

Vortex-induced vibrations of a cylindrical wind turbine tower: Application of a force partitioning method

Simon Van Hulle, Shyam VimalKumar and Axelle Viré

Faculty of Aerospace Engineering

Kluyverweg 1, 2629 HS Delft, the Netherlands

Correspondence: s.b.e.vanhulle@student.tudelft.nl

July 30, 2023

ABSTRACT

Vortex-induced vibrations occur when an eigenfrequency of a fluid-immersed body approaches the frequency of vortex-shedding around that body. The periodically changing transverse force stimulates enhanced movement and causes vibrations in the body. Breaking down the contributions of these aerodynamic forces can provide valuable insight into the origin and driving factors behind many such flow phenomena. The present work revises a method to isolate the influence of five force-generating mechanisms and a scalable implementation in OpenFOAM is presented to make the force-partitioning method (FPM) freely available for future research efforts. A 2D simulation of vortex-induced vibrations in a wind turbine tower is then analysed. The method correctly identifies the vorticity-induced force as the main contribution to these oscillations, however, a number of limitations in the implementation have yet to be resolved. Sensitivity is observed with respect to the use of parallel computation and dynamic meshes in higher Reynolds number regimes.

1. INTRODUCTION

the Strouhal number St , defined as

$$St = \frac{f_{St} D}{V_\infty}, \quad (1)$$

When a circular cylinder is immersed in a fluid, the flow around it already becomes unstable at very low Reynolds numbers ($Re > 47$). This results in alternating vortices in the wake of the cylinder, known as a Kármán vortex street. The instability also causes the side regions of the object to experience periodically changing pressures and velocities and as such unsteady lift and drag forces (Mittal & Singh, 2005). If the cylinder is elastically supported, it will start oscillating, which in turn strongly influences the behaviour of the flow. This interaction between the oscillating body and the surrounding flow becomes most significant if an eigenfrequency of the body approaches the frequency of vortex-shedding. This frequency f_{St} is usually found using

where V_∞ is the flow velocity and D the cylinder diameter. In such case where f_{St} is close to a natural frequency of the elastic system, the frequencies tend to *synchronise*. This phenomenon, which is also called *lock-in*, was first described by Bishop & Hassan (1964) and Feng (1968). It may cause significant additional fatigue loads and lead to early failure.

One specific application where these vortex-induced vibrations (VIV) are increasingly important is the design of offshore wind turbine towers. These are particularly susceptible to VIV when they are (i) standing on the quayside, (ii) being trans-

ported offshore or (iii) placed on their foundation, prior to the installation of the rotor-nacelle assembly (Livanos, 2018; Viré et al., 2020).

It has been concluded over the last decades that the behaviour of VIVs has a complex dependence (i) on structural properties like mass damping and the mass ratio - that is, the structural mass divided by the mass of moved fluid (Khalak & Williamson, 1999; Govardhan & Williamson, 2000) and (ii) on fluid dynamics parameters such as Reynolds number (Anagnostopoulos & Bearman, 1992; Meneghini & Bearman, 1995). This complicates the matter since it implies that subcritical results may not simply be used to draw conclusions for the transitional ($1.5 \times 10^5 \leq Re \leq 3.5 \times 10^6$) or fully turbulent ($Re \geq 3.6 \times 10^6$) regimes, which are relevant for most realistic applications. Both experimental research and numerical simulations have been performed within these flow conditions (Achenbach, 1968; Lehmkuhl et al., 2014), however, these regimes impose challenges: to create such Reynolds numbers in an experimental setup, very high velocities are required in very large wind tunnels. This makes the process expensive and influences the accuracy of the outcome, so that different setups have resulted in inconsistent measurements for most of the force coefficients (Schewe, 1983). For computational models, the difficulty of setting up realistic and accurate wall conditions comes into play. However, in some cases, the numerical results have been convincingly similar to the experimental data (Singh & Mittal, 2005).

One aspect of VIVs that has not been very rigorously researched is the contribution of various force generating mechanisms to drive these oscillations. Menon & Mittal (2020) recently proposed a mathematical force-partitioning method (FPM) to isolate and quantify the contributions of the added mass, the vorticity-induced and viscous shear force on the oscillating body. Moreover, their approach separates spacial regions, in order to include the effects from several flow structures (e.g. the vortex wake) on the forcing behaviour.

Other partitioning methods had already been used, including one where the force is separated into a component in phase with velocity and another in phase with acceleration. For instance, Sarpkaya et al. (1978) demonstrated that the in-line force

increases with the ratio of amplitude to cylinder diameter, whereas the transverse force seriously varies close to the Strouhal frequency f_{St} (1).

However, this last method does not isolate different physical mechanisms and neither does it take into account the spacial location of the flow structures. That is where the technique of Menon & Mittal (2020) mostly proves its worth. It combines the fundamental work of Quartapelle & Napolitano (1983) about conveniently rewriting forces and moments on a body by projecting the Navier-Stokes equations (see section 2) with the observation by Chang (1992), who recognised the partitioning into vorticity-induced, added mass and shear forces.

Notice that breaking down the force components does not yet indicate to what extent they contribute to or counteract the oscillation. One way to do just that, is to determine the energy extracted from the surrounding flow by the elastic system of the cylinder under forced vibrations. As shown by the work of Morse & Williamson (2009), there exists a close relationship between extracted energy in forced oscillations and amplitude response in free oscillations. Therefore, measuring the work done by each of the forcing mechanisms on the cylinder will quantify their respective influence on the vibration. This method has shown all the more useful, because of the highly nonlinear responses that characterise this type of system (Williamson & Govardhan, 2004).

Using this combined approach of force partitioning and energy mapping, Menon & Mittal (2020) concluded that the most important factor in the sustenance of VIVs is the vorticity-induced force, rather than the vortices in the wake behind the cylinder. It was found that the wake vortex shedding only contributes temporarily at the onset of the oscillations, whereas growth of the vibrations is very much determined by vorticity in the shear layer. In that phase, the wake vortices extract energy from the oscillations rather than adding to it.

It must be noted that the above-mentioned simulations were performed at very low Reynolds numbers ($Re = 100$), which avoids most of the complex dependencies described before. This regime is of course unrealistic for actual wind turbine appli-

cations, or any aerodynamics applications for that matter.

Clearly, there are numerous considerations that can be made about the assumptions in previous studies with this FPM. However, since the actual method is not publicly available, there is a barrier to explore these influencing factors and their impact on the results. To encourage a more frequent use of the method in flow simulations on topics including but not limited to VIVs, a standardised, scalable and extendable implementation of the FPM is required within a commonly used flow simulation framework. Cases like three-dimensional effects, turbulent regimes, etc. could then be more easily incorporated in the base of knowledge on the onset and sustenance of vortex-induced vibrations. One such software package is [OpenFOAM \(2020\)](#); it is an open source tool that is written entirely in C++ and it allows for highly customised flow calculations. Its wide user base in both commercial and academic organisations makes it a good choice for the implementation of this method.

The purpose of this paper is to present an OpenFOAM implementation of the force partitioning method by [Menon & Mittal \(2020\)](#) and to investigate its performance for variations in Re regime, in order to identify the influence of various force creating mechanisms on the severity of VIVs. To this effect, an Unsteady Reynolds-Averaged Navier Stokes (URANS) simulation adapted from the setup created by [Viré et al. \(2020\)](#) will be analysed.

The present work starts off in [section 2](#) by a detailed overview of the specific FPM and then describes its application within the OpenFOAM software. The section also briefly introduces the modelling approach used to obtain the results documented in [section 3](#). This 2D model of a wind turbine tower in a uniform cross-flow is used to illustrate the typical outcomes that can be expected while using the FPM and to provide insight in the accuracy of the implementation. Consequently, an analysis of the FPM is shown in [section 4](#), along with all the identified shortcomings and recommendations for improvement. Finally, the main conclusions are summarised in [section 5](#).

2. METHODOLOGY

This section details the derivation of the method for force partitioning, and its implementation in the OpenFOAM software is discussed in depth. For illustration, a typical case layout is then described, going over all the adjustments that must be made to ensure correct functioning of the method. The modelling approach for vortex-induced vibrations is briefly reviewed, following the work of [Viré et al. \(2020\)](#). The combination of all these methods was needed to produce the results that are presented later in this work.

2.1 Force-Partitioning Method

This study implements a force partitioning method (FPM) that starts from first principles to mathematically isolate the force generating mechanisms. The derivation below follows the original proposal of [Quartapelle & Napolitano \(1983\)](#) with adaptations from [Zhang \(2015\)](#) and [Menon & Mittal \(2020\)](#). Note that the approach in itself is not at all limited to the analysis of VIVs. The force on any object submerged in a fluid could in principle be deconstructed in this fashion.

