlan, J.C. Kurnia, B.T. Tan, and M.-Z. Ismadi. Review on optimisation methods of wind farm array under three classical wind condition problems. *Renewable and Sustainable Energy Reviews*, 135:110047, January 2021.

[9] José Herbert-Acero, Oliver Probst, Pierre-Elouan Réthoré, Gunner Larsen, and Krystel Castillo-Villar. A Review of Methodological Approaches for the Design and Optimization of Wind Farms. *Energies*, 7(11):6930–7016, October 2014.

[10] Rabia Shakoor, Mohammad Yusri Hassan, Abdur Raheem, and Yuan-Kang Wu. Wake effect modeling: A review of wind farm layout optimization using jensen's model. *Renewable and Sustainable Energy Reviews*, 58:1048–1059, May 2016.

[11] Javier Serrano González, Manuel Burgos Payán, Jesús Manuel Riquelme Santos, and Francisco González-Longatt. A review and recent developments in the optimal wind-turbine micro-siting problem. *Renewable and Sustainable Energy Reviews*, 30:133–144, February 2014.

[12] Enrico G.A. Antonini, David A. Romero, and Cristina H. Amon. Optimal design of wind farms in complex terrains using computational fluid dynamics and adjoint methods. *Applied Energy*, 261:114426, March 2020.

[13] Anshul Mittal, Kidambi Sreenivas, Lafayette K. Taylor, Levi Hereth, and Christopher B. Hilbert. Blade-resolved simulations of a model wind turbine: effect of temporal convergence. *Wind Energy*, 19(10):1761–1783, October 2016.

[14] Yuanchuan Liu, Qing Xiao, Atilla Incecik, Christophe Peyrard, and Decheng Wan. Establishing a fully coupled CFD analysis tool for floating offshore wind turbines. *Renewable Energy*, 112:280–301, November 2017.

[15] M. De Oliveira, R.C. Puraca, and B.S. Carmo. Blade-resolved numerical simulations of the NREL offshore 5 MW baseline wind turbine in full scale: A study of proper solver configuration and discretization strategies. *Energy*, 254:124368, September 2022.

[16] Zhihao Zhang, Limin Kuang, Zhaolong Han, Dai Zhou, Yongsheng Zhao, Yan Bao, Lei Duan, Jiahuang Tu, Yaoran Chen, and Mingsheng Chen. Comparative analysis of bent and basic winglets on performance improvement of horizontal axis wind turbines. *Energy*, 281:128252, October 2023.

[17] M. Raissi, P. Perdikaris, and G.E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, February 2019.

[18] Maziyar Bazmara, Mohammad Mianroodi, and Mohammad Silani. Application of physics-informed neural networks for nonlinear buckling analysis of beams. *Acta Mechanica Sinica*, 39(6):422438, June 2023.

[19] Enrui Zhang, Ming Dao, George Em Karniadakis, and Subra Suresh. Analyses of internal structures and defects in materials using physics-informed neural networks. *Science Advances*, 8(7):eabk0644, February 2022.

[20] Clara Herrero Martin, Alon Oved, Rasheda A. Chowdhury, Elisabeth Ullmann, Nicholas S. Peters, Anil A. Bharath, and Marta Varela. EP-PINNs: Cardiac Electrophysiology Characterisation Using Physics-Informed Neural Networks. *Frontiers in Cardiovascular Medicine*, 8:768419, February 2022.

[21] Chen Cheng, Hao Meng, Yong-Zheng Li, and Guang-Tao Zhang. Deep learning based on PINN for solving 2 DOF vortex induced vibration of cylinder. *Ocean Engineering*, 240:109932, November 2021.

[22] Hong Shen Wong, Wei Xuan Chan, Bing Huan Li, and Choon Hwai Yap. Strategies for multi-case physics-informed neural networks for tube flows: a study using 2D flow scenarios. *Scientific Reports*, 14(1):11577, May 2024.

[23] Qinghua Jiang, Chang Shu, Lailai Zhu, Liming Yang, Yangyang Liu, and Zhilang Zhang. Applications of finite difference-based physics-informed neural networks to steady incompressible isothermal and thermal flows. *International Journal for Numerical Methods in Fluids*, 95(10):1565–1597, October 2023.

[24] Vignesh Gopakumar, Stanislas Pamela, and Debasmita Samaddar. Loss landscape engineering via Data Regulation on PINNs. *Machine Learning with Applications*, 12:100464, June 2023.

[25] Aditi S Krishnapriyan, Amir Gholami, Shandian Zhe, Robert M Kirby, and Michael W Mahoney. Characterizing possible failure modes in physics-informed neural networks. *Advances in Neural Information Processing Systems*, 34:26548–26560, 2021.

[26] George Em Karniadakis, Ioannis G. Kevrekidis, Lu Lu, Paris Perdikaris, Sifan Wang, and Liu Yang. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, May 2021.

[27] Félix Fernández De La Mata, Alfonso Gijón, Miguel Molina-Solana, and Juan Gómez-Romero. Physics-informed neural networks for data-driven simulation: Advantages, limitations, and opportunities. *Physica A: Statistical Mechanics and its Applications*, 610:128415, January 2023.

[28] Hui Xu, Wei Zhang, and Yong Wang. Explore missing flow dynamics by physics-informed deep learning: The parameterized governing systems. *Physics of Fluids*, 33(9):095116, September 2021.

[29] Shengze Cai, Zhicheng Wang, Frederik Fuest, Young Jin Jeon, Callum Gray, and George Em Karniadakis. Flow over an espresso cup: inferring 3-d velocity and pressure fields from tomographic background oriented schlieren via physics-informed neural networks. *Journal of Fluid Mechanics*,

915:A102, 2021.

[30] Jaemin Seo. Solving real-world optimization tasks using physics-informed neural computing. *Scientific Reports*, 14(1):202, January 2024.

[31] Zhao Chen, Yang Liu, and Hao Sun. Physics-informed learning of governing equations from scarce data. *Nature Communications*, 12(1):6136, October 2021.

[32] Adhika Satyadharma, Ming-Jyh Chern, Heng-Chuan Kan, Harinaldi, and James Julian. Assessing physics informed neural network performance with sparse noisy velocity data. *Physics of Fluids (Under review)*, 2024.

[33] QiZhi He, David Barajas-Solano, Guzel Tartakovsky, and Alexandre M. Tartakovsky. Physics-informed neural networks for multiphysics data assimilation with application to subsurface transport. *Advances in Water Resources*, 141:103610, July 2020.

[34] Jun Pu, Wenfang Song, Junlai Wu, Feifei Gou, Xia Yin, and Yunqian Long. PINN-Based Method for Predicting Flow Field Distribution of the Tight Reservoir after Fracturing. *Geofluids*, 2022:1–10, April 2022.

[35] Prabhav Borate, Jacques Rivière, Chris Marone, Ankur Mali, Daniel Kifer, and Parisa Shokouhi. Using a physics-informed neural network and fault zone acoustic monitoring to predict lab earthquakes. *Nature Communications*, 14(1):3693, June 2023.

[36] Salvatore Cuomo, Vincenzo Schiano Di Cola, Fabio Giampaolo, Gianluigi Rozza, Maziar Raissi, and Francesco Piccialli. Scientific Machine Learning Through Physics–Informed Neural Networks: Where we are and What's Next. *Journal of Scientific Computing*, 92(3):88, September 2022.

[37] Zhiping Mao, Ameya D. Jagtap, and George Em Karniadakis. Physics-informed neural networks for high-speed flows. *Computer Methods in Applied Mechanics and Engineering*, 360:112789, 2020.