Final Technical Report

Federal Agency to which Report is submitted: DOE EERE – Wind & Water Power Program

Recipient:	Regents of the University of Minnesota
Award Number:	DE-EE0005482
Project Title:	High-resolution computational algorithms for simulating offshore wind turbines and farms: Model development and validation
Project Period:	Sept. 30, 2011 – Sept. 30, 2015

Principle Investigator: Fotis Sotiropoulos, fotis.sotiropoulos@stonybrook.edu, 631-632- 8380

Report Authors:
Antoni Calderer, cald0075@umn.edu
Xiaolei Yang, xyang@umn.edu
Dionysios Angelidis, dangelid@umn.edu
Christ Feist, feist020@umn.edu
Michele Guala, mguala@umn.edu
Kelley Ruehl, kmruehl@sandia.gov
Xin Guo, XinGuo@jhu.edu
Aaron Boomsma, booms002@umn.edu
Lian Shen, shen@umn.edu
Fotis Sotiropoulos, fotis.sotiropoulos@stonybrook.edu

Date of Report:	October 30, 2015
Covering Period:	Sept. 30, 2011 – Sept. 30, 2015

Working Partners: Sandia National Laboratories

DOE Project Team:
DOE HQ Program Manager – Jose Zayas
DOE Field Contract Officer – Pamela Brodie
DOE Grants Management Specialist – Jane Sanders
DOE Technical Project Officer – Gary Nowakowski
DOE Technical Monitor – Stephanie Hodge

Acknowledgment and Disclaimer

Acknowledgment
This report is based upon work supported by the U. S. Department of Energy under Award No. DE-EE0005482.

Contents

Executive Summary

The present project involves the development of modeling and analysis design tools for assessing offshore wind turbine technologies. The computational tools developed herein are able to resolve the effects of the coupled interaction of atmospheric turbulence and ocean waves on aerodynamic performance and structural stability and reliability of offshore wind turbines and farms. Laboratory scale experiments have been carried out to derive data sets for validating the computational models.

Subtask 1.1 Turbine Scale Model
A novel computational framework for simulating the coupled interaction of complex floating structures with large-scale ocean waves and atmospheric turbulent winds has been developed. This framework is based on a domain decomposition approach coupling a large-scale far-field domain, where realistic wind and wave conditions representative from offshore environments are developed, with a near-field domain, where wind-wavebody interactions can be investigated. The method applied in the near-field domain is based on a partitioned fluid-structure interaction (FSI) approach combining the curvilinear immersed boundary (CURVIB) method with a two-phase flow level set formulation and is capable of solving free surface flows interacting non-linearly with floating wind turbines. The FSI-Levelset code was validated by carrying out simulations for various simple floating bodies as well as for a wedge falling freely across the air/water interface. The computed results were in excellent agreement with measurements available in the literature. In particular the model was shown to capture complex wave phenomena, such as breaking waves and structure overtopping. For coupling the far-field and near-field domains, a wave generation method for incorporating complex wave fields into Navier-Stokes solvers has been proposed. The wave generation method was validated for a variety of wave cases including a broadband spectrum. The computational framework has been further validated for wave-body interactions by replicating the experiment of floating wind turbine model subject to different sinusoidal wave forces (task 3). The simulation results, which agree well with the experimental data, have been compared with other numerical results computed with available numerical codes based on lower order assumptions. Despite the higher computational cost of our method, it yields to results that are in overall better accuracy and it can capture many additional flow features neglected by lower order models. Finally, the full capabilities of the framework have been demonstrated by carrying out large eddy simulation (LES) of a floating wind turbine interacting with realistic ocean wind and wave conditions.

Subtask 1.1 Farm Scale Model
Several actuator type models were implemented in the computational framework allowing to simulate turbines and wind farms. In addition, the capability to simulate wind farms using large-eddy simulation loosely coupled with mesoscale simulations was implemented.

The wave effects in farm scale simulations can be included in two different approaches: (1) The fully-coupled non-linear approach, which is a straightforward extension of the two-phase flow level-set approach developed in task 1.1; (2) The one-way coupled, linear approach, in which the waves are decoupled from the rest of the problem. This was achieved with the same far-field/near-field coupling approach developed in task 1.1, but prescribing the wave field in the near-field solver using the Immersed Boundary method.

The actuator type models, actuator disk and actuator line, were validated extensively against SAFL wind tunnel measurements for a single wind turbine and wind-turbine arrays [1], SAFL main channel measurements for a hydrokinetic turbine [2], field measurements from Mower County wind farm and EOLOS wind turbine [1]. Overall good agreement with the measurements is obtained.

In the hydrokinetic turbine case, it was found that the nacelle has an important effect on the inner and outer shear layer interaction and the far wake turbulence intensities. However, the nacelle model in the literature cannot capture such effect. An improve actuator surface model for nacelle was proposed and applied to simulate the flow over a nacelle of a hydrokinetic turbine. Promising results on a very coarse grid (10 cells per diameter) were obtained in comparison with the results from a high-resolution (50 cells per diameter) immersed boundary simulation.

A turbine torque controller was implemented for the actuator line model. Validation was performed on the University of Minnesota EOLOS turbine for both uniform and turbulent inflow conditions. The computed power coefficient curve showed good agreement with that calculated from the FAST model (in which the blade element theory is employed) for the case with uniform inflow. Good agreement with the field measurement was also obtained for the computed power for the case with turbulent inflow.

In order to simulate the atmospheric stratification effects on the turbine and wind farm performance, temperature equation was solved with wall function applied at the ground. Validations were performed on turbulent channel flow with temperature differences at two walls.

Loosely coupling of the high-resolution large-eddy simulation (LES) with a meso-scale model WRF (Weather Research and Forecasting Model) was implemented. In this loose coupling algorithm, the flow field from WRF with added synthetic turbulence was employed as inflow conditions for the LES domain. This loosely coupling was applied to the simulation of the turbines in Mower County wind farm. The computed power agrees well with the field measurements. Besides the model wind turbine, hydrokinetic turbine, Mower County wind farm and EOLOS turbine cases, the present model was also applied to: (1) Simulation of a miniature wind turbine in the downstream of a three-dimensional hill with overall good agreements with the measurements obtained at SAFL wind tunnel (Yang et al. 2015); (2) Simulation of wind field at Prairie Island, MN, with/without hypothetically installed wind turbines (Yang et al. 2014); (3) Simulation of SWiFT (Scaled Wind Farm Technology) turbines with two different designs, which are located at the Reese Technology Center near Lubbock, TX, USA); (4) Simulation the three SWiFT wind turbines under three different wind directions and two different tip speed ratios.

Task 1.3 Adaptive Mesh Refinement
Performing high-fidelity numerical simulations of turbulent flow in multi-turbine wind farms remains a challenging issue mainly because of the large computational resources required to accurately simulate the large disparity of spatial scales. To address this challenge we develop herein a new Adaptive Mesh Refinement (AMR) flow solver to enhance the resolution and improve the efficiency of the Virtual Flow Simulator (VFS) code ([3], previously named VWIS), which is capable of simulating multi-turbine wind farms in complex terrain. We extend the Curvilinear Immersed Boundary (CURVIB) approach incorporated in the VFS code to unstructured Cartesian grids with strong coupling between multiple levels of refinement. The challenging issues of flux mismatching or pressure discontinuity across fine/coarse interfaces are overcome by the resulting fully unstructured approach. The efficiency and accuracy of the solver is demonstrated by solving the Navier-Stokes equations in driven cavity flows. Large-eddy simulation (LES) of turbulent flows past a stand alone wind turbine, which is modelled by using the Actuator Line Model (ALM), reveal that computed results obtained in locally refined domains are in good agreement with the experimental measurements. These simulations also show the ability of our method to simulate the rich dynamics on the wake of the turbine.

Task 2 Massively parallel implementation of the computational models
Using the RedSky/RedMesa supercomputers at Sandia National Labs, we have been able to simulate the fluid dynamics at the SWiFT site with very high fidelity. To achieve this, VFS was first ported to the Red Sky cluster, then initial simulations of the site, along with a grid refinement study, was performed. In order to efficiently utilize Sandia's computational resources, UMN researchers have been focusing on improving VFS scalability. First, a dedicated application testing time was acquired on RedMesa, where a strong scaling of approximately $20-30\%$ was achieved. Then, using this initial test and collaboration with Sandia, the Poisson solver preconditioner was modified, which increased the strong scaling to about $40-50\%$.

Task 2 Experiments
Two set of experiments in the St. Anthony Falls Laboratory wave tank facility and atmospheric wind tunnel are integrated to provide a scaled representation of a floating wind turbine under heave and pitch motions due to ocean waves. The quasi-coupling is established by controlling the turbine rotor speed to generate a thrust force mimicking steady or fluctuating wind gusts in the wave tank, and by using two actuators to oscillate a miniature turbine in the wind tunnel. Measured pitch and heave motions under varying waves are scaled down using rotor geometry and the wake meandering frequency to study the effect of the floating platform kinematics on the evolution and characteristics of the oscillating turbine wake. Results provide a phenomenological

description of the floating-turbine system under variable waves and wind gusts in a fully developed turbulent boundary layer. As compared to land-based turbines, i) wind gusts contribute to increased pitch range depending on the local interaction with the incoming waves, ii) periodic large scale flow patches of high and low momentum are generated by the oscillating rotor and advected in the wake. Both mechanisms could contribute to amplify the pitch response of downwind floating units within the offshore power plant, in particular if the wave and/or wind forcing frequencies approach the pitch natural frequency of the floating system. However, such criticality is not identified under a pure heaving motions because, in such conditions, the turbine wake is weakly modulated, and only near the blade tip region.

Public Release of the Computational Framework
The code has been released under the GNU General Public License (GPL 2.0) in the following Github repository: https://github.com/SAFL-CFD-Lab/VFS-Wind. The code comes with a detailed users manual and a set of instructional test cases.

1 Introduction

The high potential of floating wind turbines to capture part of the vast offshore wind energy resource has attracted increasing attention by the scientific community. High-fidelity numerical simulations can play a major role in the development of novel offshore wind technologies, and may be the only feasible way to tackle such a problem. However, the complexity of the problem poses a major challenge due to the need to resolve the coupled interaction of atmospheric turbulence and ocean waves, the arbitrary geometric complexity of floating structures, the inherent two-phase nature of such flows, and the dominant role of complex nonlinear phenomena such as turbulence and free surface effects. Due to these complexities most numerical methods currently rely on oversimplified assumptions. The methods for simulating floating wind turbines can be classified in two broad categories depending on the approach to solve the rigid body equations of motion (EoM) of the floating structure, frequency domain methods or time domain methods. Due to its computational expedience, frequency domain methods are usually applied to early feasibility studies (see [4, 5]). However, they are very limited in accuracy as they cannot account for key features of the problem, such as transients of the problem, non-linearities, and wave radiation and diffraction. Several commercial packages such as WAMIT, ANSYS-AQWA, AQUADYN and AQUAPLUS, have a frequency domain boundary element method (BEM) solvers. Time domain solvers, on the other hand, can account for transients and non-linear effects, and thus deliver an overall better agreement, although the computational cost is considerably higher.

The degree of fidelity of a time domain solver is determined by the approach used to handle the fluid-body interactions. The most simplified approach is to model the wave induced forces with the semi-empirical Morison equations, as in the work of [6]. The Morison equations are based on very rough assumptions that makes it only valid for studying structures with a cylindrical and slender shape. Alternatively, one can use a BEM code to solve the boundary value problem derived from linear potential theory ([7, 8, 9]). BEM codes assume low amplitude motions and inviscid and irrotational flows. This assumptions limit the applicability of the method to cases with mild wave conditions in which the wave slope is limited. Also, this approach cannot be used for investigating the flow around the floating structure as the main flow features, such as eddies, are neglected. Finally, the most physically accurate approach is the so called Computational Fluid Dynamics (CFD)-FSI. The Navier-Stokes equations are solved to obtain the flow field around the structure which is used to compute the pressure and shear stresses at the body surface. The added accuracy and the ability to capture a realistic flow field may be critical for the development and optimization of new platform concepts and designs. Obviously, the disadvantage is in the higher computational cost.

Most CFD codes for simulating FSI of geometrically complex floating structures are based on a single phase flow formulation, in which only the water domain is considered and cannot account for the wind field (see for example [10, 11, 12]). Only few CFD-FSI codes for two-phase flows are available in the literature (see for example Sanders et al. [13] and Shen and Chan [14]), however, they have not been applied to three-dimensional (3D) applications. A relevant work is that from Yang and Stern (2009) who proposed a method to perform large-eddy simulation (LES) of two-phase flows interacting with complex moving bodies. In this work, a sharp interface immersed boundary (IB) method is used for tracking the body motion and a sharp interface level set method is used for the free surface interface. The method was validated and applied to simulate several free surface-body interaction problems such as rotating ellipses, or the motion of a ship. The limitation of this method is that has not been extended to FSI problems.

The proposed research consists of three distinct components: 1) Development of multi-scale and multi-physics computational models for offshore wind turbines and farms; 2) Implementation of these models in computer codes that take full advantage of massively parallel computational platforms; and 3) validation of the models with laboratory and field-scale experiments.

2 Task 1.1 The Turbine Scale Model

2.1 Goals and Objectives

The goal in this subtask is to extend the Curvilinear Immersed boundary (CURVIB) method to simulate in high resolution an offshore floating wind turbine considering the physics of the air-water interface, the effects of waves and their impact on the floating structure, and the six degree of freedom (6DOF) dynamics of the floating structure.

2.2 Project team

Fotis Sotiropoulos James L. Record Professor, Director of the St. Anthony Falls Laboratory. Principal investigator of the project.

Lian Shen Benjamin Mayhugh Associate Professor in the Mechanical Engineering deparment of the University of Minnesota. supervising and guiding on the work related to far-field/near-field coupling in subtask 1.1.

Antoni Calderer PhD student in the Saint Anthony Falls Laboratory. Mr. Calderer worked on the turbine geometry resolving model (subtask 1.1). He obtained a PhD as a result of this work.

Xin Guo Post-doctoral researcher at the St. Anthony Falls Laboratory. Dr Guo assisted in the wave modeling part of this work.

Kelley Ruehl Ocean Renewable Energy Engineer at Sandia National Labs. Mrs. Ruehl is the coordinator from Sandia National Labs, providing her expertise in ocean engineering to the project, both in the numerical and experimental side. Performed simulations using engineering level models.

2.3 Technical Approach

The numerical framework couples an efficient large-scale model, which is referred in this work to as the far-field flow solver, and is suitable for simulating realistic ocean wave and wind conditions, with a high resolution near-field model capable of solving complex free-surface flows interacting non-linearly with arbitrarily complex real life floating structures.

2.3.1 The two-phase Navier-Stokes equations

The near-field model solves the spatially-filtered Navier-Stokes equations governing incompressible flows of two immiscible fluids in non-orthogonal generalized curvilinear coordinates. We adopt the two-fluid, level set formulation of Kang and Sotiropoulos [15], where a single equation is used in all the computational domain taking the corresponding fluid properties values in each fluid phase.

Using compact Newton notation, where repeated indices imply summation, the equations read as follows ($i, j, k, l = 1, 2, 3$):

$$J \frac{\partial U^j}{\partial \xi^j} = 0, \tag{1}$$

$$\begin{aligned} \frac{1}{J}\frac{\partial U^i}{\partial t} = & \frac{\xi_l^i}{J}\left(-\frac{\partial}{\partial \xi^j}\left(U^j u_l\right) + \frac{1}{\rho\left(\phi\right) Re}\frac{\partial}{\partial \xi^j}\left(\mu\left(\phi\right)\frac{\xi_l^j \xi_l^k}{J}\frac{\partial u_l}{\partial \xi^k}\right) - \frac{1}{\rho\left(\phi\right)}\frac{\partial}{\partial \xi^j}\left(\frac{\xi_l^j p}{J}\right) - \right. \\ & \left. -\frac{1}{\rho\left(\phi\right)}\frac{\partial \tau_{lj}}{\partial \xi^j} - \frac{\kappa}{\rho(\phi) We^2}\frac{\partial h(\phi)}{\partial \xi^j} + \frac{\delta_{i2}}{Fr^2} + S_l^w + S_l^s + S_l^{AL}\right), \end{aligned} \tag{2}$$

In the above equations: ϕ is the level set function (see below for details), ξ^k are the curvilinear components, ξ_l^i are the transformation metrics, J is the Jacobian of the transformation, U^i are the contravariant volume fluxes, u_i are the Cartesian velocity components, ρ is the density, μ is the dynamic viscosity, p is the pressure, τ_{lj} is the sub-grid scale (SGS) tensor, κ is the curvature of the interface, δ_{ij} is the Kronecker delta, h is the smoothed Heaviside function, S_i^w is the source term for wave generation, S_i^s is the source term for wave dissipation, S_l^{AL} is the actuator line body force, and Re, Fr, and We are the dimensionless Reynolds, Froude, and Weber numbers, respectively, defined as follows:

$$Re = \frac{UL\rho_{water}}{\mu_{water}}, Fr = \frac{U}{\sqrt{gL}}, We = U\sqrt{\frac{\rho_{water}L}{\sigma}} \tag{3}$$

where, U and L are the characteristic velocity and linear dimension, ρ_{water} and μ_{water}, are the density and dynamic viscosity of the water phase, g is the gravity, and σ is the surface tension. The subgrid scale stress tensor τ_{lj} in Eq. (2) is modeled in the present work as described in [16] using the dynamic Smagorinsky eddy viscosity model of [17].

In the level set method, the interface is tracked using the signed distance function $\phi(x, t)$, also known as the level set function, which is an scalar function defined in the whole computational domain, measuring the minimum distance from any point x in the fluid to the closest point of the free surface interface. The interface is located at the level $\phi = 0$, and the sign is positive in the liquid phase, and negative in the gas phase.

The jump condition of the density and viscosity fields at the interface in a level set approach is taken to be continuous, and is smeared over a thin layer of thickness 2ϵ to prevent the formation of numerical instabilities. It can be expressed as follows:

$$\rho\left(\phi\right) = \rho_{air} + \left(\rho_{water} - \rho_{air}\right) h\left(\phi\right), \tag{4}$$

$$\mu\left(\phi\right) = \mu_{air} + \left(\mu_{water} - \mu_{air}\right) h\left(\phi\right), \tag{5}$$

where the smoothed Heaviside function [18] $h(\phi)$ is

$$h(\phi) = \begin{cases} 0 & \phi < -\epsilon, \\ \frac{1}{2} + \frac{\phi}{2\epsilon} + \frac{1}{2\pi}\sin(\frac{\pi\phi}{\epsilon}) & -\epsilon \leq \phi \leq \epsilon, \\ 1 & \epsilon < \phi, \end{cases} \tag{6}$$

Typical values for ϵ are between one and three times the length of the smallest grid cell. Our experience shows, however, that for problems in which the interface undergoes rapid deformations due to complex phenomena such us air/water entrainments or wave breaking, it is necessary to employ larger values of ϵ that can be up to the length of six or eight grid cells for the most extreme scenarios. The need for an increased number of grid cells within the transition region has been investigated and discussed in [19, 20]. The study of Iafrati and Campana[19] shows that spurious velocity effects are introduced in the interior of the transition layer caused by the smearing of the interface. If the thickness ϵ of the layer is sufficiently large the spurious effects remain confined within the transition layer without altering the exterior flow. It was also shown that larger interface thickness is required for increased Reynolds number flows. The kinematic and dynamic boundary conditions ensuring continuity of the velocity and the normal and tangential stresses at the interface, are intrinsic in the current formulation and are satisfied in a smooth manner.

The momentum equations are discretized using a second-order central differencing scheme for the diffusion and advective terms, except in regions such as the vicinity of the free surface interface where a third-order WENO scheme is applied for the advective terms. The solution is advanced in time by using a second-order Crank-Nicholson scheme and the fractional step method.

2.3.2 The Level set equations

The motion of the free surface interface can be modeled by the level set method proposed by Osher and Sethian [21]. The spatially filtered advection equations in generalized curvilinear grids will assume the form:

$$\frac{1}{J}\frac{\partial \phi}{\partial t} + U^j \frac{\partial \phi}{\partial \xi^j} = -\tau^L_{ij}, \tag{7}$$

where τ^L_{ij} is the sub-grid scale stress tensor responsible of the effect of the unresolved subgrid scales on the level set field. In the present model the effect of τ^L_{ij} is neglected assuming that the residual field of ϕ is small and its overall contribution to the energy containing scales is negligible.

As Eqn. (7) is integrated in time to determine ϕ, there is no guarantee that the resulting solution will satisfy the required, for a distance function, unit gradient condition $|\nabla\phi| = 1$. Such incosistent solution will in turn lead to poor conservation of mass between the two fluids. This problem is circumvented by solving the following mass conserving re-initialization equation proposed by Sussman and Fatemi [22]:

$$\frac{\partial \phi}{\partial \tau} + S(\phi_0)(|\nabla\phi| - 1) = \lambda\tilde{\delta}(\phi)\,|\nabla\phi|\,, \tag{8}$$

where τ denotes a pseudo-time, ϕ_0 the distance function at the initial step of the pseudo-time iteration procedure, $S(\phi_0)$ is the smoothed sign function defined as

$$S(\phi_0) = \begin{cases} 1 & \phi_0 \geq \epsilon, \\ -1 & \phi_0 \leq -\epsilon, \\ \frac{\phi_0}{\epsilon} - \frac{1}{\pi}\sin\left(\frac{\pi\phi_0}{\epsilon}\right) & \text{otherwise,} \end{cases} \tag{9}$$

$\tilde{\delta}(\phi)$ is the smoothed delta function defined as $\tilde{\delta}(\phi)$ is the smoothed delta function given as

$$\tilde{\delta}(\phi) = \begin{cases} \frac{1}{2\epsilon}\left(1 + \cos\left(\frac{\pi\phi}{\epsilon}\right)\right) & |\phi| \leq \epsilon, \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

and,

$$\lambda = -\frac{\int_\Omega \tilde{\delta}(\phi) S(\phi_0)(1 - |\nabla\phi|) d\Omega}{\int_\Omega \tilde{\delta}^2(\phi) \, |\nabla\phi| \, d\Omega}, \tag{11}$$

being Ω the volume of a grid cell. A detailed description of the method in the context of curvilinear coordinates can be found in Kang and Sotiropoulos [15].

The level set equations are discretized with a third-order WENO scheme in space, and second-order Runge-Kutta in time. The re-initialization step uses a second-order ENO scheme

2.3.3 Equations of motion for rigid bodies

For computing the general 6 degrees of freedom (DoF) motion of 3D rigid bodies we use the Lagrangian form of Newton's second law, i.e. the linear and angular momentum equations. With no loss of generality we write the equations for a single body, although the formulation can be extended to multiple bodies, expressed along the principle axes. Under the above assumptions, the general form of the equations for a body mounted on an elastic and damped system can be written in the inertial frame of reference in the following form (i=1,2,3):

$$M\frac{\partial^2 X^i}{\partial t^2} + b_{t,i}\frac{\partial X^i}{\partial t} + k_{t,i}X^i = F_{fluid,i} + F_{ext,i} \tag{12}$$

$$I_i\frac{\partial^2 \Theta^i}{\partial t^2} + b_{r,i}\frac{\partial \Theta^i}{\partial t} + k_{r,i}\Theta^i = M_{fluid,i} + M_{ext,i} \tag{13}$$

where Eq. (12) represents the pure translation motion, and X^i are the components of the position vector, M is the mass of the structure, $b_{t,i}$ the damping coefficients, $k_{t,i}$ the spring stiffness coefficients, $F_{fluid,i}$ the components of the force exerted by the fluid, and $F_{ext,i}$ the external forces. The case of pure rotation is represented by Eq. (13), and Θ^i denote the components of the relative angle of rotation vector, I_i the moment of inertia, $b_{r,i}$ the damping coefficients, $k_{r,i}$ the spring stiffness coefficients, and $M_{fluid,i}$ and $M_{ext,i}$ the moments in respect to the axis of rotation exerted respectively by the fluid, and external forces. For 6 DoF motions the two sets of the equations, linear and angular, have to be solved concurrently.

The solid and fluid domains are coupled together via the Dirichlet boundary condition that the fluid velocity field must satisfy on the surface Γ of the body, as follows:

$$u_i = \frac{\partial X^i}{\partial t} + \varepsilon_{ijk} r_j \frac{\partial \Theta^k}{\partial t} \tag{14}$$

The system of second order ordinary differential equations (12) and (13) that govern the motion of the structure is solved by first transforming it into a system of first order ordinary differential equations. Then, it is integrated in time (see [23] for details).

2.3.4 Solution of the equations of motion for rigid bodies

The forces $F_{fluid,i}$ and moments $M_{fluid,i}$ that the fluid exerts to the rigid body and appear in the right hand side of the structural equations of motion (eqns. 12 and 13) are computed by integrating the pressure and the viscous stresses along the surface Γ of the body as follows:

$$F_{fluid,i} = \overbrace{\int_\Gamma -p n_i d\Gamma}^{F^i_{f,p}} + \overbrace{\int_\Gamma \tau_{ij} n_j d\Gamma}^{F^i_{f,s}} \tag{15}$$

$$M_{fluid,i} = \overbrace{\int_\Gamma -\varepsilon_{ijk} r_j p n_k d\Gamma}^{M^i_{f,p}} + \overbrace{\int_\Gamma \varepsilon_{ijk} r_j \tau_{kl} n_l d\Gamma}^{M^i_{f,s}} \tag{16}$$

where p denotes the pressure, τ the viscous stress, r the position vector, and n the normal vector. The subscripts p and s in the terms in right hand side of the above equations identify contributions from the pressure and shear forces or moments.

For IB methods in general the computation of the forces and moments in equations (15) and (16) cannot be directly performed. The mesh of the fluid domain does not conform with the structure, hence, the pressure and velocity gradients are not known on the surface of the body. A technique that was used in [23] to remedy this situation is to employ an integration surface Γ_1 (see figures 1 and 2) that encloses the body and is defined by the fluid nodes immediately adjacent to the IB nodes. At the nodes defining this surface both the pressure and velocity gradients can be calculated and thus the forces $F^i_{f,1}$ and moments $M^i_{f,1}$ can be obtained. This technique introduces an error inherent to immersed boundary methods, but, as shown in [24], the error approaches zero as the grid resolution increases.

Let Γ be the actual surface of the body and Γ_1 the aforementioned approximate surface surrounding the solid body. Let m_{IB} be the mass of the fluid delimited by Γ and Γ_1. While in previous studies for single phase problems this mass was neglected, it can be important when the equations are formulated for two-phase flow and include a gravitational force. For instance, let us assume that we have a body submerged in stagnant water. From Archimedes's principle we can easily see that if we integrate the

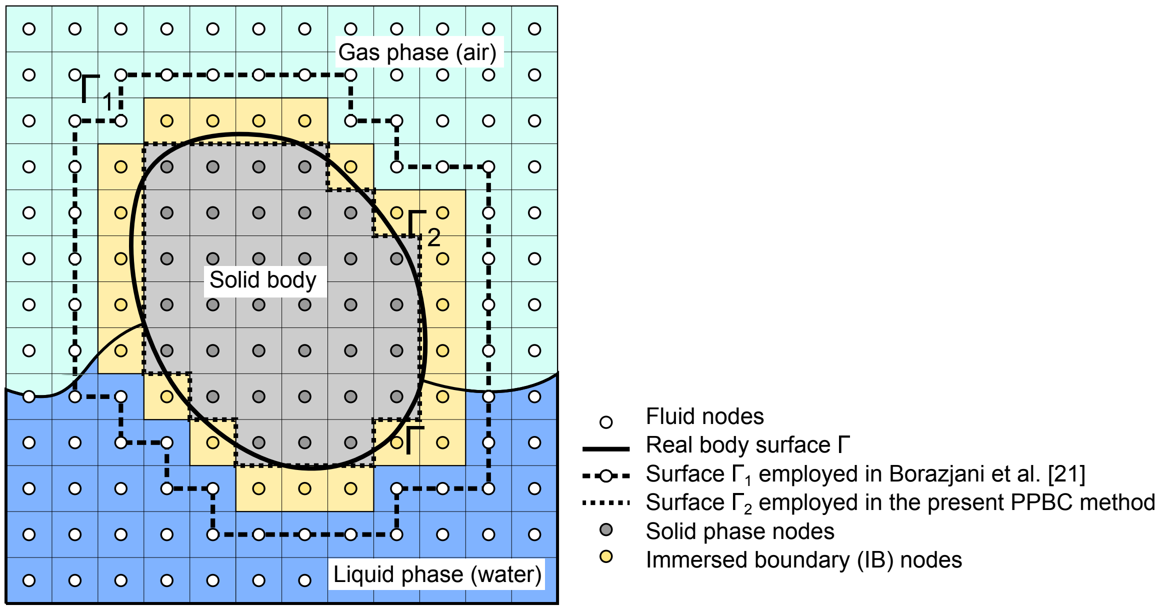


Figure 1: Schematic representation of the node classification in the CURVIB method. The surfaces Γ, Γ_1, and Γ_2 are the actual surface of the body and various approximate surfaces on which the pressure field can be integrated to calculate the pressure force acting on the body.

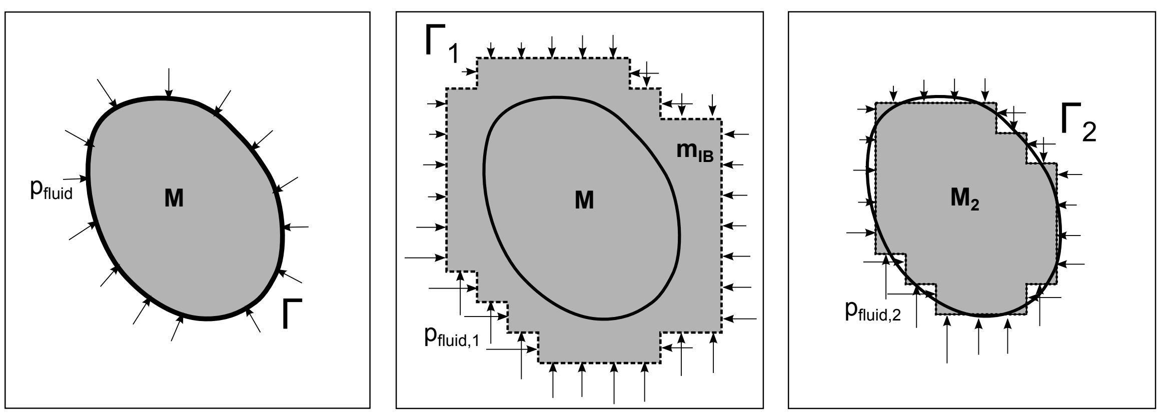


Figure 2: Schematic representation of the various approaches for calculating the pressure force by integrating the pressure field on: 1) the actual body surface Γ; 2) the surface Γ_1 outlining the volume defined by the IB cells, which was employed in Borazjani et al. [23]; and 3) the approximate surface Γ_2 employed in the proposed PPBC approach. M is the exact or approximate (M_2) mass of the body for each case.

forces along a surface larger than the actual body surface, such as Γ_1, the resulting buoyant force will be higher than the real force felt by the body. Hence, if the method from [23] is applied directly for applications involving gravity, the vertical force will be over-predicted. To overcome this situation we propose an alternative approach referred to as the PPBC method in which the resultant force and moment due to pressure is computed by integrating the pressure distribution on a surface Γ_2.

In the proposed PPBC approach, the part of the force F_{fluid} that is due to pressure is computed by directly integrating the pressure on the surface Γ_2 (see figures 1 and 2). To enable such integration, however, the pressure on Γ_2 has to be appropriately projected from the fluid nodes of the background grid where it is known. We propose a two-step approach for performing this pressure projection. First the pressure is projected to the center of the IB cells which are adjacent to the body. In a second step, the pressure at the center of a given IB cell is projected to its lateral faces belonging in Γ_2 as illustrated in the schematic shown in figure 3 and described in detail as follows.

To obtain the pressure at the IB nodes in the first step, the momentum equation (2) is projected along the direction of the wall normal as done in [25] and applied on the surface of the body. Neglecting viscous and subgrid-scale stresses, the normal momentum equation applied on the body reads as follows:

$$-\frac{dp}{dn} = \rho(\phi)\, n^i \left(\frac{Du_i}{Dt} - \frac{\delta_{i3}}{Fr^2} \right) \quad \text{on } \Gamma \tag{17}$$

where n^i denotes the unit vector normal to the body surface and u_i the velocity components of the body computed with Eq. (14). With reference to figure 3, and since the value of the pressure p_c can be readily obtained by interpolating the pressure values between neighboring fluid cells [25], we can obtain the pressure at the IB node b as follows:

$$p_b = p_c - d_{cb}\rho_a n^i \left(\frac{u_i^n - u_i^{n-1}}{\Delta t} - \frac{\delta_{i3}}{Fr^2} \right) \tag{18}$$

where d_{cb} is the distance from points c to b, and the superscripts n and $n-1$ denote the current, and previous time steps. The density value ρ_a on Γ is unknown. However, it can be set to be equal to the density ρ_b as the Neumann boundary condition is applied for the distance function ϕ normal to the wall. The above equation (18) has been obtained by combining the following two expressions, which are approximations of equation (17) applied on the surface of the body:

$$-\left(\frac{p_a - p_c}{d_{ca}} \right) = \rho_a n_a^i \left(\frac{u_i^n - u_i^{n-1}}{\Delta t} - \frac{\delta_{i3}}{Fr^2} \right) \tag{19}$$

$$-\left(\frac{p_a - p_b}{d_{ba}} \right) = \rho_a n_a^i \left(\frac{u_i^n - u_i^{n-1}}{\Delta t} - \frac{\delta_{i3}}{Fr^2} \right) \tag{20}$$

where d_{ca} is the distance from points c to a and d_{ba} the distance from points b to a. In the second step, the so-computed pressure p_b at a cell center is projected to its cell faces a' and a'' on Γ_2 in the following manner:

$$-\left(\frac{p_{a'} - p_b}{d_{ba'}} \right) = \rho_b n_{a'}^i \left(\frac{u_i^n - u_i^{n-1}}{\Delta t} - \frac{\delta_{i3}}{Fr^2} \right) \tag{21}$$

where $n_{a'}$ is taken as the unit normal to the corresponding cell face. A similar expression is used to obtain the pressure at a''. Once the pressure has been computed at the center of all the faces forming Γ_2 the forces $F^i_{f,p,2}$ and moments $M^i_{f,p,2}$ can be computed as follows:

$$F^i_{f,p,2} = \sum_{j=1}^{N_{faces}} p^j_{a'} S^j_{a'} n^j_{a',i} \tag{22}$$

$$M^i_{f,p,2} = \sum_{l=1}^{N_{faces}} \varepsilon_{ijk} r^l_j p^l_{a'} n^l_{a',k} S^l_{a'} \tag{23}$$

where N_{faces} is the total number of cell faces forming Γ_2, $S^j_{a'}$ is the area of the j-th cell face a'.

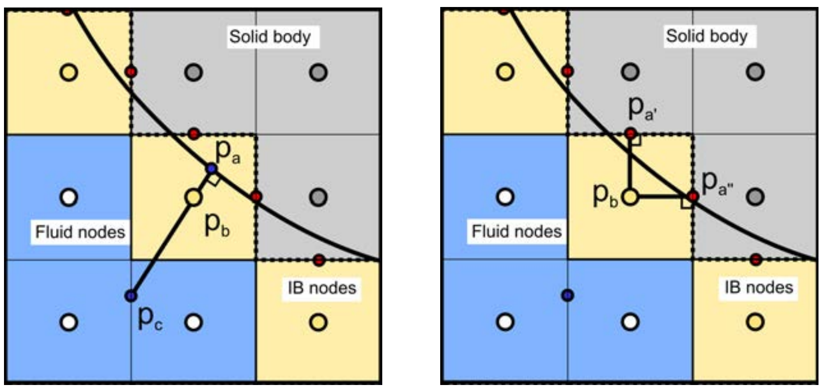


Figure 3: Schematic description of the successive pressure projections used to calculated the pressure on Γ_2 in the proposed PPBC method: step 1 (left) and step 2 (right).

The implementation of the PPBC approach is straightforward in methods such as the hybrid-Cartesian/immersed boundary (HCIB) method or the CURVIB method in which a similar algorithm is used for reconstructing the velocity boundary condition at the IB nodes. Also, since the integration of the stresses is carried out in the background mesh, the parallel implementation of the algorithm is significantly simpler than in methods based on projecting the stresses to the unstructured Lagrangian mesh of the body. The simplicity and expedience of both algorithms, i.e. the PPBC and that from [23], when compared to methods based on projecting directly on the body surface, is in expense of an additional error caused by the approximation of the surface area. In the following we discuss the relative accuracy of the two methods.

2.3.5 The FSI-CURVIB method

The method for tracking the motion of geometrically complex bodies is the sharp interface CURVIB method of Ge and Sotiropoulos [26], which has been thoroughly validated for simulating deformable bodies with large motions in various applications, including FSI problems [23, 27]. For the sake of completeness only a brief description of the method is presented herein. In the CURVIB approach, the body is represented by an unstructured triangular mesh which is embedded in the background curvilinear or Cartesian grid. An efficient searching algorithm is used to classify all nodes of the computational domain depending in their location with respect to the position of the body. The linking between the background and structure grids is done through a sharp interface approach by reconstructing the boundary conditions for the velocity field and the distance function ϕ at the IB nodes (see Ge and Sotiropoulos [26] and Kang and Sotiropoulos [15] for details). The velocity is reconstructed in the wall normal direction with either linear or quadratic interpolation in the case of low Reynolds number flows when the IB nodes are located in the viscous sub-layer, or using the wall models described by [28, 29, 30] in high Reynolds number flows when the grid resolution is not sufficient to accurately resolve the viscous sub-layer. The distance function ϕ is reconstructed by setting its gradient to be zero at the cell faces that are located between the fluid and IB nodes. This is equivalent to applying a zero Neumann boundary condition along the grid line corresponding to the aforementioned cell faces. A further description of how to reconstruct the distance function in the IB nodes is given in [15].

2.4 Results

2.4.1 FSI case 3: Free falling wedge

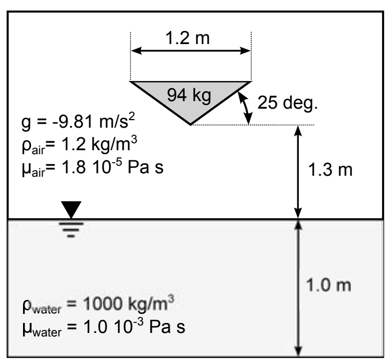
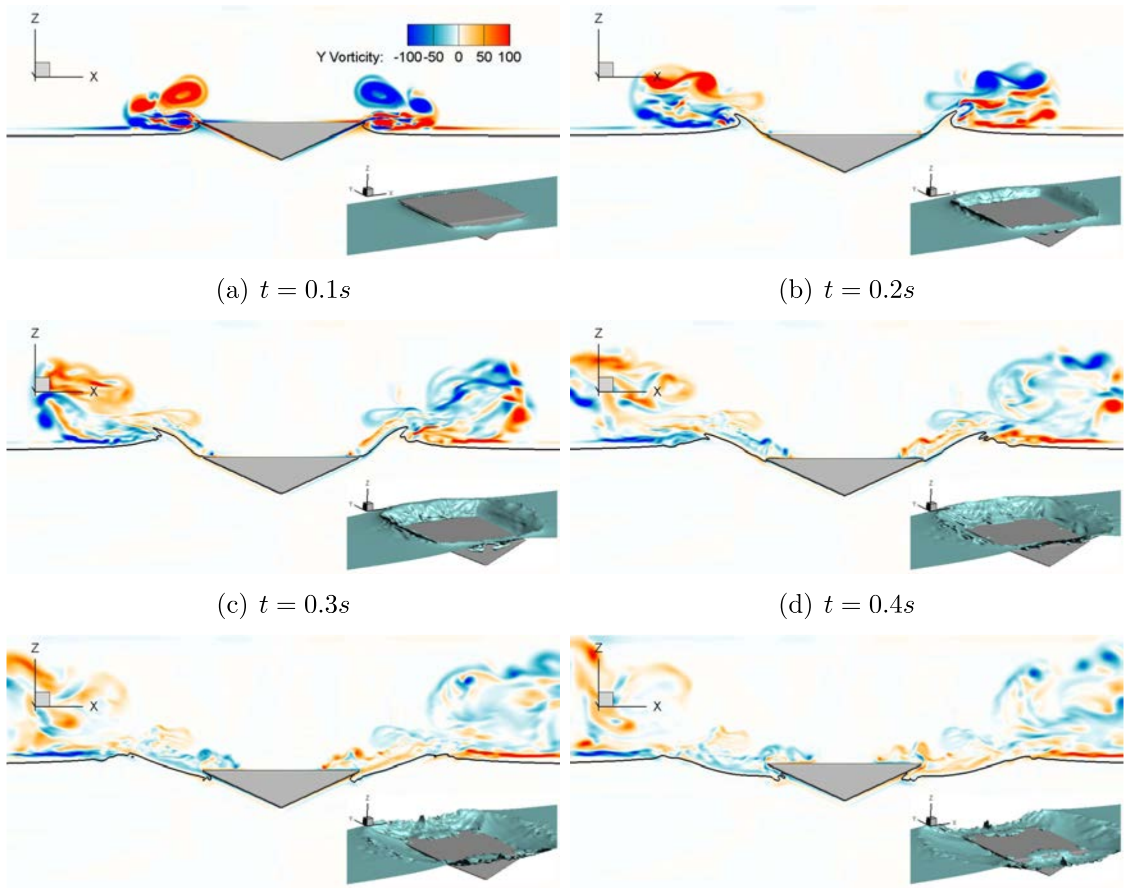


Figure 4: Schematic description of the free falling wedge configuration studied experimentally by Yettou et al. [31].

This test case is the simulation of a geometrically complex body, a 3D wedge, falling freely and impinging with its pointed keel into the free surface. The geometrical complexity of the structure along with the large pressure and velocity gradients that develop as the structure's keel impinges on the surface make this the most challenging of all the cases we have studied in this paper. The specific test case we simulate corresponds to that studied experimentally by Yettou et al. [31], who reported detailed data sets of the wedge velocity and position as function of time.



(a) $t = 0.1s$ (b) $t = 0.2s$

(c) $t = 0.3s$ (d) $t = 0.4s$

(e) $t = 0.5s$ (f) $t = 0.6s$

Figure 5: FSI simulation of a free falling wedge. Several snapshots of the calculated position of the wedge, the free surface, and corresponding out-of-plane vorticity contours are shown in these figures at the cross middle plane ($Y = 0$). A small 3D view of the wedge is superposed. The solution has been obtained on grid 1, which has near-body spacing equal to $\Delta x = \Delta z = 0.005L$, and a time step of $0.00025s$ has been used.

The simulated 3D body has a symmetric wedge-shaped section with a 25 degree dead-rise angle. Its mass is $94Kg$, which is equivalent to a structural density of $\rho = 466.6Kg/m^3$. Initially, the keel of the wedge is placed 1.3 m above the free surface and starts moving downwards under the action of gravity towards the free surface. The geometrical configuration of the simulation is illustrated in figure 4. The flow Reynolds number based on the wedge length and its maximum velocity is of the order of $Re = 5 \cdot 10^5$.