The procedure starts from the incompressible Navier Stokes equations, that can be written as

$$\rho \frac{\partial \vec{u}}{\partial t} + \rho \vec{\omega} \times \vec{u} + \frac{1}{2} \rho \nabla (\vec{u} \cdot \vec{u}) = -\nabla p - \mu \nabla \times \vec{\omega}. \quad (2)$$

In this equation, ρ is the fluid density, μ represents the dynamic viscosity, \vec{u} is the flow velocity, p is the pressure and $\vec{\omega}$ is the vorticity. The latter is defined as the curl of the velocity vector $\vec{\omega} = \nabla \times \vec{u}$.

The velocity on the boundary \mathcal{B} of the immersed body is denoted as $\vec{U}_{\mathcal{B}}$ and the velocity on the outer surface of the large but finite domain is set equal to the free stream velocity \vec{V}_{∞} . The boundary of the body $\mathcal{B}(t)$ is assumed a function of time, to allow for versatile analysis of time-dependent systems. For instance, the study by [Zhang \(2015\)](#) applied the method to deconstruct the forces on an insect hovering in the air, constantly moving its wings and thereby changing its outer boundary. For the purposes of VIV analysis in wind turbines, the fluid-structure interactions must be allowed to accurately represent the oscillating tower.

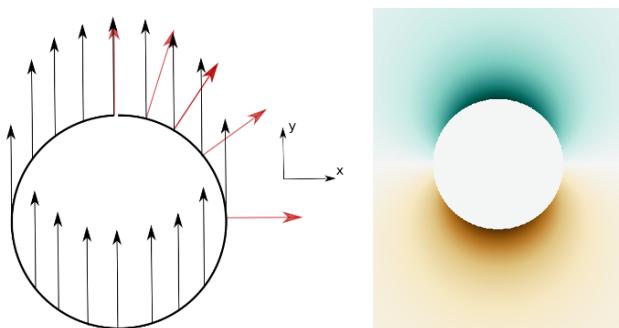
To simplify the Navier Stokes equations, a harmonic function $\phi^{(i)}$ is introduced, where $i \in \{1, 2, 3\}$ represents the direction of force partitioning. This scalar field is defined for every time $t = \tau$ by

$$\nabla^2 \phi^{(i)}(\tau) = 0, \quad \vec{n} \cdot \nabla \phi^{(i)} = \begin{cases} n_i & \text{on } \mathcal{B}(\tau) \\ 0 & \text{on } \Sigma \end{cases} \quad (3)$$

This field can be interpreted as the potential for ideal flow passing by the body in the x_i direction with unit velocity (Zheng et al., 2019). The vector \vec{n} represents the normal to the boundary, pointing into the fluid. Moreover, $n_i = \vec{n} \cdot \nabla x_i$ is the component of that normal vector in the direction of force partitioning. For a 2D case, this means that the boundary condition for \mathcal{B} can be written as

$$\frac{\partial \phi^{(i)}}{\partial \vec{n}} \Big|_{\mathcal{B}} = \begin{cases} \vec{n} \cdot \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T & \text{if } i = 0 \\ \vec{n} \cdot \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T & \text{if } i = 1 \end{cases}. \quad (4)$$

To clarify this further, the case for $i = 1$ is visualised in Figure 1a. In that image, the red arrows represent the normal face vector \vec{n} and the black ones show the direction of ∇x_i , which basically comes down to a unit vector in the direction of force partitioning. Clearly, the dot product of these two has the maximum value at the top and bottom of the cylinder. At those locations, $\phi^{(i)}$ changes the most, whereas it is constant along the x-direction. At those points, the vectors are perpendicular, so the normal gradient becomes zero.



(a) Visualisation of the boundary condition for $\phi^{(i)}$ around a cylinder, using $i = 1$. Red arrows represent the normal vectors \vec{n} to the surface and black arrows show the unit vector in the direction of force partitioning.
(b) Analytical solution of $\phi^{(i)}$ for a circular sphere: circular cross-section.

Figure 1: Auxiliary potential field $\phi^{(i)}$

Specifically for the case of a circular sphere, the field can be solved analytically by Equation (5), of which a cross-section along the xy plane is visualised in Figure 1b. Note that R represents the sphere's radius, and r and θ are the radial coordinate and zenith angle, respectively (Zhang, 2015). This visualisation indeed agrees with the described boundary conditions.

$$\phi^{(i)} = -\frac{R^3 \cos \theta}{2r^2} \quad (5)$$

This auxiliary potential field can be used to project Equation (2) and integrate the equation on both sides over the entire volume of the domain V_f :

$$\begin{aligned} & \rho \int_{V_f} \left(\frac{\partial \vec{u}}{\partial t} + \vec{\omega} \times \vec{u} \right) \cdot \nabla \phi^{(i)} dV \\ & + \int_{V_f} \left(\frac{1}{2} \rho \nabla (\vec{u} \cdot \vec{u}) + \nabla p \right) \cdot \nabla \phi^{(i)} dV \\ & + \mu \int_{V_f} (\nabla \times \vec{\omega}) \cdot \nabla \phi^{(i)} dV = 0. \end{aligned} \quad (6)$$

Using a variation of the divergence theorem

$$\int_V \nabla f dV = \int_{\partial V} f \vec{n} dS, \quad (7)$$

the expression in Equation (6) can be reformulated as shown below. Note that the integral of pressure and kinetic energy over the outer boundary Σ of the domain vanishes due to the boundary condition of $\nabla \phi^{(i)}$ (see Equation (3)).

$$\begin{aligned} & \rho \int_{V_f} \left(\frac{\partial \vec{u}}{\partial t} + \vec{\omega} \times \vec{u} \right) \cdot \nabla \phi^{(i)} dV \\ & + \int_{\mathcal{B}} \left(\frac{1}{2} \rho \vec{U}_{\mathcal{B}}^2 + p \right) n_i dS \\ & - \mu \int_{\mathcal{B}+\Sigma} (\vec{\omega} \times \vec{n}) \cdot \nabla \phi^{(i)} dS = 0 \end{aligned} \quad (8)$$

According to Zhang et al. (2015), the total force component that acts on the immersed body in the direction of force partitioning x_i can be written as

$$F_{\mathcal{B}}^i = \int_{\mathcal{B}} p n_i dS - \int_{\mathcal{B}} (\vec{\omega} \times \vec{n}) \cdot \nabla x_i dS. \quad (9)$$

If the integrated expression from Equation (8) is

then subtracted from this total force equation, it is true that

$$\begin{aligned} F_B^i &= -\rho \int_{V_f} \frac{\partial \vec{u}}{\partial t} \cdot \nabla \phi^{(i)} dV - \rho \int_{V_f} (\vec{\omega} \times \vec{u}) \cdot \nabla \phi^{(i)} dV \\ &\quad - \int_{\mathcal{B}} \frac{1}{2} \rho \vec{U}_{\mathcal{B}}^2 n_i dS + \mu \int_{\mathcal{B}} (\vec{\omega} \times \vec{n}) \cdot \nabla (\phi^{(i)} - x_i) dS \\ &\quad + \mu \int_{\Sigma} (\vec{\omega} \times \vec{n}) \cdot \nabla \phi^{(i)} dS. \end{aligned} \quad (10)$$

The only unsteady term in this equation (containing the time partial derivative of the velocity vector) can be rewritten using the definition of the total derivative

$$\begin{aligned} \frac{d\vec{u}}{dt} &= \frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} \\ &= \frac{\partial \vec{u}}{\partial t} + \frac{1}{2} \nabla \vec{u}^2 - \vec{u} \times (\nabla \times \vec{u}). \end{aligned} \quad (11)$$

With this expansion and the divergence theorem, the first term of Equation (10) can be reformulated as

$$\begin{aligned} \rho \int_{V_f} \frac{\partial \vec{u}}{\partial t} \cdot \nabla \phi^{(i)} dV &= \rho \int_{\mathcal{B}+\Sigma} \vec{n} \cdot \left(\frac{d\vec{u}}{dt} \phi^{(i)} \right) dS \\ &\quad - \rho \int_{V_f} \nabla \cdot (\vec{\omega} \times \vec{u}) \phi^{(i)} dV \\ &\quad - \rho \int_{V_f} \nabla \cdot \left(\frac{1}{2} \nabla \vec{u}^2 \phi^{(i)} \right) dV. \end{aligned} \quad (12)$$

The total force expression now contains integrals over the volume V_f , the boundary of the body \mathcal{B} , and the boundary of the domain Σ . They all involve velocity and vorticity, but clearly describe different phenomena. One final split of terms can be achieved by using the Helmholtz decomposition to expand the velocity vector into a curl-free and a divergence-free component (Batchelor, 2000):

$$\vec{u} = \vec{u}_\Phi + \vec{u}_v = \nabla \Phi + \nabla \times \vec{A}, \quad (13)$$

where Φ is a scalar potential and \vec{A} a vector potential. By the continuity equation, the divergence of the velocity vector is zero. The same is true for the divergence of $\nabla \times \vec{A}$, provided that \vec{A} has continuous derivatives. Therefore, it can be concluded that $\nabla^2 \Phi = 0$, such that Φ simply refers to the velocity

potential field.

Combining all of these considerations with Equation (10) yields the final decomposition of force that acts on the body $F_{\mathcal{B}}^i = F_{\kappa}^i + F_{\omega}^i + F_{\sigma}^i + F_{\phi}^i + F_{\Sigma}^i$. Recall that $i \in \{1, 2, 3\}$ represents the direction of force partitioning.

Complete Force Partitioning

$$\begin{aligned} F_{\kappa}^i &= -\rho \int_{\mathcal{B}} \left(\frac{d\vec{U}_{\mathcal{B}}}{dt} \cdot \vec{n} \phi^{(i)} + \frac{1}{2} U_{\mathcal{B}}^2 \vec{n} \cdot \nabla \phi^{(i)} \right) dS \\ F_{\omega}^i &= \rho \int_{V_f} \nabla \cdot \left[(\vec{\omega} \times \vec{u}) + \nabla \left(\frac{1}{2} u_v^2 + \vec{u}_\phi \cdot \vec{u}_v \right) \right] \phi^{(i)} dV \\ F_{\sigma}^i &= \mu \int_{\mathcal{B}} (\vec{\omega} \times \vec{n}) \cdot \nabla (\phi^{(i)} - x_i) dS \\ F_{\phi}^i &= \rho \int_{V_f} \nabla \cdot \left(\nabla \frac{1}{2} u_\phi^2 \phi^{(i)} \right) dV \\ F_{\Sigma}^i &= -\rho \int_{\Sigma} \vec{n} \cdot \frac{d\vec{u}}{dt} \phi^{(i)} dS + \mu \int_{\Sigma} (\vec{\omega} \times \vec{n}) \cdot \nabla \phi^{(i)} dS \end{aligned} \quad (14)$$

As one might get lost in the mathematical abstraction of this decomposition, it is good to identify the main features in each of these contributions. First of all, the kinematic force F_{κ}^i only depends on the shape of the body's boundary \mathcal{B} and its velocity and acceleration within the flow. Furthermore, all the contributions of vorticity to the total force are grouped within the F_{ω}^i component. The force F_{σ}^i is then fully due to the viscous shear with the boundary (notice the dynamic flow viscosity μ) and F_{ϕ}^i quantifies the effects that result purely from potential flow. Finally, F_{Σ}^i describes the influence of viscous shear with the domain's outer boundary. This is very similar to the F_{σ}^i force, except for an additional unsteady term. For a more detailed breakdown of these coefficients' meaning, refer to Zhang (2015).

Finally, for dimensional analysis of many applications, one might be interested in the force coefficient in a certain direction rather than the magnitude of the actual load. To this effect, the force must be divided by the free-stream dynamic pressure q_∞ and the effective area in the direction i . For the present work, only 2D cases are considered, so the latter actually becomes the reference length of the immersed body (diameter of the cylinder):

$$C_{\mathcal{B}}^i = \frac{F_{\mathcal{B}}^i}{q_{\infty} \ell_{\text{ref}}} = \frac{F_{\mathcal{B}}^i}{\frac{1}{2} \rho_{\infty} V_{\infty}^2 \ell_{\text{ref}}}. \quad (15)$$

Dimensional analysis has many advantages, including the ability to apply results in a variety of scales and reduce the number of experiments that need to be done to get an insight in the influence of certain parameters (White, 2003).

2.2 OpenFOAM Implementation

The main effort of this work is the implementation of the outlined FPM into the software package OpenFOAM (2020). This tool for computational fluid dynamics (CFD) is useful for a wide variety of computational problems, from complex fluid flows to solid mechanics and electromagnetics. The open source nature of the software makes it easy to extend the functionality and tailor the behaviour of all simulations to the need of the relevant analysis.

The present implementation of the FPM is collected in a small library of several files that can be downloaded and installed to extend OpenFOAM by following a set of straight-forward steps. All the source code, documentation and installation instructions can be found in the GitLab repository of the project (Van Hulle, 2022).

The implementation basically consists of three different components that supplement the templated library. First, there is a slightly modified version of the pimpleFoam solver, that ensures the calculation of the velocity potential field Φ and the auxiliary potential field $\phi^{(i)}$. The latter, however, requires a boundary condition that is not part of the standard OpenFOAM library. Therefore, an adaptation of the fixedGradient boundary condition is proposed to allow setting the property from Equation (3) with very limited configuration. Finally, the most important addition is a custom function object that calculates all the force contributions from Equation (14) and then stores this information for all time steps in a data file within the postProcessing directory of the case.

The custom boundary condition requires the user to provide a constant gradient vector at the boundary of interest. The dot product is then calculated with the normal vector to the surface \vec{n} to find the correct

normal gradient $\partial F / \partial \vec{n}$ as specified in Equation (4). Special care must be taken with the sign conventions of these normal vectors within OpenFOAM. In the software, these are considered positive in the direction away from the internal domain. Therefore, the default normal vector actually points into the immersed body instead of out and needs to be reversed before calculating the gradient. Apart from this, the entire boundary condition (formally referred to as a fvPatchField) is identical to the fixedGradient equivalent. Renaming all constructors was the only thing necessary to ensure the expected functionality.

With the necessary boundary condition defined to constrain the auxiliary potential field $\phi^{(i)}$, this and the velocity potential Φ can be solved by a slightly modified solver application fpmPimpleFoam. An effort was made to do this within the function object that also calculates the force contributions, but without satisfactory results. As these fields need to be known at every time step, it is considered preferable to have them within the same scope as the pressure and velocity fields. This also allows easier writing to files for later inspection and post-processing of these scalar fields.

Note that the potential fields Φ and $\phi^{(i)}$ are described by exactly the same equation and only differ in their boundary conditions. The current implementation automatically creates the velocity potential Φ with a zero gradient on the boundaries, but a manual initialisation for the auxiliary potential $\phi^{(i)}$ is still required to set the custom boundary condition described before. This is a weakness in the code that should be resolved to increase the method's ease of use.

The main part of the FPM is finally implemented in a so-called function object that can be initialised from the control dictionary of the case. It is a templated class called fpmForcePart that derives from the forces class. It was inspired by the function object that calculates the traditional force coefficients such as lift, drag and moments. As these have very similar output, the same pattern could be followed to ensure all the components that one might expect. The configuration is read from the controlDict and a check is performed to ensure that all the required fields (\vec{u} , $\phi^{(i)}$ and Φ) are present in the database. The function then proceeds to calculate

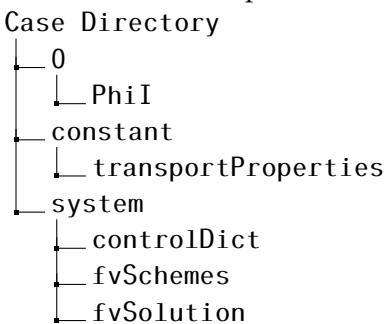
all the force contributions from [Equation \(10\)](#), using the operators from the `Foam::fvc` namespace. The results are logged to standard output, as well as to a dedicated data file for post-processing. To prevent overwriting old results, a new file is automatically created when the default name already exists.

The reader is referred to the source code for further details on the implementation and possibilities to extend the functionality.

2.3 Case Setup Requirements

After outlining all parts of the FPM implementation, it is necessary to address the additional setup that using this method requires, when creating simulation cases. This section will highlight the minimum requirements for the simulation to produce valid FPM results. For the details of additional functionality, the reader is referred to the documentation within the code repository.

The FPM is dependent on a minimal directory structure as shown below. Note that a case might need many more files, but those will not be read directly by the method of the present work.



First of all, the `0/PhiI` file defines the auxiliary potential field $\phi^{(i)}$, along with its initial and boundary conditions. This volumetric scalar field has dimensions of $\text{m}^2 \text{s}^{-1}$, the internal field can be initialised to any arbitrary uniform value as it will be solved from the boundary conditions. Finally, the boundary field must comply with [Equation \(3\)](#), meaning that all outer boundaries of the domain require the `zeroGradient` condition and all boundaries of the immersed body need to be of the custom type `fpmPotentialGradient`, with the gradient vector set to the direction of force partitioning.

For instance, if one wants to partition the lift force on a cylinder, defined as positive in the y direction,

then the boundary field must be set as

```

<CASE>/0/PhiI
...
boundaryField
{
    ...
    CYLINDER
    {
        type          fpmPotentialGradient;
        gradientVec (0 1 0);
    }
}
  
```

The next file that the method relies on is the `transportProperties` dictionary. This needs to contain a field for the kinematic viscosity ν that is used to calculate the Reynolds number. For example, if $\nu = 0.01$ for the case of interest, then the file should contain the following.

```

<CASE>/constant/transportProperties
...
nu           nu [ 0 2 -1 0 0 0 0 ] 0.01;
  
```

Then, the most important additions are required within the control dictionary of the case. To make certain that the above-mentioned custom boundary condition works as expected, the relevant library must be loaded within the case. Furthermore, a function object definition is required to call for the calculations of the FPM. This involves explicitly listing the boundary patches that describe the surface of the immersed body - the cylinder in the present example - as well as a definition of the coordinate system and a choice for the direction of force partitioning. For the results to be valid, it is essential that the boundary condition in `0/PhiI` matches the value of the `fpmForcePartDir` item within the control dictionary.