The computational domain is a $30m$ long channel, $2m$ wide with and a water depth of $1m$, and the wedge is located at the center section of the channel. The grid used has an inner/outer region structure. In the inner region, enclosing the structure, has a constant grid spacing of $0.005L$ both in the longitudinal and vertical

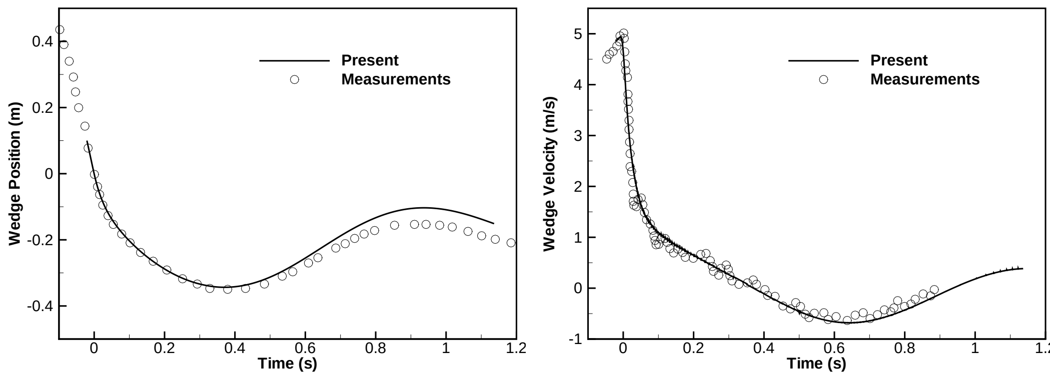


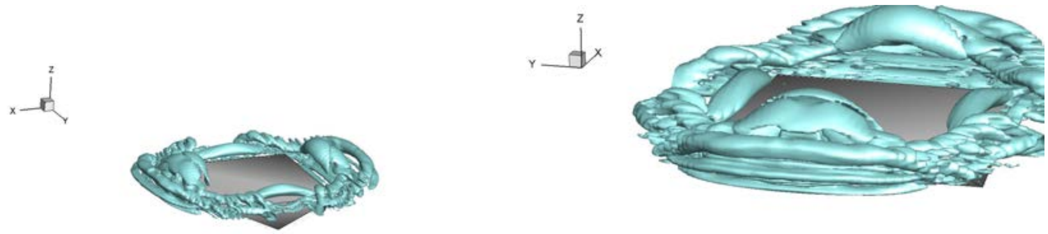
Figure 6: FSI simulation of a free falling wedge. Vertical position (left) and vertical velocity (right) of the wedge, and the experimental data of Yettou et al. [31].

directions. 60 grid nodes are used in the span-wise direction and the overall grid size is approximately 10.

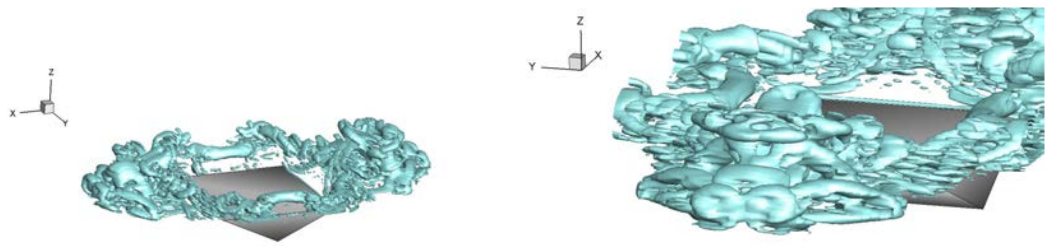
The dimensions in the horinzontal (x), spanwise (y) and vertical (z) directions is $660 \times 60 \times 250$. The inner rectangular region of constant grid spacing that encloses the wedge is of size [-0.6, 0.6] in the horizontal direction, and [-0.6, 0.4] in the vertical direction. In the outer region, the stretching ratio in the x direction has been kept reduced (1.001) and constant along the following interval ($0.6 < |x| < 2.5$) which coincides with the area of high vorticity generated by the wedge. Away from this interval the stretching ratio is increased progressively up to a limit of 1.05.

To demonstrate the complex flow patterns that develop as the wedge impinges on the free surface, we show in figure 5 a series of snapshots of the simulated free surface patterns along with contours of the Y component of the vorticity on the $Y = 0$ plane. As seen in this figure, as the wedge impinges on the surface it generates a wave on each of its sides. These waves steepen as they propagate away from the wedge and ultimately break leading to massive separation off the the wave cusp and production of large-scale vortical structures of opposite sign in the air phase. This breaking wave generated vorticity is entirely confined in the air phase and ultimately breaks up into smaller scales giving rise to a highly turbulent flow state that rises for several wedge heights above the free surface. These results are entirely consistent with the already discussed findings of Iafrati et al. [32] who showed that in breaking wave phenomena most of the energy is transferred from the wave to the air phase. [32] also showed that vortical structured generated in the air phase near the free surface during wave breaking are able to penetrate into the air phase at significant heights above the surface, a phenomenon which is also observed in our simulations. We note, however, that due to the impact of the wedge on the surface, the condition we have simulated herein is more severe than previously studied cases with enormous amounts of energy transferred from the wedge, to the breaking waves and ultimately to the air phase.

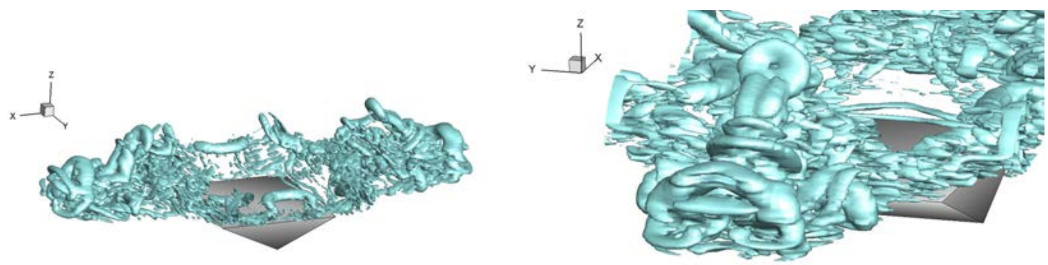
In figure 6, we compare the simulated temporal variation of the wedge position and velocity with the experimental measurements of [31]. The simulation results shown



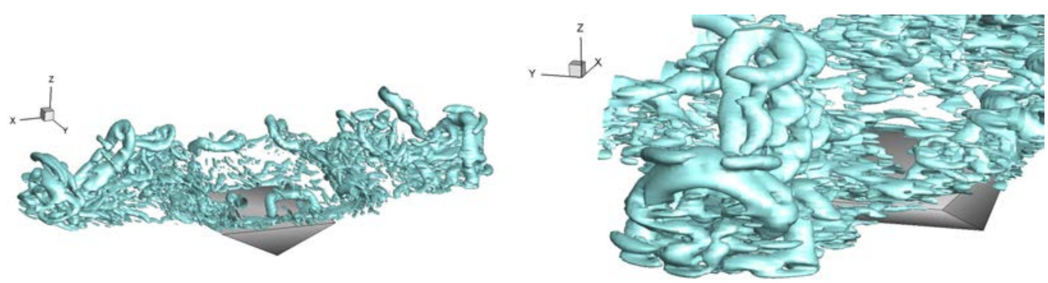
(a) $t = 0.1s$



(b) $t = 0.2s$



(c) $t = 0.3s$



(d) $t = 0.4s$

Figure 7: FSI simulation of a free falling wedge. Snapshots showing coherent structures in the flow visualized by the Q criterion (Q=-200). The right figures show a closer view of the coherent structures near the area where wave breaks.

in these figures have been obtained on grid 1. It is readily seen from figure 6 that the computed results are in very good agreement with the measurements, capturing both the frequency and amplitude of the wedge oscillation with good accuracy. These results further reinforce the accuracy of the PPBC method we developed herein for calculating the pressure force acting on floating structures.

Finally, in figure 7 we present several snapshots of the simulated 3D coherent structures visualized with iso-surface of q-criterion [33] to illustrate the richness of the ensuing air and water dynamics as the waves off the sides of the wedge steepen and break. As seen, the massive separation zone induced by the breaking waves is dominated by a series vortex loops, arch vortices, and hairpin vortices. It is also evident from sub-figures 5 (a) and (b), corresponding to times $t = 0.1s$ and $t = 0.2s$, that the 3D complexity of the simulated flow grows rapidly as the waves steepen and begin to break and the flow transitions from laminar to turbulent.

In summary, the results we presented herein show that the proposed coupled FSI level set approach is able to accurately simulate a very challenging case involving large forces on the impinging structure, wave-breaking, and overtopping. The ability of the method to also resolve the complex dynamics of the flow that emerges in the air phase during wave breaking was also illustrated.

2.4.2 Far-field/near-field coupling: simulation of a broadband wave spectrum

In this section we present a wave case aimed to validate the far-field/near-field wave coupling algorithm. We also want to demonstrate the ability of the forcing method to generate complex wave fields composed of various superposed frequencies.

In this case, the simulation is started at the HOS domain of the far-field code, by setting the initial velocity potential and free surface elevation to that of the given wave case at time zero. As soon as the far-field simulation is started a Fast Fourier Transform of the free surface elevation is applied at every time step to extract the wave frequencies and amplitudes, which are then incorporated to the near-field solver with the proposed surface forcing method. In the present cases we do not consider any airflow in order to minimize the complexity of the problem.

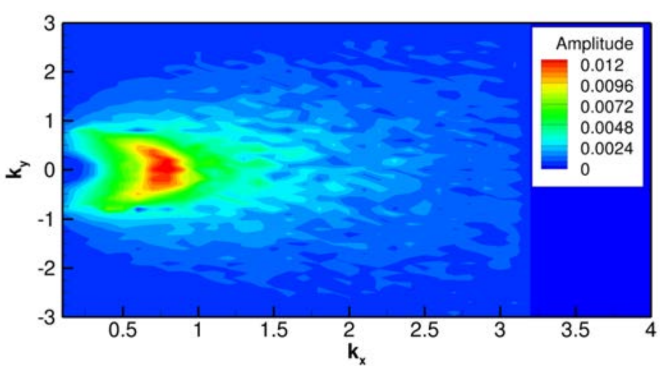
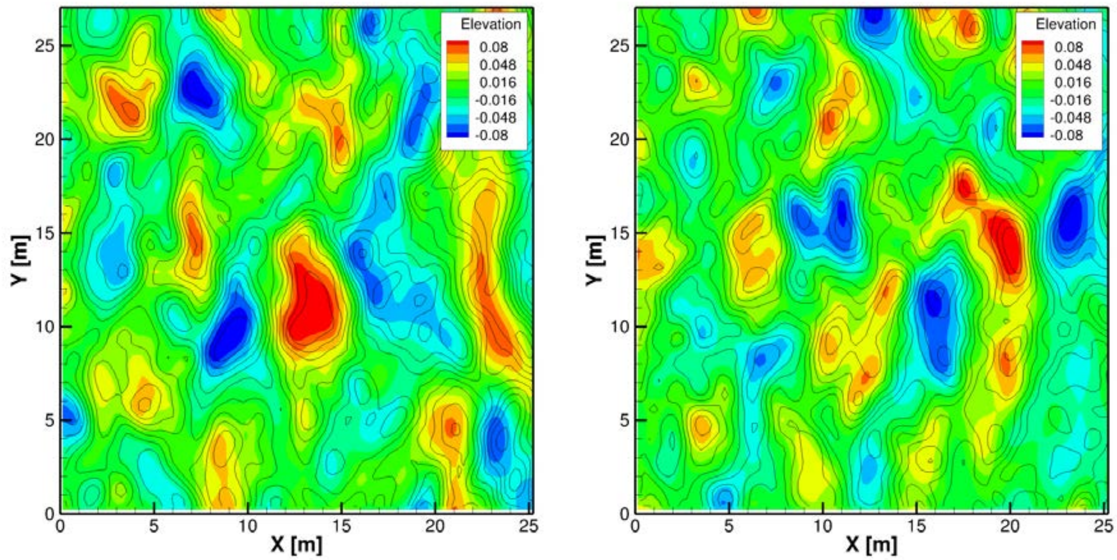


Figure 8: Far-field/Near-field coupling case 3: simulation of a broadband wave spectrum. Definition of the wave field with contours of amplitude as a function of the wavenumber computed at the far-field domain at time $t = 30s$.

The case consist of a broadband wave spectrum of peak approximately equal to $L \approx 8m$. The wavenumber distribution of the wave field that is extracted from the



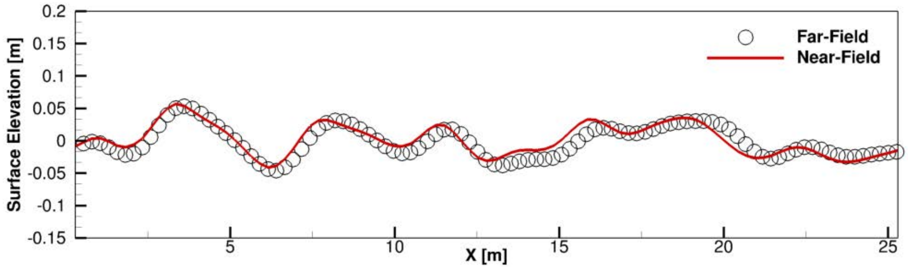
(a) Time $t = 26s$ (b) Time $t = 30s$

Figure 9: Far-field/Near-field coupling case 3: simulation of a broadband wave spectrum. Computed free surface elevation in meters at the far-field domain (contour lines) and at the near-field domain (colored contours).

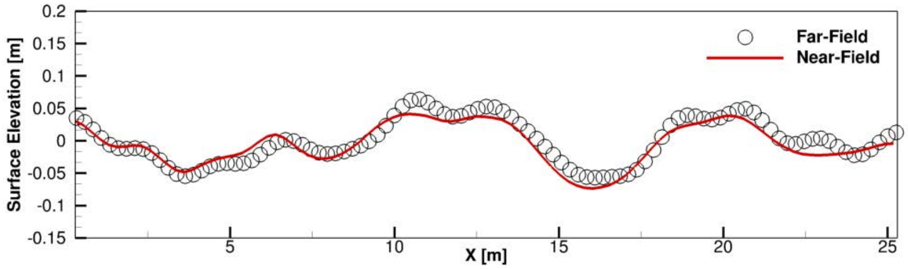
far-field domain at time $t = 30s$ is shown in figure 8.

The computational domain is a rectangular basin of length $93.5m$ ($X = [-31.16m, 62.34]$) in the stream-wise direction, 31.16m ($Y = [0m, 31.16m]$) in the span-wise direction, water depth of $10m$, and air column depth of $1m$. The grid is non-uniform with an inner/outer region structure. The inner region, including the source region, has uniform spacing and spans the following dimensions: $X = [-7.5m, 30m]$, $Y = [0m, 31.16m]$, and $Z = [-0.2, 0.2]$. The grid spacing within this area is, respectively, $0.4m$, $0.25m$, and $0.02m$. The time step is $0.0025s$ and the gravity $g = 10m/s^2$.

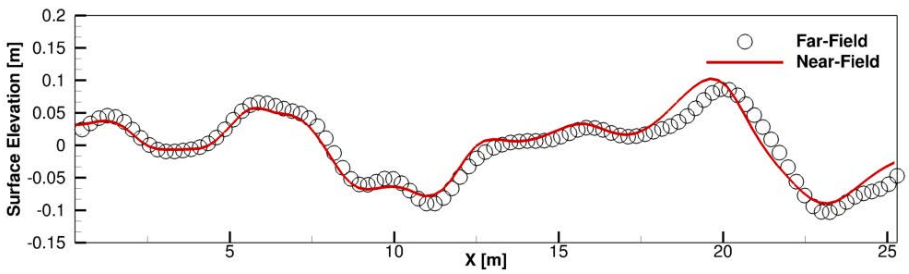
The free surface elevation results computed at the near-field domain and at the far-field domain are presented in figures 9 and 10. In particular, figure 9 shows elevation contours at different times, $t_1 = 26s$ and $t_2 = 30s$, and figure 10 several elevation profiles at different Y planes at time $t_2 = 30s$. As demonstrated in the two figures, the resulting wave field in the near-field solver, which is generated in the pressure forcing method, agree well with that simulated in the far-field domain. Obviously, there are some discrepancies which are explained due to the fact that not all wave frequency components can be resolved with the optimum number of grid nodes per wavelength. These discrepancies can be minimized with increased grid resolution.



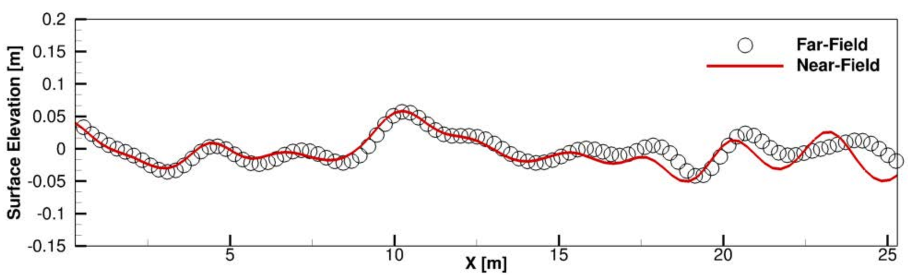
(a) $Y = 5.0m$



(b) $Y = 10.0m$



(c) $Y = 15.0m$



(d) $Y = 20.0m$

Figure 10: Far-field/Near-field coupling case 3: simulation of a broadband wave spectrum. Computed free-surface elevation profiles from both the far-field and the near-field domains at different Y planes. The results correspond to time $t = 30s$ and the time step used is the simulation is $0.0025s$.

2.4.3 Turbine-wave interactions: response amplitude operator

In this section, we present the simulation results of the floating turbine experiments presented in task 3 (experiments). First of all, we performed a series of tests of the

Table 1: Description of the three grids employed in the RAO

Grid	**Grid size**	**Near body spacing [m]**	**Source region spacing [m]**	ϵ **[m]**
G1	$214 \times 85 \times 233$	$\Delta X = \Delta Y = 0.020$,	$\Delta X = 0.08$	0.04
	(4.2M)	$\Delta Z = 0.0025$	$\Delta X = 0.08$	0.04
G2	$248 \times 85 \times 233$	$\Delta X = \Delta Y = 0.020$,	$\Delta X = 0.16$	0.04
	(4.9M)	$\Delta Z = 0.0025$	$\Delta X = 0.16$	0.04
G3	$508 \times 169 \times 217$	$\Delta X = \Delta Y = 0.010$,	$\Delta X = 0.15$	0.04
	(18.6M)	$\Delta Z = 0.0050$	$\Delta X = 0.15$	0.04

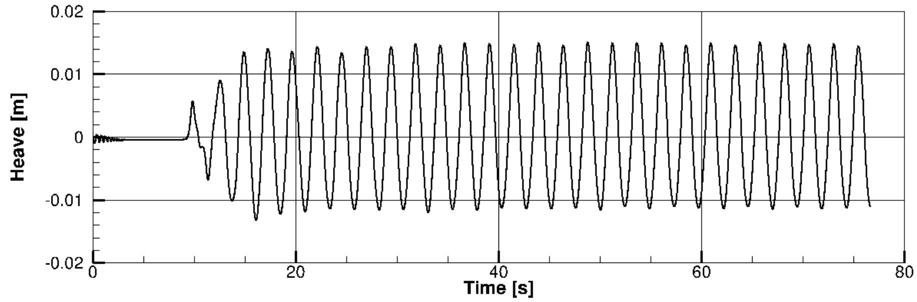
Table 2: Description of parameters used in each of the test cases to compute the RAO.

T[s]	**A[m]**	**L[m]**	**Grid used**
0.91	0.020	1.28	G1
0.97	0.012	1.47	G1
0.99	0.015	1.54	G1
1.02	0.013	1.61	G1
1.03	0.013	1.66	G1
1.50	0.009	3.52	G2
1.80	0.010	4.79	G2
2.30	0.014	6.97	G3
2.40	0.0725	7.40	G3
2.425	0.00125	7.50	G3
2.425	0.0017	7.50	G3
2.45	0.0085	7.61	G3
2.45	0.0121	7.61	G3
2.50	0.0086	7.82	G3
2.60	0.013	8.23	G3
2.70	0.0135	8.65	G3

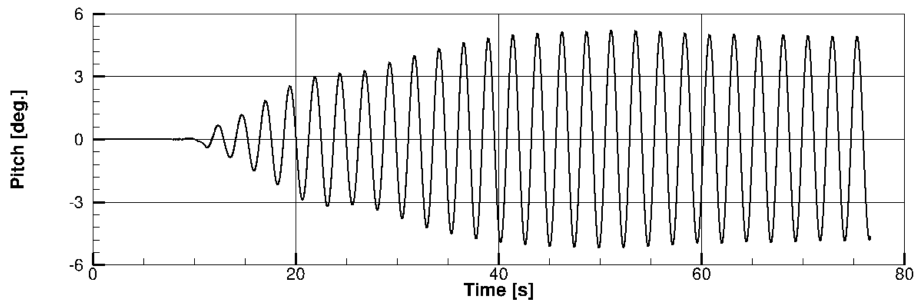
floating turbine without waves to characterize the dynamic parameters of the system in order to determine the natural frequencies and the damping ratios in the heave and in the pitch motions (not shown here). This was achieved by simulating free decay tests of the platform, first in a single DoF (in heave and in pitch), and then in the two DoF (combining heave and pitch). Then, we simulate wave-turbine interactions by performing a RAO with specific effort on the frequencies close to the fundamental oscillatory modes.

We studied the platform response when subject to a wave fields of different frequencies by performing a so called response-amplitude operator (RAO). A RAO is a transfer function used to predict the structural response that a given structure will exhibit when subject to different wave conditions.

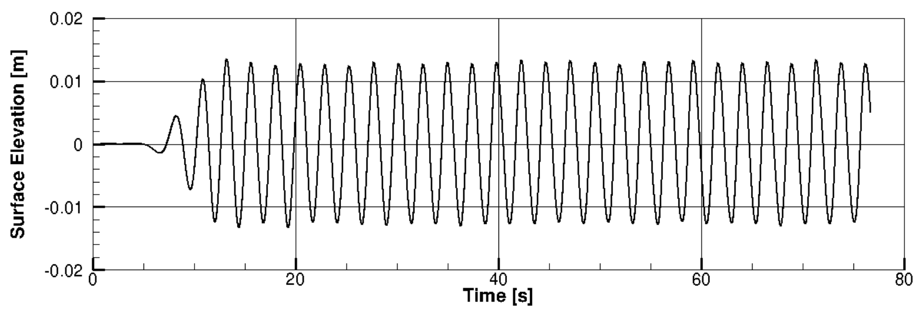
To carry out the RAO using the present computational method we first define



(a) Turbine response in heave



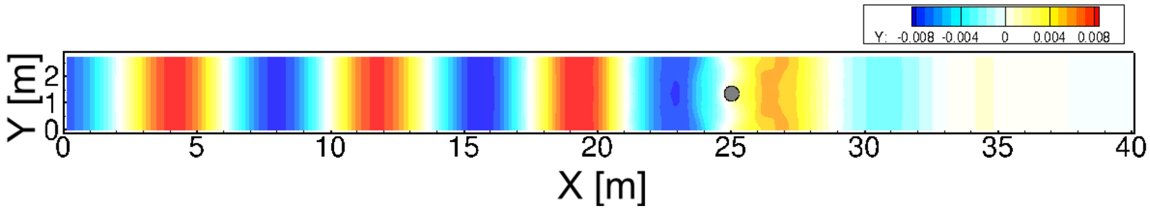
(b) Turbine response in pitch



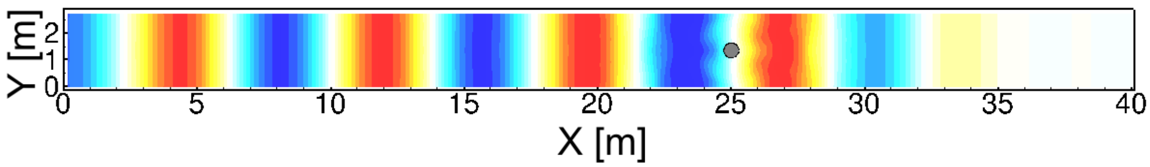
(c) Surface elevation at $Z = 18m$

Figure 11: Wave/body interactions for the case of incident waves of period $T = 2.425s$. Computed structural response of the strucure in heave (a), in pitch (b), and surface elevation at a point located at $Z = 15m$ (c). The results have been computed on grid G2 using a time step of $0.0025s$.

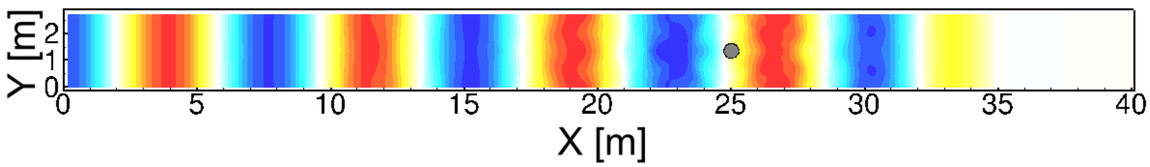
a series of test cases characterized by the period T and the amplitude A of the incident wave heading to the floating structure. For each case, the simulation is started with a calm free surface and the structure at rest. The wave forcing method, applied at the source region centered on the origin ($X = 0$), progressively begins to generate the wave field, which achieves the desired amplitude approximately after three oscillations. Then, for computing the RAO amplification factors for each of the wave cases, we take the maximum amplitude of the oscillations of the structure, measured from peak to peak, for both, heave and in pitch, after the initial transient effects have been dissipated.



(a) Time $t = 13.0s$



(b) Time $t = 15.5s$

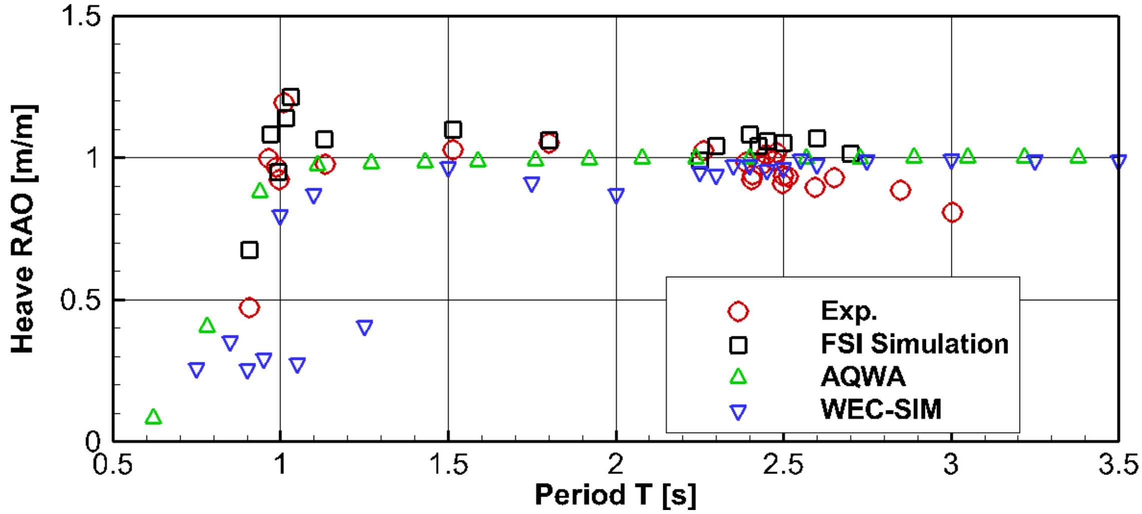


(c) Time $t = 32.5s$

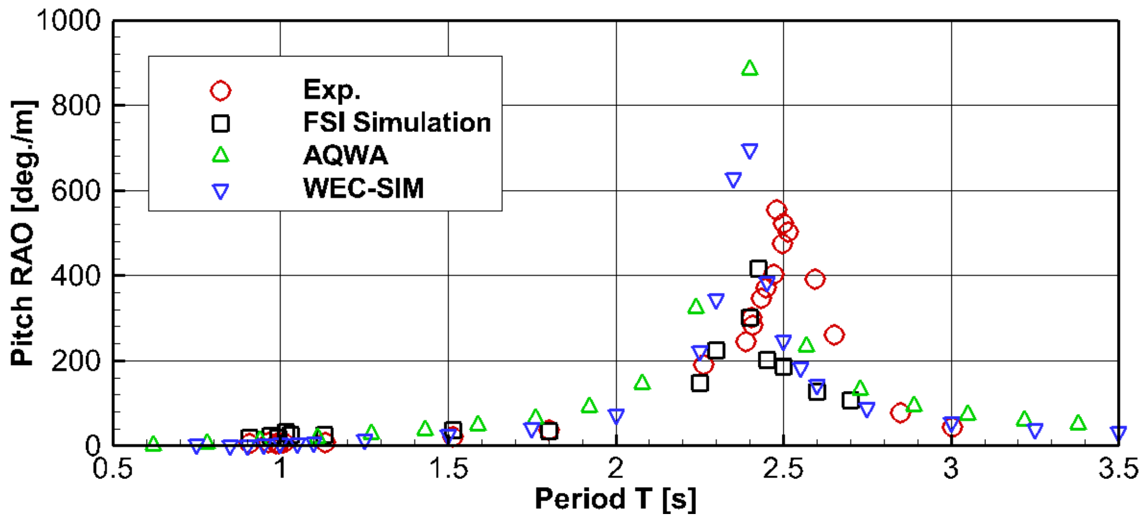
Figure 12: Wave/body interactions for the case of incident waves of period $T = 2.425s$. Surface elevation contours at different instances in time. The results have been computed on grid G2 using a time step of $0.0025s$.

The wave period of the several test cases considered for the RAO ranges from $0.9s$ to $2.7s$ to be able to capture the RAO near the heave and pitch natural periods, which with the decay tests were shown to be $1s$ for heave and $2.45s$ for pitch. The list of computed test cases is given in Table 2.

Similar to the heave decay tests, the computational domain is a fraction of the Saint Anthony Falls Laboratory (SAFL) wave basin. While the basin width of $2.743m$ and water depth of $1.37m$ are taken from the real dimensions, the length does not span the full dimension, which would be very costly. The sponge layer, applied at the beginning and end of the basin, ensures that the waves are not reflected. The basin length is defined based on the requirements of the forcing method and the sponge layer. The source region dimension ϵ_x needs to be equivalent to $L/2$, and the length of the sponge layer x_s between L and $2L$. Also, the test section where the structure is positioned needs to be located at least $2L$ from the source region to allow proper development of the wave field. Based on these restrictions, we defined three lengths of the computational domain to accommodate the different RAO cases, which have a wide range of wavelengths (from $1.28m$ to $9.26m$). Using the dispersion relation, the wavelength L corresponding to the $0.9s$ period wave is $1.28m$, and the wavelength for the $2.7s$ period wave is $9.26m$. The length of the computational domain for the low L cases is $18.6m$ ($X = [-6.0, 12.6]$), for the intermediate L cases is $36.6m$



(a) Heave RAO



(b) Pitch RAO

Figure 13: Response amplitude operator of the floating structure. Normalized structural response of the strucure in heave (a) and in pitch (b) when subject to incidint monochromatic waves of varying wave period. The simulation results from the present FSI model have been computed on grid G2 using a time step of $0.0025s$.

($X = [-12.0, 24.6]$), and for the large L cases is $75.0m$ ($X = [-30.0, 45.0]$). We define a different non-uniform grid for each of the three computational domains: G1 of size $214 \times 85 \times 233$ (4.2M) for the short domain, G2 of size $248 \times 85 \times 233$ (4.9M) for the intermediate domain, and G3 of size $478 \times 169 \times 217$ (18.6M) for the long domain. All three grids follow a three-region structure with two inner regions of constant grid spacing and an outer region within witch the spacing progressively increases towards the boundaries. One of the inner regions encloses the wave generation source region and the other region encloses the floating structure. In the source region ΔX is 0.08

for grid G1, 0.16 for grid G2, and 0.15 for grid G3. In the near-body region the grid spacing is $\Delta X = \Delta Y = 0.020$ and $\Delta Z = 0.0025$ for G1, $\Delta X = \Delta Y = 0.020$ and $\Delta Z = 0.0025$ for G2, and $\Delta X = \Delta Y = 0.010$ and $\Delta Z = 0.0050$ for G3.

The dimensions, near body spacing, and interface thickness for the three grids is summarized in Table 1. Also, in Table 2, we indicate the grid employed for each of the RAO cases.

As we stated in the introduction, in addition to validate the present FSI model for wave-body interactions, we also seek to compare its accuracy to other models, based on lower order assumptions, which typically used by the offshore industry. We thus performed a simulation of the RAO of the floating turbine model using the following two numerical codes: (1) the commercial hydrodynamic software ANSYS-AQWA [34]; and (2) the open source Wave Energy Converter Simulator (WEC-Sim) [35, 36].

ANSYS-AQWA is a potential flow solver based on the boundary element method. As such, it neglects the viscosity of the fluid and assumes linear wave theory and small amplitude motions. Although ANSYS-AQWA has a time domain solver for the structural response, we used the linear frequency domain solver. Frequency domain hydrodynamic codes are an efficient tool to compute the hydrodynamic coefficients required by other time domain solvers such as WEC-Sim.

In ANSYS-AQWA, the structural response is computed linearly by solving the following rigid body EoM in the frequency domain (i=1,2, ... , 6):

$$Y_i(\omega)\left[-\omega^2(m_{ii} + A_{ii} + j\omega(B_{ii} + \Lambda_i) + k_{ii})\right] = F_{e,i}(\omega), \tag{24}$$

where ω is the wave angular frequency, m_{ii} the diagonal terms of the mass matrix, A_{ii} is the added mass matrix, B_{ii} the radiation damping coefficients, Λ_i the visocous/additional damping coefficients, and $F_{e,i}(\omega)$ is the wave excitation force.

In the present case, Λ_i is taken from the experimental decay tests of [37]; for heave is $43.766 N \cdot s/m$ and for pitch $0.366 kg \cdot m^2/s^2$. Note that these values differ from the additional damping incorporated into the CFD-FSI method. While the CFD-FSI methods can, in principle, account for the eddy making damping and wave making damping, the potential based flow solvers completely neglects all contributions of the damping, which has to be fully incorporated artificially.

The fact that the center of mass of the floating turbine is below the pitch center of rotations is not straightforward to model in ANSYS-AQWA. We thus assume that the center of rotations of the floating system is at the center of gravity of the structure. Such assumption may have some effect in the ANSYS-AQWA pitch solution, but however, it does not alter the computation of the hydrodynamic coefficients (F_e, A_{ii}, and B_{ii}) to be used by WEC-Sim.

WEC-Sim solves the following time domain EoM (i=1,2,...,6):

$$m_{ij}\frac{\partial^2 Y^i}{\partial t^2} = F^i_{ext} + F^i_{rad} + F^i_v + F^i_B + F^i_m, \tag{25}$$

where m_{ij} are the components of the mass matrix, F^i_{ext} are the wave excitation force components, F^i_{rad} the force components due to radiated waves, F^i_v the viscous damping force components, F_B^i the net buoyancy force components, and F_m^i the force components due to mooring connections. The F_{ext}^i and F_{rad}^i forces are computed using data from the previous frequency domain solution computed using ANSYS-AQWA. The details of the method are given in [36, 38].

In WEC-Sim, we considered the actual location of the center of gravity (CG) and the pitch axis. Also, since WEC-Sim accounts for some non-linearities of the problem, we expect it to provide a more accurate solution than that form ANSYS-AQWA.

In Figure 13 we present the heave and pitch RAO results, including the experimental data, the FSI computation, and the ANSYS-AQWA and WEC-Sim results. As it is shown in the heave plot in Figure 13(a), the FSI model predicts the resonance peak when the wave period is $T = 1.03s$ with a corresponding RAO of 1.216. This values are nearly identical to the experimental results where the maximum amplification factor of 1.19 occurs for an incident wave of period $1.01s$. In contrast, the lower order methods, ANSYS and WEC-Sim, are unable to accurately predict the heave RAO near the natural frequency. Particularly, WEC-Sim has some values near the heave peak with very low amplification factor far below the experimental data.

Before analyzing the computed RAO results in pitch, we first need to look into the experimental data. As shown in Figure 13(b), the exact experimental RAO and period of the incident wave of the resonance peak is not clearly defined. The following four experimental realizations with period close to the peak where tested: $T = 2.480s$, $T = 2.498s$, $T = 2.500s$, and $T = 2.515s$, with the respective amplification factor being $553deg/m$, $475deg/m$, $521deg/m$, and $501deg/m$. The amplitude of the incident wave for these four cases is nearly identical ($0.00725m$, $0.00745m$, $0.00725m$, and $0.00725m$, respectively). The fact that the amplification factor oscillates when slightly increasing the wave period, means that the peak period and amplitude is subject to a certain degree of randomness. Thus we consider the experimental amplification factor for pitch to be within the range $[475, 553]deg/m$. The average value of these four realizations is $512.5deg/m$. Similarly, the period of the incident wave is within the range $[2.480, 2.515]deg/m$ and the average value is $2.498s$.

In the same Figure 13(b), the RAO results in pitch computed with the FSI model predict the resonance peak to occur when the period of the incident wave is $T = 2.425s$. This value, compared to the averaged experimental pitch natural period of $2.498s$, is 2.9% lower. This is consistent with the observations of the decay tests that our model was slightly under-predicting the pitch decay period by 2% as a result of considering a simplified geometry of the floating system. Looking at the RAO of the peak, the FSI result is shown to be at $417deg/m$, which is 18.6% below the experimental value taken from averaging the four near-peak cases.

With regards to the lower order models results, both ANSYS-AQWA and WEC-Sim, capture the resonance peak for a period of the incident wave of $T = 2.4s$, which is 3.9% below the experimental measurements. Comparing the amplification factor at the pitch resonance peak with that from the experiments, ANSYS-AQWA, with a value of $884deg/m$, over-predicts it by 63.5% and WEC-Sim, with a value of $696deg/m$, over-predicts it by 35.6%. As it was expected from the fact that WEC-

Sim consider non-linearities of the problem and that in the ANSYS-AQWA model the center of rotation was not taken at the actual position but at the CG position, the WEC-Sim results in pitch are significantly better than those from ANSYS-AQWA.

One of the features that we could capture with the present FSI model that is not accounted for by linear models is the effect of the amplitude of the incident wave given a fixed wave period. This phenomenon was already documented in previous works such us in Jung et al. [39]. They carried out, experimentally, the RAO in roll of a rectangular barge restricted to move in a single DoF. They observed that for two cases of incident wave with equal period, the RAO is larger for the case of lower wave amplitude. Also, this phenomenon was observed to be maximum at the peak frequency and to minimize far from the peak. In our pitch RAO results computed with the FSI model we observed the exact same phenomenon. We computed the RAO at the pitch peak period ($T = 2.425s$) with two cases of different wave amplitude, $0.0125m$ and $0.0170m$. The RAO for the case of lower wave amplitude is $417deg/m$ and for the case of larger wave amplitude $297deg/m$. We performed the same analysis to a case in which the incident wave period is far from the peak ($T = 2.45s$) to show how this effect, far from the peak, diminishes. For that wave period, a case with wave amplitude of $0.0085m$ resulted in a amplification factor of $202deg/m$ and a case with wave amplitude of $0.0121m$ to a slightly smaller amplification factor of $194deg/m$. Note that in Figure 13(b) only the wave case that results in a maximum RAO are displayed.

Finally, in Figure 11 we present the structural response in, heave (a) and in pitch (b), as well as the time evolution of the surface elevation at a point located at $X = 15m$ (c). These results correspond to the case in which the incident wave has a period of $2.425s$ and an amplitude of $0.00125m$. A common behavior observed in this figure and in most of the remaining wave cases is that while the pitch response grows progressively, the heave response grows sharply in just 3 to 4 oscillations. This sharp start of the heave motion results in a fictitious maximum amplitude usually at the fourth oscillation. Since this is an artifact effect that could have been removed by using a ramping function, we decided not to consider the first four oscillations in the calculation of the maximum amplitude of the oscillation. For the same case, we show in Figure 12 the surface elevation contours at different instances in time. The figure shows how the wave field evolves in time, going from a clean state, free of disturbances, to a state in which diffracted, radiated, and reflected waves have spread.

In summary, the results we presented herein demonstrate that the FSI model of [40] can reasonably capture wave-body interactions with overall better agreements than that predicted by the two lower order models considered, which were based on the potential flow theory.

2.4.4 FSI simulation of an offshore floating wind turbine simulation

The aim of this test case is to demonstrate the full capabilities of the proposed far-field/near-field computational framework by simulating an offshore floating wind turbine under realistic wind and wave conditions representative from a site-specific

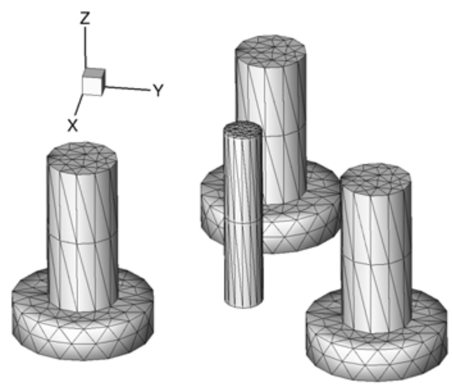


Figure 14: Mesh of the floating platform used in the IB method in the offshore floating wind turbine case.

environment such as the Pacific Northwest (PNW).

To determine the wind velocity and the wave field that is representative from the PNW we use the measurements taken from the Station 46041 of the National Data Buoy Center. Based on the data given in [41] we adopt for the simulation a wind speed of $5m/s$ taken at $5m$ height above the free surface, and a wave field with broadband spectrum of dominant peak period $T_{peak} = 12.75s$ which using the dispersion relation is equivalent to a peak wavelength of $L_{peak} = 251m$.

We consider a large floating wind turbine system, consisting of 13.2MW wind turbine installed on a tri-column triangular platform. The turbine rotor, which is a design by Sandia National Laboratories, has a $200m$ long diameter and a SNL100-00 blade (see [42] for details), and the floating platform, designed by Principle Power, is based on the OC4 semi-submersible design detailed in [43]. The hub height is $133.5m$ with respect to the mean-sea level and the turbine is simulated with a constant tip speed ratio of 8, which given an incoming velocity of about $9m/s$, is estimated to be close to the optimal value for performance. The geometrical parts of the platform considered in the simulation and its dimensions are presented in figure 14 which also shows the structural mesh used for discretizing the structure in the CURVIB method. The structural elements interconnecting the four columns have been neglected due to its small size in comparison to the large dimensions of the columns. The floating platform is secured in place using the same mooring system as in [43] but Froude-scaled by a factor of 1.4. It basically consists of a three catenary lines distributed symmetrically with respect to the platform vertical axis.