Consistently defining the free stream velocity and density as well as the body reference length finishes the setup of the function. This reference length ℓ_{ref} is used exclusively to calculate the Reynolds number and to make the coefficients non-dimensional; so for a cylinder, it would be equal to the diameter D . This is currently the only thing that prevents the immediate use of the method for 3D simulations. In such case, a reference area would be needed to make the forces non-dimensional; but the Reynolds

number would still be calculated with a reference length. Specifying different values for these two things is currently not supported. In summary, an example entry for the 2D wind turbine tower is shown for illustration:

<CASE>/system/controlDict

```

...
libs      ("libfpmBoundaryConditions.so");

funcitons
{
    ...
    fpmForcePart_object
    {
        type      fpmForcePart;
        libs      ("libfpmForcePart.so");

        patches     (CYLINDER);
        enabled     true;
        log         true;

        CofR        (0 0 0);
        liftDir     (0 1 0);
        dragDir     (1 0 0);

        fpmForcePartDir (0 1 0);

        magUInf     10;
        lRef        1.0;
        rhoRef      1.0;

        writeControl timeStep;
        writeInterval 1;
    }
}

```

Two more files are left to complete for successful use of the method. First, the `system/fvSchemes` needs default schemes for the time derivative, gradient, divergence, Laplacian and interpolation. It is of course possible to specify varying schemes for all the different calculations, however, a detailed analysis of the available schemes and their influence on the performance of the FPM is beyond the scope of the present work.

Finally, the solvers need to be specified for the two additional fields that are calculated by the adapted `pimpleFoam` application. A minimum working example of the `fvSolution` file is provided for reference, but the influence of solver choice is not considered here either.

<CASE>/system/fvSolution

```

...
solvers
{
    ...
    Phi
    {
        solver          GAMG;
        smoother        DIC;
        tolerance       1e-06;
        relTol          0.01;
    }
    PhiFinal      { $Phi }
    PhiI          { $Phi }
    PhiIFinal     { $Phi }
}

```

Provided that all the parameters and values in the previously mentioned files are set correctly and consistently, the simulation stores all results within the `postProcessing/fpmForcePart_object/` directory. A single file contains all the output for the entire run. With the details of the implementation clearly defined, it is possible to set up a model for analysing vortex-induced vibrations in a wind turbine tower.

2.4 Turbine Tower Modelling Approach

The simulation used for this study was adapted from the one introduced by [Viré et al. \(2020\)](#). It employs OpenFOAM-v2012 to solve the incompressible Navier-Stokes equations for a Newtonian fluid by means of a finite volume discretisation. That study also uses a turbulent version of the case where a $k - \omega$ shear model was applied to solve the unsteady Reynolds-averaged Navier-Stokes (URANS) equations. Since the present work is mostly focused on implementing and preliminarily validating the force partitioning approach in OpenFOAM, no turbulence model is included, and all validation is performed with the assumption of laminar flow. The applicability of the FPM in turbulent regimes is not addressed here, but could certainly be analysed in future research.

All equations are solved with second-order accurate finite-volume discretisation in space and time. In order to couple the pressure and velocity behaviour, the PIMPLE algorithm is then applied. This method combines the pressure-implicit splitting of operators (PISO) with the semi-implicit method for

pressure-linked equations (SIMPLE) (Jasak & Gosman, 2001).

When considering incompressible flow, the assumption of constant density introduces a coupling between pressure and velocity that usually requires an iterative solver to find consistent fields that fulfil the conservation of mass and momentum. Holzmann (2019) describes this problem and its solutions in rigorous detail. Within OpenFoam, SIMPLE is often used for steady-state analysis (Patankar, 1981), whereas PISO is applied for transient simulations with time steps corresponding to a Courant number $Co < 0.7$ (Issa, 1986). When larger time steps are preferred, the combined PIMPLE algorithm has the great advantage of stability at $Co > 1$ (Holzmann, 2019).

The fluid-structure coupling considered in this case is untouched from Viré et al. (2020), where it is described in detail, along with the implementation of dynamic meshes. The analysis only considers a weakly coupled scheme, which uses one iteration per time step for the motion solver. The position of the cylinder and the computational mesh are first updated based on the previous time step and then, the algorithm iterates over the flow solver to find the new velocity and pressure field.

The dynamic movement of the turbine tower in 2D is governed by a system that consists of a linear-elastic spring with constant k and a cylinder with mass m , connected together at the cylinder's centre of gravity. The system is then modelled using forced vibrations, with the aerodynamic lift force as the acting force (Menon & Mittal, 2020). Furthermore, the cylinder is constrained to move only in the direction transverse to the free-stream flow. To ensure a fair comparison between simulation cases, the dimensionless equation of motion is used where the diameter D , free-stream velocity V_∞ and density ρ are considered as the characteristic scaling parameters. This results in

$$m^* \ddot{y}^* + k^* (y^* - y_0^*) = C_L, \quad (16)$$

where C_L is the total lift force coefficient. The parameters y^* , m^* and k^* represent the dimensionless forms of the displacement, cylinder mass and spring stiffness constant, respectively. These are defined as

$$y^* = \frac{y}{D}, \quad m^* = \frac{2m}{\rho D^2} \quad \text{and} \quad k^* = \frac{2k}{\rho V_\infty^2} \quad (17)$$

One final parameter that is often used is the reduced velocity $V^* = V_\infty / (f_s D) = 1/f_s^*$. In that relation, f_s^* is the dimensionless form of the structural system's natural frequency. For all simulations in this work, a cylinder with a diameter of 1 m is used, where the dimensionless mass is kept constant at $m^* = 10$.

3. RESULTS FOR VORTEX-INDUCED VIBRATIONS

The VIV results of this work mainly serve to validate a selection of the findings reported by Menon & Mittal (2020) and conversely, the insights are used to verify the presented implementation by comparison. First, Figure 2 shows an illustration of the vortex-induced vibrations scenario. The coloured plane represents the pressure distribution behind a 2D section of a moving cylinder. The Kármán vortex street is clearly formed, and its oscillating nature causes a changing resultant force component in the direction transverse to the airflow.

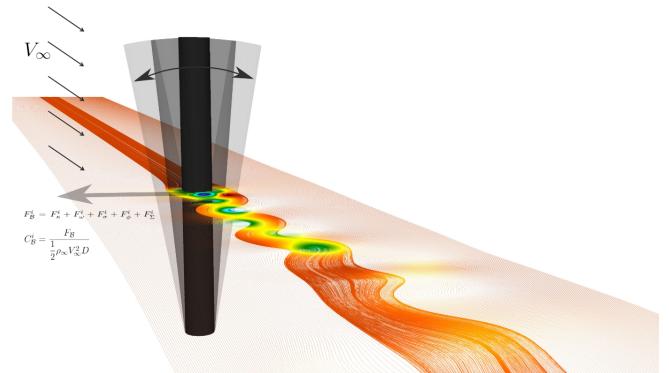


Figure 2: Situation sketch of the VIV case. 2D simulation around a transverse-moving cylinder at $Re = 100$ with reduced velocity $V^* = 5$. Note that the 3D movement in the image is an exaggerated impression and solely serves for illustrative purposes.

The 2-dimensional behaviour is highlighted more closely by the images of Figure 3. Notice that the regions of maximum vorticity magnitude coincide with the abrupt transitions between higher and lower velocity magnitudes. Since the free stream flows from left to right, high-velocity regions above low-velocity regions result in clockwise vorticity

and vice-versa.

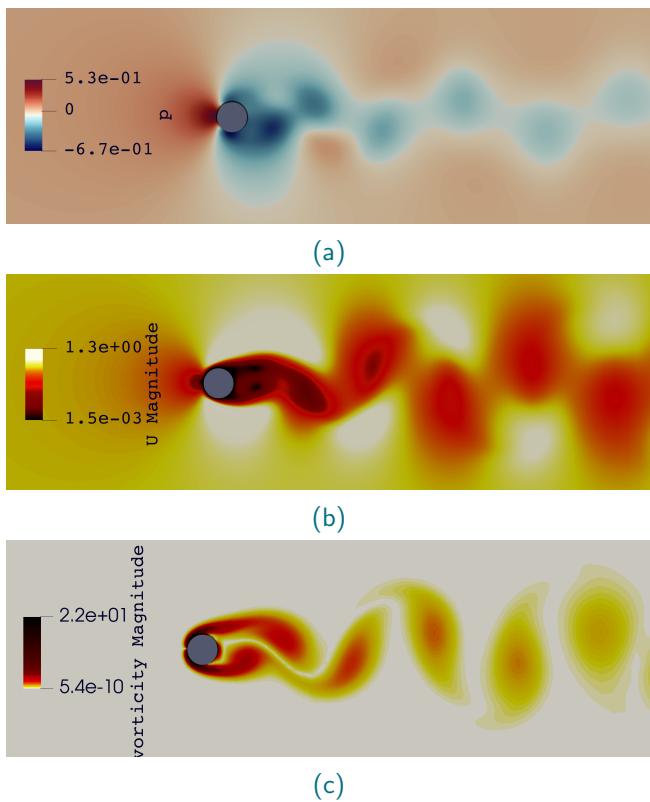


Figure 3: Contour plot of the flow characteristics behind the cylinder for $Re = 100$. (a) pressure field, (b) magnitude of the velocity vector field and (c) magnitude of the vorticity in the vicinity of the turbine tower.

Furthermore, the maximum dimensionless amplitude is plotted for a range of reduced velocities V^* . The maximum is observed at $V^* = 5$, after which the lock-in regime continues up to a reduced velocity of 7. The trend of the results in Figure 4a mostly agrees with the data from Menon & Mittal (2020), although the absolute errors between the individual data points cannot be ignored. The oscillating frequency is then plotted in Figure 4b for the same range of V^* , normalised by the natural frequency of the mass-spring system that represents the tower. This ratio generally follows a linear trend, as indicated by the Strouhal frequency f_{St} , with $St \approx 0.2$. However, when the frequency of vortex shedding approaches the tower's natural frequency, it tends to synchronise; this results in the extended lock-in regime. When increasing the velocity beyond the lock-in regime, the frequency resumes its linear growth with the same slope, although a negative offset is observed with respect to f_{St} . This data agrees with the findings of Viré et al. (2020).

The results above are based solely on the total lift

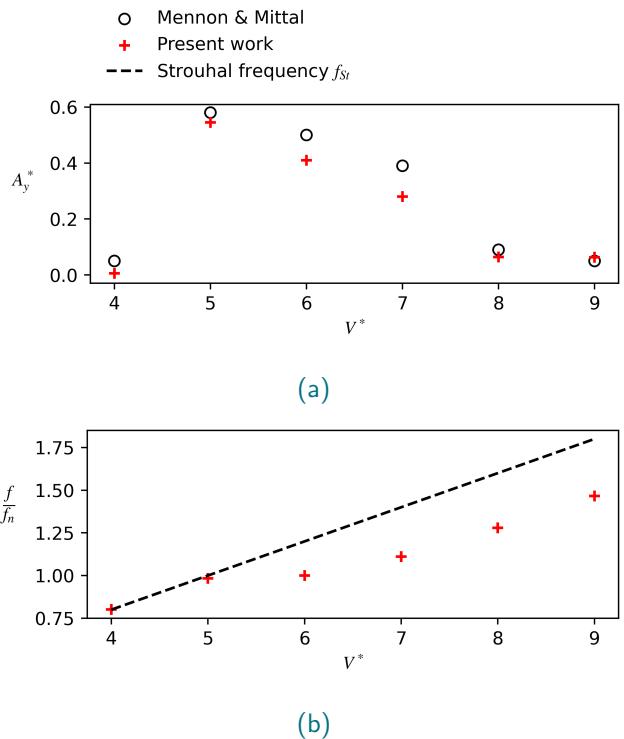


Figure 4: Variations in parameters for varying reduced velocity V^* at $Re = 100$. (a) reports the maximum dimensionless amplitude in the transverse direction A_y^* and (b) shows the ratio of oscillating frequency f to the natural frequency f_n of the mass-spring system.

coefficient, and don't yet consider the outcome of the FPM. Figure 5 shows the response partitioning during the onset of the oscillation with $Re = 100$. It is immediately clear that the main driving coefficient is C_ω , representing the vorticity-induced force. Less predominant are the contributions due to the kinematics of the cylinder and the shear with the body's boundary. These are of similar magnitude and their contribution is reduced further when the Reynolds number rises. All of this is in line with the observations by Menon & Mittal (2020). In comparison, the force due to potential flow is negligibly small. This again confirms the discussions by Zhang (2015) and Menon & Mittal (2020). The final component C_Σ is excluded from the present work, as the literature considers it negligible as well, provided there is a sufficiently large spatial domain. For this purpose, none of the results that follow in the next section feature those two coefficients; it is assumed that $C_\phi \approx 0 \approx C_\Sigma$.

Although the work by Menon & Mittal (2020) does not include the full response data for a circular cylinder, it is still valuable to compare the find-

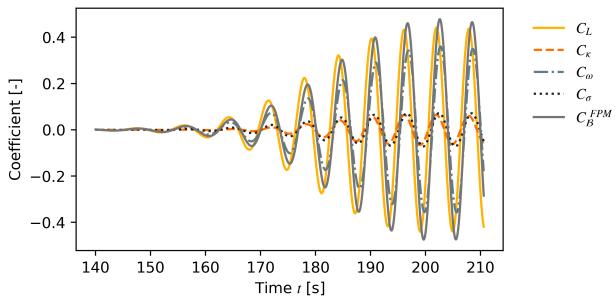


Figure 5: Full response with dynamic mesh at $Re = 100$.

ings on a qualitative level. Figure 6 shows their C_L results for a super-ellipse-shaped cylinder with aspect ratio $AR = 1.15$, i.e. the diameter parallel to the flow is 1.15 times the transverse diameter. As seen in Figure 6a, this shape introduces amplitude fluctuations that are not found in the circular case of the present work. This figure also shows the accurate matching of C_L and the summed FPM coefficient C_B . Then Figure 6b confirms the finding that C_ω accounts for the large majority of the lift coefficient what its amplitude is maximal.

Additional results are presented in the next section, where the verification procedures for the FPM implementation are described, along with the main findings and a number of recommendations for improvements in the near future.

4. VERIFICATION AND LIMITATIONS OF THE FPM IMPLEMENTATION

Using the case of vortex-induced vibrations, extensive tests were performed on the FPM code, to iteratively solve problems and correct bugs. Unfortunately, not all problems are currently fixed and a number of weaknesses remain that need to be addressed before the results can be trusted. It is strongly recommended to further investigate them while finalising the code.

4.1 Potential fields

First of all, the solution of the potential fields is verified qualitatively by visualising them in a program called ParaView (Ahrens et al., 2015). The harmonic auxiliary potential $\phi^{(i)}$ is shown for the lift ($i = 2$) on a cylinder in Figure 7. This field was evaluated for many different simulations, varying the flow velocity, viscosity, cylinder diameter. As expected, the

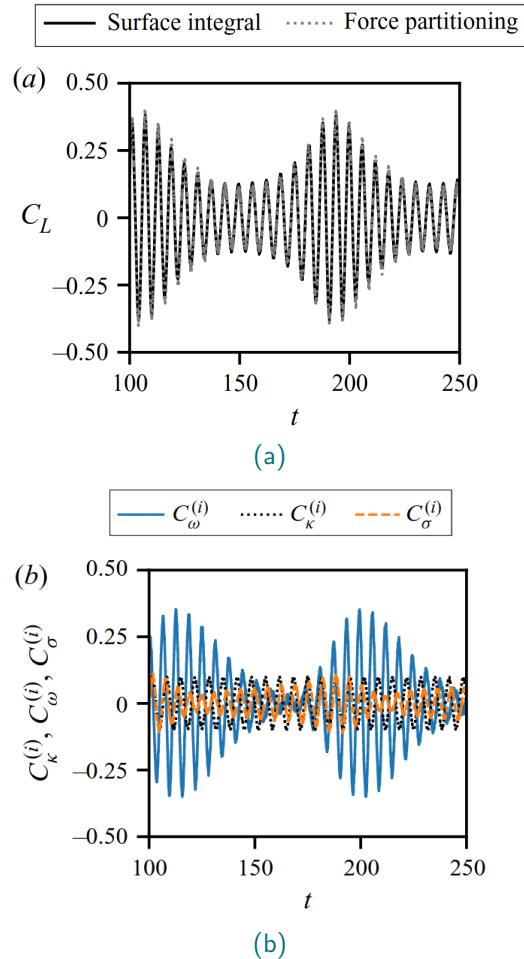


Figure 6: Reference data from Menon & Mittal (2020), showing the partitioning of the lift force on a cylinder with aspect ratio $AR = 1.15$. (a) shows the match of C_L and C_B and (b) highlights the main contributing coefficients.

outcome is consistently the same with respect to the current position of the cylinder. When allowing a movement of the dynamic mesh, the solution is correctly recalculated and the expected field found. Note that this solved field agrees with the analytical solution for a sphere that was shown in Figure 1b. Furthermore, it is observed that the field is not recalculated when the mesh does not move, which is as expected. The confidence in the implementation of the boundary condition and the solution of $\phi^{(i)}$ is high.