The near-field computational domain is $2675m$ long in the stream-wise direction ($X = [-550m, 2125m]$) and $1750m$ wide in the span-wise direction ($Y = [-875m, 875m]$), the water depth is $280m$ and the air column above the free surface is $1000m$. The source region has a length ϵ_x of $224m$ and is centered on $X = 0$. The floating turbine is positioned downstream of the sponge layer at $X = 900m$ and centered on $Y = 0m$. We use a non-uniform mesh of size $163 \times 207 \times 259$ that is schematically described in figure 15. In the stream-wise direction, the spacing Δx is constant at the following

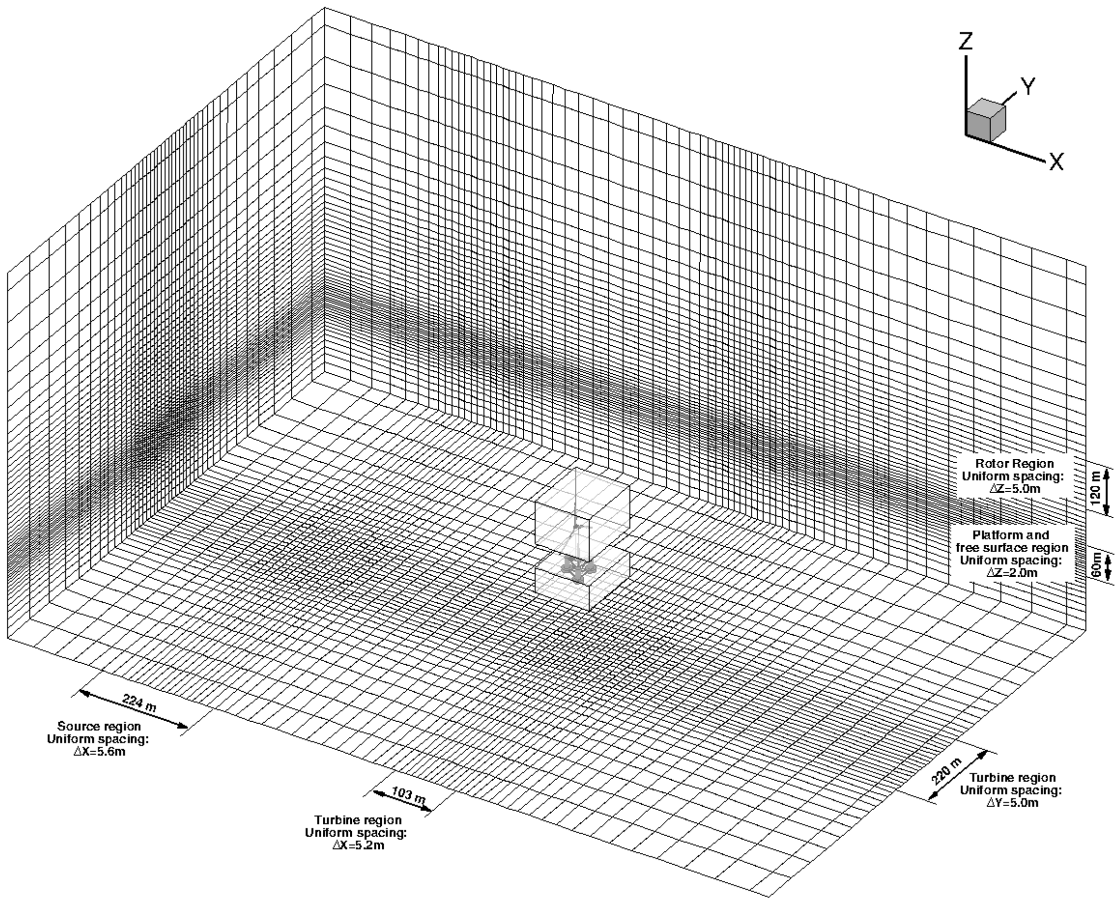


Figure 15: Schematic description of the fluid mesh used in the far-field domain of the offshore floating wind turbine case. The rectangular boxes indicate the two regions of constant grid spacing where the floating turbine is located. In this figure, for every grid line shown four are skipped.

two regions: (1) at the source region ($X = [-112m, 112m]$) in which the spacing is $\Delta x = 5.6m$; and (2) at the region containing the floating structure and defined by $X = [848.2m, 951.8m]$ in which the spacing is $\Delta x = 5.18m$. From the end of the first region ($X = 112m$) to the beginning of the second region ($X = 848.2m$), Δx varies smoothly across the two values, and outside of these two regions the spacing increases progressively towards the inlet and outlet boundaries. In the vertical direction the spacing Δz follows the same spacing pattern also with two regions of constant spacing: (1) the region from $Z = -40m$ to $Z = 20m$ which has spacing Δz equal to $2m$ and comprises the floating platform and the free surface; and (2) the region from $Z = 130m$ to $Z = 250m$ which has spacing Δz equal to $5m$ and comprises most of the rotor. Finally, in the span-wise direction the spacing Δy is constant and equal to $5m$ at a single region spanning from $Y = -110m$ to $Z = 110m$ and enclosing the

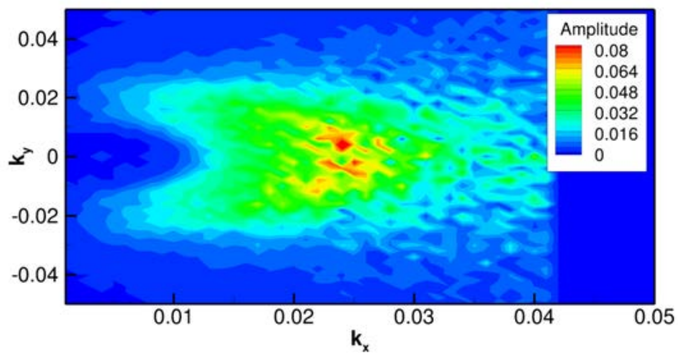
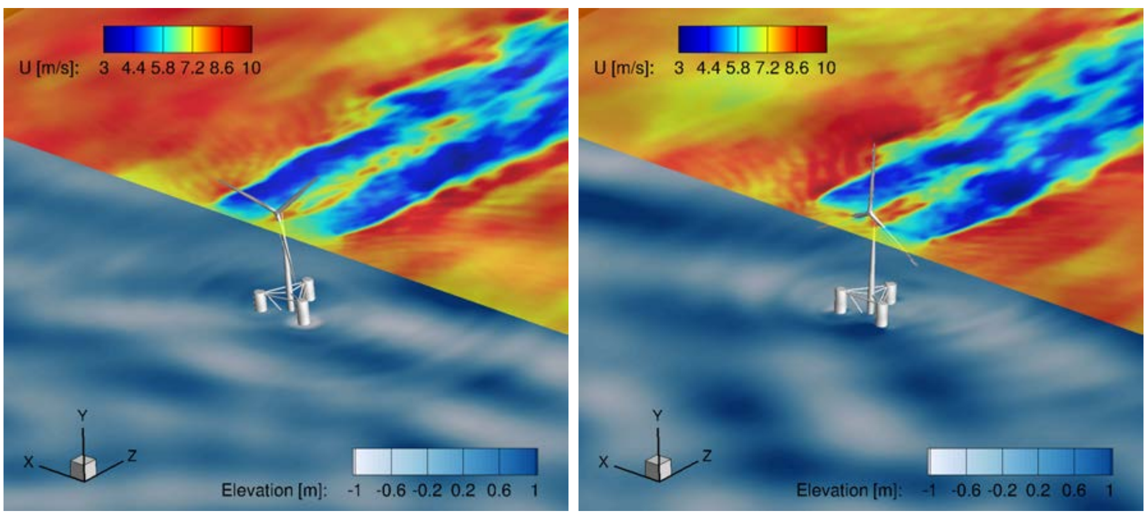


Figure 16: Definition of the broadband wave spectrum represented by wave amplitude contours as function of the directional wavenumbers. Computed at the far-field domain at a time for which the flow is fully developed ($t = 163500s$).



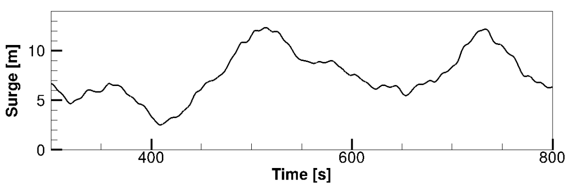
(a) Time $t = 273s$

(b) Time $t = 685s$

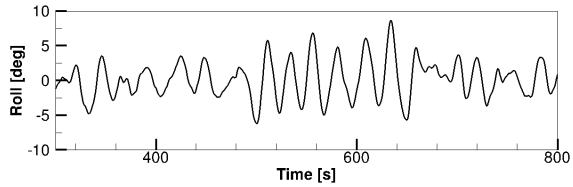
Figure 17: Offshore floating wind turbine case. 3D view of the floating wind turbine with the free surface colored with elevation contours and a horitzonal plane at hub heigth of stream-wise velocity.

floating turbine. The stretching ration used in all direction is always limited to 1.05 and its variation follows a hyperbolic function. The thickness of the interface is set to $\epsilon = 4m$, the gravity to $g = 9.81m/s^2$, and the density and dynamic viscosity for the water to $1000kg/m^3$ and $1.0 \times 10^{-3}Pas$, respectively, and for the air $1.2kg/m^3$ and $1.8 \times 10^{-5}Pas$. Slip-wall boundary conditions is adopted at the 6 boundaries, and the sponge layer method with thickness $200m$ is applied at the four lateral walls.

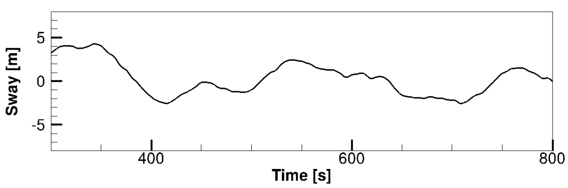
The turbine rotor in the present simulation is treated with the actuator line model implementation of [1]. The thrust force computed with the actuator model is incorporated in the EoM as an external force and moment. In addition to the thrust force, we also add as external moment the gyroscopic effect induced by the rotation of the



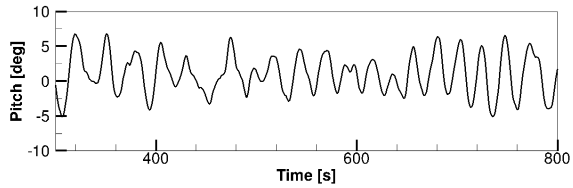
(a) Response in Surge



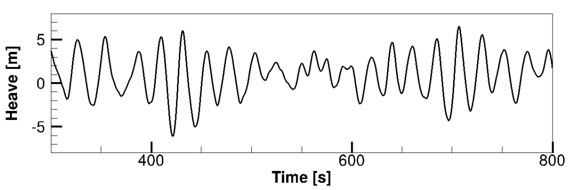
(b) Response in Roll



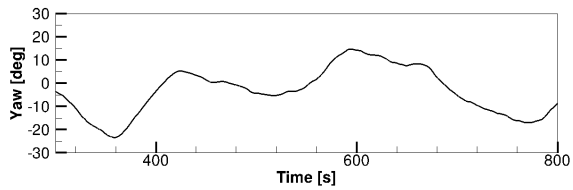
(c) Response in Sway



(d) Response in Pitch



(e) Response in Heave



(f) Response in Yaw

Figure 18: Offshore floating wind turbine case. Structural response of the floating turbine system in the six DoF.

turbine rotor using the approach of [44]. For the mooring system we use the linearized mooring model described by [43].

The wind and wave conditions are developed with a precursor simulation using the far-field model. For the air phase of the far-field model, the domain size is $6280m$, $3140m$, and $1000m$, in the stream-wise, span-wise, and vertical direction, respectively. A non-uniform mesh of size $128 \times 128 \times 128$ is used. While in the stream-wise and span-wise directions the spacing is constant, in the vertical direction the grid cells are clustered near the free surface. The mesh of the HOS for simulating the wave field is uniform and of size 769×769. The free surface initial condition is a broadband wave spectrum of type Joint North Sea Wave Observation Project (JONSWAP) and wave period peak $T_{peak} = 12.75s$. The wind field, which is driven by a constant pressure gradient such that the velocity at height $5m$ is $6m/s$, has been solved in a coupled manner with the wave field. The far-field simulation has been advanced about 300000 time steps, with a time step size of $0.545s$, before start feeding to the near-field domain. That was to ensure that fully developed wind and wave conditions were achieved. The wave spectrum for the developed wave field is shown in figure 16.

In figure 17 we present near-field results of the floating wind turbine including the free surface and the stream-wise velocity at a horizontal plane at hub-height. As seen in the figure, the wave field clearly shows the formation of radiated waves induced by the motion of the platform. The structural response of the floating structure in the six DoF is given in figure 18. Note that surge, sway, and heave correspond to the transnational DoF in the stream-wise, span-wise, and vertical direction, respectively, and roll, pitch, and yaw, rotations with respect to the X, Y, and Z axis, respectively. Looking at the pitch response in sub-figure 18(b), the turbine is slightly inclined towards positive angles as a result of the wind effect.

2.5 Summary and Conclusions

We developed, validated, and demonstrated the predictive capabilities of a novel computational framework that can simulate real life complex floating structures and its interaction with realistic ocean waves and wind fields. To efficiently deal with the computational challenge of large disparity of scales associated to such type of problems we adopted a far-field/near-field domain decomposition approach, i.e., a large-scale domain with periodic boundary conditions, known as far-field domain, in which the offshore ocean conditions are efficiently developed, and a reduced scale domain with high grid resolution, known as near-field domain, where the floating structure is located. In the far-field domain we applied the two-fluid method of Yang and Shen [45, 46], allowing to obtain fully developed wind and wave condition in an efficient manner as the waves are resolved with a potential flow based high-order spectral (HOS) method. In the near-field domain, a novel FSI model for simulating arbitrarily complex floating rigid bodies interacting with non-linear free surface flows was developed and validated.

The proposed near-field model, published in [40], integrates the FSI-CURVIB method of Borazjani et al. [23] with a level set approach along with a new method for calculating forces due to pressure on submerged structures in two-phase flows. The so-called pressure projection boundary condition (PPBC) method was shown to mitigate the difficulties encountered when calculating the force on the structure using the standard method developed by Borazjani et al. [23], which employs integration of the pressure on the surface of the volume consisting of all immersed boundary grid cells in the CURVIB method around the structure. While this standard approach works well in single-phase flows, in two-phase flow problems with submerged bodies it does not account for the force imparted on the structure due to the large density difference between the air and water enclosed between the immersed boundary surface and the body and gives rise to a first-order accurate calculation of the pressure force. The proposed PPBC approach employs the normal momentum equation to the body to obtain a more accurate representation of the pressure field on the body via a series of successive projection steps. Numerical tests clearly showed that the proposed method not only reduces the error in the calculation of the force by nearly one order of magnitude relative to the standard approach but also yields near second-order accurate convergence rate for the force and results that are in significantly better agreement with experimental measurements. While the PPBC method was developed and demonstrated herein in the context of the CURVIB approach it is general and should be readily applicable to other sharp-interface immersed boundary methodologies.

To demonstrate the accuracy of the coupled FSI, level set implementation for free surface-body interactions we simulated a series of test cases, including forced motion problems and coupled FSI problems. We showed that for all simulated cases the proposed method is able to replicate with good accuracy the structural response of several laboratory experiments including a free decay test of a circular cylinder, a roll decay test of a rectangular structure, and a free falling wedge impacting the free surface.

In order to prevent the formation of instabilities in problems involving complex air-water interface phenomena, the use of a large interface strip thickness ϵ as well as reduced reinitialization time steps sizes $\Delta\tau$ was found necessary. To reduce the computational cost for implementing these remedies, we systematically investigated the influence of using an overall reinitialization time lower than the time required for full reinitialization of the interface strip. We showed that for the falling cylinder case using a time equivalent to reinitializing 10% of the interface strip was able to predict in sufficiently high accuracy the formation of the breaking wave formed at the side of the cylinder.

The most challenging case we simulated was that of a wedge impinging on the free surface. Large pressure gradients and forces develop on the structure as it impinges on the surface and decelerates rapidly, which made the solution of the FSI problem especially challenging. Even for this case, however, our method was able to obtain converged solutions but required the use of strong coupling FSI in conjunction with the Aitken method. The latter technique was critical for efficient FSI iterations, since it reduced the number of strong-coiling sub-iterations required for convergence by fifty percent or more.

Our simulations elucidated the rich 3D dynamics resulting in the air phase as the waves induced by the impinging on the free surface wedge break, and showed that most of the energy from the breaking waves is ultimately transferred to the air phase. Massive separation off the cusp of the breaking waves gave rise to complex coherent structures dominated by loops, arch, and hairpin vortices forming an intertwined web of vortical structures that ultimately lead to the flow transitioning to turbulence. The resulting turbulent flow in the air phase was found to grow upward and persist at a significant elevation above the free surface. All these findings are in accordance with the recent findings of Iafrati et al. [32] who also showed that flow separation off breaking waves is the key mechanism for producing turbulence in the air phase. Overall the computed results demonstrated the potential of our method as a powerful tool for simulating the coupled interaction of complex floating structures with a free surface.

Our simulations for the wedge case exposed a limitation of the method when structure over-topping occurs. Our method was able to capture this complex phenomenon, which leads to pockets of water entrapped on the flat surface of the wedge in the air phase. As these pockets, however, spread laterally and grow thinner ultimately reach the scale of resolution of the level set method (the thickness of the interface that can be resolved on a given grid) and spuriously disappear. This is an inherent limitation of the level set approach and can only be resolved by local grid refinement, which can be made practical via an adaptive mesh refinement approach.

To couple the two decomposed domains, the far-field and the near field domains, we adopted a one-way coupling approach, feeding the wind and wave fields that have been fully developed at the far-field domain, to the near-field domain. We opted for a one-way loosely coupled approach, in contrast to a two-way strongly coupled approach, considering the fact that the far-field domain is applied to generate large-scale offshore flows in which the presence of a single or several marine structures should only have local effects and not alter the ocean environmental conditions.

To incorporate the far-field waves into the near-field domain, we employed an internal wave generation method consisting in applying a pressure force on the free surface in form of source term in the momentum equation. This approach, known as pressure forcing method, was initially proposed by Guo and Shen [47] to generate, suppress, and maintain water waves in a computational approach in which the free surface is treated with a sharp interface method. In this work, we extended the pressure forcing method by adapting it to the diffused interface level set method of Kang and Sotiropoulos [15]. Wave reflections at the lateral boundaries are prevented by using a sponge layer method. We validated the forcing method by applying it to simulate various wave cases of increasing complexity including simple monochromatic waves, monochromatic waves with several frequencies, three-dimensional directional waves with a single or multiple frequencies, and a broadband spectrum. Some of these cases involved transferring the wave field from the far-field domain to the near-field domain, which showed that the two domains were successfully coupled. On the other hand, the coupling of the wind field between the two domains was implement by feeding, at every time step, the velocity at the inlet plane of the near-field domain.

Once the wave generation method was implemented and validated we showed the capability of the method to study wave-body interactions. We applied the framework to replicate the experiments of [37] investigating a barge style floating wind turbine under different wave cases. By simulating a set of free decay test, we showed that the method can capture the natural periods of the system in heave and in pitch with very good agreement when compared to the experimental measurements. A very small discrepancy of less than 2% was observed between the natural period of the simulations and experiments in the pitch DoF. The discrepancy could be explained by the geometry simplification, which may have a small effect on the drag force, and thus on the computed natural period. This discrepancy was not observed on the heave DoF. For one of the heave decay cases we also compared the free surface elevation at two nodal locations showing excellent agreement. The decay tests also served to calibrate the amount of artificial damping introduced in the system to account for the frictional damping of the experimental apparatus and the damping due to the fact of using a simplified geometry. In contrast to the methods based on potential flow theory in which all damping needs to be added artificially, we showed that our approach already captures most part of the damping, and only a small contribution needs to be added artificially.

Once the optimum value of additional damping was obtained, as in [40], we performed a grid sensitivity study for all decay cases. For the single DoF free decay tests for which experimental data was available, we demonstrated that the computed solution was converging towards a grid independent solution. This solution was in very good agreement with the experimental data. An important conclusion that we could extract from these tests is that when the amplitude of the motion diminishes, approaching very small values, coarse grids tend to be inaccurate or insensible to such small amplitude motions, and finer grid resolution are required. This effect was taken in consideration when selecting the grid resolution to be used in the subsequent simulations of the wave-body interactions. After validating the code for each of the single DoF, we presented simulation results for a decay test, combining an initial heave displacement with an initial inclination. In that case, in addition to the structural response, we showed vorticity contours, illustrating the vortex shedding around the body edges and demonstrating the ability of the method to capture complex flow features around the floating body. Also, we presented free surface elevation contours showing the evolution of the platform radiated waves forming complex free surface elevation patterns.

With the RAO results, we demonstrated the validity of the method for simulating wave-body interactions. We compared the FSI results of the RAO with the experimental results and the simulation results performed using two widely used methods based on the potential flow theory, ANSYS-AQWA and WEC-Sim. We showed that the FSI solution is able to predict the results in overall better agreement with the experimental data. Also, the CFD-FSI method was shown to provide additional information not captured by the lower order methods such as the flow features formed at the vicinity of the structure.

Both, high order CFD-FSI models and lower order based methods can have a mutual benefit if employed in a wise manner. On the one hand, the CFD-FSI framework is able to provide a great deal of information which may be useful for developing and calibrating low-dimensional dynamic models of floating platforms. On the other hand, since the CFD-FSI model is computationally very intensive, lower order models may be employed to obtain a quick first estimation of the floating platform response under different scenarios of wind and wave conditions. This first approximation can be useful to select the most relevant wind/wave scenarios to be computed in high resolution with the CFD-FSI framework.

It is important to note that the turbine model case we studied was characterized by low amplitude motions, which are conditions favorable for the lower order models. We expect that for more severe environmental conditions only CFD-FSI models would provide reasonable results. The cost of the FSI model is significantly higher, but, can certainly be justified with the higher level of accuracy, the wider range of applicability, and extra features of the problem that can be captured.

To investigate the effect of turbine rigid body motions to the turbine wake we simulated the case of an oscillating wind turbine model studied experimentally in the SAFL wind tunnel in the second phase of the experiments of [37]. In particular, we simulated two cases, a static turbine case and a pitching turbine case. In line with the experimental data, we showed through simulations that the turbine pitching motion has minimal effect to the averaged stream-wise velocity profiles and turbulence statistics. We leave as a future work to further study this case by looking into phase averaged results and to simulate more cases involving motions in the other DoF.

Finally, to demonstrate the full potential of the framework. We applied it to simulate a $13.2MW$ offshore floating wind turbine under realistic site-specific ocean wind and wave conditions. The method was able to capture the turbine response in the 6 DoF, considering the platform-wave interactions, the effect of the turbulent wind on the turbine, and the gyroscopic effect of the spinning rotor.

2.6 Publications and Presentations

2.6.1 Publications

Calderer, A.; Kang, S.; Sotiropoulos, F. Level Set Immersed Boundary Method for Coupled Simulation of Air/Water Interaction with Complex Floating Structures. J. Comput. Phys. 2014, vol. 277, 201-227.

Calderer, A.; Guo, X.; Shen, L.; Sotiropoulos, F. Coupled fluid-structure interaction simulation of floating offshore wind turbines and waves: a large eddy simulation approach. Journal of Physics: Conference Series. 2014, vol 524., 012091.

Calderer, A.; Kang, S.; Sotiropoulos F. Large Eddy Simulation of Fluid Structure Interaction for Complex Floating Structures. ERCOFTAC Symposium on Unsteady Separation in Fluid-Structure Interaction. 2013.

Calderer, A. Fluid-Structure Interaction Simulation of Complex Floating Structures and Waves. Thesis Dissertation. 2015.

Calderer, A.; Guoa, X.; Shen, L.; Sotiropoulos, F. Fluid-structure interaction simulation of floating structures interacting with complex, large-scale ocean waves and atmospheric turbulence. To be sumbitted in the Journal of Computational Physics.

Calderer, A.; Feist, C.; Rueh, K.; Guala, M.; Sotiropoulos, F. Experimental validation of a Fluid-Structure interaction model for simulating offshore floating wind turbines. To be sumbitted in the Journal of Fluds and Structures.

2.6.2 Presentations

Calderer, A.; Kang, S.; Sotiropoulos, F. Level Set Immersed Boundary Method for Coupled Simulation of Air/Water Interaction with Complex Floating Structures. J. Comput. Phys. 2014, vol. 277, 201-227.

Calderer, A.; Feist, C.; Ruehl, K.; Guala, M.; Sotiropoulos, F. Experimental validation of a Fluid-Structure interaction model for simulating offshore floating wind turbines. 67th Annual Meeting of the APS Division of Fluid Dynamics 59 (20), 2014.

Calderer, A.; Guo, X.; Shen, L.; Sotiropoulos, F. Coupled fluid-structure interaction simulation of floating offshore wind turbines and waves: a large eddy simulation approach. The Science of Making Torque from Wind Conference. Poster presentation. Copenhagen, Denmark, June 2014

Calderer, A.; Guo, X.; Shen, L.; Sotiropoulos, F. Fluid-structure interaction simulation of floating wind turbines interacting with complex, large-scale ocean waves. 66th Annual Meeting of the APS Division of Fluid Dynamics 58 (18). 2013.

Calderer, A.; Guo, X.; Shen, L.; Sotiropoulos, F. Fluid-Structure interaction simulation of floating wind turbines interacting with complex, large-scale ocean waves. World Congress on Computational Mechanics. Barcelona, Spain. July 2014

Calderer, A.; Kang, S.; Sotiropoulos, F. Coupled level-set CURVIB method for fluid-structure interaction simulations of arbitrarily complex floating bodies. 65th Annual Meeting of the APS Div. of Fluid Dynamics. San Diego, CA. Nov. 2012

Calderer, A. Fluid-Structure Interaction Simulation of Complex Floating Structures and Waves. Thesis Final Defense. October 2015.

3 Task 1.2: The Farm Scale Model

3.1 Goals and Objectives

The goal of this subtask was to implement in the existing model: 1) the capability to simulate turbines and wind farms using actuator type models; 2) the capability to simulate wind farms using large-eddy simulation loosely coupled with mesoscale simulations. The wave effects in farm scale simulations can be included in two different approaches: 1) The fully-coupled non-linear approach, which is a straightforward extension of the two-phase flow level-set approach developed in task 1.1; 2) The one-way coupled, linear approach, in which the waves are decoupled from the rest of the problem. This was achieved with the same far-field/near-field coupling approach developed in task 1.1, but prescribing the wave field in the near-field solver using the Immersed Boundary method.

3.2 Project team

Fotis Sotiropoulos James L. Record Professor, Director of the St. Anthony Falls Laboratory. Principal investigator of the project.

Xiaolei Yang Research associate in the Saint Anthony Falls Laboratory. Dr Yang developed and implemented the farm scale models and performed the simulations presented in this subtask (subtask 1.2).

3.3 Technical Approach

In the computational framework, the rotor blades can be either parameterized using the actuator models or directly resolved using the CURVIB method. Three different actuator models, i.e., actuator disk model, actuator line model and actuator surface model, were implemented in the VFS code. In the actuator disk model, the rotor is modelled using a permeable disk with a induction factor or thrust coefficient to be specified. In the actuator line model, each blade is represented by a rotating line. Forces are distributed on each line (blade) to represent the effects of wind on the blades. The forces are calculated based on a blade element approach, which divides the blade into discrete elements in the radial direction and employs tabulated airfoil data (chord, twist, drag and lift coefficients). In the present actuator surface model, the forces calculated from the actuator line model are distributed on a surface formed by the foil chord instead of a line. It was shown in an axial-flowing hydrokinetic turbine simulation that a nacelle plays an important effect in the wake behaviour. In order to account for the nacelle effects more accurately, an actuator model for the nacelle was developed in VFS, where the nacelle is represented by its actual surface with distributed forces. A schematic of various turbine models in VFS is shown in Figure 19.

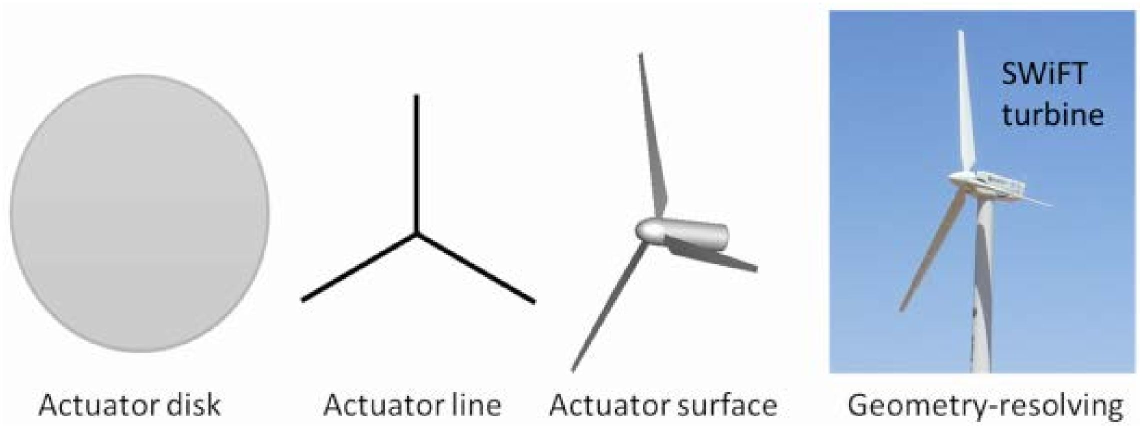


Figure 19: Schematic of various turbine models in VFS (from left to right: actuator disk model, actuator line model, actuator surface model with actuator model for nacelle and geometry-resolving model using CURVIB).

3.3.1 Actuator Disk Model

In the actuator disk model, the turbine rotor is represented by a circular disk that is discretized with an unstructured triangular mesh. The body force of the disk per unit area is the following

$$F_{AD} = -\frac{F_T}{\pi D^2/4}, \tag{26}$$

where D is the rotor diameter and F_T is the thrust force computed as

$$F_T = \frac{1}{2}\rho C_T \frac{\pi}{4} D^2 U_\infty^2, \tag{27}$$

where U_∞ is the turbine incoming velocity, $C_T = 4a(1-a)$ is the thrust coefficient taken from the one-dimensional momentum theory, and a is the induction factor. The incoming velocity U_∞ is also computed from the one-dimensional momentum theory as

$$U_\infty = \frac{u_d}{1-a}, \tag{28}$$

where u_d is the disk-averaged stream-wise velocity computed as

$$u_d = \frac{4}{\pi D^2} \sum_{N_t} u(X) A(X), \tag{29}$$

where N_t is the number of triangular elements composing the disk mesh, $A(X)$ is the area of each element, and $u(X)$ is the velocity at the element centers. The fluid velocity at the disk ($u(X)$) requires interpolation from the velocity values at the surrounding fluid mesh points, as the nodes from the fluid and disk meshes do not necessarily coincide. If we consider X to be the coordinates of the actuator disk nodes and x the coordinates of the fluid mesh nodes, the interpolation, which uses a discrete delta function, reads as follows

$$u(X) = \sum_{N_D} u(x)\delta_h(x - X)V(x), \tag{30}$$

where δ_h is a 3D discrete delta function, $V(x)$ is the volume of the corresponding fluid cell, and N_D is the number of fluid cells involved in the interpolation.

Finally, the body force F_{AD}, which is computed at the disk mesh nodes, needs to be distributed over the fluid cells located in the immediate vicinity using the following expression:

$$f_{AD}(x) = \sum_{N_D} F_{AD}(X)\delta_h(x - X)A(x). \tag{31}$$

3.3.2 Actuator Line Model

In the actuator line method, each blade of the rotor is modeled with a straight line, divided in several elements along the radial direction. In each of the elements, the lift (L) and drag (D) forces are computed using the following expressions:

$$L = \frac{1}{2}\rho C_L C V_{rel}^2, \tag{32}$$

$$D = \frac{1}{2}\rho C_D C V_{rel}^2, \tag{33}$$

where C_L, C_D are the lift and drag coefficients, respectively, taken from tabulated two-dimensional (2D) airfoil profile data, C is the chord length, and V_{ref} is the incoming reference velocity computed as

$$V_{rel} = (u_z, u_\theta - \Omega r) \tag{34}$$

where u_z and $u_\theta - \Omega r$ are the components of the velocity in the axial and azimuthal directions, respectively, Ω the angular velocity of the rotor, and r the distance to the center of the rotor.

To compute the reference velocity at the line elements, similarly to the actuator disk method, the velocity at the fluid mesh is transferred to the line elements using a discrete delta function as given by equation (30). Once the lift and drag forces are computed at each of the line elements, the distributed body force in the fluid mesh can be computed using the following equation:

$$f_{AL}(x) = \sum_{N_L} F(X)\delta_h(x - X)A(x). \tag{35}$$

where N_L is the number of segments composing one of the actuator lines, $F(X)$ is the projection of L and D, expressed in actuator line local coordinates, into Cartesian coordinates.

3.3.3 Actuator surface model for blades

In the actuator surface model for blades, the blade geometry is represented by a surface formed by the chord lines at every spanwise locations of the blade. The forces are calculated in the same way as in the actuator line model. The lift ($\boldsymbol{L}$) and drag ($\boldsymbol{D}$) at each radial location are calculated as follows:

$$\boldsymbol{L} = \frac{1}{2}\rho C_L c |\boldsymbol{V}_{rel}|^2 \boldsymbol{n}_L, \tag{36}$$

and

$$\boldsymbol{D} = \frac{1}{2}\rho C_D c |\boldsymbol{V}_{rel}|^2 \boldsymbol{n}_D, \tag{37}$$

where C_L and C_D are the lift and drag coefficients, correspondingly, c is the chord, $\boldsymbol{V}_{rel}$ is the relative incoming velocity, and $\boldsymbol{n}_L$ and $\boldsymbol{n}_D$ are the unit vectors in the directions of lift and drag, respectively. The relative incoming velocity $\boldsymbol{V}_{rel}$ is computed by

$$\boldsymbol{V}_{rel} = u_x \boldsymbol{e}_x + (u_\theta - \Omega r)\boldsymbol{e}_\theta \tag{38}$$

where Ω is the rotational speed of the rotor, $\boldsymbol{e}_x$ and $\boldsymbol{e}_\theta$ are the unit vectors in the rotor rotating and axial flow directions, respectively. The $u_x = \boldsymbol{u}(\boldsymbol{X}_l) \cdot \boldsymbol{e}_x$, and $u_\theta = \boldsymbol{u}(\boldsymbol{X}_l) \cdot \boldsymbol{e}_\theta$ are the axial and azimuthal components of the flow velocity at the leading edge. Generally, the leading edge point LE does not coincide with any background nodes. In the present work, we employ the smoothed discrete delta function (i.e. the smoothed four-point cosine function) proposed by Yang et al. [48] to interpolate $\boldsymbol{u}(\boldsymbol{X}_l)$ from the background grid nodes as follows:

$$\boldsymbol{u}(\boldsymbol{X}_l) = \sum_{\boldsymbol{x} \in g_{\boldsymbol{x}}} \boldsymbol{u}(\boldsymbol{x}) \delta_h (\boldsymbol{x} - \boldsymbol{X}_l) V(\boldsymbol{x}) \tag{39}$$

where $\boldsymbol{X}_l$ are the coordinates of the actuator line along the leading-edge of the blade in this paper, $g_{\boldsymbol{x}}$ is the set of the background grid cells, $V = h_x h_y h_z$ (h_x, h_y, and h_z are the grid spacings in the x, y and z directions, respectively) is the volume of the background grid cell, $\delta_h(\boldsymbol{x} - \boldsymbol{X}) = \frac{1}{V_{\boldsymbol{x}}}\phi\left(\frac{x-X}{h_x}\right)\phi\left(\frac{y-Y}{h_y}\right)\phi\left(\frac{z-Z}{h_z}\right)$ is the discrete delta function, and ϕ is the smoothed four-point cosine function [48], which is expressed as

$$\phi(r) = \begin{cases} \frac{1}{4\pi}\left(\pi + 2\sin\left(\frac{\pi}{4}(2|r|+1)\right) - 2\sin\left(\frac{\pi}{4}(2|r|-1)\right)\right), & |r| \leq 1.5, \\ -\frac{1}{8\pi}\left(-5\pi + 2\pi|r| + 4\sin\left(\frac{\pi}{4}(2|r|-1)\right)\right), & 1.5 \leq |r| \leq 2.5, \\ 0, & 2.5 \leq |r|, \end{cases} \tag{40}$$

in which $r_i = (x_i - X_i)/h_i (i = 1, 2, 3)$.

The blade rotation causes the stall delay phenomenon at the inboard sections of the blades, which increases the lift coefficients, and decreases the drag coefficients as compared with the corresponding two-dimensional airfoil data. To account for such three-dimensional rotational effect, the stall delay model developed by Du and Selig [49] is employed to correct the lift and drag coefficients from two-dimensional experiments or computations. In Du and Selig's model, the corrected lift and drag coefficients ($C_{L,3D}$ and $C_{D,3D}$) are calculated as follows:

$$C_{L,3D} = C_{L,2D} + f_L \left(C_{L,p} - C_{L,2D} \right), \tag{41}$$

and

$$C_{D,3D} = C_{D,2D} - f_D \left(C_{D,2D} - C_{D,0} \right), \tag{42}$$

where $C_{L,p} = 2\pi \left(\alpha - \alpha_0 \right)$, $C_{D,0} = C_{D,2D}$ for $\alpha = 0$, and the correction functions f_L and f_D are determined by

$$f_L = \frac{1}{2\pi} \left(\frac{1.6(c/r)a - (c/r)^{\frac{d}{\Lambda}\frac{R}{r}}}{0.1267b + (c/r)^{\frac{d}{\Lambda}\frac{R}{r}}} - 1 \right), \tag{43}$$

and

$$f_D = \frac{1}{2\pi} \left(\frac{1.6(c/r)a - (c/r)^{\frac{d}{2\Lambda}\frac{R}{r}}}{0.1267b + (c/r)^{\frac{d}{2\Lambda}\frac{R}{r}}} - 1 \right), \tag{44}$$

respectively, where $\Lambda = \Omega R / \sqrt{V_w^2 + (\Omega R)^2}$, R is the rotor radius, a, b and d are empirical correction factors. In this work, a, b and d are equal to 1 as in Du and Selig's paper [49].

Non-zero force can exist at the blade tip when the pitch angle is zero or a chambered foil is used [50]. This is in contradiction with the physics that the force should tend to zero at the tip due to pressure equalization as discussed by Shen et al. [50]. To correct this non-physical force behaviour, the tip-loss correction proposed by Shen et al. [51, 50] is also applied to the drag and lift coefficients computed from Eqs. (41) and (42). The final C_D and C_L employed to calculated the forces on the blade are then calculted as

$$C_L = F_1 C_{L,3D}, \tag{45}$$

and

$$C_D = F_1 C_{D,3D}, \tag{46}$$

where

$$F_1 = \frac{2}{\pi} \cos^{-1} \left(\exp \left(-g \frac{B(R-r)}{2r \sin\phi} \right) \right), \tag{47}$$

in which B is the number of blades, g is computed by

$$g = \exp \left(-c_1 \left(B\Omega R / V_w - c_2 \right) \right) + c3, \tag{48}$$

where c_1, c_2 and c_3 are the correction coefficients, which are equal to 0.125, 21 and 0.1 as in [51], respectively.

After calculating the lift ($\boldsymbol{L}$) and drag ($\boldsymbol{D}$) at every radial locations, the force per unit area on the actuator surface are then calculated as

$$\boldsymbol{F}(\boldsymbol{X}_s) = \left(\boldsymbol{L}(\boldsymbol{X}_l) + \boldsymbol{D}(\boldsymbol{X}_l) \right) / c, \tag{49}$$

where $\boldsymbol{X}_s$ are the coordinates of the actuator surface. For each triangular cell, the corresponding actuator line point is selected as the one when $|(\boldsymbol{X}_s - \boldsymbol{X}_l) \cdot \boldsymbol{e}_r|$ is minimum ($\boldsymbol{e}_r$ is the unit vector in the radial direction following the leading edge of the blade).

To calculate the forces on the background mesh for the flow field, the computed forces on the are then distributed from the actuator surface mesh as follows:

$$\boldsymbol{f}(\boldsymbol{x}) = \sum_{\boldsymbol{X}_s \in g_{\boldsymbol{X}_s}} \boldsymbol{F}(\boldsymbol{X}_s)\delta_h\left(\boldsymbol{x} - \boldsymbol{X}_s\right) A_s(\boldsymbol{X}_s), \tag{50}$$

where $g_{\boldsymbol{X}_s}$ is the set of the actuator surface grid cells and A_s is the area of the actuator surface mesh. The same discrete delta function as in Eq. (39) is employed.

3.4 Results

3.4.1 Flow over a stand-alone wind turbine

The VFS model was first validated by applying it to simulate the flow over a stand-alone wind turbine in a wind tunnel. The corresponding experiment can be found in [52]. A miniature turbine with diameter of 0.15 meters and hub height of 0.125 meters is employed in the experiment. The effect of the nacelle is modeled by extending the blade geometry at the root to the center of the rotor. The effects of the tower are neglected. The inflow velocity at hub height is 2.2 m/s. The tip-speed ratio is 4.1.

The Reynolds number based on the rotor diameter D and the incoming velocity at hub height U_h is 4.2×10^4. The size of the computational domain is $30D$, $12D$ and $3D$ in the streamwise (x), spanwise (y) and vertical (z) directions, respectively. The turbine is located 2D from the inlet boundary and in the middle of the spanwise direction. The hub height of the wind turbine is $z_h = 0.833D$. We carry out a grid refinement study using two different grids (N_x, N_y and N_z are the number of grid nodes in x, y and z directions, respectively.): G1 with $N_x = 201$, $N_y = 121$ and $N_z = 31$, G2 with $N_x = 401$, $N_y = 241$ and $N_z = 61$.The grid spacing near the turbine and the near wake region are D/10, D/20 for grids G1, G2.

Figure 20 shows the time-averaged streamwise velocity profiles at the inlet (20 (a)) and different downstream locations (2D to 14D downstream of the turbine) for the three different grids. As shown in Figure 20 (a) good agreement with the wind tunnel measurements is obtained for all three grids at the inlet. At the 2D and 3D downstream locations on the other hand (see Figures 20 (b) and 20 (c)), the simulation underpredicts the velocity deficits. This is probably because of the simple model we employ for the nacelle and the fact that no attempt has been made to model the tower, the effects of which should be significant at these near wake locations.

The streamwise turbulence intensities at the inlet and different downstream locations are shown in Figure 21. As shown in Figure 21 (a), the calculated turbulence intensities agree very well with the wind tunnel measurements for all the three grids at the inlet except for the several points adjacent to the ground, where the streamwise

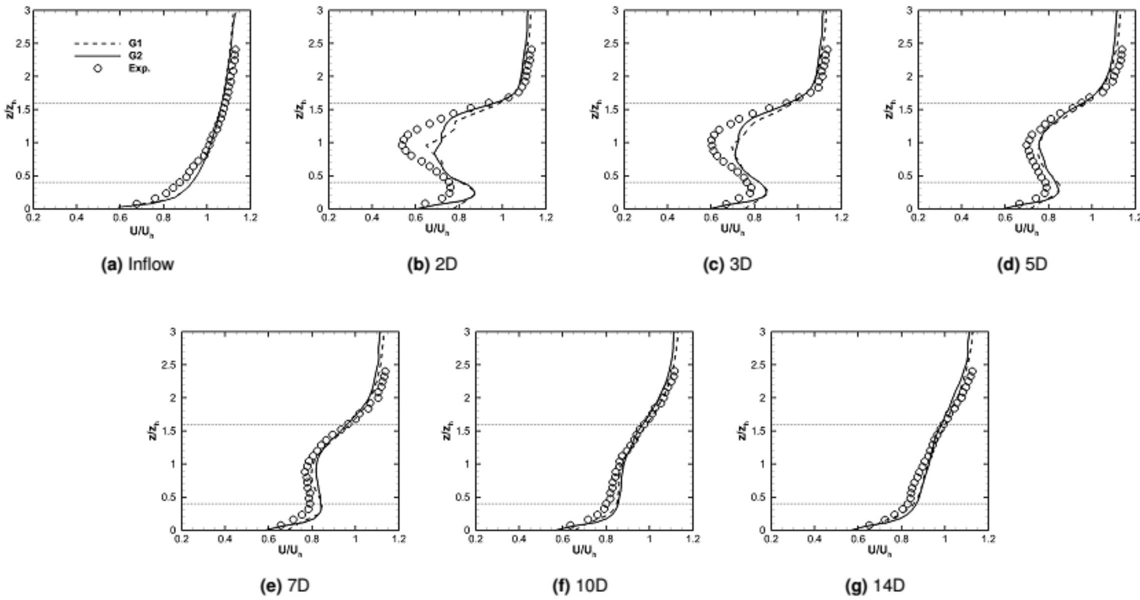


Figure 20: Time-averaged streamwise velocity profiles at different downstream locations for different grids. The experimental data is digitally extracted from [53].

turbulent intensities are over predicted for all the grids, which was probably because of the wall model and also observed in [28].

The computed primary Reynolds shear stresses ($-\langle uw \rangle$) are compared with measurements in Figure 22. As seen in Figure 22 (a), the calculated Reynolds shear stress at the inflow boundary agrees very well with that from the experiments for all the three grids. At the 2D downstream location, as shown in Figure 22 (b), the LES fails to predict the two peaks within the wind turbine region on all the three grids. The peak at the top tip of the wind turbine, on the other hand, is captured on all the three grids. At the 3D downstream location (Figure 22 (c)), good agreement with the wind tunnel measurements is observed for grid G2 at all locations. At further downstream locations (from 5D to 14D as shown in Figures 22 (d) to 22 (g)), good agreements with the wind tunnel measurements are obtained for all the three grids.

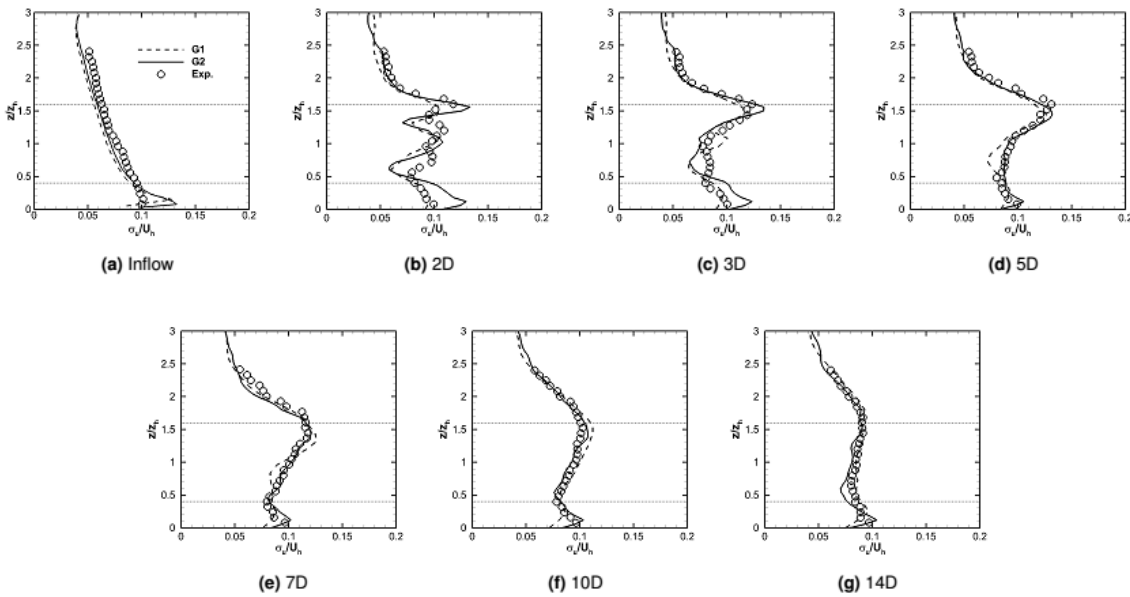


Figure 21: Streamwise turbulence intensities at different downstream locations for different grids. The experimental data is digitally extracted from [53].