The same cannot be said about the ordinary velocity potential Φ . As discussed in subsection 2.2, the solver combines aspects from the PIMPLE solver and the potential solver. In principle, this should allow the boundary conditions to be set automatically and the field solved accordingly. However, the output obtained by a typical run looks similar to Figure 8. Note that the value of Φ is between 0

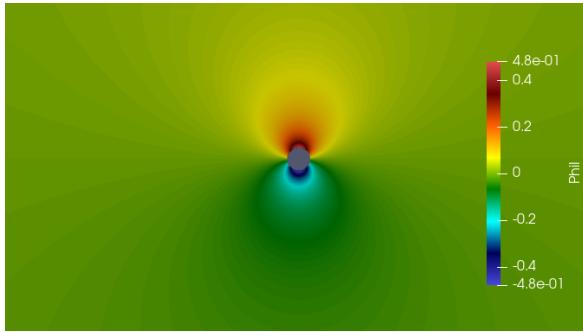


Figure 7: Visualisation of the auxiliary potential field $\phi^{(i)}$ for $i = 2$ (lift) around a circular cylinder

and $1.1 \times 10^{-27} \text{ m}^2 \text{ s}^{-1}$, which indicates that something is wrong with the absolute magnitude of the field. However, it must be stressed that for the purpose of the FPM, only the gradient of this field is important, which might not be affected by the incorrect magnitude of the field values. Nonetheless, further debugging efforts are required to remove this problem.

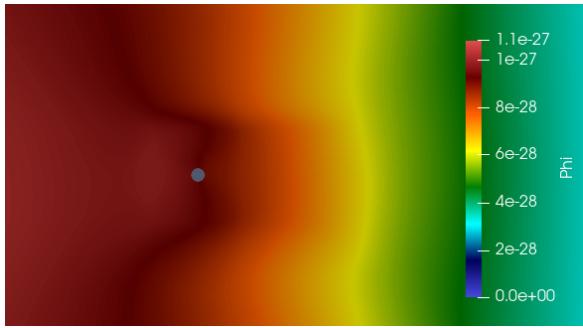


Figure 8: Calculated velocity potential field Φ in a part of the domain around the cylinder. The magnitudes provided are unrealistically low.

4.2 Static vs Dynamic Mesh

Another shortcoming that has not yet been resolved is how to deal with the moving nature of the body. The present approach is to make abstraction from the changing characteristics of the mesh, and simply calculate the integrals for all coefficients based on the instantaneous state of the mesh at every time step. Provided that the time increment is sufficiently small, this should theoretically result in the desired output. This is, however, not the observed behaviour.

In the previous section, [Figure 5](#) already showed the transverse force response on a moving cylinder in a flow with $Re = 100$. At first glance, the sum C_B of all FPM coefficients that act on the body seems

to agree more or less with the total lift coefficient that is broken down in this analysis. The correct frequency is observed and the amplitude of the force oscillations remains close to what is expected. However, zooming in on a 20 s range within this simulation (see [Figure 9](#)), it is found that a phase shift is present between C_L and C_B ; apparently caused by the contribution of the vorticity-induced force coefficient C_ω . Moreover, a small difference in the amplitude is also apparent.

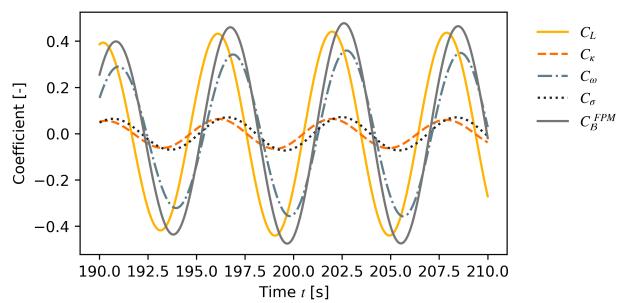
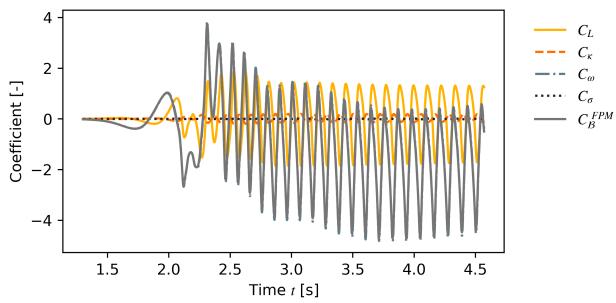


Figure 9: 20 s portion of the response with dynamic mesh at $Re = 100$.

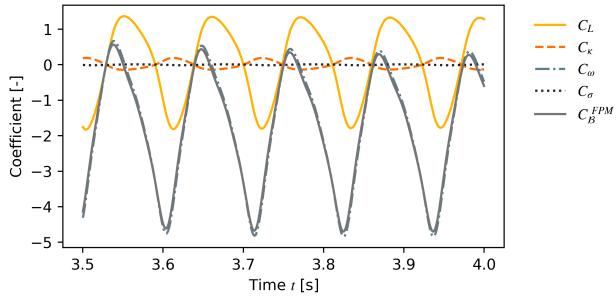
The issue becomes much more pronounced when increasing the Reynolds number to $Re = 4000$. [Figure 10](#) visualises the resulting coefficients for that case, where it becomes very obvious that the problem indeed resides in the calculation of C_ω . An offset with respect to the average lift (x-axis in the images) is now also observed for that coefficient, and its amplitude is almost twice what it is expected to be. It must be noted that the frequency of the vibration matches flawlessly, even though the phase difference is still there.

While investigating the origin of this problem, it was found that it does not occur when disabling the fluid-structure interactions; i.e. if the body is kept stationary and the simulation mesh does not update over time. Of course, a dynamic mesh is essential to accurately represent the VIV phenomenon. However, for the sake of verification for the FPM, the plots in [Figure 11](#) show results from the same simulation as before, but using a static mesh instead of the moving cylinder.

It is immediately clear that this modification eliminates the issue of phase difference. Both coefficients C_L and C_{sum} cross the x-axis simultaneously for any of the simulated Re -regimes. Moreover, the downward shift that was observed for the higher



(a)



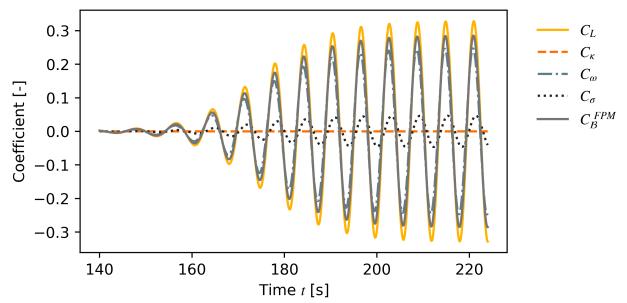
(b)

Figure 10: Lift coefficient response and its partitioning around a moving cylinder (dynamic mesh) for $Re = 4000$. (a) shows the full response and (b) zooms in on a smaller time domain.

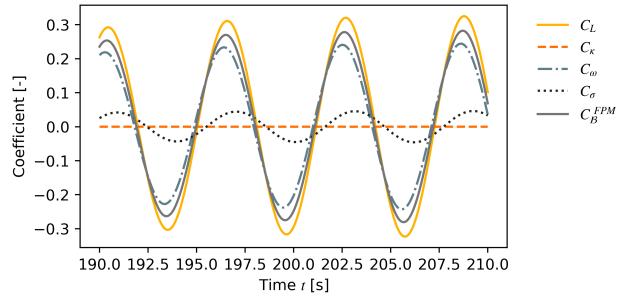
Reynolds numbers is not present anymore. The only issue that remains in this static case is the factor difference in amplitude.

The (dis)-agreement between C_L and C_B signals can be quantified using their normalised cross-correlation (NCC) (Bracewell & Bracewell, 1986). This measure is very useful when describing the delay of complicated signals, but even for these simple near-harmonic oscillations, it provides a good indication for how well the results match. In Figure 12, the NCC is reported over a range of Reynolds numbers for in the dynamic and static case. For all the simulations that exclude the movement of the tower, an NCC value above 0.99 is found, along with very little performance degradation by increasing Re . Conversely, when the dynamic mesh is reintroduced, the NCC already drops to 0.82 with $Re = 100$ and decreases further to 0.55 at larger Reynolds numbers. In conclusion, the method is valid for cases where the immersed body is stationary but should be used with care when dynamic meshes are enabled, certainly for increasing Re .

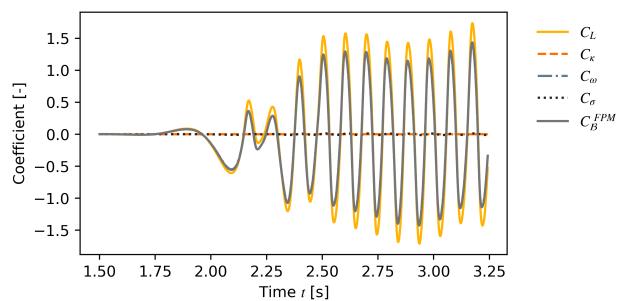
As a final note on this problem, there is a possibility that the case setup for the dynamic mesh influences the FPM calculations. In this scenario, the issue



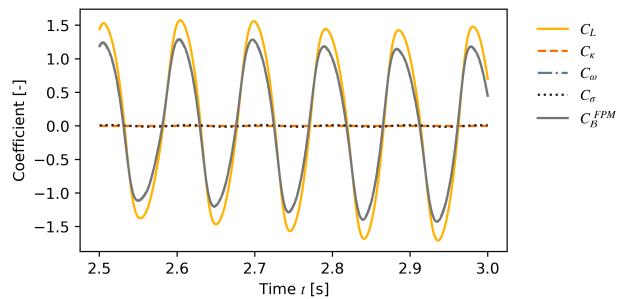
(a)



(b)



(c)



(d)

Figure 11: Lift coefficient response and its partitioning around a stationary cylinder (static mesh) for different Reynolds numbers: (a-b) $Re = 100$, (c-d) $Re = 4000$. (a, c) show the full response and (b,d) zoom in on a smaller time domain.

might be resolved by appropriately changing the configuration in the `constant/dynamicMeshDict` file of the OpenFOAM case. Investigating this influence is left for future work.

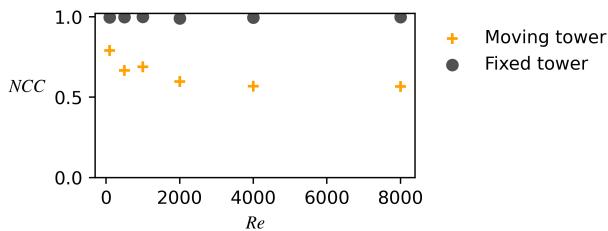


Figure 12: Correlation coefficients for static and dynamic mesh at various Reynolds numbers. The two sets of data points represent the situation with static and dynamic mesh, respectively.

4.3 Parallel Processing

For all previously presented results, the simulations were run without parallelisation. However, the correct working of the method with parallel processing is vital to its usefulness. It is very rare for this type of simulations to ever be done sequentially, as optimal use of multi-node processing is what enables simulations to be completed relatively quickly. Therefore, it is a massive drawback if one function prevents this. For the present work, it has been the priority to get as close as possible to valid results, such that the aspect of parallel performance got less attention.

Yet another comparison between C_L and the summed FPM coefficients is shown in Figure 13. The case setup for this is identical to the one that produced Figure 5, except for the fact that this simulation was distributed over four separate computation nodes, by splitting the domain and using MPI for communicating between the processors (MPI Forum, 1993).

Since the simulation setup, domain and its boundary conditions are symmetric, the lift coefficient and its contributions must be oscillating around a value of zero. At $Re = 100$, the kinematic contribution C_κ shows the same behaviour as before, however, the viscous component C_σ now has a constant offset in the negative direction and the vorticity-induced coefficient C_ω oscillates around an even more negative value. This observation suggests that the problem resides in the calculation methods for spatial derivatives. As indicated in Equation (14), only a temporal derivative is necessary to find C_κ , whereas the definition of C_σ and C_ω include the curl of the velocity and a divergence calculation. The spatial derivatives introduced by

the ∇ operator require successful communication at the boundaries between processor domains. As expected (regarding the previous discussion), increasing the Reynolds number makes the dependence on C_ω even larger, and thereby makes the mismatch worse.

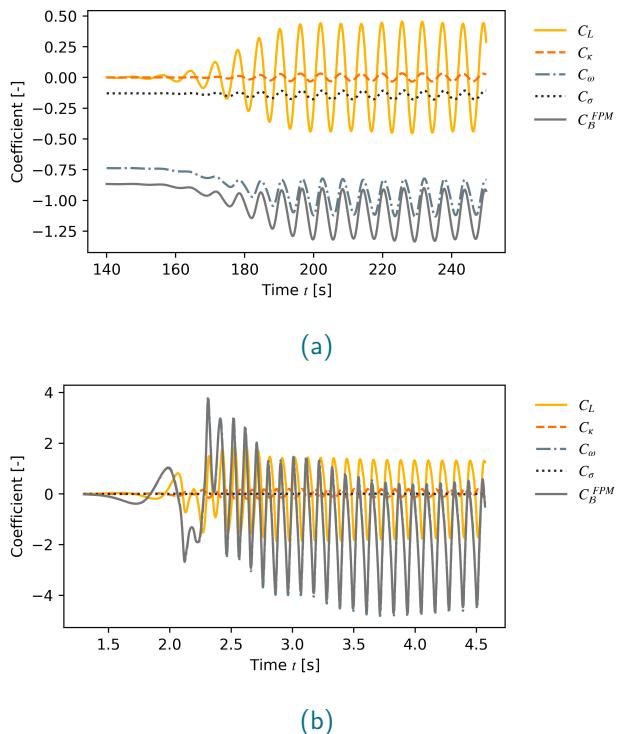


Figure 13: Full response of C_L and its partitioned coefficients using parallel processing for (a) $Re = 100$ (b) $Re = 4000$.

Remarkably, the phase shift and frequency of the oscillations are not affected; only their amplitude and absolute offset. It is strongly recommended to investigate further what causes these differences, as it might be at the basis of some of the other issues presented here.

4.4 Usability and Completeness

Several other limitations have been identified that relate to the ease of use and range of applicability that the method provides.

First of all, the current implementation requires a user to manually define the boundary conditions for the $\phi^{(i)}$ field (see subsection 2.3). However, these boundary conditions are solely determined by the direction of force partitioning, which should already be specified within the control dictionary. This is sub-optimal, as a user of the method should not be concerned with the intricacies of the calculation method. Moreover, twice specifying the

same information increases the risk of mismatched configurations and invalid results.

Furthermore, a note must be made on the completeness of the calculation. Up to this point, one force component has been neglected, as its contribution was assumed negligible. The effects of viscosity shear on the outer boundary F_Σ^i can be neglected for sufficiently large domains. Implementing the calculation should not be very complicated, but it would require the user to specify all patches of the outer boundary, which adds unnecessary work if it is discarded later anyway. Nonetheless, the calculation should be part of the method to make it complete. It is recommended to implement the functionality with the possibility to switch it on and off, to avoid useless configuration if one is not interested in the C_Σ coefficient.

Another point that leaves room for improvement is that only laminar flow regimes were used for verifying the code. The integration with turbulence models might expose additional deficiencies in the work presented here.

Finally, there is a small element in the code that prevents its use directly for 3-dimensional analysis. Currently, a single variable exists to describe the so-called reference length ℓ_{ref} . This is used both to calculate the Reynolds number and to make the coefficients dimensionless (see [Equation \(15\)](#)). As mentioned, the latter actually requires a reference area A_{ref} instead of length. However, for the present 2-dimensional analysis, unit width was assumed for the out-of-plane dimension of the nodes, such that $A_{\text{ref}} = 1 \cdot \ell_{\text{ref}}$. Apart from this, the FPM is in principle ready to be verified in a realistic 3D simulation.

5. CONCLUSION

This work presented the implementation of a method for force partitioning (FPM) in the CFD code OpenFOAM, starting from first principles up until actual use within the open-source simulation tool. A case of vortex-induced vibrations (VIV) in a 2D circular cylinder was then used for verifying the implementation.

In a simulation with forced vibrations, transverse to the free stream of the fluid, a convincing match

of amplitude and frequency was found for a range of reduced velocities, correctly identifying the frequency synchronisation within the lock-in regime.

The lift force was then partitioned using the presented FPM into contributions from five force generating mechanisms; a quantitative analysis showed that the vorticity-induced force is the driving factor for the vibrations in the cylinder; the influences of viscous and kinematic forces are also significant, but diminish with increasing Reynolds numbers. This behaviour also agrees with the literature.

When applying the FPM to the forces on a stationary cylinder, a correlation above 0.99 was found between the total lift coefficient and the sum of its FPM contributions, for all simulated Reynolds numbers. Only an amplitude error is observed in that situation. However, the implementation fails to produce valid results for higher Reynolds numbers if fluid-structure interactions are enabled through a dynamic mesh.

Some other shortcomings were identified in relation to the compatibility with parallel processing and the overall ease of use of the method. These problems should be addressed before applying the method in further OpenFOAM simulations.

Once these final issues are resolved, future work could use the method to analyse VIVs in realistic Re -regimes for wind turbine towers, including 3D effects and in-line oscillations of the towers.

DISCLAIMER

The implementation of the force-partitioning method used for this work is publicly available for download and adaptation to the needs of other research. However, it should be used with care as insufficient verification and validation were performed to ensure expected behaviour in all situations.