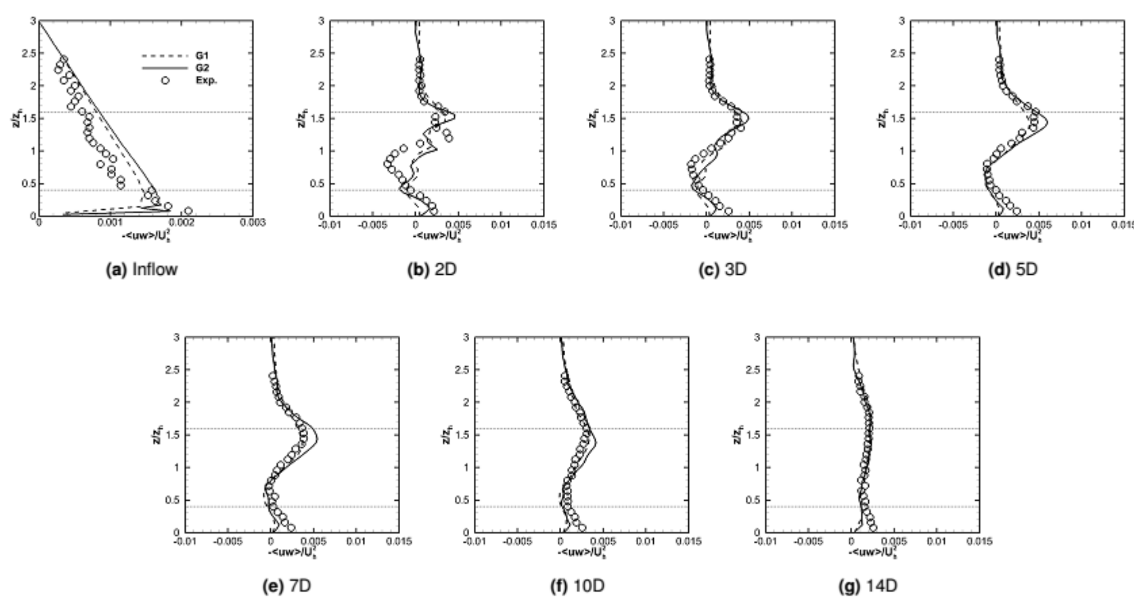


Figure 22: Primary Reynolds shear stresses at different downstream locations for different grids. The experimental data is digitally extracted from [53].

3.4.2 Flow over an aligned wind-turbine array

In this section, the VFS model was further validated by simulating the flow over an aligned wind-turbine array. The details of the wind tunnel experiment can be found in [54]. The same miniature turbine as that used for the previously discussed single turbine case was employed in this experiment. The inflow velocity at hub height is 2.1 m/s. The tip-speed ratios are 4.1 for all the turbines. The Reynolds number based on the rotor diameter D and the incoming velocity at hub height U_h is 2.5×10^4. The size of the computational domain is $30D$, $12D$ and $5D$ in the x, y and z directions, respectively. Two grids are employed: one grid G1 with 601, 241 and 101 grid nodes in x, y and z directions, respectively, which are uniformly distributed in all three directions; the other grid G2 with the same grid nodes in x and y direction but with 121 grid nodes in the vertical direction, in which the grid is uniform for $z < 2D$ with grid spacing $D/40$ and then stretched to top boundary. The streamwise and spanwise turbine spacings are 5D and 4D, respectively. Six and three wind turbines are placed in the streamwise and spanwise directions, respectively. The first row (y direction) of wind turbines is located 2D downstream from the inlet boundary. The turbulent flow field at the inlet is generated from a precursory fully developed turbulent flow. Simplified boundary layer equations are employed as wall model [55, 56, 57].

Figure 23 compares the calculated profiles of time-averaged streamwise velocity along the streamwise direction at various vertical locations with the wind tunnel measurements. At the bottom tip of the wind turbines (Figure 23 (a)), good agreement with the wind tunnel measurements is obtained for the first 4 rows of turbines. On the other hand, some discrepancies are observed for the wakes behind the 5th and 6th rows of wind turbines. For the comparisons at the turbine hub height (Figure 23 (b)), overall good agreement is obtained. However, the simulations underpredict somewhat the velocity deficit in the near wake locations, which is consistent with the previously discussed results for the single wind turbine simulation (Figure 20). At the top tip of the wind turbine (Figure 23 (c)), some discrepancies are observed at the locations in the near wake of the first row turbine and in the far wake (3D to 5D downstream locations) locations of the 3rd and 4th rows of the wind turbines. Very good agreement is obtained for all the other locations.

Comparisons of the streamwise turbulence intensities at the bottom tip, hub height and top tip of the wind turbines are shown in Figure 24. At the bottom tip of the turbines (Figure 24 (a)), the turbulence intensity is under predicted. With refined mesh in the vertical direction, no significant improvements were observed. The reasons for this under prediction are not very clear. Similar under prediction of the turbulence intensity in the region below the hub at 3D and 4D turbine downstream locations was also observed in [58] for the turbines in the 4th rows and further, in which the turbines are parameterized as actuator disks with/without turbine rotations (in which more assumptions are needed), and the effects of nacelle and tower are modelled using drag coefficients. At the hub height of the wind turbine (Figure 24 (b)), the computed turbulence intensities agree well with the wind tunnel measurements at 1D and 2D downstream locations but are under predicted at 3D and 4D downstream locations.

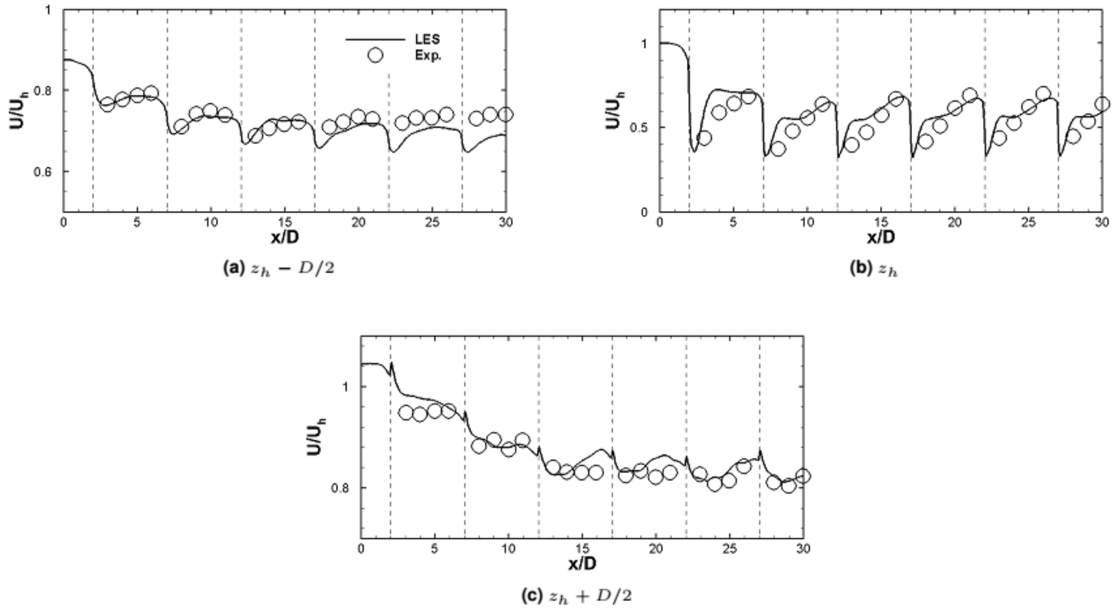
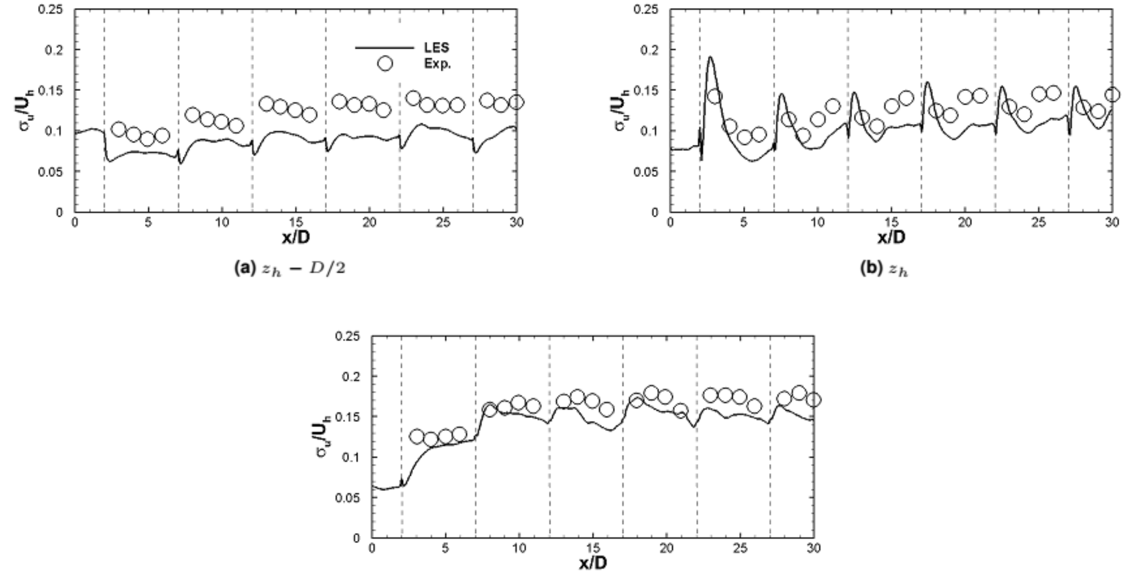


Figure 23: Variations of the mean streamwise velocity at different vertical locations. Solid line: G1, dashed lineL G2. The locations of wind turbines are marked using dotted lines. The experimental data is digitally extracted from [54].

At the top tip of the wind turbines (Figure 24 (c)), on the other hand, good overall agreement is obtained.

The contours of the mean velocity and turbulence statistics are shown in Figure 25. As seen in Figure 25 (a), the velocity deficits from the first wind turbine are much weaker than those from the further downstream wind turbines. The wakes from the further downstream turbines, on the other hand, are nearly the same. The contours of the time-averaged spanwise velocity shown in Figure 25 (b) reveal that significant spanwise velocity is induced in the wake by the rotating blades. The spanwise velocity in the upper part above the hub height is larger than the one in the lower part, and remains significant at the downstream turbines. The contours of the streamwise turbulence intensity are shown in Figure 25 (c). One region with large turbulence intensity is located just behind the wind turbine, which decays rapidly within 1.5D (for the turbines in the first row) or 1D downstream of the wakes. The other region with relatively large turbulence intensity is located within the shear layer generated from the bottom tip of the wind turbines. A third region with large turbulence intensity is located within the shear layer at the top tip of the wind turbines. Within this region turbulence intensities are larger than those at the bottom tip shear layer and persist for all turbines. The strength of the turbulence intensity within the top tip shear layer of the turbines in the first row is much weaker than that from the further downstream turbines. The contours of Reynolds shear stress are shown in Figure 25 (d).

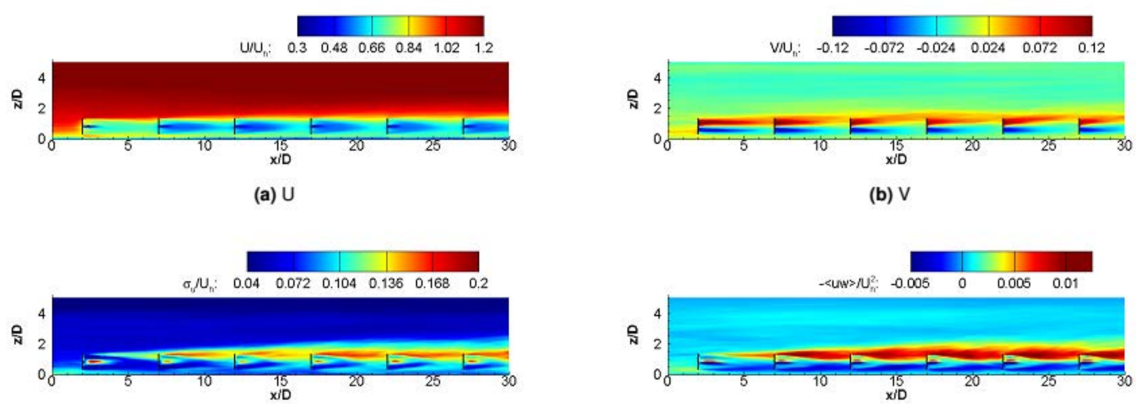


(a) $z_h - D/2$

(b) z_h

(c) $z_h + D/2$

Figure 24: Variation of streamwise turbulence intensity at different vertical locations. Solid line: G1, dashed lineL G2. The locations of wind turbines are marked using dotted lines. The experimental data is digitally extracted from [54].



(a) U

(b) V

(c) σ_u

(d) $-\langle uw \rangle$

Figure 25: Contours of the time-averaged streamwise velocity (a), spanwise velocity (b), streamwise turbulence intensity (c) and Reynolds shear stress (d) on a x-z plane through the center of the wind turbines in the middle column.

3.4.3 Flow over a hydrokinetic turbine

In this part, we simulate the turbulent flow over an axial-flow hydrokinetic turbine in a water tunnel using different actuator blade and actuator nacelle models. The corresponding experiment of this case was carried out by Chamorro et al. [59] at Saint Anthony Falls Laboratory, University of Minnesota. Kang, Yang and Sotiropoulos [2] carried out geometry-resolving and actuator line simulations of this case.

Figure 26 shows the comparisons between the measured mean velocity and turbulence statistics profiles [60] and the LES-AD and LES-AL results, respectively. As one would anticipate, given the simplicity of the AD and AL models, the overall agreement with the measurements is worse than that of the turbine-geometry-resolving (TR) LES. This trend is especially apparent in the near wake region where the geometrical details of the turbine greatly impact the flow structure. It is evident, for instance, from figure 26 that for $z/D \leq 2$ both the AD and AL models are not able to resolve the low velocity region near the hub height ($y/D = 0.8$). Furthermore, the pockets of high TKE measured around the hub height at $z/D = 0.5$ and $z/D = 1$ are either completely missing (in LES-AD) or under-predicted (in LES-AL) in the present simulations and the TKE level near the blade tip positions is largely underpredicted especially by the LES-AD simulation. Overall, the LES-AL simulations yield better agreement with the measurements than the LES-AD, which confirms previously reported conclusions in the literature that including the tip vortices in the simulations improves the predictive capability of actuator type models [53]. We also note that the LES-AL simulation resolves with good accuracy the structure of the swirl velocity profiles in the near wake and more specifically captures the two-layer rotating flow pattern with the counter-rotating inner region immediately downstream of the turbine hub.

It was also found that the regular actuator line model underpredicts the turbulence intensity in the far wake because it cannot capture the inner-outer wake interaction and thus the wake meandering in the far wake accurately.

We test now using the same test case the new class of actuator surface models of blades and nacelle developed herein for better incorporating the geometrical effects of the blades and nacelle. Several combinations of different blade models and nacelle models are tested on this case. The blade models considered include the actuator line model and the actuator surface blade model. The nacelle models considered include a disk model and an actuator surface nacelle model with $C_f = 0$, 1 and 2. In the simulations, all the cases are first carried out until the total kinetic energy reached a quasi-steady state, and subsequently the flow fields were averaged for $90T$, where $T = 2\pi/\Omega$ is the rotor revolution period. The rotational speed of the rotor is 9.28 rad/s, which gives a tip-speed ratio of 5.8 based on the bulk velocity.

We examine the performance of the actuator surface blade model with and without the nacelle model. Two different grid resolutions are employed: 1) D/50 the one employed in most cases; and 2) D/100 from a refined grid. From the vertical profiles of the time-averaged streamwise velocity shown in Fig. 27 (a), we can see the wake from the nacelle is well captured by employing the actuator surface nacelle model.

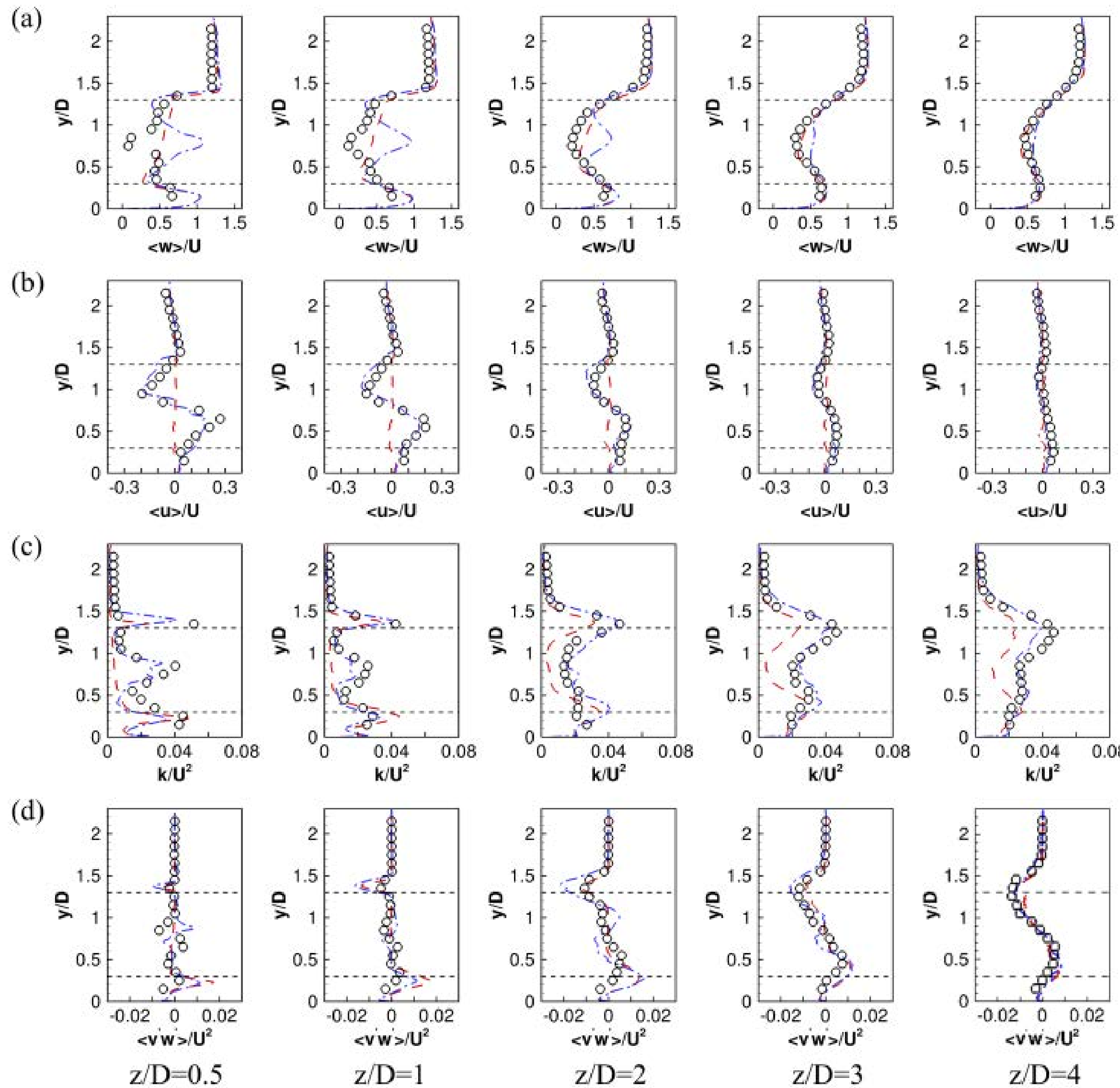


Figure 26: Comparisons of the vertical profiles of flow quantities computed by the actuator disk model (red dashed line) and actuator line model (blue dash-dotted line) with the measurements (symbols). (a) streamwise velocity, (b) transverse velocity, (c) turbulence kinetic energy, (d) primary Reynolds shear stress.

For the cases with the nacelle model, the time-averaged streamwise velocity computed on the grid with D/50 agrees well with the measurements, while the velocity deficits are somewhat overpredicted at 3D rotor downstream on the grid with spacing D/100. For the rotational velocities at 1D rotor downstream shown in Fig. 27 (b), underpredictions of their magnitudes are observed for both D/50 and D/100 grid resolutions with the actuator surface nacelle model. The prediction capability of the proposed model on the turbulence kinetic energy is demonstrated in Fig. 27 (c). As seen, the turbulence kinetic energy profiles computed from the cases with the actuator surface blade and nacelle models on both grid resolutions are in good agreement with the measurements at all downstream locations, which are underpredicted significantly by the actuator surface blade model only case. The comparison of the primary Reynolds shear stress $\langle v'w' \rangle$ profiles computed from different setups is shown in Fig. 27 (d). One observation is that the magnitude of $\langle v'w' \rangle$ computed on the grid with spacing D/100 is overpredicted at 3D turbine downstream. Other than that, good agreement is obtained at further downstream locations. One thing needs to be emphasized is that the prediction of the $\langle v'w' \rangle$ magnitude at 1D turbine downstream for the grid with spacing D/50 is improved notably. Although the proposed model on both grid resolutions gives improved predictions, which are in overall good agreement with the measurements, it is still hard to conclude the grid independent solution has been obtained. This inherent shortage of the actuator-type models depends on the asymptotic behaviour of the turbine parametrizations, such as the geometry representation of the blades using forces and the employed tip-loss correction and the correction to the three-dimensional and rotational effects, when we refined the mesh. Systematic investigation of such problems deserves several papers on it and is beyond the scope of this work.

To compare the spatial extent of the wake meandering region obtained by the TR and AL approaches, we superimpose in figure 28 the same TKE contour level on the turbine symmetry plane (x=0). The selected TKE contour level ($k/U^2 = 0.03$) outlines the zones of high TKE in the computed flowfields, including the near hub, turbine tips and wake meandering regions, and succinctly captures in a single figure all previously discussed differences among the three LES results. For the LES-TR simulation we include in this figure results obtained on two grids, Grid I (coarser) and II (finer), to show that insofar as the high TKE zone in the wake meandering region ($3 < z/D < 8$) is concerned the two LES-TR simulations yield essentially identical results. The main difference between the two TR simulations in this regard is that on the coarse grid the wake meandering region starts closer to the turbine and as a result it is somewhat longer than that obtained on the fine grid (we will elaborate on the reason for this discrepancy in the next paragraph). As seen in figure 28, in the LES-AL simulation wake meandering is confined within a shear layer that emanates immediately downstream of the turbine top tip, starts growing in thickness at approximately $z/D = 1$, reaches maximum thickness at $z/D \sim 3.5$ ends at $z/D \sim 5.5$. In the LES-TR flowfield, on the other hand, wake meandering is marked by an elliptical-like region of high TKE levels along the turbine top tip shear layer that starts between at $z/D \sim 2.5$, reaches maximum thickness at $z/D \sim 5.5$ and extends up to $z/D = 8$. Furthermore, the maximum thickness of wake meandering region in the LES-TR flowfield , which is reached where the wake meandering region in LES-AL has essentially diminished, is approximately equal to the turbine radius.

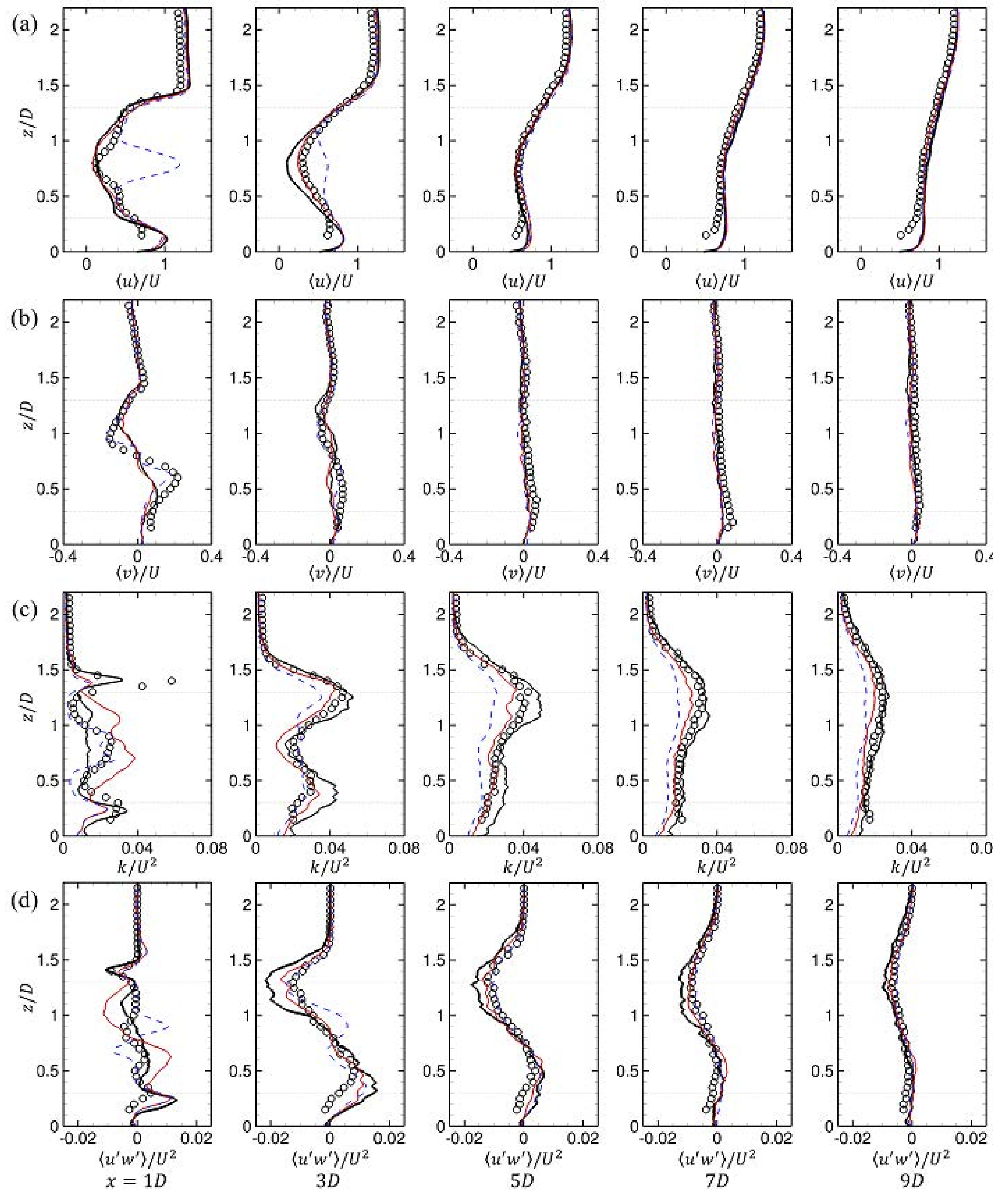


Figure 27: Vertical profiles of (a) the time-averaged streamwise velocity $< w >$ and (b) the spanwise (rotational) velocities hui, (c) the turbulence kinetic energy k and (d) the Reynolds shear stress $< v'w' >$ at different downstream locations. Symbols: measurements from [59]; Thick black solid lines: actuator surface models for both blades and nacelle on the finer grid D/100; Red solid lines: actuator surface models for both blades and nacelle; Blue dashed lines: actuator surface blade model only.

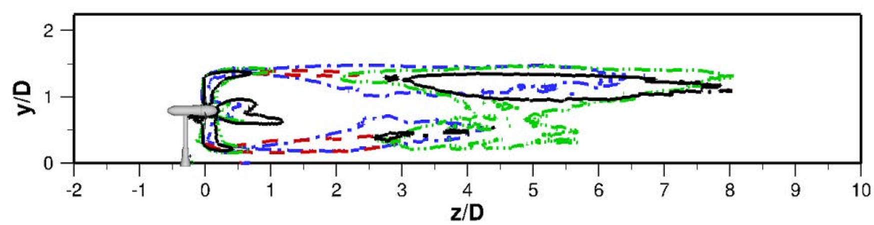


Figure 28: Comparisons of the contours of $k/U^2 = 0.03$ at the $x = 0$ plane from LES-TR (red dashed lines for Grid I; black solid lines for Grid II) and LES-AL (blue dash-dotted lines).

3.4.4 Validation of the actuator model for nacelle

Validation of the actuator model for the nacelle is carried out for flow past a sphere at $Re = UD/\nu = 3700$, where U is the velocity of the incoming flow and D is the sphere diameter. The size of the computational domain is $L_x \times L_y \times L_z = 19D \times 8D \times 8D$. The number of grid nodes is $N_x \times N_y \times N_z = 192 \times 82 \times 82$.

The time-averaged flow field is shown in Figure 29 for (a) the streamwise velocity contours and (b) the crosswise velocity contours. The profiles of the time-averaged streamwise velocity (a) and radial velocity (b) are compared with DNS data [61] in Figure 30. The major discrepancy happens in the near wake, where the size of the separation bubble is not captured accurately (The separation bubble from the current simulation is 1.9D, which it is 2.28D in the direct numerical simulation (DNS) [61].). For further downstream locations, overall good agreement is obtained. In Figure 31, the streamwise and radial turbulence intensities are compared with DNS data. Overall good agreement is obtained, except in the near wake of the sphere, where the radial turbulence intensity is under predicted by the present actuator model.

Considering the very coarse mesh employed in the actuator simulation, the accuracy obtained is acceptable although some discrepancies are observed in the near wake. This actuator model for the nacelle will be employed in most turbine simulations.

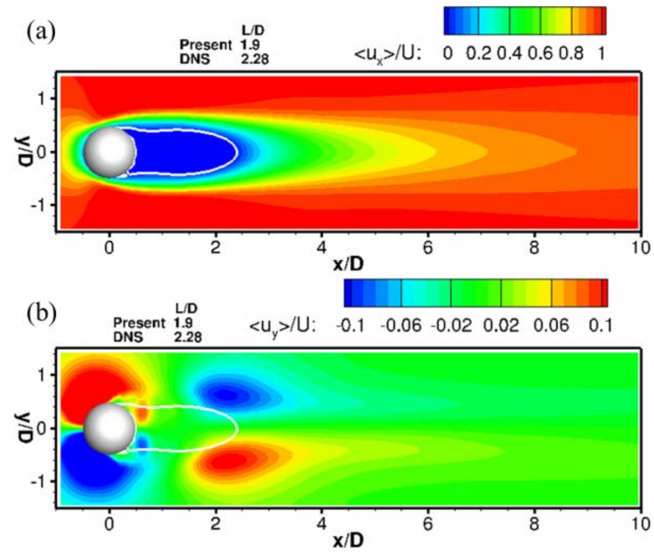


Figure 29: Time-averaged flow fields ((a): streamwise velocity; (b): crosswise velocity) for flow past a sphere at Re=3700 simulated using the present nacelle model on grid spacing D/10.

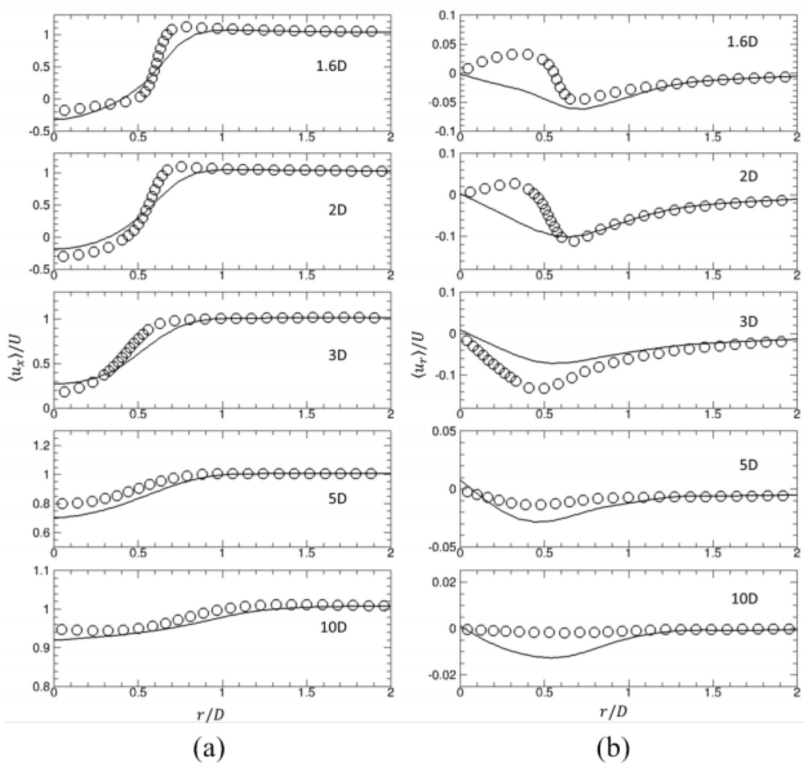


Figure 30: Comparison of profiles for time-averaged streamwise velocity (a) and radial velocity (b) simulated using the present actuator model for nacelle (solid lines) with direct simulation results in [61] (circles).

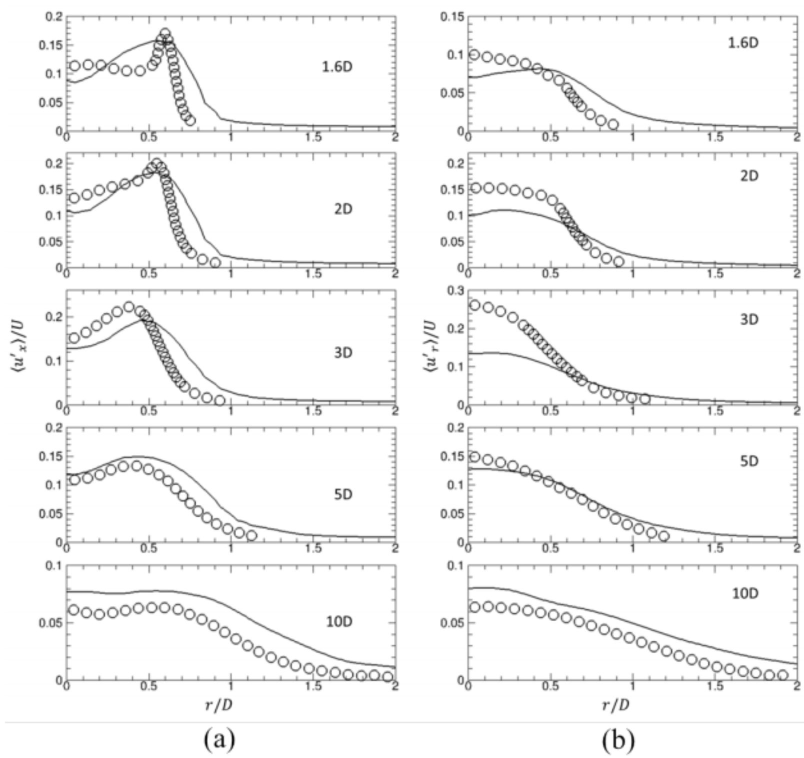


Figure 31: Comparison of profiles for streamwise turbulence intensity (a) and radial turbulence intensity (b) simulated using the present actuator model for nacelle (solid lines) with direct simulation results in [61] (circles).

3.5 Summary and Conclusions

We implement actuator type models in the existing computational framework to simulate turbines and wind farms and we added the capability to incorporate conditions from a mesoscale model. Combining the actuator type models developed in this task with the capabilities from task 1.1, the code can be applied to wind farm scale simulations using two different approaches: 1) The fully-coupled non-linear approach, which is a straightforward extension of the two-phase flow level-set approach developed in task 1.1 but using multiple turbines; and 2) The one-way coupled, linear approach, in which the waves are decoupled from the rest of the problem. This was achieved with the same far-field/near-field coupling approach developed in task 1.1, but prescribing the wave field in the near-field solver using the Immersed Boundary method.

The actuator type models were validated extensively against SAFL wind tunnel measurements for a single wind turbine and wind-turbine arrays ([1]), SAFL main channel measurements for a hydrokinetic turbine ([2]), field measurements from Mower County wind farm and EOLOS wind turbine ([1]). Overall good agreement with the measurements is obtained.

In the hydrokinetic turbine case, it was found that the nacelle has an important effect on the inner and outer shear layer interaction and the far wake turbulence intensities. However, the nacelle model in the literature cannot capture such effect. An improve actuator surface model for nacelle was proposed and applied to simulate the flow over a nacelle of a hydrokinetic turbine. Promising results on a very coarse grid (10 cells per diameter) were obtained in comparison with the results from a high-resolution (50 cells per diameter) immersed boundary simulation.

A turbine torque controller was implemented for the actuator line model. Validation was performed on the University of Minnesota EOLOS turbine for both uniform and turbulent inflow conditions. The computed power coefficient curve showed good agreement with that calculated from the FAST model (in which the blade element theory is employed) for the case with uniform inflow. Good agreement with the field measurement was also obtained for the computed power for the case with turbulent inflow.

In order to simulate the atmospheric stratification effects on the turbine and wind farm performance, Temperature equation was solved with wall function applied at the ground. Validations were performed on turbulent channel flow with temperature differences at two walls.

Loosely coupling of the high-resolution large-eddy simulation (LES) with a mesoscale model WRF (Weather Research and Forecasting Model) was implemented. In this loose coupling algorithm, the flow field from WRF with added synthetic turbulence was employed as inflow conditions for the LES domain. This loosely coupling was applied to the simulation of the turbines in Mower County wind farm. The computed power agrees well with the field measurements.

Besides the model wind turbine, hydrokinetic turbine, Mower County wind farm and EOLOS turbine cases, the present model was also applied to: 1) Simulation of a miniature wind turbine in the downstream of a three-dimensional hill with overall good agreements with the measurements obtained at SAFL wind tunnel (Yang et al. 2015); 2) Simulation of wind field at Prairie Island, MN, with/without hypothetically installed wind turbines (Yang et al. 2014); 3) Simulation of SWiFT (Scaled Wind Farm Technology) turbines with two different designs, which are located at the Reese Technology Center near Lubbock, TX, USA); 4) Simulation the three SWiFT wind turbines under three different wind directions and two different tip speed ratios.

3.6 Publications and Presentations

3.6.1 Publications

Yang, X.; Kang, S.; Sotiropoulos, F. Computational study and modeling of turbine spacing effects in infinite aligned wind farms. Phys. Fluids. 2012, 24, p. 115107.

Yang, X.; Howard, K.; Guala, M.; Sotiropoulos, F. Effects of a three-dimensional hill on the wake characteristics of a model wind turbine. Physics of Fluids. 2015, 27, no. 2, 025103.

Kang, S.; Yang, X.; Sotiropoulos, F. On the onset of wake meandering for an axial flow turbine in a turbulent open channel flow. Journal of Fluid Mechanics. 2014, vol. 744, p. 376-403.

Yang, X.; Sotiropoulos, F.; Conzemius, R. J.; Wachtler, J. N.; Strong, M. B. Large-eddy simulation of turbulent flow past wind farms in complex terrains: The Virtual Wind Simulator (VWiS). Wind Energy (2014).

Yang, X.; Boomsma, A.; Barone, M.; Sotiropoulos, F. Wind turbine wake interactions at field scale: a LES study of the SWiFT facility. Journal of Physics: Conference Series. 2014. Vol 524, 012139

Yang, X.; Khosronejad, A.; Chawdhary, S.; Calderer, A.; Angelidis, D.; Shen, L.; Sotiropoulos, F. Simulation-based approach for site-specific optimization of marine and hydrokinetic energy conversion systems. E-proceedings of the 36th IAHR World Congress. 2015.

Yang, X.; Boomsma, A.; Sotiropoulos, F.; Kelley, C.; Maniaci, D.; and Resor, B. Effects of spanwise blade load distribution on wind turbine wake evolution. Proceedings of the AIAA SciTech 33rd Wind Energy Symposium. 2015.

Yang, X.; Annoni, J.; Seiler, P.; Sotiropoulos, F. Modeling the effect of control on the wake of a utility-scale turbine via large-eddy simulation. In Journal of Physics: Conference Series. 2015, Vol. 524, No. 1, p. 012180.

Yang, X.; Sotiropoulos, F. LES investigation of in infnite staggered wind-turbine arrays. In Journal of Physics: Conference Series. 2014, Vol. 555, No. 1, p. 012109.

Yang, X.; Kang, S.; Khosronejad, A.; Sotiropoulos, F. Towards Simulation of Hydrokinetic Turbine Arrays with Sediment Transport in Natural Waterways. Proceedings of 2013 IAHR Congress. 2013.

Yang, X.; Kang, S.; Sotiropoulos, F. Toward a simulation-based approach for optimizing mhk turbine arrays in natural waterways. In Proceedings of the 1st Marine Energy Technology Symposium. 2013.

3.6.2 Presentations

Yang, X.; Howard, K.; Guala, M.; Sotiropoulos, F. Effects of a three-dimensional hill on the wake characteristics of a model wind turbine. 67th Annual Meeting of the APS Division of Fluid Dynamics 59 (20). 2014

Yang, X.; Sotiropoulos, F. An improved e ffective roughness height model for optimization of wind farm layout. 65th Annual Fall DFD Meeting, November 18-20, 2012, San Diego, California.

Yang, X.; Sotiropoulos, F. On the predictive capabilities of les-actuator disk model in simulating turbulence past wind turbines and farms. The 2013 American Control Conference, June 17 - 19, Washington, DC. 2013.

4 Task 1.3: Coupling Across Scales: Adaptive Mesh Refinement Code

4.1 Background

In most real life problems where complex geometries are involved, the nature of a conforming grid's topology may influence the robustness of the algorithm. Bad quality body fitted meshes, consisting of cells having large aspect ratio, skewness or even possible singularities may decrease the rate of convergence and could cause numerical instability. Composite overlaid/Chimera grids [62] have been developed to take advantage of the positive properties of structured and unstructured conforming grids. Even though the latter method is attractive leading to the generation of good quality grids around complex moving geometries, the interpolation between component grids is not conservative and the method may fail whenever overlap between components is insufficient. This is why numerous Cartesian Adaptive Mesh Refinement (AMR) methods have been developed to take advantage of the increased stability and accuracy provided by the inherent orthogonality and simplified layout of Cartesian meshes.

The idea of adaptive mesh refinement (AMR) is not new; non-conforming Cartesian AMR methods have been developed to take advantage of the increased stability and accuracy provided by grids orthogonality, but also of the capability to control the discretization error due to the adaptation potential. Caruso et al. [63], based on the hierarchical data structure, presented the first study for solving the laminar and Reynolds-averaged incompressible Navier-Stokes equations, while Coelho et al. [64] adopted a strong coupling approach between successive levels of refinement and discretized the flow equations on fully unstructured collocated grids. Anagnostopoulos [65] and Papadakis and Bergeles [66] extended the idea of multi-resolution simulations by introducing fully unstructured methods for solving 2D and 3D incompressible flows, respectively, and more recently flow simulations using unstructured Cartesian grids have been utilized to resolve real life problems including the flow field within urban street canyons [67]. The increased potential of AMR methods has encouraged scientific groups to develop general toolkits which can manage dynamic data structures and simplify the solution of the governing equations. Characteristic packages include AMR++/OVERTURE [68], SAMRAI [69], an object oriented Structured Adaptive Mesh Refinement (S-AMR) algorithm [70, 71] parallelized according to the Berger-Rigoutsos method [72] and PARAMESH [73]. The latter package has been used to enhance the efficiency of demanding calculations across a wide range of scientific fields including astrophysical studies [74] and fluid-structure interaction simulations for laminar and turbulent incompressible flows [75]. Lately, grid refinement techniques have been utilized to resolve the rich dynamics on the wakes of wind turbines. Zehle and Sørensen [76] used the EllipSys3D [77] solver to simulate the flow over a 2-blade NREL Phase VI rotor at different wind speeds and Tran et al.[78] performed aerodynamic analysis on the rotating rotor blades of a floating support platform by using the CD-Adapco STAR-CCM+ software package. In both studies, the finite volume flow solvers utilized overset grids and solved the Reynolds-Averaged Navier-Stokes equations using the $k-\omega$ closure model. Recently, Martinez et al. [79] used the OpenFOAM toolbox [80] to study the effect of the Actuator Line Model (ALM) and the Actuator Disk Model (ADM) with respect to the grid resolution of nested grids, on the wake of wind turbine rotors.

4.2 Goals and Objectives

The goal of this subtask was develop an Adaptive Mesh Refinement (AMR) flow solver for incompressible flows according to which the adaptively refined grid is employed as a unique and unified single-block on which the governing equations are discretized. The capabilities of the recently developed VFS code[3] are here enhanced by adopting an unstructured hybrid staggered/non-staggered approach; thus avoiding any flux mismatch or pressure discontinuity issues. In contrast to S-AMR solvers, the proposed grid layout facilitates the satisfaction of the divergence free constraint of the continuity equation and no special treatment for mass conservation across grid refinement interfaces is needed. By doing so, all the dependent variables are calculated on one grid, avoiding any existence of overlapped information which could increase memory requirements and raise discontinuity problems. The parallel fractional step solver is utilizing hanging nodes for second order finite differencing calculations. Concerning the parallelization of the algorithm, having established the adjacency list makes partitioning of the computational domain straightforward. This is contrast to conventional multi-level methods which require partitioning of each level individually; not ensuring load balance. The pressure-Poisson equation is solved using the Krylov subspace solver while the Algebraic Multi-Grid (AMG) method is chosen as a preconditioner for accelerating the convergence rate of the solver and the matrix-free Newton-Krylov method is utilized for solving the momentum equation. PETSc libraries [81, 82] were chosen to parallelize the overall algorithm.

4.3 Project team

Fotis Sotiropoulos James L. Record Professor, Director of the St. Anthony Falls Laboratory. Principal investigator of the project.

Dionysios Angelidis Post-doctoral researcher in the Saint Anthony Falls Laboratory. Dr Angelidis worked implemented the adaptive mesh refinement code of the present subtask (subtask 1.3).

4.4 Technical Approach

4.4.1 The governing equations

We solve the spatially filtered (for LES) incompressible Navier-Stokes equations in Cartesian unstructured grids. Assuming that $\{u_i\}$ are the Cartesian velocity components in $\{x_i\}$ coordinates the governing continuity and momentum equations, formulated in primitive variable form, are defined as:

$$\frac{\partial \bar{u}_i}{\partial x_i} = 0 \tag{51}$$

$$\frac{\partial \bar{u}_i}{\partial t} + \frac{\partial \bar{u}_i \bar{u}_j}{\partial x_i} = -\frac{\partial \bar{p}}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 \bar{u}_i}{\partial x_j \partial x_j} - \frac{\partial \tau_{ij}}{\partial x_j} + f_i \tag{52}$$

where p is the pressure divided by the density ρ and Re is the Reynolds number based on characteristic length scales. The sub-grid scales (SGS) are modeled by utilizing the Smagorinsky-Lilly model [83] while the calculation of the subgrid tensor τ_{ij} is based on the assumption that the filter width is equal to the grid spacing and the Smagorinsky constant is C_s=0.1. Wind turbine rotors are modelled by using an Actuator Line Model (ALM) and the effect of the rotation of the blades is represented by distributed forces f_i to the flow field [84]. A wall model is employed near walls based on a dynamic procedure for calculating the eddy viscosity from the mixing length with near wall damping [85].

4.4.2 Grid topology

Grid topology is relevant to the arrangement of the computational cells and the location of the dependent variables related to the discretization procedure. A major feature of the approach we develop herein is that even though the grid generation or regeneration algorithm may adopt a hierarchical octree-tree structure, the flow solver uses a fully unstructured data structure. The multi-level computational grid which covers the entire computational domain is therefore a unique single-block, in contrast to approaches employing hierarchical nested and locally structured sub-blocks [75] (fig. 32). The grid arrangement of the computational grid is based on the idea of the hybrid staggered/non-staggered grid layout [86] which facilitates the satisfaction of the divergence free constraint for the velocity field and the imposition of boundary conditions, avoiding at the same time pressure-velocity decoupling. For that reason, the dependent variables of the velocity components, which are normal to the cell faces, are located in the middle of the interfaces between adjacent cells and the pressure variables are cell-centered. The latter arrangement corresponds to the standard staggered layout; however, the velocity field is reconstructed at the cell centers and the discretization schemes can be applied in a collocated manner.

4.4.3 Lagrange reconstruction around the fine/coarse interface

The calculations around the fine/coarse interfaces of the mesh are here facilitated by adopting the concept of auxiliary/hanging nodes. They are located at the normal intersection of a line passing from the center of the fine cell and a plane passing from the center of mass of cells with the lower level of refinement (fig. 33). Hence, a

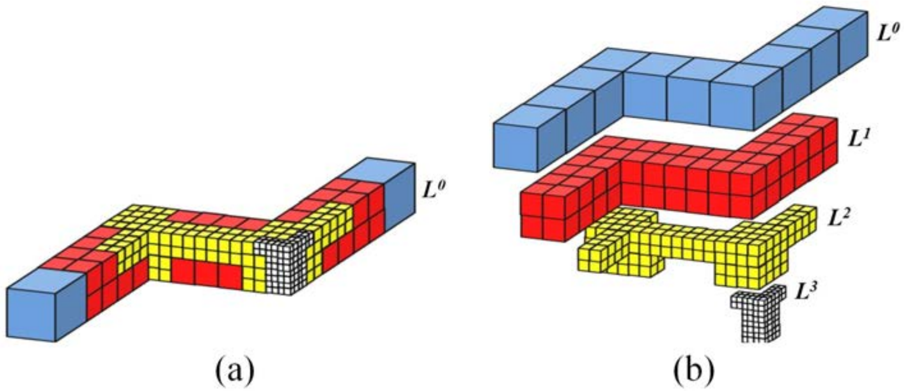


Figure 32: (a) Fully unstructured and (b) hierarchical grid structure.

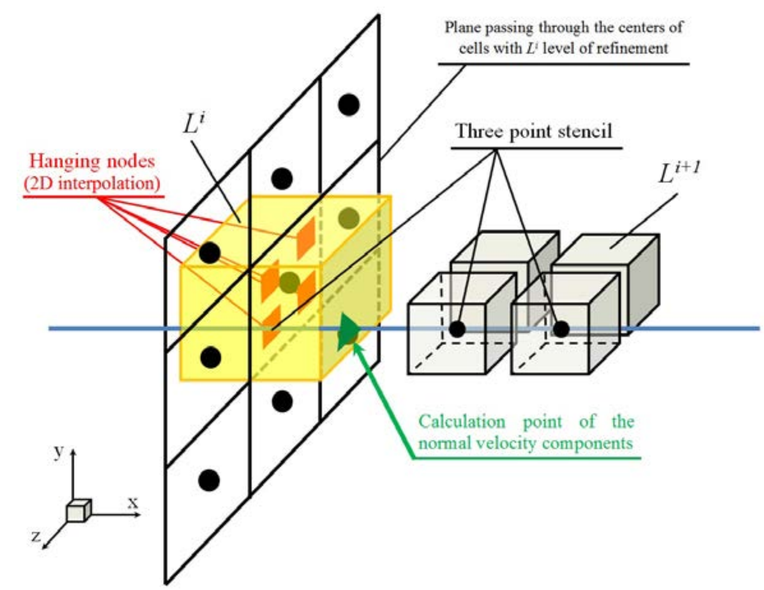


Figure 33: Three point stencil consisting of collinear cell centered and hanging nodes.

collinear three point stencil is constructed; enabling high order calculations in the middle of the interface shared between cells with different level of refinement. Note that the above described technique of locating hanging nodes wherever the cell centers of surrounding cells are not collinear with the reference one, allows the implementation of arbitrary sized stencils.

We select the Lagrange interpolation formula because it ensures that the interpolated function goes through the given data set and the accuracy of the calculations is determined by the size of the computational stencil. Two dimensional reconstruction is performed to evaluate the contribution of the surrounding coplanar values to the hanging nodes and one dimensional reconstruction is employed to calculate a variable in the middle of the shared interface between cells with varying levels of refinement. The number and location of the points of the 1D stencil determine the stability and accuracy of the discretization scheme. For example a four-point stencil may be used or an upwind calculation can also be performed.

Provided that the Lagrange basis polynomials can be described as:

$$L_i(\omega) = \prod_{s=0,s\neq 1}^{n_\omega} \frac{\omega - \omega_s}{\omega_i - \omega_s}, 0 \leq i \leq n_\omega \tag{53}$$

where ω can be any Cartesian coordinate ξ, η and $(n_\omega + 1)$ the number of data points, the 1-D interpolation can be defined as:

$$\phi(\xi) = \sum_{i=0}^{n_\xi} f(\xi_i) L_i(\xi) \tag{54}$$

and the 2-D interpolation will be:

$$\phi(\xi, \eta) = \sum_{i=0}^{n_\xi} \sum_{j=0}^{n_\eta} f(\xi_i, \eta_j) L_i(\xi) L_j(\eta) \tag{55}$$

The above Lagrange interpolant polynomials interpolate $(n_\xi + 1)$ or $(n_\xi + 1)(n_\eta + 1)$ data points $f(\xi_i)$ or $f(\xi_i, \eta_j)$ respectively. In essence, the interpolated values derived from the Lagrangian basis polynomials constitute a correlation between weighting factors and the information from the data points. Those weighting factors have the same values on any isotropically refined uniform meshes, adding no additional cost to their calculation.

It has already been stated that the hanging nodes are used whenever a grid cell is surrounded by other cells with higher level of refinement. In such a case, a two-dimensional 3x3 patch forms the two dimensional reconstruction stencil (fig. 34). The

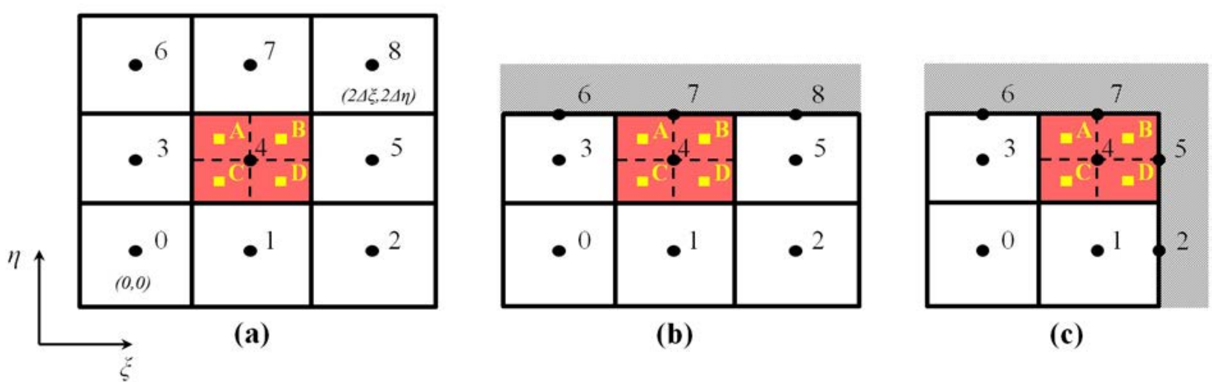


Figure 34: Reconstruction stencils utilized to calculate values at the auxiliary nodes A, B , C and D within a reference cell. (a) No walls around the cell; (b) Cell adjacent to one wall and (c) Corner cell. • Cell centers; □ auxiliary nodes

latter stencil consists of a 3x3 uniform subgrid ($3\Delta\xi$, $3\Delta\eta$) with $\Delta\xi$, $\Delta\eta$ being the dimensions of the current coarse cell along the plane were reconstruction is applied. Obviously, the 3x3 values of an arbitrary variable ϕ of the stencil coincide with the cell centered values of the computational grid if the surrounding cells have equal level of refinement with the reference cell. Otherwise, the nodes of the stencil may be located at the center of an isotropically split cell and an inverse distance function is utilized to calculate a value at the corresponding location of the stencil. The 3x3 2-D Lagrange interpolation, which practically results in a quadrilateral bi-quadratic interpolation, enables the calculation of the 9 weighting factors $L_i(\xi)L_j(\eta)$, $i, j = 0, 2$ even if some points of the data set belonging to the reconstruction stencil lie along the boundaries of the domain governed from Dirichlet conditions. In case of Neumann boundary conditions, the weighting factors are calculated by satisfying the gradient constraint on the polynomial variation.

4.4.4 Second order differencing operator

In this section we describe the differencing operators used on the unstructured grid. A central difference scheme can be used when neighbor cells have the same level of refinement. On the other hand, if the calculation point is surrounded by cells of varying levels of refinement, hanging nodes enable calculations along collinear data points that consist a three point stencil (fig. 35). Let's now assume that a dependent variable $\phi(\xi)$, may be represented from a function $f_i = f(\xi_i) = (.)_i$ which is continuous and n times differentiable in an interval $\xi \in [-h_{i-2}, h_{i+1}]$; the Taylor expansion around the point i for f_{i+1}, f_{i-1}, f_{i-2} would give:

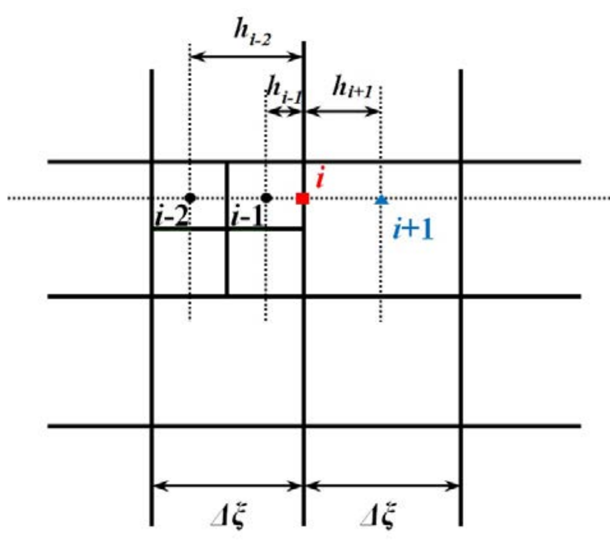


Figure 35: Second order derivation using a non-uniform three point stencil. • Cell centers; □ reference location △ hanging node.

$$f_{i+1} = f_i + h_{i+1} f_i^{'} + \frac{h_{i+1}^2}{2} f_i^{''} + \frac{h_{i+1}^3}{6} f_i^{'''} + \frac{h_{i+1}^4}{24} f_i^{IV} + ... + \frac{h_{i+1}^n}{n!} f_i^n \tag{56}$$

$$f_{i-1} = f_i - h_{i-1} f_i^{'} + \frac{h_{i-1}^2}{2} f_i^{''} - \frac{h_{i-1}^3}{6} f_i^{'''} + \frac{h_{i-1}^4}{24} f_i^{IV} + ... + \frac{h_i^n}{n!} f_i^n \tag{57}$$

$$f_{i-2} = f_i - h_{i-2} f_i^{'} + \frac{h_{i-2}^2}{2} f_i^{''} - \frac{h_{i-2}^3}{6} f_i^{'''} + \frac{h_{i-2}^4}{24} f_i^{IV} + ... + \frac{h_i^n}{n!} f_i^n \tag{58}$$

From (56)-(58), if we neglect the contribution of the terms $f^{'''}$ and higher, $f_i^{'} = \delta_\xi(.)_i$ can be evaluated from a three-point stencil having non uniform intervals with 2^{nd} order accuracy. By doing so, the differencing operator δ_ξ will be:

$$\delta_\xi(.)_i = \frac{c_0(.)_{i+1} + c_1(.)_{i-1} + c_2(.)_{i-2}}{\Delta\xi} + O(\Delta\xi^2) \tag{59}$$

with c_0,c_1,c_2=$g(h_{i+1}, h_{i-1}, h_{i-2})$ Apparently, the c_i, $i = 0, 2$ coefficients, which are function of the grid topology, obtain unique values on isotropically refined uniform meshes. The above differencing operator degenerates to a central differencing scheme with c_0=1, c_1=-1, c_2=0, whenever $|h_{i-1}| = |h_{i+1}|$, which is valid if the nodes $i+1$ and $i-1$ are at the center of cells with the same level of refinement.

4.4.5 The hybrid staggered/non-staggered approach on unstructured Cartesian grids

The idea of developing a grid arrangement which facilitates the divergence free constraint of the continuity equation and imposition of boundary conditions, avoiding at the same time pressure-velocity decoupling[86, 87] is extended to unstructured Cartesian meshes. The latter approach consists of discretizing the continuity and momentum equations on a hybrid staggered/non-staggered grid layout; using second order accurate finite-difference schemes. According to the present work, dependent variables of the velocity components which are normal to the faces are located in the middle of the surfaces cells' and the variables for pressure are stored at the cell centers. Whenever a cells is surrounded by other cells having different levels of refinement, the fluxes are located in the middle of the face of the smallest cells. The latter arrangement corresponds to an unstructured staggered layout; however, the velocity field is defined at the cell centers, as well, as in the collocated grids (fig. 36). By doing so, the hybrid formulation enables the satisfaction of both the continuity equation and the momentum equations on arbitrary refined cells.

The continuity equation can be satisfied by integrating it over a control volume of the unstructured grid. The discrete divergence operator over a given cell, k, in this case, will be:

$$\mathcal{D}_k(\mathbf{u}) = \frac{\sum\limits_{N=1}^{N_{D1}} u_{k,D1,N} S_{k,D1,N} - \sum\limits_{N=1}^{N_{U1}} u_{k,U1,N} S_{k,U1,N}}{\Delta x \sum\limits_{N=1}^{N_{D1}} S_{k,D1,N}} + \frac{\sum\limits_{N=1}^{N_{D2}} v_{k,D2,N} S_{k,D2,N} - \sum\limits_{N=1}^{N_{U2}} v_{k,U2,N} S_{k,U2,N}}{\Delta y \sum\limits_{N=1}^{N_{D2}} S_{k,D2,N}} +$$

$$+ \frac{\sum\limits_{N=1}^{N_{D3}} w_{k,D3,N} S_{k,D3,N} - \sum\limits_{N=1}^{N_{U3}} w_{k,U3,N} S_{k,U3,N}}{\Delta z \sum\limits_{N=1}^{N_{D3}} S_{k,D3,N}} = 0 \tag{60}$$

where $\mathbf{u} = (u, v, w) = ({}^1u, {}^2u, {}^3u)$ are the Cartesian velocity components, Dj and Uj $(j = 1, 2, 3)$ indicate downwind and upwind surfaces and N_{Uj}, N_{Dj} the total number of common surfaces between adjacent cells. Given that whenever adjacent cells have a different level of refinement the velocity component is located in the middle of the smallest cells' common interface, one cell may be governed from 6 to 24 normal velocity components, depending on level of refinement of the surrounding cells. It is a natural choice that reconstruction of the velocity field from the cell-face components to the cells centers (k) is implemented by averaging the corresponding volume fluxes and dividing by the normal surface area, as follows:

$${}^ju_k = \frac{\sum\limits_{N=1}^{N_{Dj}} {}^ju_{k,Dj,N} S_{k,Dj,N} + \sum\limits_{N=1}^{N_{Uj}} {}^ju_{k,Uj,N} S_{k,Uj,N}}{2 \sum\limits_{N=1}^{N_{Uj}} S_{k,Uj,N}} \tag{61}$$

After calculating the Cartesian velocity components at the cell centers, discretization schemes can be implemented in the same manner as in collocated grids.

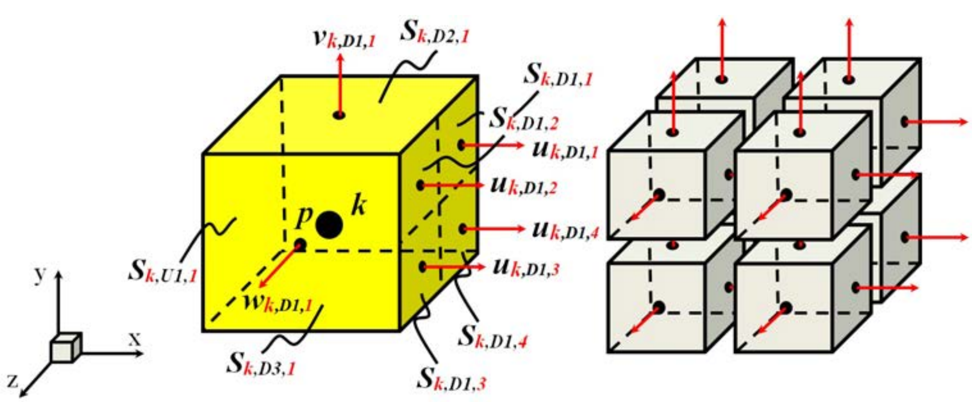


Figure 36: Staggered grid layout extended on refined Cartesian grids.

The momentum equations can be formulated in discrete operator form as follows (52):

$$\frac{\partial u_j}{\partial t} = -\mathcal{A}(u_j) - \mathcal{G}_j(p) \tag{62}$$

where $\mathcal{G}$ is the pressure gradient operator and $\mathcal{A}$ is a summation of the convective $\mathcal{C}$ and viscous $\mathcal{V}$ operators:

$$\mathcal{A}(.) = \mathcal{C}(.) - \frac{1}{Re}\mathcal{V}(.) \tag{63}$$

which are defined as:

$$\mathcal{C}(.) = \frac{\partial}{\partial x_i}(u_i.) \tag{64}$$

$$\mathcal{V}(.) = \frac{\partial}{\partial x_i}\left(\frac{\partial}{\partial x_i}.\right) \tag{65}$$

Discretizing the momentum equation in the middle of the shared inteface between cells with different levels of refinement (Fig. 36), m, we obtain the following semi-discrete approximation:

$$\left.\frac{\partial u_j}{\partial t}\right|_{k,D_j,m} = -\mathcal{A}(u_j)\Big|_{k,D_j,m} - \mathcal{G}_j(p)\Big|_{k,D_j,m} \tag{66}$$

The pressure gradient term on unstructured Cartesian meshes is calculated by using the differencing operator δ_ξ, defined in (59), as follows:

$$\mathcal{G}_j(p)\Big|_{k,D_j,m} = \delta_{x_j}(p) \tag{67}$$

Suppose that the convective terms $\mathcal{A}$ have been calculated at the cells centers and that $\tilde{\mathcal{A}}$ includes the above terms in a space of collinear data points with values at the cell centers or hanging nodes, as described above, we can obtain:

$$\mathcal{A}(u_j)\Big|_{k,D_j,m} = \sum_{i=0}^{N_{st}}[\tilde{\mathcal{A}}(u_j)]_i L_i(x_j) \tag{68}$$

which is in fact one dimensional interpolation of the convection and viscous terms from a stencil with N_{st} data points. The above calculation involves a two point stencil, unless the adjacent cells have varying levels of refinement; in such a case $N_{st}\geq 3$. The discrete continuity and momentum equations are integrated in time using a fractional step method. The intermediate velocity components at the surface centers of the unstructured mesh, which do not satisfy the continuity equation, are obtained by solving the momentum equation using a second-order backward scheme in time. The projection step is then applied to satisfy the continuity equation and the numerical algorithm can converge to machine zero, since consistent numerical operators are used for the discretization of the Poisson equation and the projection calculations.

4.4.6 Time integration method

The fractional step method is here utilized to integrate the governing equations in time, on unstructured Cartesian grids. The momentum equations are solved implicitly by employing a second order temporal backward Euler differencing scheme:

$$\frac{3\mathbf{u}^{*} - 4\mathbf{u}^{n} + \mathbf{u}^{n-1}}{2\Delta t} = -\frac{\nabla p^{n}}{\rho} - \mathbf{u}^{*}\nabla\mathbf{u}^{*} + \frac{1}{Re}\nabla^{2}\mathbf{u}^{*} \tag{69}$$

where n denotes the time step and $\mathbf{u}^{*}$ represents the intermediate staggered velocity field, which does not satisfy the continuity equation. The right hand side of the above equation is discretized in space according to the hybrid staggered/non-staggered approach described above. The pressure correction step is employed to enforce satisfaction of the continuity equation at the time step $n{+}1$:

$$-\frac{3\rho(\mathbf{u}^{n+1} - \mathbf{u}^{*})}{2\Delta t} = \nabla\Pi^{n} \tag{70}$$

$$\nabla\mathbf{u}^{n+1} = 0 \tag{71}$$

and the pressure correction Π:

$$\Pi = p^{n+1} - p^{n} \tag{72}$$

can be calculated after solving the Poisson equation resulted by the combination of the Eqs. (70) and (71):

$$\nabla^{2}\Pi = \frac{3\rho\nabla\mathbf{u}^{*}}{2\Delta t} \tag{73}$$

The velocity field that satisfies the continuity equation is then obtained as follows:

$$\mathbf{u}^{n+1} = \mathbf{u}^{*} - \frac{2\Delta t\nabla\Pi}{3\rho} \tag{74}$$

The Jacobian-free Newton-Krylov method is employed to solve the momentum equation in a fully implicit manner. The discretization of pressure (Eq. 67) following the strong coupling approach, leads in the generation of a non-symmetric sparse matrix. For that reason, the Algebraic Multi-Grid (AMG) serves as the preconditioner to accelerate the convergence rate of the Newton-Krylov solver, when solving the Poisson equation (Eq. 73).

4.5 Results

4.5.1 Driven cavity flows

The typical lid-driven cavity problem consists a classical benchmark problem which has been extensively studied over the last decades and used to validate the accuracy of incompressible flow solvers. It has attracted the interest of several works since it is governed by a simple geometry while complex flow patterns are found. Three decades ago Koseff and Street[88] studied experimentally three-dimensional cavity flows with Spanwise Aspect Ratios (SAR) ratios varying from SAR=1 to SAR=3 and for Re ranging from 1000 to 10.000; while in many previous works the same problem for SAR=1 was numerically studied [89, 90, 91, 3]. Anagnostopoulos [65] solved the 2D lid-driven cavity problem in order to demonstrate the robustness of an unstructured 2D solver on arbitrarily refined refined grids.

The robustness of the 3D unstructured flow solver is demonstrated by solving the laminar lid-driven cubical flow with SAR=1 and Re=1000. Three different grids were considered for the particular problem, a coarse uniform grid with 25^3 cells, an adaptively refined grid having as a primary grid the coarse and a fine uniform grid with 150^3 cells. Concerning the grid adaptation procedure, the flow field was initially solved on the coarse grid and the rotationality criterion[92, 65] τ_R was used to flag cells for refinement. The latter sensor is essentially the vorticity magnitude weighed by the cells' length scale; thus promoting large cells for further refinement:

$$\tau_R = |\nabla \times \vec{u}| h_r^{\frac{3}{2}} \tag{75}$$

with

$$h_r = \sqrt[3]{Vol} \tag{76}$$

The refinement of the 25^3 coarse grid resulted to the unstructured grid presented in fig. 37. It can be observed that high levels of refinement appear close to the walls and essentially along the zones of high shear layer. In fig. 38 the U and V velocity profiles along the vertical and horizontal centerlines of the symmetry planes are compared with the calculations of Jiang *et al*. [89] It is shown that the numerical results of the uniform fine grid compare well with the results of Jiang *et al*., revealing that the new flow solver can provide convenient results without refinement. More importantly, the adaptively refined grid provide much more accurate solution compared with the one derived from the primary structured coarse grid; which was insufficient to capture the correct solution. High spatial resolution along the regions of high vorticity allowed the correct prediction of the strength of the primary vortex. Fig. 39 illustrates the streamlines on the midplanes x-y and y-z respectively. The secondary vortices appear in the two lower corners on the symmetry plane with z=0.5. A pair of contra-rotating vortices are located close to the upper corners and the bottom wall of the cavity respectively, on the symmetry plane with x=0.5.

A second test case related to the lid-driven cavity flow for SAR=3 and Re=3200, is also investigated. For this problem two grids were used, a coarse uniform 20x20x60 and a locally refined by applying 2 levels only close to the cavity walls. The grid refinement zone had a width which was equal to the 20% of the height of the cavity. Fig. 40 presents the normalized mean U and V velocity profiles along the vertical and horizontal centerlines of the symmetry plane, respectively, and the results are

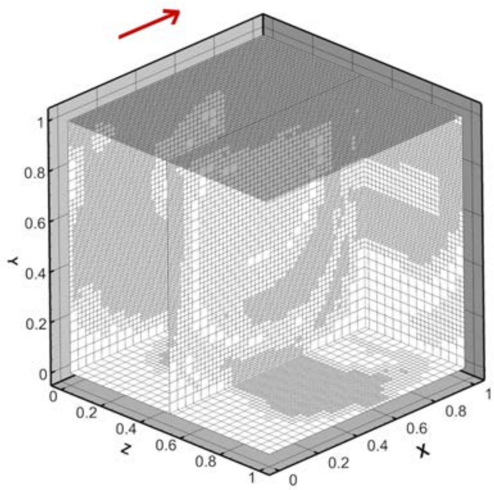


Figure 37: 2-Level adaptively refined grid based on a rotationality criterion.

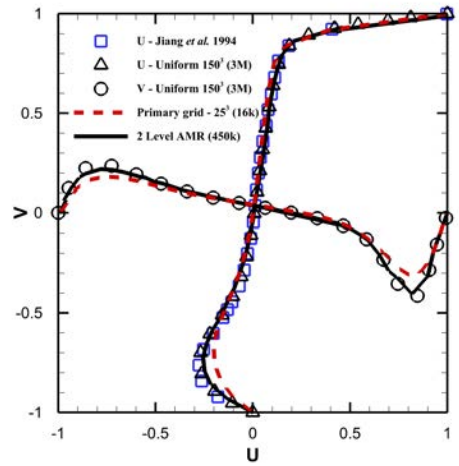


Figure 38: Normalized U and V velocity profiles along the vertical and horizontal centerlines of the symmetry plane, respectively; *Re*=1000, SAR=1.

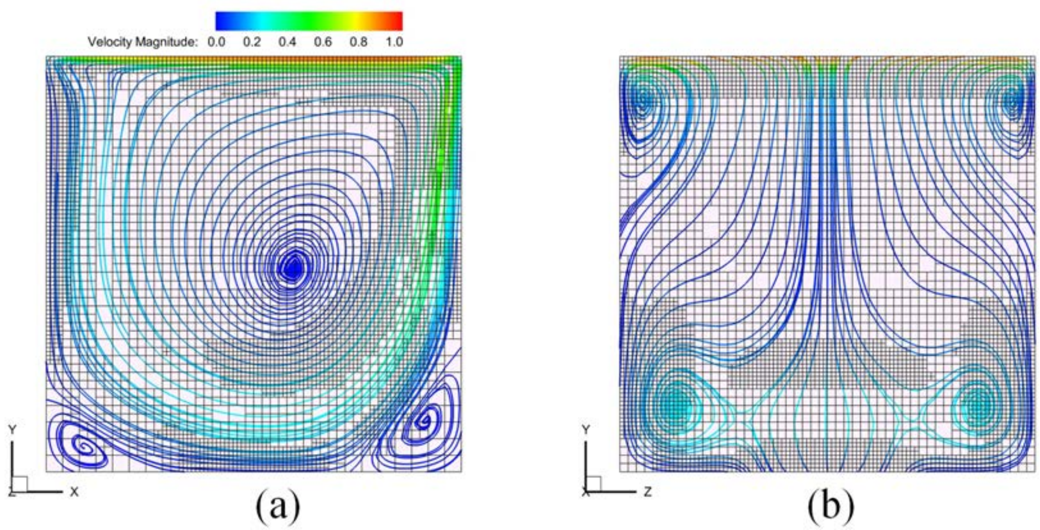


Figure 39: Streamlines on the midplanes (a)x-y; (b)y-z for *Re*=1000.

compared with the experimental measurements of Koseff and Street [88]. The flow field at this *Re* is unsteady and the Taylor-Görtler (TG) vortex pairs are found at the bottom wall of the cavity. It is evident that the coarse uniform grid was not sufficient to predict the pair of contra-rotating vortices close to the walls. On the other hand the 2-level locally refined grid seems to be sufficient to resolve the TG vortices formation and the steep gradients at the end-wall plane.

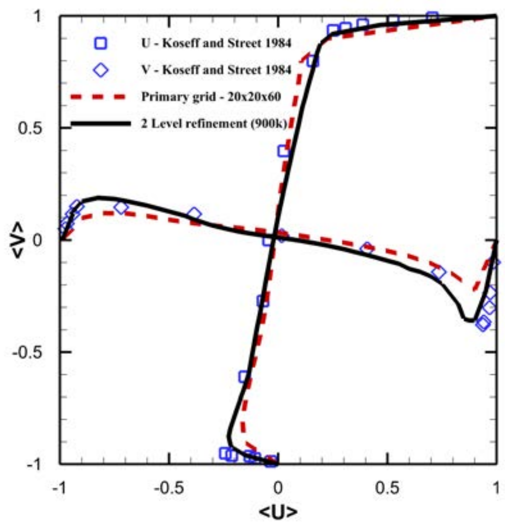


Figure 40: Normalized mean U and V velocity profiles along the vertical and horizontal centerlines of the symmetry plane, respectively; *Re*=3200, SAR=3.

4.5.2 Wind turbine simulations

The new LES unstructured flow solver is used to simulate the flow over a stand-along wind turbine. In fact, the numerical simulations correspond to the experimental work performed by Chamorro and Pórte-Agel[93]. The latter work investigated the flow field over a miniature turbine consisted of a three-blade GWS/EP-6030x3 rotor attached to a small DC generator. The diameter of the turbine is D=0.15m and the hub height is $h/D = 5/6$. In this work, the effect of the nacelle is simplified by extending the blade geometry to the center of the rotor. The tip speed ratio is λ=4.1 and the Reynolds number of the problem is Re=42.000, based on the incoming velocity at the hub height W_h and with characteristic length equal to the rotor diameter D. The same problem was simulated by Wu and Pórte-Agel [94] who treated the blades as a flat plate and Yang *et al.* [84] who represented the blades by a NACA0012; deriving data for the data and lift coefficients from Sheldahl and Klimas [95]. Here we followed the latter more realistic approach and the radial distribution of the pitch angle and chord length of the blades are shown in Table 3.

The turbulent flow field over a stand-alone wind turbine located 3D from the inlet, is simulated by discretizing the governing equations on a computational domain sized 20D, 5D and 3D in the streamwise (z) spanwise (x) and vertical (y) directions, respectively. Two simulations were carried out; the first was performed on a uniform structured grid (G1) of 135x50x30 ($\sim$200k) cells and the second (G2) which was generated after remeshing the first coarse mesh in an ellipsoidal manner around the turbine by applying 2 levels (fig. 41); the refined region extends from 2D upstream to 14D downstream the turbine. By doing so, the refined grid consisted of 1.7M cells and it is worth mentioning that the equivalent uniform structured grid having grid spacing equal to the finest spacing achieved with the G2 grid would consist of 13M cells. The above numerical setup allowed distribution of the forces from the actuator lines to the background mesh along 10 and 40 gridnodes for grids the G1 and G2 respectively. The pre-computed turbulent flow field generated from a fully developed channel flow is imposed as boundary condition at the inlet, the boundary layer at the bottom is modelled by utilizing a wall mode [85] and free slip condition is applied at the spanwise walls and at the top of the vertical direction. The time step for both grids is $0.001D/W_h$; thus the same inflow can be used in both cases and any comparison between the two simulations can be straightforward.

Figures 42-44 present the distribution of time-averaged values at different locations

	Radius (cm)							
	1.0	2.0	3.0	4.0	5.0	6.0	7.0	7.5
Pitch (^{0})	20.50	20.90	19.83	16.91	13.19	10.67	9.12	6.66
Chord (cm)	1.39	1.47	1.47	1.41	1.31	1.18	1.02	0.60

Table 3: Pitch angle and chord length as a function of the radial variation for the miniature turbine[94].

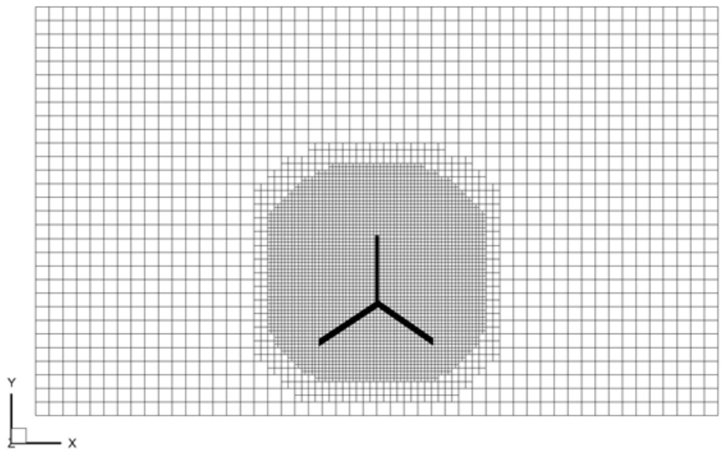


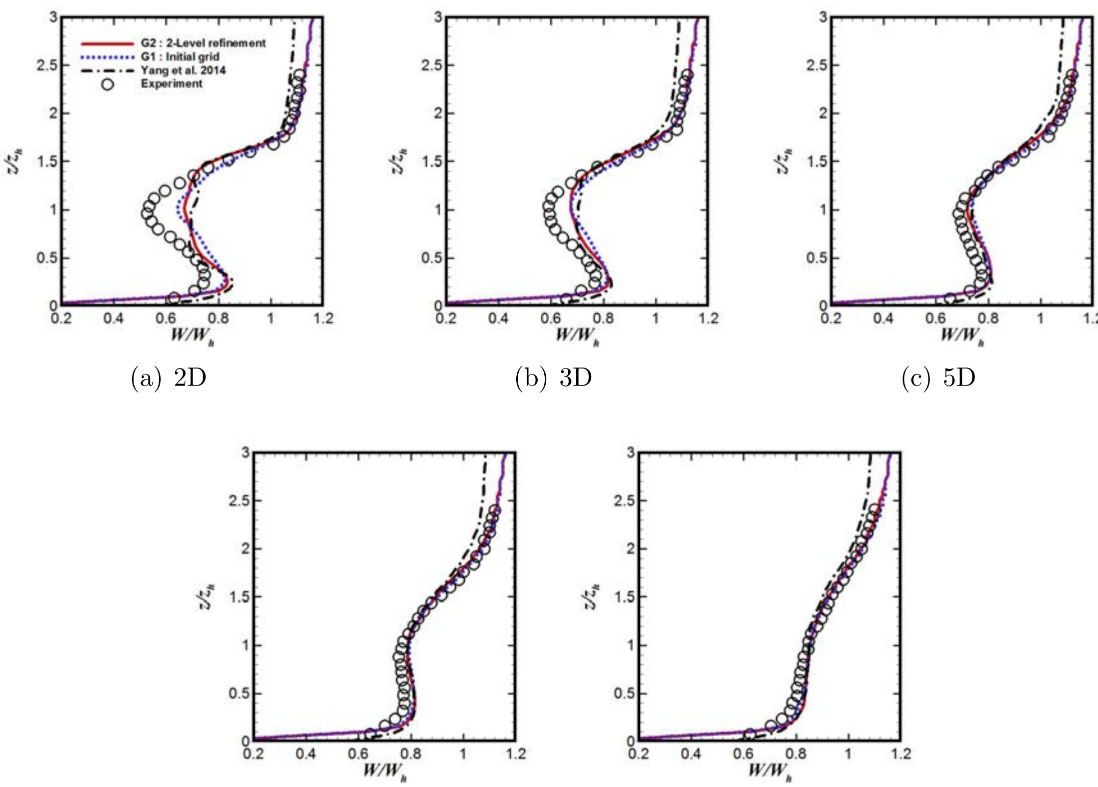
Figure 41: Spanwise view of the 2-level locally refined grid around the wind turbine.

downwind the turbine and the numerical simulations are compared against experimental measurements [93] and previous simulations [84]. Yang *et al.* solved the same problem by using a 480x240x120 stretched grid along a 30Dx12Dx3D domain resulting in grid spacing around the wind turbine which is comparable to the discretization obtained with the G2 grid. Figure 42 displays the distribution of the time-averaged streamwise velocity at different locations downstream the turbine. The numerical calculations are in very good agreement with the experimental measurements, mainly in the far-wake regions with $z/z_h > 5D$. The velocity overprediction in the near wake region at the hub height, is attributed to the fact that both the nacelle and the tower are not modelled, while the actuator line parametrization is insufficient for accurate near wake calculations with $z/z_h < 2D$. Overall a really good prediction of the distribution of the streamwise turbulence intensity is achieved on the refined grid (fig. 43). It is evident that applying 2 levels of refinement on the G1-initial structured grid results in improvement of the LES calculations. Essentially, the overprediction of the turbulent intensity at the top tip of the turbine further downstream at $z/z_h > 7D$ is eliminated and the overall trend of the distribution is in better agreement with the measurements. The calculations on the unstructured grid compare well with the simulations of Yang *et al.*; however the numerical calculations coupled with ALM parametrization in any case fail to provide reliable results at the near wake region and at $z/z_h < 2D$. Concerning the distribution of the Reynolds shear stresses ($- < w'v' >$) (fig. 44) good agreement is found in all the simulations and particularly further downstream the wind turbine, while further improvement can be achieved by extending the spanwise length of the computational domain.

Further physical insight of the turbine wake can be obtained by plotting the streamwise velocity, spanwise velocity, streamwise turbulence intensity and kinematic shear stress contours, as shown in figure 45. The velocity deficit at the near wake is related to the extracted energy of the turbine; an effect which influences the flow field even in the far wake. In line with the results from Yang *et al*. the turbine rotation induces significant spanwise velocity (fig. 45(b)). Large shear stresses are generated from the rotation of the turbine. In fact two zones of high shear stresses originate from near the top and bottom tips of the turbine and extend almost 10D downstream the turbine. The strong shear is found in the experimental measurements as well [94] while the two zones with lower stresses at the hub height within 2D downstream the wind turbine are generated because the nacelle is not modelled and instead the blade geometry is extended to the center of the rotor (fig. 45(d)). Also, high levels of streamwise turbulence intensity are found at the top tip level even 10D downstream the turbine. The rich dynamics resolved with the unstructured grid are shown in figure 46 where the instantaneous streamwise velocity contours are displayed.

The flow dynamics over the same wind turbine are also investigated under uniform inlet conditions to better illustrate the ability of our solver to simulate the helical tip vortices and their break up downwind of the rotor. In that case, the wind turbine is located 3D downstream the inlet and in the middle of a 6Dx6Dx8D domain. An initially uniform structured grid 120x120x160 was refined around the turbine resulting in a resolution of 100 gridpoints per diameter. In fact, 2 level circular refinement was applied on the primary grid with radii of R_1=0.8D, R_2=D for the first and second level, respectively 47. The refinement zone extended from 1D upstream to 4D downstream the turbine resulting in an unstructured grid of 7M cells. The equivalent structured grid achieving the same grid resolution around the wind turbine would require 147M cells; 20 times the number of the cells of the unstructured mesh.

Figure 47 shows the isosurfaces of vorticity magnitude coloured with the velocity magnitude, on the wake of the turbine. The rotation of the blades clearly results in the formation of the tip-vortex ring which is convected downstream. Those coherent structures consist three individual spirals rotating in phase with the turbine. The deceleration of the axial velocity on the far wake excites the vortex breakdown and instability of the flow around 3D downstream the rotor. The tip helical vortices are accompanied with contra-rotating root vortices (ω_Y); thus satisfying conservation of vorticity.



(a) 2D (b) 3D (c) 5D

(d) 7D (e) 10D

Figure 42: Distribution of the time-averaged streamwise velocity at different locations downstream the wind turbine.