ACKNOWLEDGEMENTS

I want to thank my supervisor, Dr. Axelle Viré, for introducing me into the field of aerodynamic research and for her guidance and input along the way. Furthermore, I am grateful to Shyam VimalKumar for a number of very interesting discussions

and debugging sessions; hopefully he will be able to use some of this work in his PhD thesis. Finally, I thank the organisation of the honours programme for this great opportunity and learning experience.

NOMENCLATURE

CFD	= Computational Fluid Dynamics
FPM	= Force Partitioning Method
MPI	= Message Passing Interface
NCC	= Normalised Cross-Correlation
PIMPLE	= PISO + SIMPLE
PISO	= Pressure-Implicit Splitting of Operators
SIMPLE	= Semi-Implicit Method for Pressure-Linked Equations
VIV	= Vortex-Induced Vibration
\mathcal{B}	= Boundary of the imersed body
μ	= Dynamic viscosity
Φ	= Velocity potential field
$\phi^{(i)}$	= Auxiliary potential field
ρ	= Density
θ	= zenith angle in spherical coordinates
ℓ_{ref}	= Reference length of the body along the flow
C_{ref}	= Sum of all FPM coefficients
D	= Diameter
f_s^*	= Dimensionless natural frequency
f_{St}	= Frequency of vortex shedding
i	= Direction of force partitioning
k^*	= Dimensionless spring stiffness
m^*	= Dimensionless cylinder mass
p	= Pressure
R	= Body radius
Re	= Reynolds number
St	= Strouhal number
V^*	= Reduced velocity
V_∞	= Free stream velocity
r	= Radial coordinate
\vec{A}	= Arbitrary vector potential field
\vec{n}	= Normal vector pointing out of the surface
\vec{u}	= Velocity
$\vec{\omega}$	= vorticity

LIST OF REFERENCES

- Achenbach, E. (1968). Distribution of local pressure and skin friction around a circular cylinder in cross-flow up to $Re = 5:106$. *Journal of Fluid Mechanics*, 34, 625–639. <https://doi.org/10.1017/S0022112068002120>
- Ahrens, J., Geveci, B., handbook, C. L. T. v., & 2005, u. (2015). Paraview: An end-user tool for large data visualization. *data-science.dsccale.org*.
- Anagnostopoulos, P., & Bearman, P. (1992). Response characteristics of a vortex-excited cylinder at low reynolds numbers. *Journal of Fluids and Structures*, 6(1), 39–50. Cited By 188. [https://doi.org/10.1016/0889-9746\(92\)90054-7](https://doi.org/10.1016/0889-9746(92)90054-7)
- Batchelor, G. K. (2000). An introduction to fluid dynamics. *An Introduction to Fluid Dynamics*. <https://doi.org/10.1017/CBO9780511800955>
- Bishop, R. E. D., & Hassan, A. Y. (1964). The lift and drag forces on a circular cylinder oscillating in a flowing fluid. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 277, 51–75. <https://doi.org/10.1098/rspa.1964.0005>
- Bracewell, R., & Bracewell, R. (1986). *The Fourier transform and its applications*, vol. 31999. New York: McGraw-Hill. <https://www.academia.edu/download/44001876/34957138.pdf>
- Chang, C. C. (1992). Potential flow and forces for incompressible viscous flow. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 437, 517–525. <https://doi.org/10.1098/rspa.1992.0077>
- Feng, C. C. (1968). The measurement of vortex induced effects in flow past stationary and oscillating circular and d-section cylinders. Tech. rep., University of British Columbia. PhD Thesis.
- Govardhan, R., & Williamson, C. H. (2000). Modes of vortex formation and frequency response of a freely vibrating cylinder. *Journal of Fluid Mechanics*, 420, 85–130. <https://doi.org/10.1017/S0022112000001233>
- Holzmann, T. (2019). *Mathematics, Numerics, Derivations and OpenFOAM®*. Holzmann CFD, 7 ed. https://www.researchgate.net/publication/307546712_Mathematics_Numerics_Derivations_and_OpenFOAMR
- Issa, R. I. (1986). Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62(1), 40–65. [https://doi.org/10.1016/0021-9991\(86\)90099-9](https://doi.org/10.1016/0021-9991(86)90099-9)
- Jasak, H., & Gosman, A. D. (2001). Residual error estimate for the finite-volume method. *Numerical Heat Transfer, Part B: Fundamentals*, 39(1), 1–19. <https://doi.org/10.1080/104077901460650>
- Khalak, A., & Williamson, C. H. (1999). Motions, forces and mode transitions in vortex-induced vibrations at low mass-damping. *Journal of Fluids and Structures*, 13, 813–851. <https://doi.org/10.1006/jfls.1999.0236>
- Lehmkuhl, O., Rodríguez, I., Borrell, R., Chiva, J., & Oliva, A. (2014). Unsteady forces on a circular cylinder at critical reynolds numbers. *Physics of Fluids*, 26. <https://doi.org/10.1063/1.4904415>
- Livanos, D. (2018). Investigation of vortex induced vibrations on wind turbine towers. Tech. rep., Delft University of Technology. MS Thesis.
- Meneghini, J., & Bearman, P. (1995). Numerical simulation of high amplitude oscillatory flow about a circular cylinder. *Journal of Fluids and Structures*, 9(4), 435–455. Cited By 130. <https://doi.org/10.1006/jfls.1995.1025>

- Menon, K., & Mittal, R. (2020). On the initiation and sustenance of flow-induced vibration of cylinders: insights from force partitioning. *Journal of Fluid Mechanics*, 907, A37. <https://doi.org/10.1017/jfm.2020.854>
- Mittal, S., & Singh, S. (2005). Vortex-induced vibrations at subcritical re. *Journal of Fluid Mechanics*, 534, 185–194. Cited By 66. <https://doi.org/10.1017/S0022112005004635>
- Morse, T. L., & Williamson, C. H. (2009). Prediction of vortex-induced vibration response by employing controlled motion. *Journal of Fluid Mechanics*, 634, 5–39. <https://doi.org/10.1017/S0022112009990516>
- MPI Forum (1993). MPI. *Proceedings of the 1993 ACM/IEEE conference on Supercomputing - Supercomputing '93*, (pp. 878–883). <https://doi.org/10.1145/169627.169855>
- OpenFOAM (2020). Openfoam. [Computer Software]. <https://www.openfoam.com/>
- Patankar, S. (1981). Numerical heat transfer and fluid flow. *Hemisphere Publishing Corporation*, 53(3), 225–225. <https://doi.org/10.1002/CITE.330530323>
- Quartapelle, L., & Napolitano, M. (1983). Force and moment in incompressible flows. *AIAA Journal*, 21, 911–913. <https://doi.org/10.2514/3.8171>
- Sarpkaya, T., Sarpkaya, & T. (1978). Fluid forces on oscillating cylinders. *STIA*, 104, 275–290.
- Schewe, G. (1983). On the force fluctuations acting on a circular cylinder in crossflow from subcritical up to transcritical reynolds numbers. *Journal of Fluid Mechanics*, 133, 265–285. <https://doi.org/10.1017/S0022112083001913>
- Singh, S. P., & Mittal, S. (2005). Flow past a cylinder: Shear layer instability and drag crisis. *International Journal for Numerical Methods in Fluids*, 47, 75–98. <https://doi.org/10.1002/fld.807>
- Van Hulle, S. (2022). Force partitioning method. [Source Code]. https://gitlab.com/simonvanhulle01/HP_B_VIV.git
- Viré, A., Derksen, A., Folkersma, M., & Sarwar, K. (2020). Two-dimensional numerical simulations of vortex-induced vibrations for a cylinder in conditions representative of wind turbine towers. *Wind Energy Science*, 5, 793–806. <https://doi.org/10.5194/wes-5-793-2020>
- White, F. M. (2003). *Fluid Mechanics*. WCB McGraw-Hill . <https://books.google.com/books/about/Fluid.Mechanics.html?id=egk8SQAACAAJ>
- Williamson, C., & Govardhan, R. (2004). Vortex-induced vibrations. *Annual Review of Fluid Mechanics*, 36, 413–455. Cited By 1587. <https://doi.org/10.1146/annurev.fluid.36.050802.122128>
- Zhang, C. (2015). MECHANISMS FOR AERODYNAMIC FORCE GENERATION AND FLIGHT STABILITY IN INSECTS. Tech. rep., John Hopkins University. [PhD Thesis]. <https://jscholarship.library.jhu.edu/handle/1774.2/37840>
- Zhang, C., Hedrick, T. L., & Mittal, R. (2015). Centripetal Acceleration Reaction: An Effective and Robust Mechanism for Flapping Flight in Insects. *PLOS ONE*, 10(8), e0132093. <https://doi.org/10.1371/JOURNAL.PONE.0132093>
- Zheng, C., Liu, Z., Wu, T., Wang, H., Wu, Y., & Shi, X. (2019). Experimental investigation of vortex-induced vibration of a thousand-meter-scale mega-tall building. *Journal of Fluids and Structures*, 85, 94–109. Cited By 11. <https://doi.org/10.1016/j.jfluidstructs.2018.12.005>