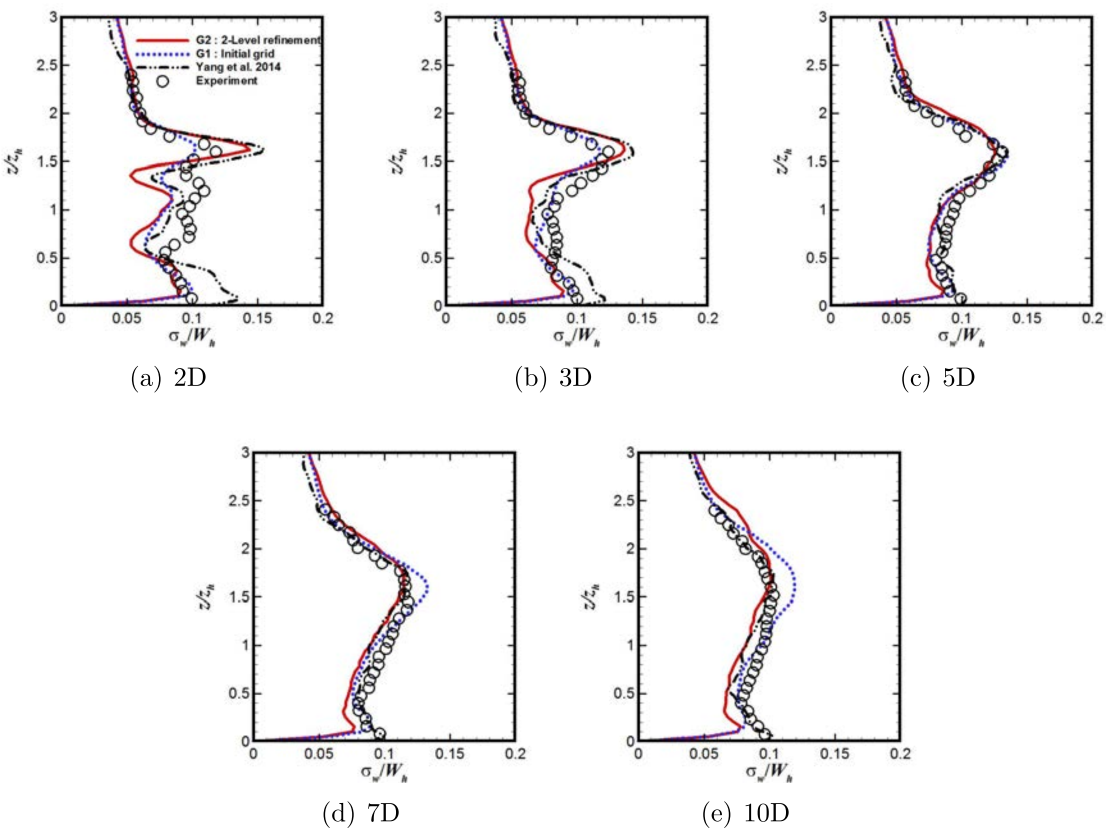
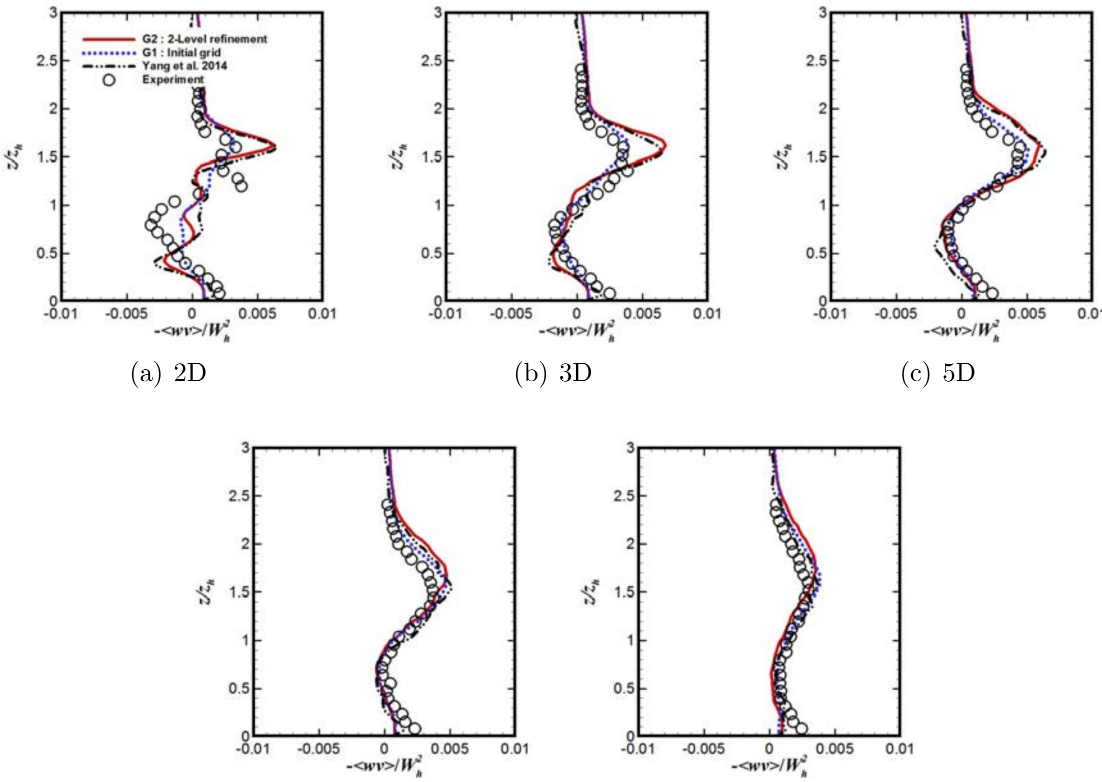


Figure 43: Distribution of the streamwise turbulence intensity at different locations downstream the wind turbine.



(a) 2D

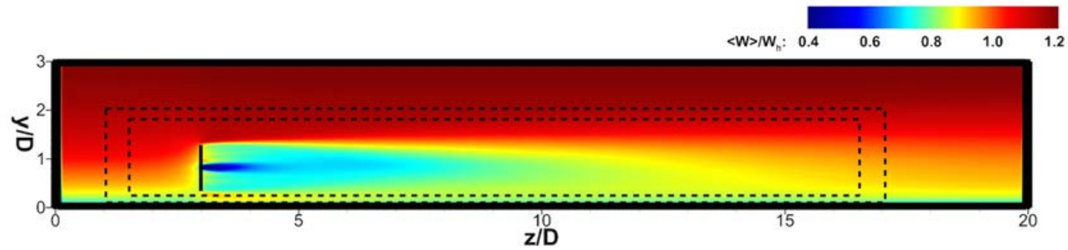
(b) 3D

(c) 5D

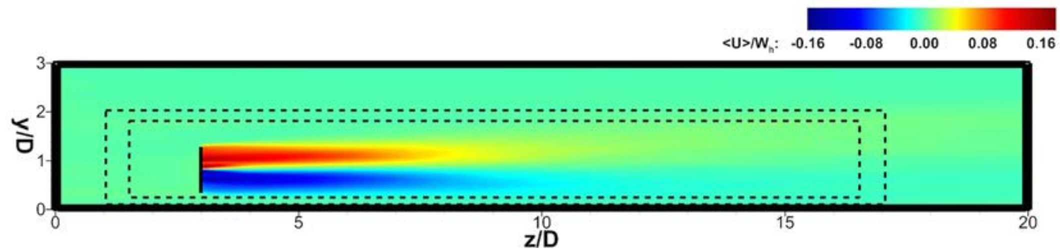
(d) 7D

(e) 10D

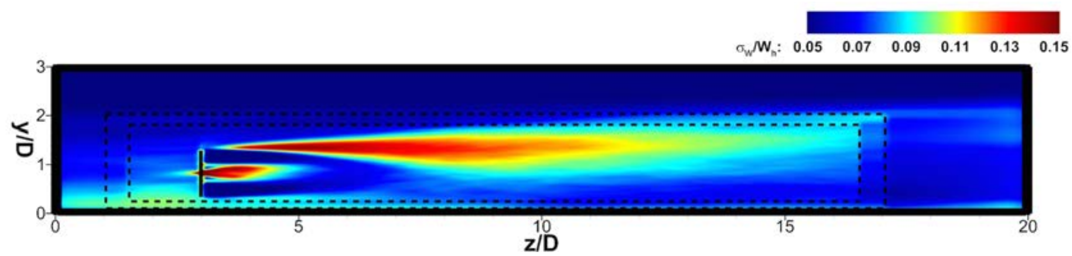
Figure 44: Distribution of the Reynolds shear stresses at different locations downstream the wind turbine.



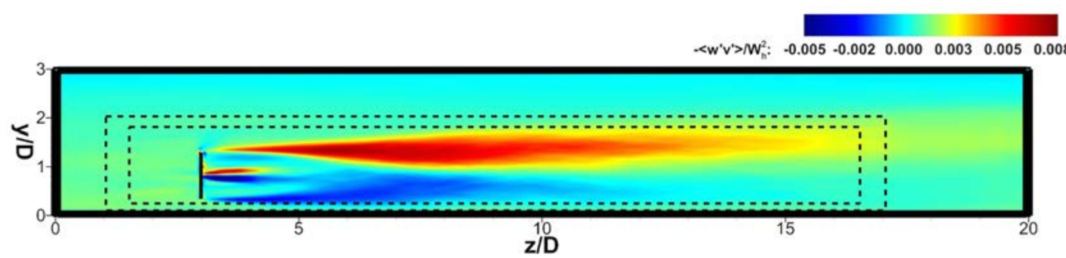
(a)



(b)



(c)



(d)

Figure 45: Contours of the time averaged normalized values of: (a) streamwise velocity; (b) spanwise velocity; (c) streamwise turbulence intensity and (d) kinematic shear stress, in the middle vertical plane perpendicular to the turbine. The regions of refinement are marked with dashed lines.

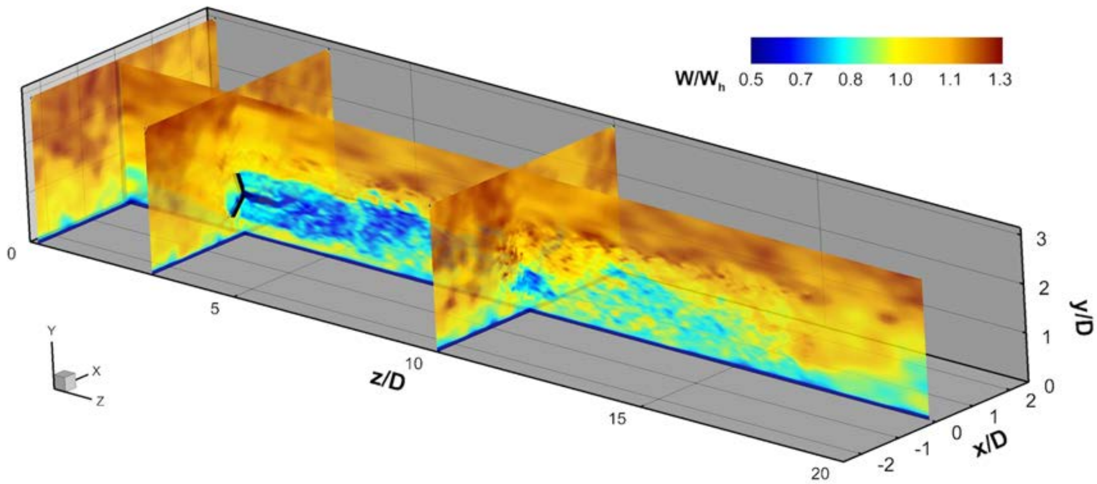


Figure 46: Instantaneous streamwise velocity field calculated on the refined grid.

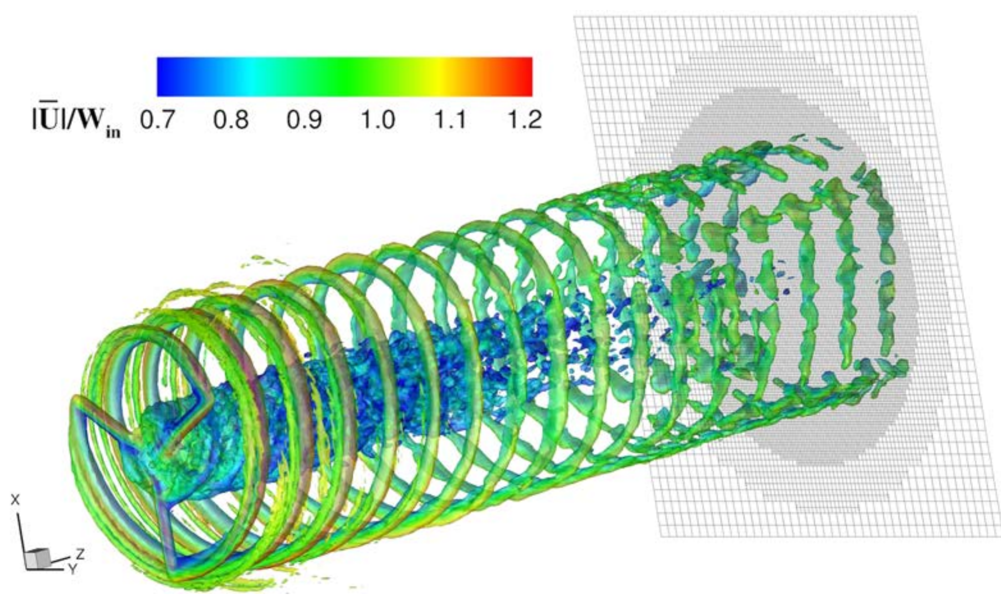


Figure 47: Normalized isosurfaces of vorticity magnitude with value equal to 7.5 coloured with the velocity magnitude, induced in uniform-idealized inflow.

4.6 Conclusions

We have developed a hybrid immersed boundary/adaptive mesh refinement (IB-AMR) method for 3D unsteady incompressible flows within and around complex topographies. Adaptive mesh refinement in the vicinity of three-dimensional boundaries of arbitrary complexity improves the numerical accuracy with low computational cost. The benefit of extending the idea of hybrid staggered/non-staggered grid layout on unstructured meshes is twofold: All the benefits of the staggered arrangement are retained, including the inherent satisfaction of the divergence-free constraint, and the implementation of boundary conditions, which may or may not conform with the physical grid's boundaries, is greatly facilitated. Additionally, flux mismatch or pressure discontinuity issues across fine/coarse interfaces are overcome since no prolongation/restriction operators are required. The fully unstructured approach makes the parallel domain decomposition procedure straightforward and load balance can be achieved after calling a conventional repartitioning function. The discrete momentum equation is solved with the matrix-free Krylov method and the algebraic multigrid method (AMG) served as a preconditioner for the Krylov-subspace method employed to solve the Poisson equation.

The discretization method we propose is second order accurate and yields accurate solutions for benchmark problems such as the flow in shear-driven cavities. The potential of the current approach is also demonstrated by discretizing the governing equations on a grid adapted around a wind turbine parameterized using the actuator line approach. LES on locally refined grids enabled accurate and low computational cost calculations, providing results in good agreement with the experimental measurements. The Actuator Line Model performs well starting at 3D downstream the turbine, where both the mean flow and the turbulence quantities are correctly captured. High levels of streamwise turbulent intensity are found to persist up to 7D downstream at the top tip of the turbine, while high levels of the Reynolds shear stresses exist both at the top and bottom tips around the same location. Lastly, the current numerical setup allowed the calculation of the root and tip helical vortices under uniform inflow.

4.7 Publications and Presentations

4.7.1 Publications

Angelidis, D.; Sotiropoulos, F. Sharp interface immersed boundary method with adaptive mesh refinement for 3D unsteady incompressible flows. Ready for submission in the Journal of Computational Physics

Angelidis, D.; Sotiropoulos, F. Simulation of wind turbine wakes on locally refined Cartesian grid. Proceedings of the AIAA SciTech, 33rd Wind Energy Symposium. 2015.

4.7.2 Presentations

Angelidis, D.; Sotiropoulos, F. Simulation of wind turbine wakes on locally refined Cartesian grid. 33rd Wind Energy Symposium. 2015.

Angelidis, D.; Sotiropoulos, F. Large-eddy simulation of wind turbine wake interactions on locally refined Cartesian grids. 67th Annual Meeting of the APS Division of Fluid Dynamics 59 (20). 2014.

Angelidis, D.; Sotiropoulos, F. Parallel Cartesian grid refinement for 3D complex flow simulations. 66th Annual Meeting of the APS Division of Fluid Dynamics, November 24-26, Pittsburh, 2013.

Angelidis, D.; Sotiropoulos, F. A local grid refinement curvilinear immersed boundary method for multi- resolution simulations of complex turbulent flows. 65th Annual Fall DFD Meeting, November 18-20, San Diego, California. 2012.

5 Task 2 Massively parallel implementation of the computational models

5.1 Goals and Objectives

A major objective of the proposed research is to develop computational tools that enables the industry to take full advantage of high-performance computing to revolutionize the design of efficient and reliable offshore turbines and optimize turbine arrays on a site specific basis. We seek investigate code-development strategies for exhibiting strong scaling on massively parallel platforms. These strategies include: I/O optimization and parallel I/O implementation; algorithm restructuring for data locality and cache reuse; scaling and use of PETSc toolkit (which is used in the computational framework); and performance optimization on heterogeneous systems.

5.2 Project team

Fotis Sotiropoulos James L. Record Professor, Director of the St. Anthony Falls Laboratory. Principal investigator of the project.

Aaron Boomsma Dr. Boomsma was a graduate student at the Saint Anthony Falls Laboratory and obtained his PhD. He Worked in improving the efficiency of the code in massive parallel implementations.

5.3 Technical Approach and Results

Using the RedSky/RedMesa supercomputers at Sandia National Labs, we have been able to simulate the fluid dynamics at the SWiFT site with very high fidelity. To achieve this, VFS was first ported to the Red Sky cluster, then initial simulations of the site, along with a grid refinement study, was performed. In order to efficiently utilize Sandia's computational resources, UMN researchers have been focusing on improving VFS scalability. First, a dedicated application testing time was acquired on RedMesa, where a strong scaling of approximately $20-30\%$ was achieved. Then, using this initial test and collaboration with Sandia, the Poisson solver preconditioner was modified, which increased the strong scaling to about $40-50\%$.

CURVIB is coded with the C language and utilizes PETSc libraries [81] for data structure logistics, matrix/vector operations, and momentum equation solves. The BoomerAMG library [96], which is linked within PETSc, is used to condition and solve the Poisson equation. CURVIB has been developed for distributed memory systems and uses openMPI for communication among supercomputer nodes and cores.

As problem sizes increase, it is appealing to utilize very large (tens-of-thousands of cores) compute-clusters. With these systems, application efficiency (especially on systems with hundreds-of-thousands of cores) is of great importance. To evaluate the efficiency and performance of CURVIB, we were granted access to Sandia National

Labs RedMesa cluster. RedMesa is currently rated 114^{th} on the worlds fastest supercomputers. It has approximately 24,000 Intel X4470 (2.93Ghz) cores connected via Infiniband. Each node has eight cores with 12Gb of shared memory.

To evaluate the efficiency of CURVIB, two parameters were assessed: strong and weak scaling. They are defined in Equations (77) and (78), respectively:

$$S(n) = \left.\frac{t_{min}}{t_n}\right|_{fixed} \tag{77}$$

$$W(n) = \left.\frac{t_{min}}{t_n}\right|_{variable} \tag{78}$$

where t_{min} refers to the solution time with the minimum required cores to run the simulation and t_n refers to the solution time with n cores. The true definitions of strong and weak scaling do not use t_{min}. Instead they use t_{seq}, which refers to the solution time achieved with a single, sequential process. However, with such large problem sizes, it would likely be impossible to even achieve a solution with one process (due to lack of memory), so instead, we use t_{min}. Each function is evaluated using a different method. Strong scaling is computed with the problem size remaining fixed. Weak scaling is computed by using a variable problem size.

If the user of an application is focused on acquiring solutions in less time (i.e., simulation turnover), strong scaling is most important. To compute $S(n)$, the problem size (say, number of grid nodes) is held constant, while the number of cores, n, is increased. $S(n)$ is an evaluation of the applications ability to decrease computation time with increasing resources. Ideally, $S(n)$ scales linearly with n. If the user is focused on solving larger or more problems, then weak scaling is most important. When computing $W(n)$, the problem size per core is held constant, while n is increased. This parameter is an evaluation of the applications ability to distribute work and communicate effectively. Ideally, $W(n) = 1$.

For the scaling tests, a turbulent open channel flow was simulated. The timestep was held constant among tests, as was Reynolds number, grid spacing, and bulk velocities. To increase problem sizes, the length of the domain width (spanwise direction) was increased. The grid was uniform in the streamwise and spanwise directions and clustered in the wallnormal direction to resolve the boundary layer. A fourth-order modified Runge-Kutta method was used to solve the momentum equations. Each case was a DNS and thus required no turbulence closure model. First, weak scaling results from RedMesa are shown in Figure 48.

These scaling results are quite typical and show good weak scaling. As the number of grid cells per node increases, the scaling improves. This is due to less MPI communication among nodes. It should be noted that good weak scaling is easier to achieve than good strong scaling. Strong scaling results from RedMesa are shown in Figure 49.

The different colored lines in Figure 49 correspond to certain grid cell counts (in millions). From this figure, it i seen that at about a speedup of three, a plateu occurs.

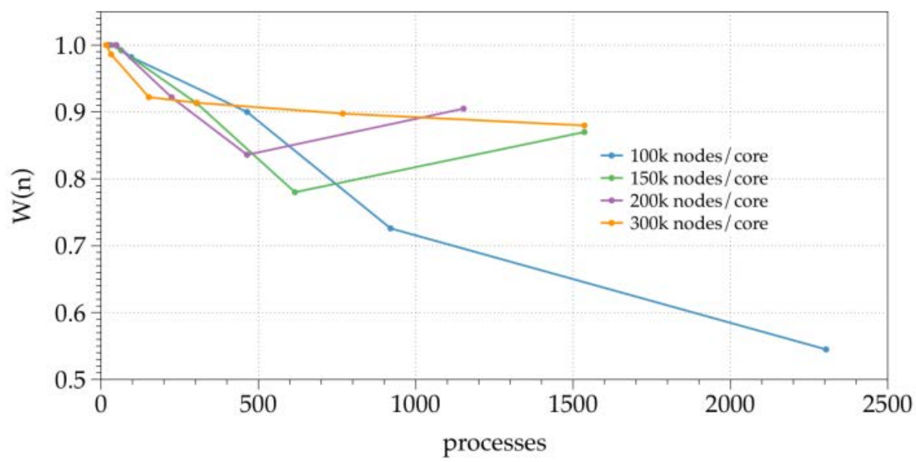


Figure 48: Weak scaling on RedMesa, 8 cores per node.

This translates to an efficiency, E of about $30 - 40\%$, where

$$E = S/n_{ratio} \tag{79}$$

and $n_{ratio} = n_{max}/n_{min}$. This result is reasonable, but should be improved. To explore possible avenues for increasing E, strong scaling results were calculated by only using half of the cores available per node. Results are illustrated in Figure 50.

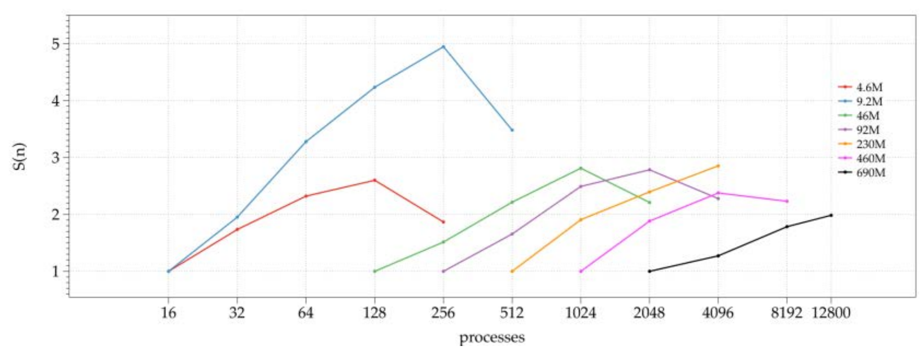


Figure 49: Strong scaling on RedMesa, 8 cores per node.

When running with only half the number of cores per node, there is a drastic difference in strong scaling. The reason for completing this test is to examine the effect of intra-node memory bandwidth. Strong scaling is much improved because the four cores now have double the memory bandwidth that eight running cores would have. This result conclusively reveals that CURVIB suffers from an intra-node memory bandwidth bottleneck. Unfortunately, this bottleneck is common and a challenge to address. One solution is to employ both MPI and openMP coding paradigms. For intra-node communication, openMP could be used and inter-node communication could be accomplished with MPI. This type of communication is the

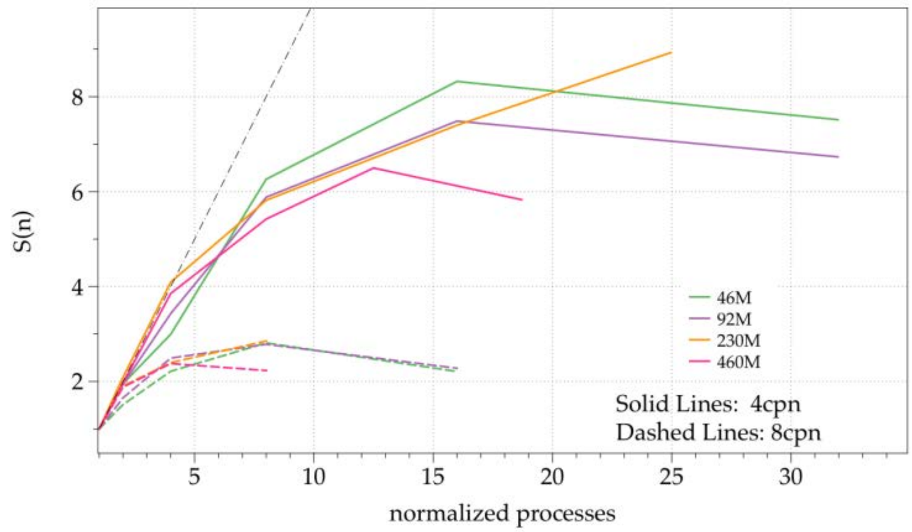


Figure 50: Strong scaling on RedMesa, 8 cores per node.

state-of-the-art in the field. Second, acceleration with a Graphics Processing Unit (GPU) could be of help. GPUs perform extremely well with embarrassingly parallel algorithms. Even though CURVIB is not so parallel, some sections likely are. Finally, it is important to note that the poor strong scaling results are also due to the Poisson equation. Testing different preconditioners can be of help, as shown in Figure 51.

Two different preconditioners are used for the strong scaling results in Figure 51: BoomerAMG method and PETScs AMG method. When using the algebraic multi-grid written in native PETSc, the efficiency is improved to $40 - 50\%$. Finally, another possible solution is to move away from using the PETSc libraries. Although time consuming, fine-tuning of the matrix operations and solver could be possible.

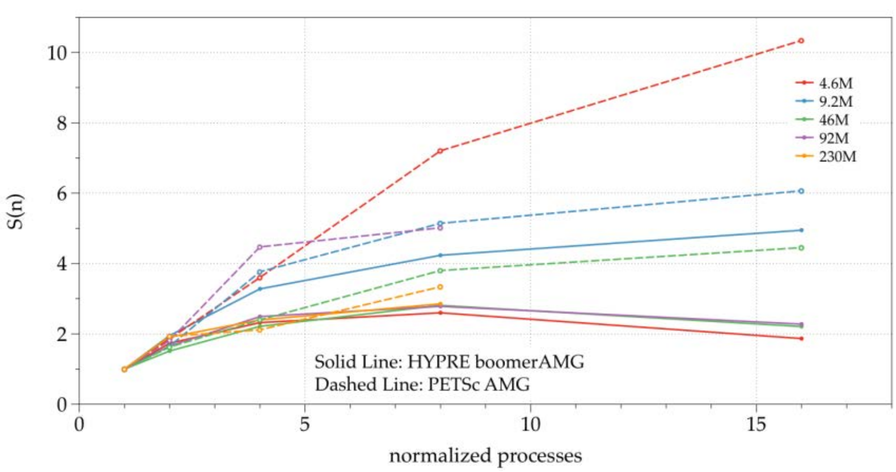


Figure 51: Strong scaling on RedMesa, 8 cores per node.

5.4 Publications

Aaron Boomsma. Drag Reduction by Riblets and Sharkskin Denticles: A Numerical Study. Thesis Disertation. July, 2015.

6 Task 3 Experiments

6.1 Background

There is a considerable body of research on land-based wind turbines that are relevant to this experimental setup and investigation. The works on the aerodynamic wake of land based wind turbines were reviewed by [97], while wind power estimates and wind turbine blade aerodynamics are addressed by [98]. Most of the experimental work on wind turbine wakes and turbine siting has been performed using miniature models deployed in a wind tunnel [99, 52, 100, 101, 102, 103, 104], which is an approach we will follow in this paper to study the effect of a moving rotor on a turbine wake. Flow measurements around utility scale turbine are only recently achieving high spatial resolution thanks to LiDAR measurements [105, 106, 107] or image-based techniques [108]. Numerical efforts provided valuable insight thanks to Large Eddy Simulations, implementation of immersed boundary conditions, and the increased computational power [109, 110, 111, 112, 2], as well as theoretical studies on turbine-induced instabilities in the near and far wake regions [113, 114, 2, 115, 116, 117]. Most of the above contributions have realistic inflow conditions, i.e. a canonical representation of the flat terrain atmospheric boundary layer, at least up to $\approx 25\%$ of the boundary layer thickness in which wind turbines are typically located. For offshore wind power plants, however, two major additional complications have to be accounted for: i) the influence on boundary layer flow characteristics by ocean waves (see the numerical works of [118] and [1], focusing on how oceanic waves affect the marine atmospheric boundary-layer and the wind farm performance) and ii) the effects induced by a moving rotor on the spatial evolution of turbine wakes. All the above works consider the case of a fixed rotor plane position, implying that the only factors affecting the turbine wake are related to the complexity of inflow or boundary conditions, e.g. varying turbulence intensity, introducing turbine or orographic wakes, surface waves, thermal stability or roughness effects. In the present study we address the second complications, keeping the inflow conditions for land-based turbine, while investigating the effect of a rotor in realistic, wave-induced, kinematic conditions. Because wind power production estimates strongly depends on turbine wake spatial evolution and turbine siting, we carried on an experimental study on floating wind turbines able to encompass not only the dynamics and kinematics of the turbine-platform floating system, but also the velocity distribution in the wake of the oscillating rotor.

6.2 Goals and Objectives

This work covers a set of state of the art experiments designed to 1) measure the response of a model floating wind turbine to different waves and simulated wind conditions and 2) investigate how the turbine wake respond to different platform motions in a turbulent boundary layer. The primary goals are:

- define the scaling parameter space used to relate wind tunnel and wave channel experiments

- identify the range of platform oscillations under prescribed incident waves, steady or gusty wind
- characterize the differences between the spatial evolution of the aerodynamic wake of a static turbine and of an oscillating turbine, in a turbulent boundary layer.

The problem is tackled through two, quasi-coupled experiments and limited to a two-degree of freedom type of motion, namely heave and pitch in the vertical plane aligned with the direction of wave propagation. Heave is the simple oscillatory motion along the vertical direction, while pitch has a variable angular components, with no vertical displacement. The two-degree of freedom motion, in fact, imposes that the wind and the waves are coming from the same direction, which is a common situation. The two key facilities employed in this work are the 80m long SAFL wave testing facility (Phase I) and the SAFL atmospheric wind tunnel (Phase II). The major challenge consisted in setting the correct scaling and boundary conditions in each class of experiments such that: i) the floating turbine model is tested under realistic waves large enough to exclude surface tension effects, and ii) a consistent, though differently scaled, oscillating turbine model is tested under realistic, turbulent, atmospheric boundary layer conditions with the rotor extending up to 25% of the boundary layer height.

Phase I experiments include model tests of a 13.2 MW prototype floating offshore horizontal axis wind turbine (HAWT) developed at Sandia National Laboratories (SNL), at 1/100 scale in the SAFL wave channel. This experimental investigation is intended to describe the basic and complex kinematics of a floating turbine system under the effect of both waves and variable wind. While the waves are reproduced in the renovated SAFL wave testing channel, wind is accounted for through a variable thrust force induced by a controlled spinning rotor, with opportunely pitched blades. The characteristics of the simulated periodic wind gusts are consistent with the signature of meandering wake motions generated by upwind turbine(s). The parameter space investigated thus includes variability in wave height, length, and period, wind magnitude, wind gust amplitude and period, thus covering a wide range of platform heave and pitch motions that will serve as benchmark experiments for coupled wind-wave numerical simulations of task 1.1. Phase II experiments are designed to investigate the spatio-temporal evolution of the wake of a pitching and heaving model turbine in the full developed turbulent boundary layer of the SAFL atmospheric wind tunnel. These model tests are performed at 1/1562.5 prototype to wind tunnel scale, using an actuated miniature wind turbine. Constant temperature anemometer (CTA) cross-wire and wall-parallel particle image velocimetry (PIV) are used to measure the wake in different planes. The investigated parameter space includes pitch and heave amplitudes, and frequency of the oscillatory motion. Motion data sets recorded in Phase I are used to prescribe the wind turbine scaled motion in Phase II, implemented through a synchronized linear actuator system.

6.3 Project team

Fotis Sotiropoulos James L. Record Professor, Director of the St. Anthony Falls Laboratory. Principal investigator of the project.

Michele Guala Assistant Professor at the Department of Civil Engineering and the St Anthony Falls Laboratory. Prof. Guala directed the experimental work.

Christ Feist Graduate Student in the Saint Anthony Falls Lab. Mr. Feist worked on the experiments and completed a Masters degree as a result of this work.

Kelley Ruehl Ocean Renewable Energy Engineer at Sandia National Labs. Mrs. Ruehl is the coordinator from Sandia National Labs, providing her expertise in ocean engineering to the project, both in the numerical and experimental side. Performed simulations using engineering level models.

6.4 Technical Approach: Prototype offshore wind turbine

The SNL 13.2 MW floating offshore horizontal axis wind turbine (HAWT), developed by Sandia National Laboratories, is the reference prototype turbine for this experimental study. Table 4 gives the significant geometric dimensions, mass properties, and stability characteristics of this turbine; more details on turbine development and specifications of rotor blade design are included in [42, 119]. The prototype turbine has a D=200 m rotor diameter with a hub height z_{hub}=119.5 m, as measured from mean sea level (MSL). The barge-type platform used as floating structure is a cylinder with radius R= 28.0 m and height h=15.0 m with a draft (height of the platform below sea level) d_p=7.0 m. Performance characteristics of the 13.2 MW were investigated using NREL's FAST (Fatigue, Aerodynamics, Structures, and Turbulence), which is an aeroelastic model used to predict performance and loading of horizontal-axis wind turbines. A second computational tool, IECWind, is integrated with FAST to provide realistic wind conditions for the prototype turbine and, in particular, to generate wind profiles for the simulation of the steady-state response over the operating wind speed range. Mean wind speeds ranging from cut-in velocity, 3 m s^{-1}, to cut-out velocity, 25 m s^{-1}, are modeled with the wind turbine response shown in figure 52 and 53. Figure 52 shows the aerodynamic steady-state thrust, ranging from 695.1 kN, at cut-in speed, to 2334.7 kN at rated power. Figure 53 gives rotor speed, varying from 4.34 rpm at cut-in to 7.44 rpm at rated power, as a function of wind speed. Figure 54 shows the tip-speed ratio (TSR) as a function of wind speed, where TSR is defined as the ratio between the tangential speed of the rotor tip and the hub height velocity V_{hub}. Note that TSR, typically in the range between 4 and 12 for land based turbine, is the key dimensionless parameter used in turbine wake modeling and turbine control and operation. In these figures the error bars represent the standard deviation of the thrust force, resulting from the dynamic contribution of aerodynamic forces exerted along the three spinning blades of the rotor, under a steady incoming velocity profile.

	Prototype model	Scale factor	Experimental model	
Platform radius	28.0	λ^{-1}	0.28	[m]
Platform height	15.0	λ^{-1}	0.15	[m]
Draft	8.0	λ^{-1}	0.08	[m]
Hub height	119.5		0.986	[m]
Rotor diameter	200.0		0.60	[m]
Mass	19704	λ^{-3}	19.704	[kg]
Center of gravity	8.580	$\sim \lambda^{-1}$	0.051	[m]
Center of buoyancy	-4.00	λ^{-1}	-0.04	[m]
Thrust overturning moment	1.08×10^{8}	λ^{-4}	1.077	[N-m]
Mass moment of inertia I_y	3.82×10^{10}	λ^{-5}	3.82	[kg-m^2]
Restoring coefficient (C_{55})	2.30×10^{9}	$\sim \lambda^{-4}$	30.120	[N m rad^{-1}]

Table 4: Prototype and experimental model properties for the SNL 13.2 MW with geometric scaling parameter $\lambda = 100$. Relevant properties, moments and distances are derived imposing Froude number similitude and are defined with respect to the mean sea level. The Moment of inertia is defined with respect to the pitch axis.

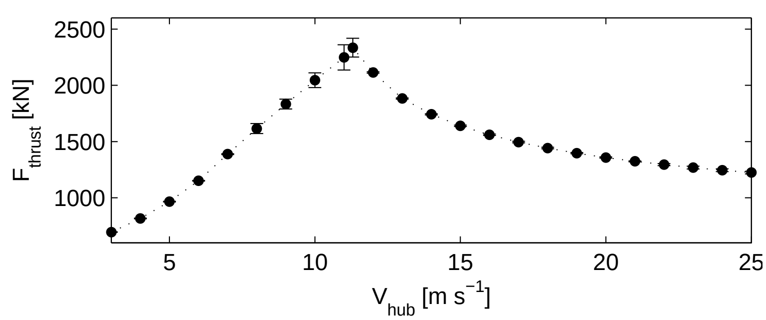


Figure 52: SNL 13.2 MW prototype scale turbine subjected to steady-state wind profile: thrust F_thrust as a function of wind speed at hub height V_{hub}; error bars indicate thrust root mean square (r.m.s.) values.

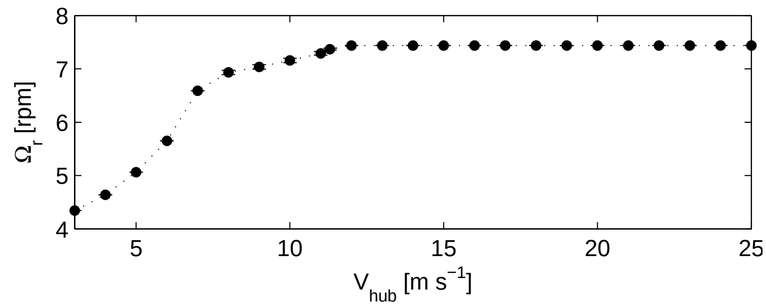


Figure 53: SNL 13.2 MW prototype scale turbine subjected to steady-state wind profile: rotor speed (or angular velocity) Ω_r as a function of wind speed at hub height V_{hub}; error bars indicate Ω_r r.m.s. values.

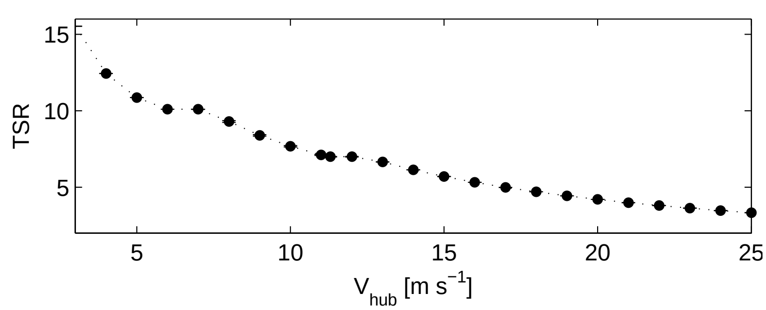


Figure 54: SNL 13.2 MW prototype scale turbine subjected to steady-state wind profile: Tip speed ratio TSR as a function of wind speed at hub height V_{hub}; error bars indicate TSR r.m.s. values.

6.5 Technical Approach Phase I: Wave Basin Experimental Setup

6.5.1 Wave channel

Phase I experiments were conducted in the SAFL wave testing facility consisting of a rectangular channel (2.75 m wide, 1.8 m high, 84 m long), a wake-maker employing an oscillating paddle hinged at the bottom of the channel and spanning its entire width and height, and an artificial beach located at the opposite end. All the experiments were run at a fixed water depth d of 1.37 m. A moving stroke-arm attached to the paddle is connected through gear reducers to a 72 hp electric motor, which is controlled by a frequency drive. The angular velocity Ω of the rotating arm defines the period of the paddle oscillations and thus of the generated wave. The length of the arm S is adjustable in order to set the paddle stroke length and obtain different wave heights. A partial investigation of the $\Omega - S$ parameter space produced a desired wave height H and period T_w in a range of $[0.01 - 0.15]$ m and $[0.6 - 3]$ s respectively. The wave testing facility is capable of testing deep water waves, i.e. satisfying $d/L \geq 1/2$ [120], for wave lengths L up to 2.74 m and corresponding to wave period of 1.3 s. The remaining wave periods tested are classified as intermediate water depth waves, defined by $1/20 < d/L < 1/2$. The sloped beach, spanning the entire width of the channel and located 80 m from the paddle, was 6.1 m in length with an adjustable slope. The beach consisted of an impermeable aluminum panel covered in permeable CavClear Masonry Mat with a thickness of 1.75 in and was inclined at a 7° angle. However, preliminary tests with short trains of waves revealed that the beach only absorbed a fraction of the wave energy, with reflected wave height averaging to approximately 15% of the incident wave height. Therefore, following a conservative protocol, we imposed that all experiments had to be performed in a time window where reflected waves were not present, an option enabled by the 80m length of the channel. This clean wave window is a function of wave period and thus changed depending on the wave parameters of interest for each test. Details on the clean wave window are discussed in §6.7.2. A schematics of the instrumented wave basin is provided in figure 55.

6.5.2 Scaled turbine model

Platform design

Model tests in the wave facility were performed on a 1/100 scale model of the SNL 13.2 MW prototype floating offshore wind turbine. Froude number similitude is used to properly scale the gravitational and inertial forces which dominate offshore platform motion, following typical scaling arguments used in offshore platform and wave basin testing [120]. The Froude number for free surface wave is $Fr = \frac{C}{\sqrt{gL}}$, where C is the wave celerity, g is the gravitational acceleration, and L is the characteristic length scale, which, in this specific case, is taken as the platform diameter D, as usually done in floating platform design. The imposed scaling relationship between the prototype scale and the model scale is $Fr_p = Fr_m$ where p denotes prototype and m model. Table 4 list the prototype and model properties, with the corresponding scaling factors derived from the the $\lambda = L_m/L_p = 1/100$ geometric scaling and the dynamic Froude number similarity. In particular, the parameters governing the two-dimensional (2D) heave and pitch, rigid body motion of the platform and turbine system, are the system mass, center of gravity, mass moment of inertia, overturning moment due to rotor thrust, and structure geometry in contact with the water (platform radius and draft). This gives a degree of freedom on hub height allowing the overall inertia and center of gravity to be tuned, to some extent. Due to technical limitations we however could not exactly match the scaled position of the center of gravity of the floating turbine model to the one of the prototype. Therefore the scaling of the center of mass and of the restoring coefficient is only approximated (indicated as $\sim \lambda$ in table 4 as it remains correct in terms of orders of magnitude). The floating platform used in these experiments is a barge type which pertains to the buoyancy stabilized category of floating platforms [121]. This class of platform was chosen for its simple geometry which aided the ease of construction, implementation and numerical validation. The platform designed for this experiment is based on the Massachusetts Institute of Technology/National Renewable Energy Laboratory (MIT/NREL) Shallow Drafted Barge (SDB) designed for the NREL 5 MW Offshore Baseline Wind Turbine model. The SDB model was scaled up to the SNL 13.2 MW prototype to obtain a starting point for platform design (geometry and mass properties). The second step in the design accounts for the effect of a mean thrust force due to steady wind condition, based on the static performance of the wind turbine system following the approach outlined in [122]. The static performance is defined as the vertical position and inclination of the floating turbine-platform system under steady-state forces, i.e. constant thrust and no incident waves. In general the full 6 degrees of freedom, rigid body motion of the floating structure subject to regular (harmonic) waves, in matrix form is defined as:

$$(M + A)\ddot{\xi} + B\dot{\xi} + C\xi = aXe^{\mathrm{i}\omega t}, \tag{80}$$

where M is the mass matrix, A is the added mass matrix, ω is the wave frequency, ξ represents the six modes of displacement along the x,y,z, axis respectively: translational (surge, sway, and heave) and rotational (roll, pitch, and yaw); x and z are the wave propagation and vertical directions, respectively, B is the damping matrix, and C is the restoring matrix, which specifically depends on the stabilizing forces maintaining the platform in equilibrium, e.g. buoyancy distribution for barge type platform.

For steady-state performance, i.e. no waves or platform acceleration, equation 80 is solved for static overturning moments and results in the static equilibrium equation $C\xi = F_{steady-state}$; this can then be solved for the steady-state pitch response due to a static overturning moment (the mean wind thrust) about the pitch axis, resulting in

$$\xi_5 = \frac{M_5}{C_{55}}, \tag{81}$$

where ξ_5 is the static pitch response [rad] and C_{55} is the platform restoring property in pitch coefficient [N m rad^{-1}]; M_5 is the overturning moment in pitch $M_5 = F_{thrust}Z_{hub}$ [N m] resulting from the aerodynamic thrust generated by the wind loading F_{thrust}, acting on the torque arm defined by the distance from the hub to the MSL, Z_{hub}. A maximum pitch angle ξ_5= 10 degrees is used in equation 81 as a design limit to maintain aerodynamic efficiency [122] and to prevent water overflow on the platform top surface (not a desired boundary conditions for experimental-numerical comparisons). The minimum pitch restoring moment $C_{55\ min}$ is then determined by applying the maximum wind loading, $F_{thrust} = 2330$ kN at rated speed, to equation 81 and solving for C_{55} which gives at the prototype scale

$$C_{55\ min} = \frac{F_{thrust}Z_{hub}}{\xi_5} = \frac{2330 \times 10^3 \times 119.5}{0.1745} = 1.782 \times 10^9 \quad [N\ m\ rad^{-1}]. \tag{82}$$

representing a case limit situation based on the above design specification and forcing assumptions. The hydrostatic properties that govern the restoring moments, contributing to bring the platform back at steady state conditions, consist of the waterplane area moment and the location of the center of buoyancy. The inertial properties are defined by the location of the center of mass and ballast planar placement. These processes result in the following equation for pitch restoring coefficient (the subscript $_E$ indicating the estimated value) valid for a circular platform of radius R:

$$C_{55,E} = F_B Z_B - M_{11} g Z_G + \rho g \pi \frac{R^4}{4}, \tag{83}$$

where F_B is the buoyant force, Z_B is the vertical position of the center of buoyancy, M_{11} is the system mass, Z_G is the vertical position of the center of gravity, g is the acceleration due to gravity, and ρ is the water density. Using $C_{55\ min}$ as a reference case limit (Eq. 82), the geometric (R) and mass (M_{11}) properties of the platform at prototype scale are tuned through equation 83 to achieve an acceptable steady-state response in pitch. The geometric and dynamic platform properties summarized in Table 4 (for the prototype model) lead to $C_{55,E} = 2.30 \times 10^9\ \ [N\ mrad^{-1}] > C_{55\ min}$, ensuring that the platform will not exceed the 10 degree threshold under the prescribed wind thrust.

Platform positioning system

Figure 56 comprises an image of the experimental setup and coordinate system and a schematics of the key designing components. With the z-axis aligned with the turbine platform center and center of gravity, heave is the displacement along the z-axis with zero position corresponding to the platform turbine system at rest. Pitch is defined as rotation about the y-axis with positive pitch being counter-clockwise rotation when looking down the y-axis toward the origin. Zero pitch identifies the system at rest (no wind, no waves). The platform is restricted to two degrees of freedom, pitch and

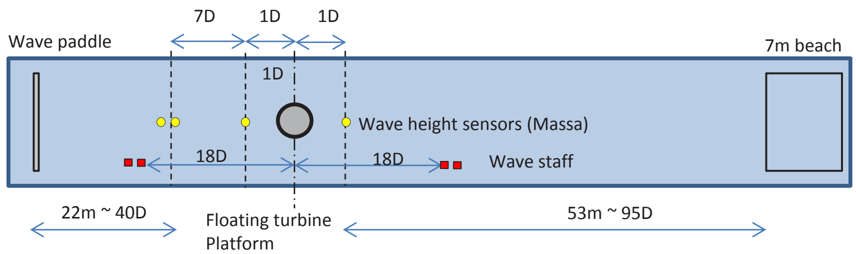


Figure 55: Schematics of the wave basin experiments and deployed instrumentation

heave, which are allowed by a combination of radial and linear roller bearings. Two radial bearings, one on each side, are located along the pitch axis at the center of gravity. The radial bearings allow for pitching motion and secure the platform to the linear roller bearings. Two linear roller bearings, one located on each side and directly above their respective radial bearings, constrain the platform in the y and x axis, and allow heaving motion. The combination of the two bearing styles prevents rotation about the z-axis, yaw, and the x-axis, roll, while keeping the turbine at the same x, y position. As a constrain of the platform positioning system, the radial bearings had to be attached to the upper surface of the platform, thus keeping the axis of rotation well above the water level. The center mass position was adjusted using two submerged masses at adjustable depths attached to the platform bottom plate. In order to provide an additional stable contribution, the center of mass was located $3.4cm$ below the theoretically scaled center of mass coincident with the actual axis of rotation. Note that By limiting the distance between the bearing system rotation axis and the center of gravity of the floating turbine structure, we ensure that only small, additional forces and moments are introduced in the kinematics of the platform to keep the platform in place under the action of the waves.

6.5.3 Scaled wind-wave environment

For the floating test of the 1/100 offshore turbine model, realistic wind and wave conditions recorded at one site in the Pacific Northwest (Buoy 46041 from the National Data Buoy Center, NDBC) were scaled according to the imposed Froude similitude. Bouy 46041, located nearby Cape Elizabeth, approximately 45 nautical miles Northwest of Aberdeen, WA, is equipped with a wind anemometer at 5 m height, and has gathered wind and wave data since 1987. Water depth at this location is approximately $d_y = 114$ m which allows deep water waves ($(L/d_y) < 1/2$) for a range of wave lengths up to L=230 m. The wave channel facility utilizes a water depth of 1.37 m providing a similar scaled range of deep water waves. Hub height mean wind velocity is extrapolated from the measured and averaged 5 m anemometer velocity to a height of 119.5 m, assuming a power law distribution according to the International Electromechanical Commission (IEC) standard [123], :

$$V(z) = V_{hub}(z/z_{hub})^{\alpha}, \tag{84}$$

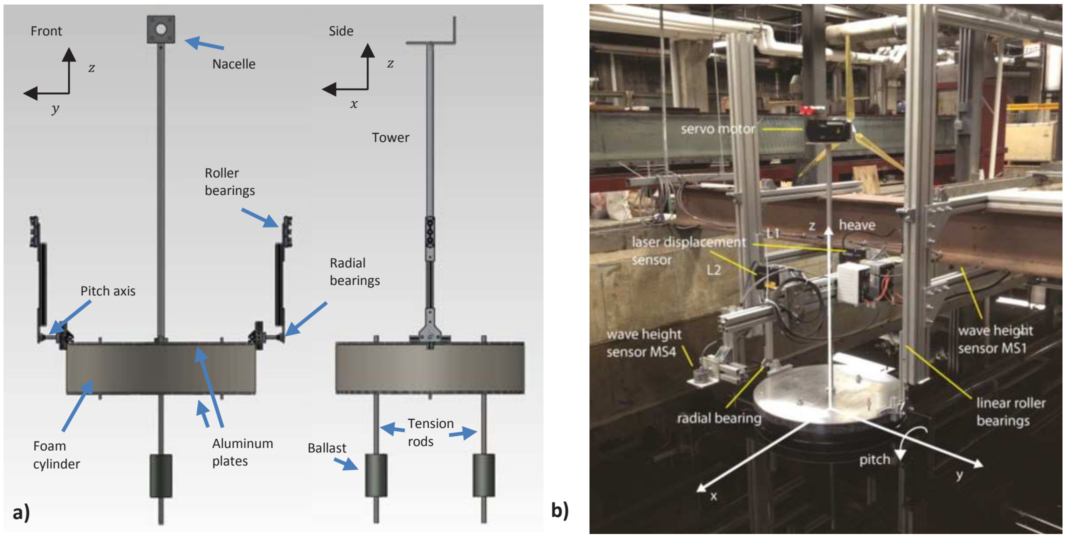


Figure 56: a) Platform drawing and key components, b) wave Basin experimental setup and coordinate reference system

where $V(z)$ is the wind speed at height $z = 5m$, V_{hub} is the wind speed at hub height and $\alpha = 0.2$ is the wind shear power law exponent. Equation 84 led to an averaged wind speed V_{hub}= 8.21 m s^{-1}. Within the same averaging time used for the wind statistics, the mean wave peak period and significant wave height were T_p=10.96 s and $H_s = 2.18m$, respectively, with H_s defined by the mean wave height, trough to crest, of the highest third of the waves. Since the significant waves, once scaled down to the wave basin experiments, are reproduced as sinusoidal waves, we adopt hereinafter the simplified notation T_w, H_w. The annual joint probability distribution of wave characteristics, scaled down to the model basin following Froude number scaling, is given in table table 5. The most common waves, with periods $T_w = [1.0-1.29]$ s and heights $H_w = [1.5-2.4]$ cm, define the range of wave characteristics explored in our experiments, with average wave period of 1.10 s and average wave height of $0.02m$.

6.5.4 Data acquisition system and control

Data acquisition and control is performed by a National Instruments cRIO-9024. The cRIO-9024 is a stand-alone measurement and control device running LabVIEW Real-Time allowing deterministic motor control and data logging. The system was designed to perform i) wave channel experiments, where motor control and analog measurements would be recorded at 100 Hz, and ii) wind tunnel experiments, where the system would need to control two linear actuators, integrate with a PIV capture system, record motor positions and hotwire voltages up to 10,000 Hz. In the wave channel the cRIO is integrated with an 8-slot reconfigurable chassis, NI 9112, where an analog I/O module is added. The 32 input channels analog module, NI 9205, has 16 bit precision, capture rates up to 250 kS-s^{-1}, and programable measurement range.

Wave measurements: Model tests started with a partial investigation of the

	H_w [cm]										
T_w [s]	0.5-1.4	1.5-2.4	2.5-3.4	3.5-4.4	4.5-5.4	5.5-6.4	6.5-7.4	7.5-8.4	8.5-9.4	9.5-10.4	TOT%
<0.3	*	-	-	-	-	-	-	-	-	-	0
0.3-0.39	0.1	-	-	-	-	-	-	-	-	-	0.1
0.4-0.49	0.5	0.1	-	-	-	-	-	-	-	-	0.6
0.5-0.59	1.5	0.6	0.1	*	-	-	-	-	-	-	2.1
0.6-0.79	8.6	4.5	0.9	0.2	*	-	-	-	-	-	14.2
0.8-0.99	8.1	8.5	2.3	0.7	0.2	*	*	-	-	-	19.9
1.0-1.29	8.1	18.2	10.6	4.4	1.5	0.5	0.1	*	*	-	43.5
1.3-1.59	2.6	3.2	3.0	2.0	1.0	0.4	0.1	*	*	*	12.4
1.6-2.0	1.4	1.1	0.7	0.4	0.2	0.1	*	*	*	*	6.0
≥2.0	0.3	0.4	0.3	0.1	*	*	*	*	*	-	1.2
TOT %	31.8	36.9	18.3	8.1	3.1	1.1	0.3	0.0	0.0	0.0	100
TOT N	45910	53167	26448	11762	4439	1590	472	153	40	15	

Table 5: Joint Distribution of wave period T_w and amplitude H_w of the model wave environment, scaled from the data recorded by the Buoy 46041 (Pacific Northwest National Data Buoy Center).

wave maker frequency-stroke, $\Omega - S$, parameter space without any floating structure to reproduce the desired, scaled, environmental wave conditions presented in §6.5.3. Four Massa ultrasonic M-300/150 sensors, designed to measure water elevation within a vertical range of ±12 cm, with a vertical resolution of 0.25 mm and a surface area defined by a full cone angle of 15 deg, were simultaneously acquiring wave height measurements at 20 Hz. Two sensors were located near the platform, at ±1D from its center, with the platform diameter D=56cm. The incident waves were measured at -8D (considered as far upstream) with two sensors. The sensors were spaced 80 cm in the x direction to allow the estimates of wave celerity c, through wave peak tracking. In addition to the four M-300/150 sensors, during initial tests also four Ocean Sensor Systems Wave Staff OSSI-010-002E (fast capacitive type, water level sensor with ±2.5 mm accuracy) were employed. A total of 8 time resolved measurements of water elevation, at different locations, were used to monitor wave propagation characteristics in the x direction and wave distortion along the cross (y) direction, near the location where the floating wind turbine would be installed.

Platform kinematics measurements: The offshore floating platform position is measured at 100 Hz using two Keyence laser displacement sensors, L1 (model LK-501) and L2 (model LK-G502), located on the x-axis at ±220 mm from the platform center. Restricting the motion to heave and pitch and assuming rigid body motion allows the two displacement sensors, with measurement repeatability of 50 μm, to fully define the turbine position and motion (figure 56).

Wind thrust generation and calibration: In the absence of real wind in Phase 1 experiments, one way to create an equivalent thrust force and overturning moment consisted in spinning a three blades rotor, with blade pitch adjusted to work as a fan, rather than a turbine. A 7.26 Nm brushless DC servo motor, AKM32H, is used to control the rotor speed of the floating turbine. The servo motor is connected to the cRIO which sends control commands via EtherCAT to a 6 Amp AKD motor

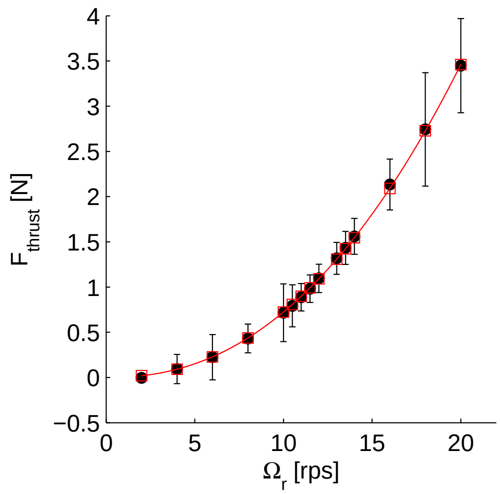


Figure 57: Model thrust curve F_{thrust} as a function of the rotor speed Ω_r: averaged experimental results shown as black dots with r.m.s. value shown as bars; using a least-squares log-fit, the mean theoretical thrust $F_{thrust,t} = 10^{-2.40}\Omega_r^{2.26}$ was obtained. (red square)

drive. Motor position is captured via a Smart Feedback Device (SFD) at the motor shaft and sent back to the cRIO, with updated speed control at a rate of 100 Hz. The rotor speed was varied during specific sets of experiments exploring the effects of constant, or variable, wind thrust on the platform kinematics. Simulated rotor thrust is calibrated using an Advanced Mechanical Technology Inc. (AMTI) multi-axis force transducer, MC3A-100-6691. The horizontal thrust component, F_x, in the x-direction is recorded for rotor speeds ranging from 2 to 20 rps. The model tower, nacelle, DC servo motor, and rotor are rigidly attached to the force-balance sensor, providing mean and fluctuating torque for given angular velocity and resulting in the calibration depicted in figure 57. Note that the range covered in the calibration $(0 - 4N)$ is about twice as large as compared to the range of the scaled thrust force (scale factor λ^{-3}) estimated at prototype scale and shown in figure 52.

6.6 Technical Approach Phase II: Wind Tunnel Experimental Setup

Phase II work investigates the turbine-wind interaction of a scale model under pitching and heaving motions in the zero pressure gradient turbulent boundary layer generated in a controlled wind tunnel. The goal is to determine what is the effect of a variable pitch and heave on the wake of a floating turbine and what are the resulting inflow conditions experienced by downwind turbines in offshore power plants.

D [m]	z_{hub} [m]	δ [m]	T [deg C]	ν [m^2 s^{-1}]	U_{hub} [m s^{-1}]	u_* [m s^{-1}]	Re_D [-]
0.128	0.104	0.4	20	1.6e-5	2.26	0.10	18276

Table 6: Turbulent boundary layer flow characteristics and scaling parameters

6.6.1 Wind tunnel setup

Phase II experiments were performed in the closed-circuit atmospheric wind tunnel at SAFL. The closed loop wind tunnel has a 16 m long test section with cross-section of 1.5 x 1.7 m (W x H) and an adjustable roof in order to enforce zero pressure gradient in the flow direction. At the beginning of the test section a 0.04 m picket fence is used to raise the turbulent boundary layer to keep the turbine height within $\simeq$25% of the boundary layer thickness. Wind tunnel temperature of the air and test section floor are controlled to create a neutrally-stratified boundary layer. Air temperature is controlled via heat exchanger, downwind of the flow driving fan and upwind of the test section contraction, to $\pm 0.1\ ^O$ C. Floor temperature is controlled in the same range through circulating a water-glycol mixture through a second heat exchanger before flowing into the aluminum test section floor panels (see [52, 103] for more details). Constant temperature anemometer (CTA) cross-wire, wall-parallel PIV, are used to spatiotemporally characterize the wake behind a miniature offshore floating wind turbine (OFWT).

Baseline flow

Figure 58 shows the vertical profiles of the mean velocity (a), turbulence intensity (b), and turbulent shear stress (c), measured with a CTA cross-wire at 10 kHz, in the baseline turbulent boundary layer flow. The mean velocity at hub height, $U_{hub} = 2.26$ m s$^-1$, is used for normalization. The zero pressure gradient, neutrally stratified boundary layer has a Reynolds number, based on the shear velocity, of $Re_\tau = u_*\delta/\nu \approx 0.3 \times 10^4$. The shear velocity, $u_* = 0.10$ m s^{-1}, was estimated by fitting the logarithmic law of the wall, $U = \frac{u_*}{\kappa} \ln \frac{z}{z_0}$ to the measured velocity profile. The Von Kármán constant, κ is taken as 0.4, while the aerodynamic roughness length resulted $z_0 = 0.03$ mm. The boundary layer height, $\delta \simeq 0.4$ m roughly corresponds to four times the model wind turbine hub height, a factor typically observed at utility scale. Table 6 summarizes the boundary layer characteristics as well as scaling parameters i.e. rotor diameter D, and hub height z_{hub}. Re_D is the Reynolds number associated with the rotor diameter and is calculated as $Re_D = U_{hub}D/\nu$.

Wind turbine model

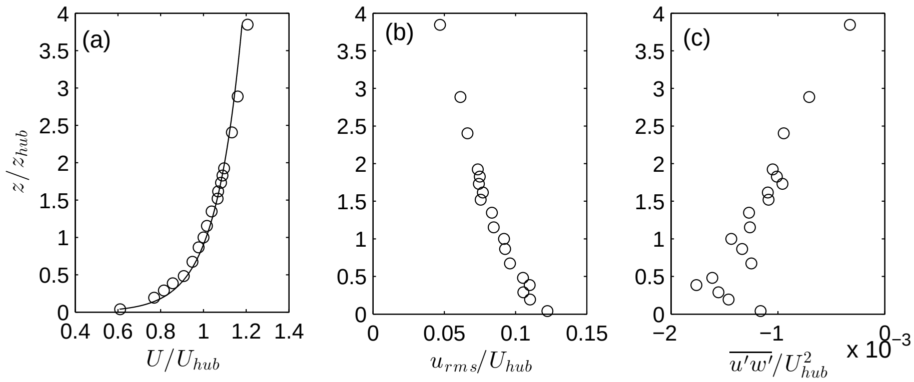


Figure 58: Turbulent boundary layer characteristics: (a) mean vertical velocity profile at turbine location with logarithmic law of the wall fit, (b) turbulence intensity, and (c) kinematic Reynolds stress. Height, z, is normalized by the turbine hub height, z_{hub}. Velocity characteristics are normalized by the velocity at hub height, U_{hub}

The model wind turbine used in this set of experiments, originally designed and built by [52], and later modified by [124, 117], utilizes a 2V DC generator with the ability to either capture voltage or apply a torque to manipulate rotor speed and vary the tip speed ratio [103]. We remind that $TSR = \omega r/U_{hub}$, where r is the rotor radius ($0.5D$) and ω is rotational speed of the rotor [rad s^{-1}]. The rotor is fixed to the shaft of the DC motor generating a voltage linearly proportional to the angular velocity and the incoming flow speed [103]. Constant TSR results from a proportional increase in both electrical torque and friction with increasing aerodynamic torque. Wind tunnel and engine dynamometer tests were conducted to evaluate the model wind turbine performance, in terms of power coefficient C_p, as performed in [103]. We includes results with the turbine operating in frictional torque condition (i.e. with no electrical current applied to the DC generator to increase torque and slow down the rotor), resulting in $TSR_{fr} = 4.5$ and $C_p \approx 0$, as compared to the optimal tip speed ratio conditions with $TSR_{opt} = 3.2$ and $C_p = 0.27$.

The Geometrical characteristics of the miniature turbine include the hub height, $z_{hub} = 0.104$ m, and a three-blades (GWS/EP-6030x3) rotor diameter $D = 0.128$ m, defining the top-tip position, $z_{top-tip} = 0.168$ m and the bottom-tip position, $z_{bottom-tip} = 0.040$ m. Geometric similarity between the prototype wind turbine and the wind tunnel model, using the rotor diameter, gives a scaling ratio of 1/1562.5. Figure 59 provides a picture and a schematic of the model turbine reference system. The vertical z axis, has the origin coinciding with the wind tunnel surface, and it is aligned with the tower and the wall normal W velocity component. The y axis is aligned with the spanwise velocity, V, and has a zero position along the centerline of the nacelle. The x axis is aligned with the streamwise velocity, U. Pitch is defined as a rotation about the y axis, around a pivot point located 0.039 m below the surface. Zero pitch implies the turbine is vertical, aligned with the z axis, while positive pitch

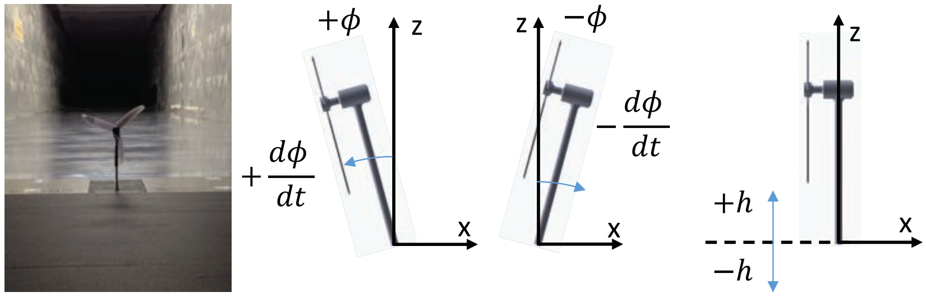


Figure 59: Photograph and schematic of wind turbine motion phases

indicates a rotation against the flow, as illustrated in figure 59.

6.6.2 Temporal scaling of waves and spinning rotor

One of the major challenges in studying the wake behind a pitching and heaving wind turbine is to define the temporal scaling from prototype to wind tunnel scale. The question in particular is, at what frequency should the wind turbine model fluctuate in pitch and/or heave in order to correctly represent the key, time-dependent, forcing processes and the related timescales (incident wave period, wind gust period, and the blade rotational period) at utility-scale. By neglecting wind unsteadiness and choosing the rotor frequency, f_r, to scale the wave frequency, f_w, the temporal scaling factor is defined through $\frac{f_{r,p}}{f_{w,p}} = \frac{f_{r,m}}{f_{w,m}}$, where p and m subscripts denote prototype and model scale, respectively. By applying a voltage, the turbine model has some control on the rotor frequency $f_{r,m}$, though in a narrow range, consistent with the limited variation in tip speed ratio. $f_{w,m}$ is controlled by setting the oscillation frequency of the actuator system, which also presents mechanical limitations. At prototype scale, the rotor frequency at rated speed is $f_{r,p} = 0.12$ Hz. Using results from the wave basin testing, specifically the range of wave periods used to generate the Response Amplitude Operator, and scaling up to prototype scale, the wave frequency has a range of $f_{w,p} = [0.04 - 0.1]$Hz, consistent with field data from the PNW buoy. This implies that the rotor period ranges from the wave period to about 1/2 of it: based on orders of magnitude the two forcing timescales are comparable. Wind tunnel tests performed with a relatively low hub velocity $U_{hub} = 2.5$ m s^{-1}, would reduce the rotor speed within the operational range of the model turbine, and allow a slower oscillations in pitch and heave. At this wind speed, the adjustable rotor frequency ranges from $f_{r,m} = 20$Hz, corresponding to the optimal $TSR = 3$, to $f_{r,m} = 28$Hz corresponding to the frictional loading $TSR = 4.5$. The resulting wave frequency in the model $f_{w,m}$ from the scaling equation, gives a range of oscillations of $f_{w,m} = [5.6 - 25.9]$Hz, which is beyond the $4Hz$ limiting frequency that the model turbine was observed to withstand due to intense accelerations. While this limitation is a major constraint in the above scaling approach, it is noteworthy that the rotor frequency would define the flow timescale only in the turbine near wake, where the tip vortex is a dominant flow feature [108]. As one of our major goals is to study the incoming flow conditions experienced by floating turbines in offshore wind power plants, the far wake evolution consistent with the typical $5 \simeq 10D$ turbine spacing, is considered more relevant than the near wake. For these two reasons, a different time scaling scheme was developed. Of specific interest in the wake study of an oscillating wind turbine is how the large scale meandering flow structure related to the vortex shedding of the rotor, would be effected, as compared to a static, land-based, turbine. Recent experimental work on the topic of wake meandering for horizontal axis turbines [2, 117, 113, 114, 59, 115] have suggested a range of Strouhal numbers $(0.2 - -0.3)$ which characterize this large-scale instability in the far wake. Our work specifically looks at oscillating a model turbine at a frequency similar to that of the wake meandering to explore the potentially critical, resonant conditions in which the wake may be amplified by the platform oscillations. This approach, using the St definition to provide an estimate for the frequency of the large-scale instabilities in the turbine wake to define the the pitching and heaving motion of the turbine model, is equivalent to the approach followed in the wave basin experiments to define the fluctuation period of the rotor thrust and simulate wind gusts. Based on the inflow velocity at hub height $U = U_{hub} = 2.5$ ms^{-1}, and on the diameter of the rotor $D = 0.128$m, the range $St_m = [0.1-0.3]$, lead to a range of frequencies $f = [2-6]$ Hz. Taking mechanical limits into consideration and the range of frequencies determined from the Strouhal number range, an oscillation frequency $f_{w,m} = 3$ Hz was chosen for the test cases.

6.6.3 Data acquisition and control

The data acquisition and control system used in the wave basin experiments is also used in the wind tunnel tests, with the following updates: two electric motors were controlled (one per actuator), an additional digital input/output module (NI 9403 with 140 kS-s^{-1}) was used, and the sampling frequency was increased to 10,000 Hz. The objectives of the data acquisition and control system are to provide: i) precise control of a linear actuator system (two electric motors) which determines the heave and pitch motion of the model wind turbine with a vertical position accuracy within 0.05 mm; ii) high temporal resolution (10k Hz) velocity measurements by CTA cross-wire anemometer, synchronized with the oscillating turbine position, velocity, and acceleration; iii) high spatial resolution velocity measurements by PIV, synchronized as well with the turbine motion. A custom LabVIEW program was written to control the position of the turbine (motor position) and acquire flow measurements/signals. Model turbine motion in pitch and heave are imposed as sinusoidal oscillations with a defined amplitude and frequency. Turbine motion time histories for pitch, heave, and combined pitch and heave are pre-generated, converted to motor revolution positions and loaded into the LabVIEW motion control program. The system is then able to simultaneously move the model turbine and acquire either cross-wire velocity measurements, or PIV laser pulse time histories for synchronizing PIV vector fields with the rotor position.

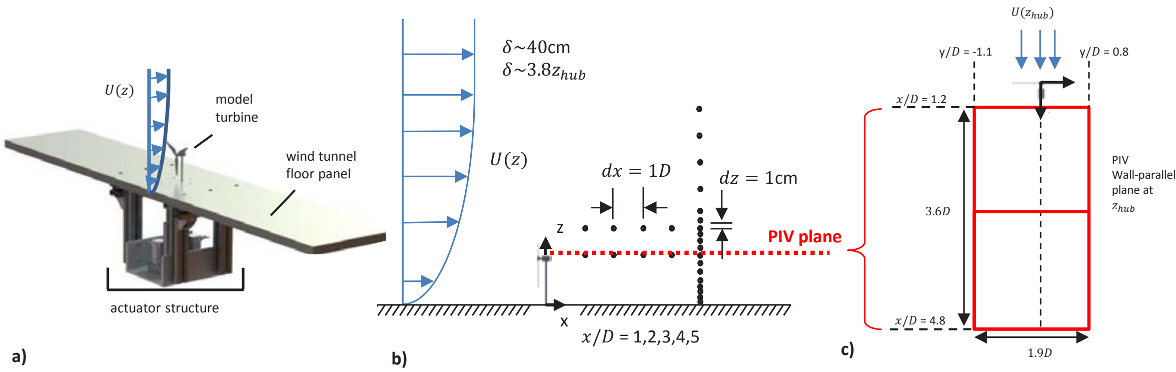


Figure 60: setup for the wind tunnel experiments a) actuator system, b) $x - z$ plane view of CTA crosswire measurements locations (dots) , c) wall parallel PIV field of view.

Hotwire setup

The hotwire sensor consisted of two tungsten wires with diameter of 5 μm and was connected to an A.A. Lab Systems AN-1003 10-channel CTA/CCA amplifier system. Output from the amplifier was recorded by the data acquisition and control system at 10 kHz. The hotwire was calibrated using nine velocities from 0.5 m s^{-1} to 5 m s^{-1}, and seven angle rotations from +30 to -30 degrees in 10 degree increments. Temperature during the calibration was maintained to within $\pm$0.2 degree C of the test temperature. Figure 60 gives a schematic of the model wind turbine located in the turbulent boundary-layer within the wind tunnel. Hotwire measurements are taken in the wake of the model turbine at $y/D = 0$ and various x and z positions. Spatial measurements in the streamwise direction are taken at $x/D = 1, 2, 3, 4$ and 5 for $z = z_{hub}$ and $z_{top-tip}$. A refinement in vertical measurement positions is provided at $x/D = 5$ with measurements starting at 4 mm above the floor and continuing up to $\delta \simeq 0.4$m, allowing for a detailed vertical profile at $x/D = 5$, generally considered as the minimum streamwise spacing in wind farms.

Particle Image Velocimetry setup

PIV measurements were also made in the turbine wake horizontal plane with $z = z_{hub}$ for spatially resolved streamwise and spanwise velocity components. Figure 60 gives a schematic of the PIV measurement window. The measurement windows spans from $y/D = -1.1$ to $y/D = 0.8$ and in the streamwise direction from $x/D = 1.2$ to $x/D = 4.8$. The PIV system consisted of two CCD cameras with 2048 by 2048 pixel resolution. These cameras simultaneously captured images at 7.25 Hz to generate the large field of view observed in figure 60. Additional components included TSI Insight

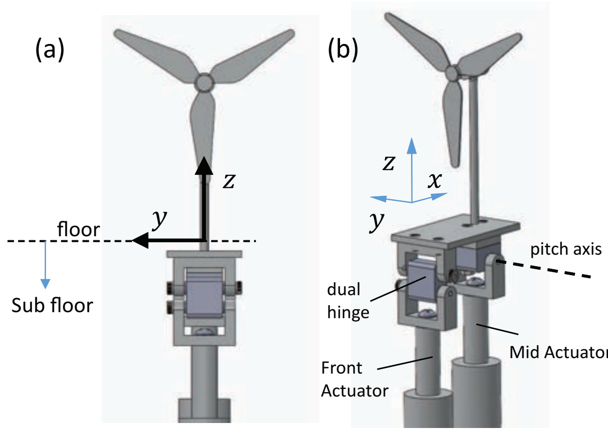


Figure 61: Detail view of actuator attachments for the model wind turbine. (a) front view ($y-z$ plane) with wind tunnel floor represented by the dotted line. (b) isometric view showing the mid and front actuators

4G software coupled with a TSI synchronizer controlling the timing of the laser and cameras. The laser used was a Big Sky dual-head Nd:YAG laser, illuminating olive oil droplets (seeding particles with diameter on the order of 10 microns) generated by a Laskin nozzle and entrained into the flow.

Linear actuator system

A dual linear actuator system, located under the wind tunnel floor, was used to perform the desired motions of pitch and heave on the model wind turbine. Figure 60 shows the wind tunnel floor panel where the actuator system was attached to, with the turbine model above the wall and within the boundary layer, while all other moving components of the actuator system were not disturbing the flow. The system is comprised of two linear actuators by Ultra Motion. Each linear actuator is driven by a brushless DC servo motor, AKM11B, capable of applying 0.61 Nm and rotational speeds up to 4000 rpm. Precise motor position is captured by a Smart Feedback Device (SFD), with a vertical displacement of 0.01016 m per revolution. A detailed view of the linear actuator system, specifically the rotation mechanisms, is shown in figure 61. The mechanism consists of a mid actuator and a front actuator. Each actuator is attached to a plate through a bracket and hinge. The mid actuator, which is the primary driver for heaving motion, includes a single hinge which acts as the pitch axis (located 0.039 m below the wall). The front actuator, which comprises a dual-hinge mechanism, is the primary driver of the pitching motion. The two actuators work in tandem to provide the desired heave in time, $h(t)$. The front actuator matches the heave value of the mid actuator and then deviates to create the pitching motion in time, $\phi(t)$.

Turbine Configuration	Test Type
Baseline; no wind, no motor cables	heave and pitch free decay
No wind, motor cables attached	heave and pitch free decay
Wind; 8, 12, 16 rps	pitch free decay

Table 7: Free decay test summary

6.7 Phase I: Wave Basin Results

6.7.1 Free decay test

Free decay tests were performed on the floating wind turbine to measure the response to initial vertical or angular displacements, in pitch and heave respectively, and determine the system natural periods and damping. Heave decay tests were performed by raising the platform-turbine system to a specific vertical position, holding it in place until equilibrium was reached, and then releasing it by means of a mechanical switch. The initial position was set at different height, while always maintaining the lower surface of the floating platform submerged. Pitch decay tests were performed by using a marked rod to i) push and hold the platform at an initial pitch steady-state and ii) release it , allowing the platform to oscillate in pitch. Consistent with the heave decay tests, for the maximum initial pitch the platform remained partially submerged with no portions of the upper surface covered by water.

In the pitch free decay tests, the platform-turbine system is subject only to an initial angular displacement. Likewise, for heave tests, only a vertical displacement from equilibrium position is imposed. A minimum of three different initial displacements are used in pitch and heave to test the turbine-platform system response to different perturbations. Additionally, this set of tests was performed on three different turbine configurations, table 7 summarizes the free decay tests. The first set considered the wind turbine without wind and thus without motor cables attached. This setup would be used to validate the computational model and simulate motion in the wind tunnel experiments. The motor cables were then attached and tests were performed using this configuration, again without wind. The third and last set of free decay tests employed a spinning rotor with motor cables connected. Decay tests in pitch only were performed at mean rotor speeds of 8, 12, and 16 rps, corresponding to rotor thrusts of 0.44, 1.09, and 2.10 N at the model scale, respectively, to investigate if variations in rotor speed (thrust) had an effect on pitch damping.

For the free decay tests of each heaving or pitching mode, the system is treated as having a single degree of freedom. Viscous and frictional damping are small resulting in an underdamped system with oscillatory motion. For the free decay response the resulting displacement x due to an initial displacement X is described by the following equation [120]:

$$x = X \exp(-\zeta\omega_n t)\sin(\omega_d t) = X \exp(-\zeta\omega_n t)\sin(\sqrt{1-\zeta^2}\omega_n t) \tag{85}$$

where ω_n is the system natural frequency and ζ is the damping ratio, or damping factor. The solution for the underdamped system is an exponential decay function with an oscillating term. The oscillations have a frequency $\omega_d = \sqrt{1-\zeta^2}\omega_n$ defined as the damped natural frequency. The damping ratio is found by determining the logarithmic decrement, δ, which uses a ratio of two successive peaks from the time-domain response of the system. As applied to the initial displacement, X, and the immediate successive peak, δ can be defined by

$$\delta = \ln\left(\frac{x_0}{x_1}\right), \tag{86}$$

where $x_0 = X$ is the initial displacement, x_1 is the first damped oscillation amplitude after one period T, which is estimated from two consecutive oscillations peaks and used to determine $\omega_d = 2\pi/T$. The damping ratio ζ is found from the logarithmic decrement by

$$\zeta = \frac{1}{\sqrt{1+\left(\frac{2\pi}{\delta}\right)^2}}. \tag{87}$$

With ζ and ω_d known, the system natural frequency is determined by $\omega_n = \frac{\omega_d}{\sqrt{1-\zeta^2}}$. Figure 62 shows the measured platform pitch response to two different initial angles of 0.5 and 6 degrees. The decaying oscillations with time can be observed. Similarly, figure 63 shows the heave response to two initial displacements, 10 and 20 mm. As compared to the heave decay response, the pitching motion has a relatively low damping factor resulting in many oscillations before coming to rest. For comparison, the damping factor is 8 times larger for heave than pitch. This reflects the larger frictional forces within the linear bearing rails system as compared to the ball bearing system, and the larger mass of water displaced under heaving motion. Results for averaged ω_d and ζ over multiple tests of both pitch and heave decay responses are presented in table 8. The damped natural frequency in pitch is 0.41 Hz with standard deviation of 0.008 Hz (2% of the mean) for the baseline test with $\zeta \approx 0.02$, r.m.s. of 0.007 (30% of the mean). The addition of cables increases the pitch damping ratio to $\zeta \approx 0.05$ (under different, but steady, simulated wind thrust conditions) and results in slower oscillations with $\omega_d = 0.38$ Hz. The latter average frequency is consistent for all tests in which motor cables were attached. The perceived change in pitch damping ratio with different simulated wind cases is a function of the average calculation. As seen in figure 65 (a) and discussed below, ζ is fairly consistent over the range of rotor speeds tested, indicating that a steady wind is not expected to change the turbine system frequency response (the cables only are responsible for the observed differences).

Figures 64 (a) and 65 (a) show the damping ratio as a function of initial platform pitch and heave displacement, respectively, resulting from experiments with or without simulated wind. The variability in the damping ratio ζ is observed to be mostly caused by small initial pitch angles (below 2 degrees, corresponding to relatively large damping values) and by the presence or absence of the cables, inducing

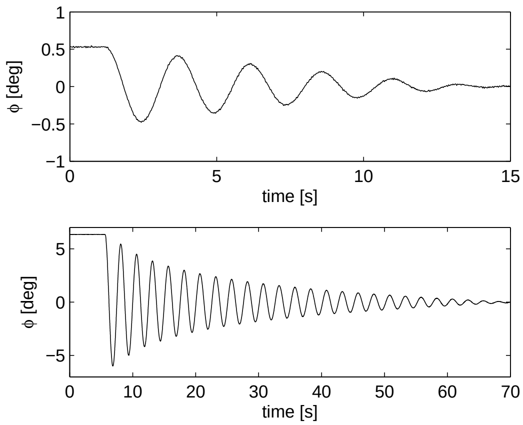


Figure 62: Pitch response in time due to an initial pitch of 0.5 (top) and 6 (bottom) degrees.

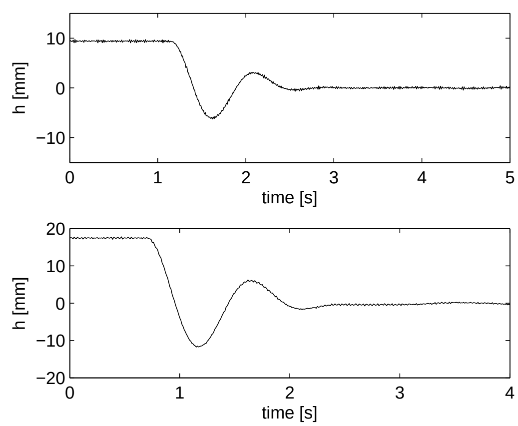


Figure 63: Heave response in time due to an initial displacement of 10 (top) and 20 (bottom) mm.

Turbine Configuration	Pitch ω_d [Hz]	Heave ω_d [Hz]	Pitch ζ	Heave ζ	Pitch ω_n [Hz]	Heave ω_n [Hz]
baseline	0.41 (2.45 s)	1.09 (0.92 s)	0.02	0.16	0.41 (2.45 s)	1.10 (0.91 s)
cables	0.38 (2.62 s)	1.06 (0.94 s)	0.06	0.15	0.38 (2.61 s)	1.07 (0.93 s)
cables + wind 8 rps	0.38 (2.62 s)	-	0.04	-	0.38 (2.62 s)	-
cables + wind 12 rps	0.38 (2.66 s)	-	0.05	-	0.38 (2.66 s)	-
cables + wind 16 rps	0.38 (2.61 s)	-	0.05	-	0.38 (2.61 s)	-

Table 8: Free decay response under initial heave or pitch: damped (ω_d) and natural (ω_n) frequencies, damping ratios (ζ)

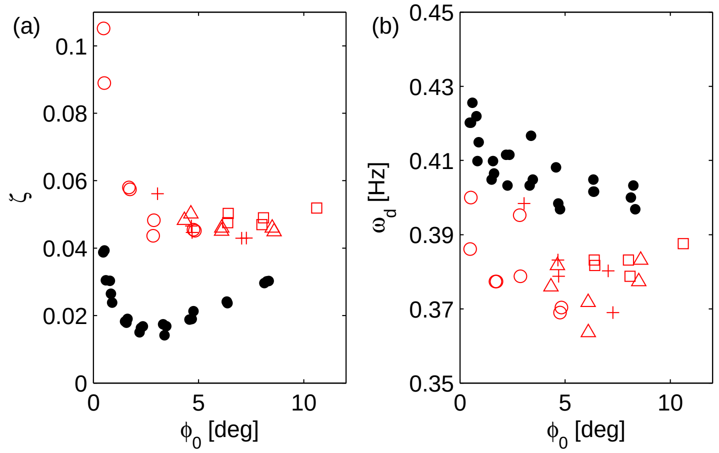


Figure 64: (a) damping ratio ζ for varying intial pitch displacements ϕ_0 and (b) damped natural frequency for varying intial pitch displacements ϕ_0. • indicate the baseline case, while red symbols mark experiments with cables attached: ∘ no wind; + wind 8 rps; △ wind 12 rps; □ wind 16 rps.

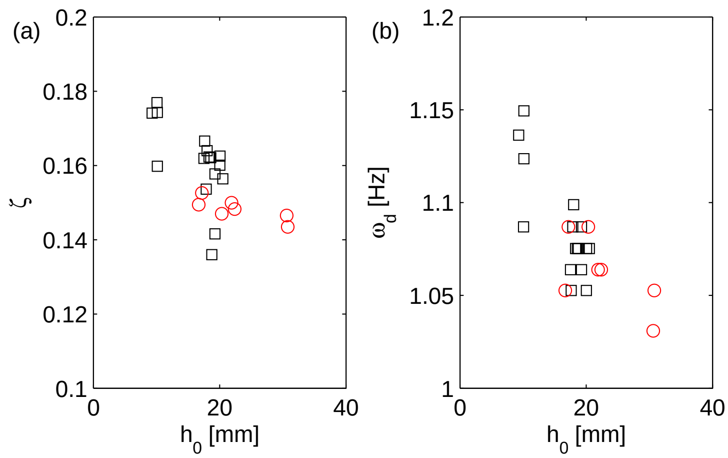


Figure 65: (a) Damping ratio ζ for varying intial heave displacements and (b) damped natural frequency for varying intial heave displacements. □ baseline case, ∘ cables

a larger damping regardless of the actual simulated wind speed. Although ζ in pitch shows a wide range of values, the effect on ω_d is actually very small (Figure 64 b). Results suggests that the floating system under steady wind and wave forcing will tend to pitch at similar ($\simeq 7\%$) frequency as compared to the wave only case, with a decreased amplitude due to the sole effect of the cables, not of the wind. As with the pitch response, also the heave damping coefficient shows a dependency on initial heave displacement, with however negligible effects by the cables or the simulated steady wind. Damped natural frequencies as a function of initial heave displacements are shown in figure 64 (b) within a $\pm 5\%$ interval

6.7.2 Response amplitude operator

Once the free decay tests were completed, with damped natural frequencies for the pitch and heave modes known, a measuring campaign to characterize the response of the floating wind turbine system subject to regular sinusoidal waves was performed. Heave and pitch of the floating turbine are used to calculate a response amplitude operator (RAO), which characterizes the response of the floating wind turbine while operating in a given sea state. RAO in pitch is defined as the maxima to minima pitch angular displacement, normalized by the wave height [deg m^{-1}]. The heave RAO is defined as heave displacement normalized by wave height [m m^{-1}]. The RAOs were generated using wave periods ranging from 0.91 to 3.00 s and wave heights from 0.93 to 3.5 cm, at model scale. RAOs represent transfer functions enabling the estimate of the floating platform pitch and heave for given waves characteristics. We however stress that RAOs obtained at laboratory-scale facilities may be used only with extreme caution to predict platform kinematics at the prototype scale, as their response is not expected to be linear within such a wide range of wave height [125].

Figure 66 gives an example of a clean-wave window for one of the RAO realization. The top graph shows time resolved measurements from two sensors of the incident wave height at -8D. The measurement window includes a transitional period, (a), where the paddle frequency ramps up, creating waves of varying height and period, followed by the clean-wave window, denoted by (b). Region (c), influenced by marginally reflected waves, has been determined by wave train tests before the actual measurements and was not included in the reported statistics. The clean-wave window, in which wave statistics are estimated, is characterized by consistent wave height and wave period between two sensors MS2, MS3 located at at -8D, and -9.28D respectively. The figure depicts waves generated with $T_w = 1.01$ s and $H = 2.60 \pm 0.12$ cm as measured from MS2 and $H = 2.54 \pm 0.09$ cm using MS3 measurements. RAOs in pitch and heave are determined by using the average wave height from MS2 and MS3 sensors and the maximum measured pitch or heave within the corresponding clean-wave period.

RAOs for pitch and heave were determined for two floating wind turbine configurations: i) the baseline case with no wind or motor cables present; ii) a steady wind case with the rotor spinning at 12 rps, equivalent to a wind generating 1.09 N of thrust, corresponding, at prototype scale, to 901.26 kN of thrust achieved at

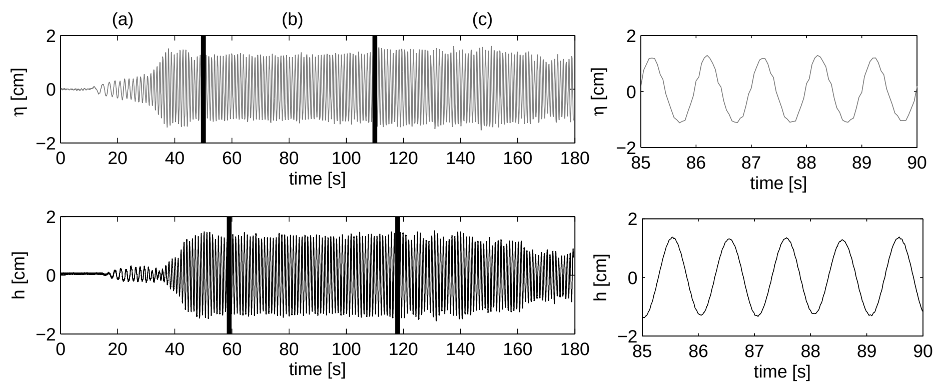


Figure 66: Depiction of the clean wave window with (a) transitional wave period, (b) clean wave window and corresponding quasi-steady-state response in heave, h, and (c) reflective waves present. Top figures show water elevation η as measured at -8D, while bottom figures depict heave, h

approximately 6 m/s wind speed. This condition was chosen because i) the resulting thrust had relatively low variance, as seen in figure 57; ii) the rotor spinning in at such intermediate range of rps value, safely allows periodic thrust variations at frequencies similar to the incident wave's ones, to simulate wind gusts.

Baseline and constant thrust RAO

Waves are varied from $T_w = [0.91 - 3.00]$ s with H between 0.96 and 3.41 cm. The dynamic response of the model platform in pitch and heave due to wave (and wind) forcing consists of the RAOs shown in figure 67. Table 9 describes all the experimental runs, each one representing a point in the RAO map, for the baseline turbine case. The results indicate a strong response in both pitch and heave near their respective resonant frequencies.

Heave: Baseline turbine results in heave are shown as black circles in (a) of figure 67. The heave natural frequency is 1.10 Hz ($\omega_d = 1.09$ Hz) for the baseline turbine. The heave RAO has a peak response of 1.21 at $T_w = 1.01$ s, which is fairly consistent with the period of 0.91 s obtained from the free decay tests. Heave response quickly drops for frequencies higher than the peak heave frequency. Lower frequencies result in a near unity response in heave with a drop off beginning near the pitch peak at $T_w = 2.48$ s. It is important to note that the heave natural frequency is near the typical wave periods obtained by the PNW data, with the most common waves between $T_w = 0.8$ s and $T_w = 1.59$ s, causing, at full scale, heave amplitudes of 1.0 and 2.0 m, respectively. While such near-resonant conditions would be avoided in the design of a platform, the overall response showed heave RAO values limited to the interval between 1 and 1.2 for a relatively wide range of wave periods. Resonant conditions in pitch represent a much more serious threat for the platform lifetime.

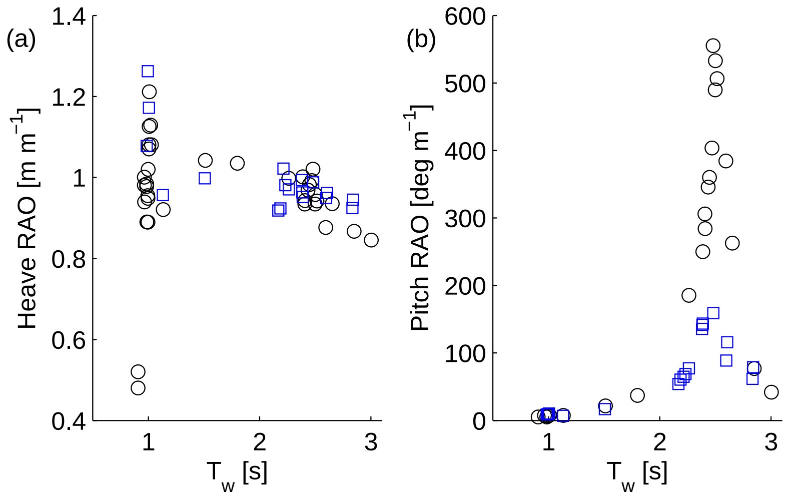


Figure 67: Response Amplitude Operator for (a) Heave and (b) Pitch
$\circ$: baseline turbine configuration, $\square$: constant wind (cables) $\Omega_r = 12$ rps ($F_{thrust} = 1.09$ N)

Pitch: The RAO in pitch for the baseline turbine is shown as black circles in (b) of figure 67. The pitch response exhibits a much larger excitation near its natural frequency ($T_w = 2.45$ s), as compared with the heave response. The RAO shows a peak response at $T_w = 2.48$, very close to the 2.45 s period obtained from pitch decay tests. Pitch response quickly diminishes at wave periods above and below the resonant period. Platform pitch response to typical waves, $T_w = [0.8 - 1.59]$s, is < 1 deg, which is a very desirable response. Wave periods near the resonant frequency can induce large pitch amplitudes i.e. 4-5 deg from relatively small waves (1-2 cm in height at laboratory scale). Note that the RAO peak value of 553.5 deg m^{-1}, representing the platform response to wave heights of the order of centimeters, cannot be scaled up to prototype scale, as the RAO is non-linear with the wave height within the two order of magnitude difference between wave basin and ocean data. Figure 67 also shows the RAOs for the case with steady, simulated wind corresponding to 1.09 N of thrust. Results for pitch and heave are shown as blue squares in (a) and (b) respectively. Heave peak response occurs at $T_w = 1.00$ s with a peak value of 1.26. Peak response in pitch occurs at $T_w = 2.48$ s with a value of 159.1 deg m^{-1}. As compared to the baseline wind turbine, the addition of wind and cables results in a similarly shaped RAO with pitch amplitude being significantly diminished, consistent with the free decay test results, showing a decrease in the damping coefficients ζ but limited effect on the natural frequency ω_d (figures 64 and 65). As was discussed in §6.7.1, this is predominantly an effect of the motor cables, not of the simulated wind or the related non-zero averaged pitch due to the steady effect of the wind thrust force.

T_w [s]	H [cm]	H_{rms} [cm]	Heave RAO [m m^{-1}]	Pitch RAO [deg m^{-1}]
3.00	0.96	0.07	0.85	41.97
2.85	1.09	0.17	-	76.94
2.65	0.99	0.10	0.94	262.84
2.60	1.26	0.04	-	384.29
2.52	1.43	0.09	-	506.53
2.50	1.42	0.10	-	532.85
2.50	1.45	0.11	-	498.62
2.48	1.45	0.15	-	555.18
2.47	1.37	0.11	0.99	403.58
2.45	1.39	0.10	-	360.02
2.43	1.37	0.09	-	345.76
2.41	1.37	0.09	-	305.95
2.41	1.41	0.07	-	284.33
2.39	1.43	0.09	1.00	249.97
2.26	1.19	0.08	1.00	185.26
1.80	2.22	0.07	1.04	37.18
1.51	1.94	0.05	1.04	21.50
1.13	2.84	0.09	0.92	7.38
1.03	2.42	0.10	1.08	-
1.01	2.43	0.09	1.21	8.12
1.00	2.90	0.11	0.95	6.58
0.99	2.90	0.10	-	5.13
0.96	2.77	0.12	0.98	7.40
0.91	3.41	0.12	0.47	5.15

Table 9: Pitch and Heave RAOs for baseline wind turbine system

6.7.3 Variable rotor thrust

As first introduced in §6.5.4 a subset of experiments was performed to investigate the effect of a fluctuating aerodynamic thrust on system dynamics. In these experiments the rotor speed was varied sinusoidally with periods of 5 and 10 seconds and amplitude of $\pm 10\%$ and $\pm 20\%$ from a mean value of ≈ 12 rps. Using the curve in fig 57(a), the above $\pm 10\%$ and $\pm 20\%$ Ω_r variations correspond to thrust oscillation in the range of [0.86 - 1.36] N and [0.66 to 1.65] N respectively, simulating repeated wind gusts. The oscillation period of the rotor speed variation was chosen to be consistent with the vortex shedding period exerted by an hypothetically upwind turbine generating a meandering wake. Recent experimental work suggested that this instability is well captured within a limited range of Strouhal number St, which is a dimensionless parameter describing a bluff body wake:

$$St = \frac{fD}{V_\infty}, \tag{88}$$

where f is the vortex shedding frequency, D is the characteristic length (rotor diameter in this case), and V_∞ is the free stream, or incoming flow velocity. In the near wake behind a model wind turbine, where measurements were made one diameter downstream, [126] suggests a Strouhal number ranging from 0.1 to 0.3. Far wake measurements in a study performed on a hydro-kinetic turbine by [59], and computational modeling by [2], gave a Strouhal number of approximately 0.28. Most recently, measurements in the far wake of a model wind turbine found a Strouhal number of 0.23 for instabilities associated with the rotor [115]. These values were further confirmed in a wind tunnel experiments with miniature turbines [117]. The vortex shedding frequency at the prototype scale is determined by solving equation 88 for f and using $D = 200$ m, $V_\infty = 8.7$ m s^{-1} which is the approximate average wind speed during the year at elevation of $z = 119.5$ m from the PNW environmental data, and $St = 0.23$. This gives $f \approx 0.01$ Hz which corresponds to a ≈ 100 s oscillation period of wake meandering induced gusts at prototype scale. The scaled vortex shedding period for the wave experiments resulted in ≈ 10 s, which defined the period of the rotor speed oscillations.Wave periods for the variable thrust tests were chosen to investigate a broad spectrum of platform responses. Because the clean wave window is shortened under long waves, the rotor oscillation period was also set to 5s to allow for a longer representative dataset of wind-wave forcing. (see Table 10 for a list of investigated wave periods and varying rotor speed configurations).

Two cases with varying wind and wave forcing are highlighted here: they are characterized by simulated wind variation, along a 5s period, with $\pm 10\%$ amplitude variation, corresponding to wind fluctuations of approximately 2 m/s at prototype scale. The two cases have similar wave height but different wave periods, respectively chosen as $T_w = 2.21$ s, and $T_w = 0.83$ s to investigate the platform motion both near and far from the pitch natural frequency. Figures 69 and 68 show the platform pitch energy spectral densities normalized by the pitch variance (S_ϕ/σ^2) for the two wave periods, comparing the steady (baseline) case with the variable wind thrust. Under

Wave Period [s]	Variable Rotor Speed
2.52	±20 % at 10 s
2.21	±10, 20 % at 5 and 10 s
1.51	±20 % at 10 s
0.99	±10, 20 % at 10 s
0.82	±10, 20 % at 5 s

Table 10: Summary of variable thrust test cases, indicating the peak to peak % variation of the rotor speed (wind gust amplitude) and the period of the imposed amplitude modulation (wind gust period)

steady wind and $T_w = 0.83$s period waves, the platform pitch energy is concentrated at the wave frequency, 1.21 Hz, producing a dominant peak with some energy represented at the natural frequency, 0.37 Hz. In contrast, the pitch spectra associated with fluctuating thrust (dot-dashed red), subject to the same incident waves, exhibit a dominant peak at the wind gust frequency (0.2 Hz corresponding to a 5s period) with very small contributions at the wave and natural frequencies. Figure 69 shows S_ϕ/σ^2 for the $T_w = 2.21$ s period waves. As with the $T_w = 0.83$s case, the dominant frequency is the wave frequency (at 0.45 Hz) for the baseline case, while for the variable thrust case at this wave frequency, two equivalent peaks are detected: one in the proximity of the wave and natural frequencies, and a second one at the wind gust frequency. From the two cases, we learn that despite of the addition of fluctuating wind, wave periods near the pitch natural frequency ω_n maintain a significant contribution to platform pitch oscillations. However, when the wave period is far from ω_d, thrust fluctuations may dominate the pitching motion. In general, under both waves and wind, the results of the pitch RAO can be used to predict the dominant contribution in the spectral platform pitch response; in fact, the platform at the natural frequency can be exited by one or the other source of forcing, depending on how close the two frequencies are to ω_n, and obviously on the amplitude of the wind gust fluctuations as compared to the wave height. In figure 70 we point at the increased variability in platform pitch motions under variable thrust force. In this specific case, the wind thrust was varied with a 10s period with amplitude of ±20% in angular velocity, while the waves had a period of 2.21 s with a height of approximately 1.2cm. This is an environmental condition where pitch is expected to be mostly dominated by wind variability; however it leads to a wide range of platform pitch angles. Because of the potentially different periods and phases of the wave and wind forcing, the platform experiences indeed a wide range of oscillations, with instantaneous pitch governed by the superimposition of the two forcing mechanisms. Note that we did not compensate for the constant wind thrust using ballast mass distribution, so the pitch platform varies around a negative value $\phi \simeq 2.3$ which represents the equilibrium pitch conditions of the platform under constant wind without waves.

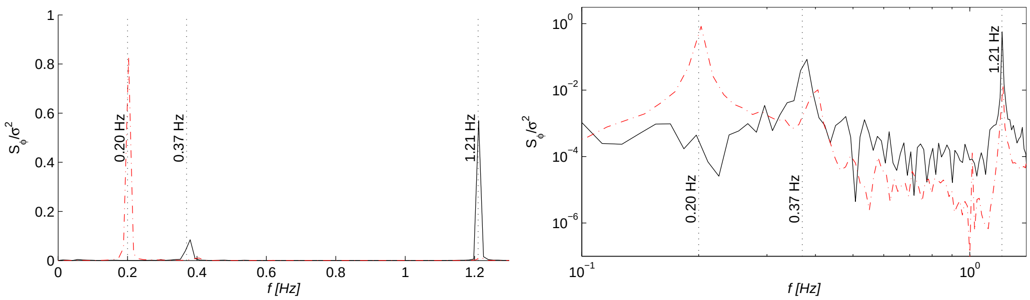


Figure 68: Power spectral density of the platform pitch S_ϕ under waves of period $T_w = 0.83$ s with static thrust (solid black) and variable thrust: $F_{thrust} = [0.86 - 1.36]$ N at 5s period (dot-dash red); linear axis in the left panel, and logarithmic axis in the right panel.

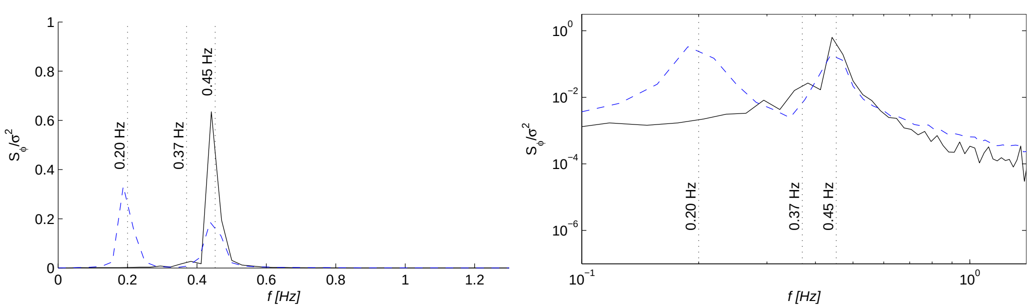


Figure 69: Power spectral density of the platform pitch S_ϕ under waves of period $T_w = 2.21$. Steady and variable thrust conditions are the same of figure 68; ; linear axis in the left panel, and logarithmic axis in the right panel.

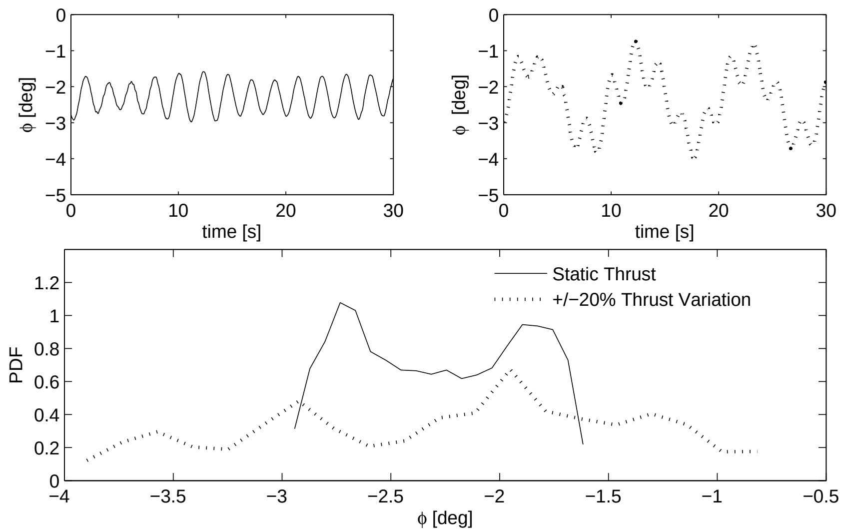


Figure 70: Time history samples of the platform pitch ϕ under constant or fluctuating wind thrust and resulting probability distribution of ϕ. Note that with constant wind but without waves the platform equilibrium position is at $\phi = -2.3$ deg.

6.8 Phase II: wind tunnel results

6.8.1 Wind turbine kinematics based on specific wave conditions

The response amplitude operator (RAO) generated in the wave basin study is used to predict the wind turbine motion response due to a known wave period, T_w, and wave height, H. A specific point, with respect to T_w, was chosen on the RAO to replicate the predicted motion of the miniature turbine in the wind tunnel (see the solid circle in Figure 71). The main factor behind the choice of this incident wave period was to observe the platform responding to both pitch and heave motions. In particular, under the specific conditions defined by the RAO in pitch, the limiting pitch amplitude of $\phi_{amp} = 4$ degrees is expected for a relatively large wave, characterized by $H = 0.1$m and $T_w \simeq 2$s, at the wave basin scale. The RAO for heave allows us to estimate vertical oscillations of the model platform of approximately 0.11m, corresponding to a heave amplitude $h_{amp} = 0.055$ m, at the wave basin scale, = 5.5 m at prototype scale, and 3.5 mm at wind tunnel scale ($h_a mp \approx 0.03 z_{hub}$). Essentially, we are reproducing in the wind tunnel the prototype floating turbine under the effects of $\simeq$ 10 m wave exciting both heave and pitch motions. Such high waves were dictated by the imposed pitch amplitude $\phi_{amp} = 4$ chosen to investigate the floating turbine wake under extreme environmental conditions, i.e. when the combination of wave and unsteady wind forcing may lead to structural failure or over tipping. The wind turbine motion investigated in the wind tunnel here are the following:
i) *static* TSR_{fr}: Static turbine with frictional torque, $TSR_{fr} = 4.5$
ii) *pitch* TSR_{fr}: Pitch only with oscillating frequency at 3 Hz and pitch amplitude

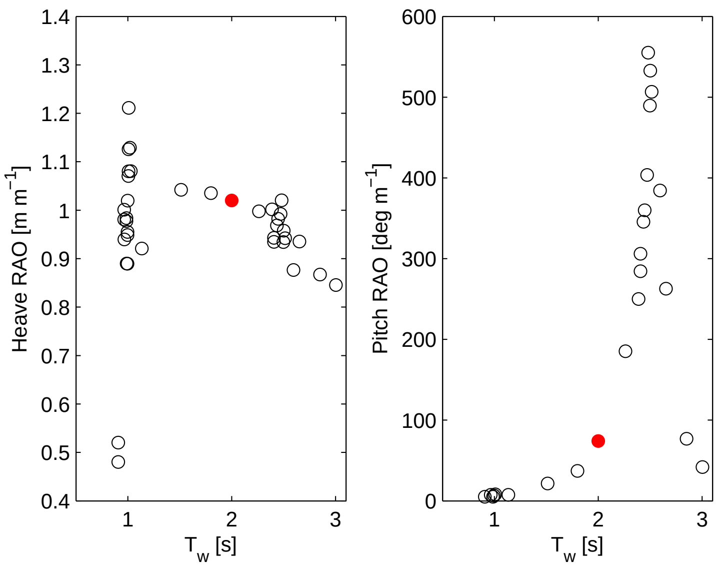


Figure 71: Response amplitude operator (RAO) for heave (left) and pitch (right). Solid circle indicates the point chosen for wind tunnel testing

$\phi_{amp} = \pm 4$ degrees, $TSR = 4.5$
iii) *heave* TSR_{fr}: Heave only with oscillating frequency at 3 Hz, and heave amplitude $h_{amp} = \pm 0.03 z_{hub}$, $TSR = 4.5$
iv) *static* TSR_{opt}: Static turbine at optimal tip speed ratio, $TSR = 3$
v) RAO: Coupled heave and pitch according to the chosen configuration in the RAO, with oscillating frequency at 3 Hz and optimal TSR

Under the RAO investigated conditions the turbine oscillations were defined by a pitch amplitude $\phi_{amp} = \pm 4$ degrees, a heave amplitude $h_{amp} = \pm 0.03 z_{hub}$, and a phase shift (obtained by wave basin experiments) defined by $\phi(t)$ leading $h(t)$ by $0.34T$, where T is the turbine oscillation period. Separating the heaving and pitching motions and comparing to the full RAO response would give insight into which degree of freedom would most contribute to large scale variability, if any, in the mean and fluctuating velocity statistics of the turbine wake. Table 11 summarizes the test cases performed in the wind tunnel as well as the employed flow measurement technique. The static turbine, pitch only, and heave only cases are compared, for the same $TSR = 4.5$, to study which type of platform motion has a greater effect on the aerodynamic wake. The static turbine and RAO case, where the optimal $TSR = 3$ was used, compares a static turbine to an oscillating turbine in both pitch and heave. This case is of significant importance for verifying computational models, providing insight into any coupling effects of the two oscillating motions in the wake, assessing if wake models developed for land-based turbines can be used for turbine siting in offshore wind power plants.

case	f_w [Hz]	ϕ_{amp} [deg]	h_{amp}	TSR	measurement method
static TSR_{fr}	0	0	0	4.5	PIV, hotwire
pitch TSR_{fr}	3	± 4	0	4.5	PIV, hotwire
heave TSR_{fr}	3	0	$\pm 0.03 z_{hub}$	4.5	PIV
static TSR_{opt}	0	0	0	3	PIV, hotwire
RAO TSR_{opt}	3	± 4	$\pm 0.03 z_{hub}$	3	PIV, hotwire

Table 11: Summary of platform pitch control test cases.

6.8.2 Mean streamwise flow profiles

Figure 72 shows the vertical profile of the average streamwise velocity $U(z)$ from hotwire measurements, normalized by U_{hub}, for the different turbine operating conditions in the wake centerplane at $x/D = 5$. Results are shown for the static turbine cases at optimal and frictional tip speed ratios, for the pitch only case (at $TSR = 4.5$), and for the RAO case (at $TSR = 3$). The results show similar velocity deficit for all turbine cases, outlining however the dominant role played by the turbine tip speed ratio as opposed to its pitching and heaving motion. Specifically, at optimal TSR the static and RAO case, show an increased velocity deficit as compared with the frictional torque cases due to increased energy capture. Under the same optimal tip speed ratio $TSR = 3$, minimal differences are observed between the static and the RAO cases. To enhance the comparison we define the velocity deficit as $\Delta U(z) = U_{case}(z) - U_{base}(z)$ where $U_{case}(z)$ is the average streamwise velocity for a specific turbine case, as outlined in table 11, and $U_{base}(z)$ is the average streamwise velocity of the incoming boundary-layer (baseline). We first acknowledge that slight differences in the free stream velocity $U(\delta)$ accounts for approximately a 1% difference of incoming hub velocity U_{hub} between the two cases. In the wake near the top-tip, $z/z_{hub} = [1.5 - 2]$, however, we record an appreciable difference of $\approx 4\%$ of U_{hub} between static and the RAO case with a slight increase in velocity deficit for the oscillating turbine. Near the hub and in the range $z/z_{hub} = [0.5 - 1.5]$, differences are again comparable with those in the inflow conditions. In the region of the bottom-tip ($z/z_{hub} = [0.2 - 0.5]$), an increase in the velocity deficit for the RAO case is again observed ($\approx 2\%$ of U_{hub}). This increased variability in the wake mean flow near the top-top and bottom-tip are likely attributed to the dynamic location of the rotor tips due to the pitching and heaving motions. Aside of this weak effects, minimal differences between the turbine cases are also observed in the mean streamwise velocity measurements at z_{hub} and $z_{top-tip}$ for varying distance to the rotor ($x/D = 1, 2, 3, 4,$ and 5) as depicted in Figure 73. Again, the only appreciable differences under the same tip speed ratio are concentrated at the top tip for the static vs. heaving and pitching turbine.

We switch now to the mean velocity evolution in the wall parallel plane $U(x, y)$, as measured by PIV. In figure 74, spanwise profiles of U/U_{hub} are shown at hub height ($z = z_{hub}$) for the frictional torque cases: static turbine (black solid line), pitch

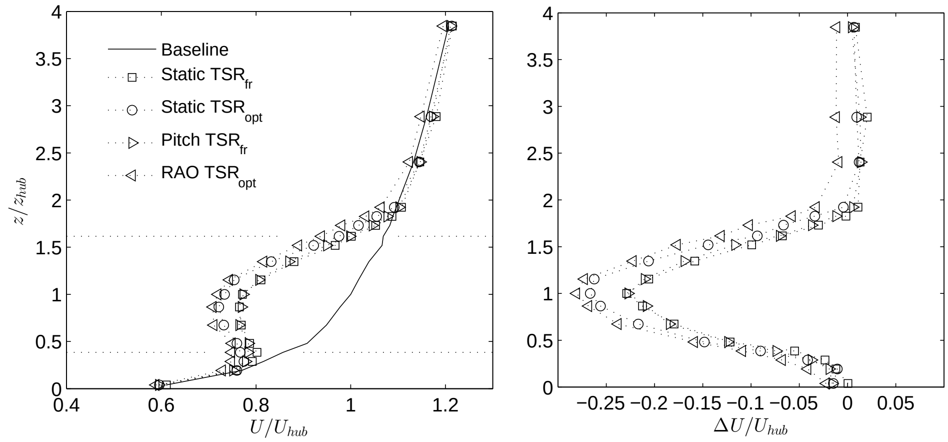


Figure 72: Vertical profile of average streamwise velocity $U(z)$ normalized by U_{hub} in the wake measured at $x/D = 5$ and $y/D = 0$ for different turbine operating configurations. Horizontal dotted lines represent the vertical rotor extension between top and bottom tips. Baseline - no turbine present. On the right panel: velocity deficit profile $\Delta U/U_{hub}$.

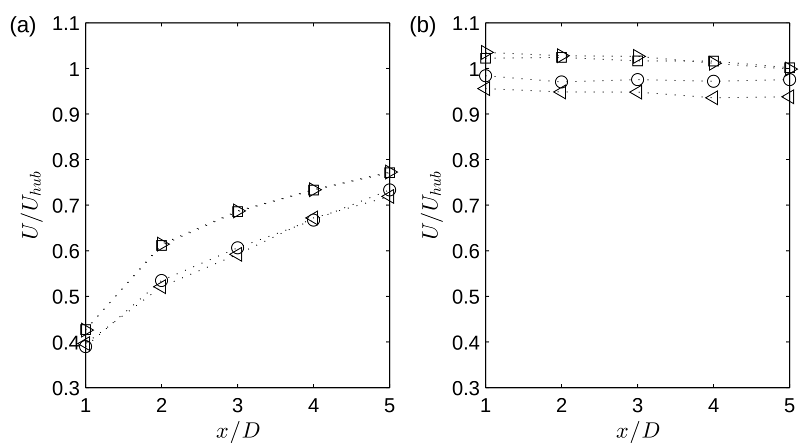


Figure 73: Streamwise profile of average streamwise velocity U normalized by U_{hub} in the wake measured at z_{hub} (a) and $z_{top-tip}$ (b) for $x/D = 1, 2, 3, 4,$ and 5 and $y/D = 0$. Symbols used are the same as those for figures 72.

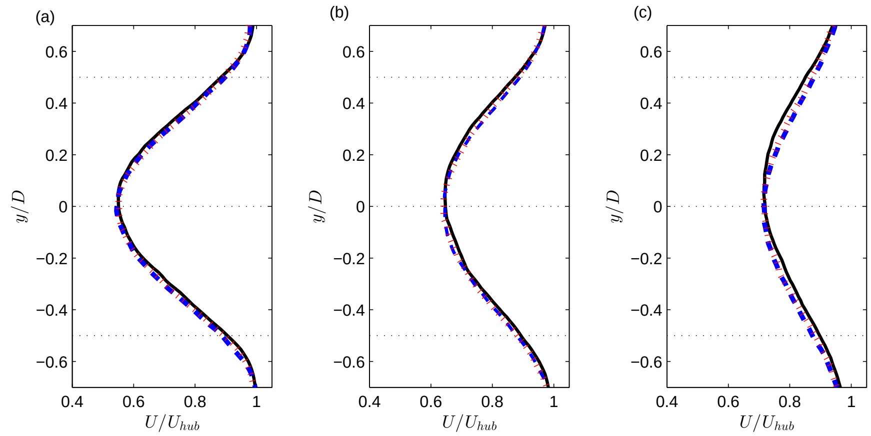


Figure 74: Spanwise profiles of average streamwise velocity $U(y)$ normalized by U_{hub} in the wake measured at (a) $x/D = 2$, (b) $x/D = 3$, (c) and $x/D = 4.5$ at the $z = z_{hub}$ plane for turbine operating configurations with $TSR = 4.5$. Horizontal dotted lines represent the lateral tip locations and the hub axis. Turbine cases shown are the static case (solid black line), pitch only case (dashed blue line), and heave only case (dot-dashed red line).

only (dashed blue line), and heave only (dot-dashed red line) at (a) $x/D = 2$, (b) $x/D = 3$, and (c) $x/D = 4.5$. No appreciable distinction can be observed between the pitch only and heave only cases for the investigated x locations. At $x/D = 4.5$ we observe a mild increase in wake symmetry (with respect to the x-axis) for the moving turbine cases as opposed to the static turbine, even though differences are smaller than 2% of U_{hub}. The asymmetric velocity distribution of the static case is consistent with the rotational direction of the rotor, bringing high momentum fluid in the wake for $y > 0$ (see, for the same turbine model [117]). Measurements suggests that the turbine oscillatory motion has a mild effect on the wake spatial evolution, specifically increasing the wake symmetry as a result of increase mixing in the near wake region close to the blade tip.

6.8.3 Streamwise velocity fluctuations

The vertical profile of the streamwise velocity fluctuations, estimated as the standard deviation of the wind velocity component u_{rms}, is presented in the dimensionless turbulence intensity form $I_u = \frac{u_{rms}}{U_{hub}}$ in Figure 75, for $x/D = 5$ and $y = 0$. With the turbine, a large increase in turbulence intensity, with respect to the baseline flow, is seen in correspondence of the top-tip shear layer, for all cases. As for the mean velocity profiles, weak differences are again observed when comparing the static case to an oscillating case, with variations in u_{rms} of less than 2% in the rotor area. However, a relatively larger variation is observed just above the top-tip location with an increase in u_{rms} of 8% for the RAO case as compared to the static turbine. Figure

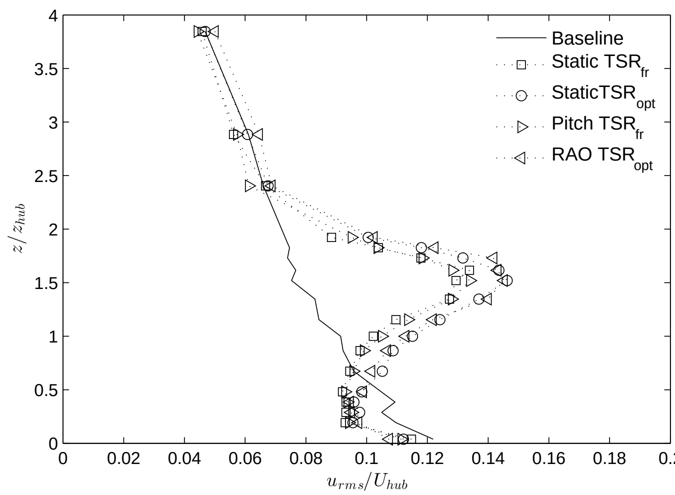


Figure 75: Vertical profile of streamwise turbulence intensity, u_{rms}/U_{hub}, in the wake measured at $x/D = 5$ and $y = 0$. Baseline case indicates measurements with no turbine present.

76 shows the streamwise evolution of u_{rms} at $x/D = 1, 2, 3, 4$, and 5 at $y = 0$ and (a) $z = z_{hub}$ and (b) $z = z_{top-tip}$. At the top-tip location, the RAO motion case shows an increase in u_{rms} of 3% over the static turbine with optimal tip speed ratio at $x/D = 3$ and 4. Along the hub axis, figure 76 (a), results are more complex to interpret as the oversized hub has a large influence on the wake structures in the near wake ($x/D < 2$).

Overall, both streamwise mean and fluctuating velocity statistics show weak differences between static and oscillating cases under the same tip speed ratio. A maximum difference of 8% in u_{rms} and 3% in U is observed in the proximity of the top tip, which is likely a result of the dynamic top-tip location of the oscillating turbine. In other words, the turbine wake oscillate with respect to the fixed hotwire position, which remains periodically in or out of the wake, thus perceiving an appreciable variability in the streamwise velocity field.

6.8.4 Turbine phases statistics

The above results seem to indicate that the wake of an offshore wind turbine could be modeled as the wake of a land based turbine, with minimal differences expected in the incoming flow of downwind units. However basic drag mechanisms associated with the varying relative velocity between the wind and the rotor suggest that both turbine motion phases, or position states and direction of travel in heave and/or pitch, should affect velocity statistics. For example, a pitching turbine into the wind will impinge the flow creating an increased velocity deficit whereas a turbine pitching with the flow will reduce the velocity deficit. To explore such dependencies, streamwise velocity statistics will be investigated through conditional averages on turbine motion phases (figure 59). Specifically, for pitch oscillations, the most relevant phases based

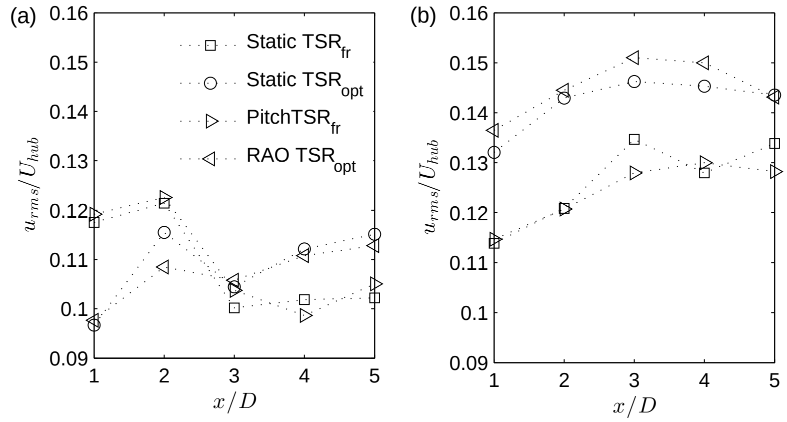


Figure 76: Streamwise profiles of turbulence intensity, u_{rms}/U_{hub}, in the wake measured at $x/D = 1, 2, 3, 4$, and 5 at $y = 0$ and (a) $z = z_{hub}$ and (b) $z = z_{top-tip}$.

on their effect on the wake flow are:

– positive direction of travel in pitch, $+d\phi/dt$, and positive values of pitch, $+\phi$ (positive phase)

– negative direction of travel in pitch, $-d\phi/dt$, and negative values of pitch, $-\phi$ (negative phase).

while for heave we have:

– positive direction of travel in heave, $+dh/dt$, and positive values of heave, $+h$ (positive phase)

– negative direction of travel in heave, $-dh/dt$, and negative values of heave, $-h$ (negative phase).

Instantaneous velocity measurements, $u(t)$, from PIV and hotwire results are binned into subsets based on the turbine motion and phase. In particular for the pitch only case, two categories are defined, termed $u_{\phi+}$ for positive pitch phase and $u_{\phi-}$ for the negative one, while for the heave only case, we have u_{h+} and u_{h-}, respectively (The RAO case includes binning based on heave and pitch separately). Statistics are then performed on each bin of instantaneous velocities for each motion and phase, leading to U_ϕ averages conditioned on pitch, U_h averages based on heave, as well as to $u_{rms,\phi}$ and $u_{rms,\phi}$ for pitch and heave conditioned standard deviation of the velocity fluctuations, respectively. To easy the comparative analysis, for each measurement position x_i, we computed the difference between the phase average velocity $U_\xi(x_i)$, depending on the specific turbine motion ξ (e.g. RAO case), and the overall average $U(x_i)$:

$$\Delta U_\xi(x_i) = U_\xi(x_i) - U(x_i), \tag{89}$$

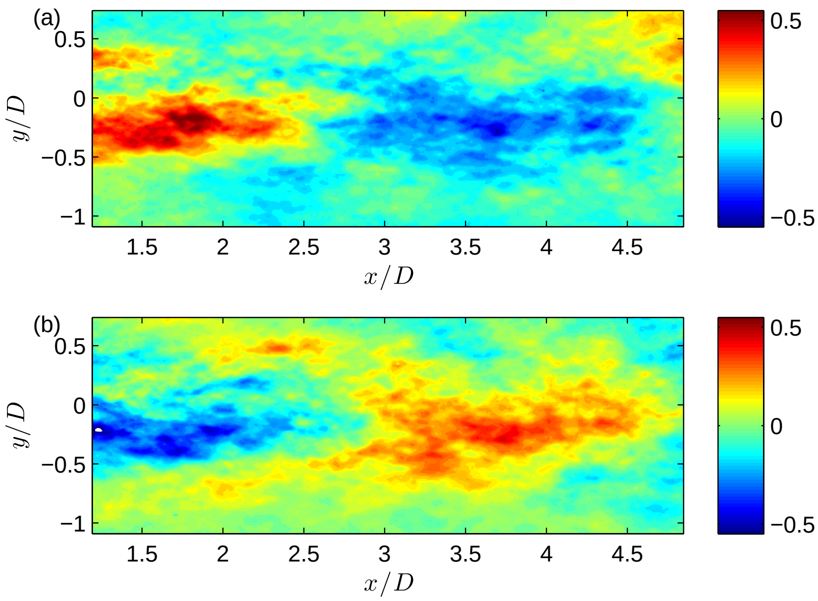


Figure 77: Contours of phase average streamwise velocity difference, $\Delta U_{\phi}(x, y)/u_{rms}$, for pitch only case at $z = z_{hub}$ plane, (a) positive phase, (b) negative phase. ΔU_{ϕ} is normalized by $u_{rms}(x = 5D, y = 0, z = z_{hub})$ of the static turbine.

where ξ represents either pitch, ϕ, or heave, h. The phase average difference is used to provide any insight of large scale periodic variation in the mean flow. $\Delta U_{\xi}(x_i)$ is non-dimensionalized by $u_{rms}(x = 5D, y = 0, z = z_{hub})$ of the static turbine in order to compare such turbine motion-induced oscillations with the turbulence intensity observed in the wake of the static turbine (under the same TSR operating condition). The same normalization applies to $u_{rms}(x = 5D, y = 0, z = z_{hub})$.

Figures 77, 78, 79, and 80 show the contours of the phase average streamwise velocity difference, $\Delta U_{\xi}(x, y)$, from wall-parallel PIV at the hub plane for the cases summarized in table 11. In each of the figures, (a) gives the positive phase and (b) the negative phase of the turbine position. When pitch oscillations contribute to the turbine motion, i.e. pitch-only and RAO (figures 77, 79, and 80), clear, opposing trends appear in the spatial distribution of the streamwise velocity between turbine phases: a pocket of velocity reduction is seen in figure 77 (a) located in the range $x/D = [3 - 4.5]$ with a peak value of $(-)30 - 40\%$ of static turbine u_{rms}. The same flow region is characterized by a velocity increase of $(+)30-40\%$ when conditioned on the negative phase (figure 77b). These oscillations represent phase specific large scale velocity fluctuations that are essentially averaged out in the unconditioned statistics. The phase averaged velocity evolution in the wake can be also observed in figure 81, representing streamwise cuts from the contour plots for $\Delta U_{\xi}(x)$ at $y = 0$ (hub axis). The opposing effects of positive and negative pitching motions is evidenced in subfigures (a), (c), and (d) which correspond to the pitch-only and RAO cases,

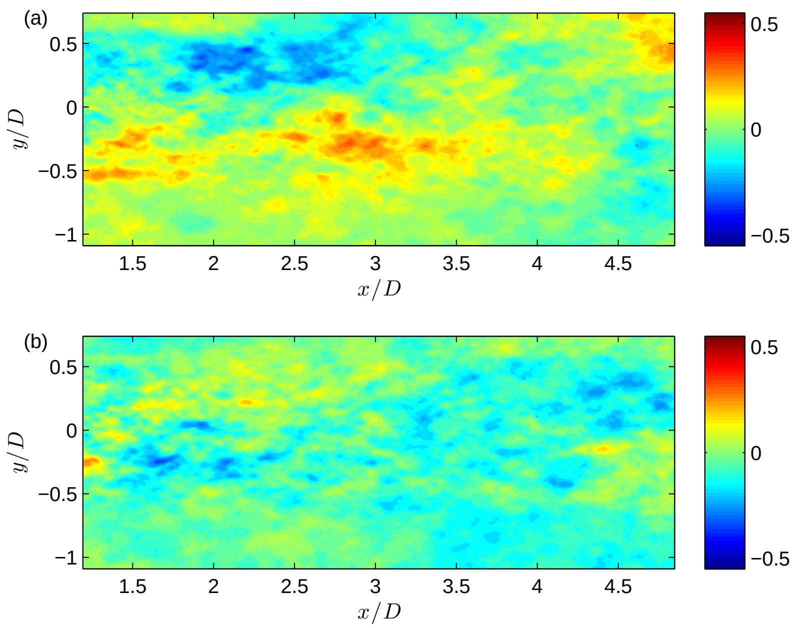


Figure 78: Contours of phase average streamwise velocity difference, $\Delta U_h(x, y)/u_{rms}$, for heave only case at $z = z_{hub}$ plane, (a) positive phase, (b) negative phase. ΔU_h is normalized by $u_{rms}(x = 5D, y = 0, z = z_{hub})$ of the static turbine.

respectively. Figures 78 and 81(b) illustrates $\Delta U_h(x, y)$ for the heave only case, suggesting that, in the hub plane, the heaving motion alone has limited effect on the velocity in the hub plane. It is inferred that the turbine wake in the two-degree of freedom RAO case is governed mostly by pitching motion, as the difference in $\Delta U_\xi(x, y)$, reported in figures 79 and 80, is due to the temporal lag between the pitch and heave conditions.

Vertical profiles of ΔU_ϕ at $x/D = 5$ and streamwise profiles of ΔU_ϕ at $z = z_{hub}$ or $z_{top-tip}$, are computed from hotwire measurements and reported in figures 82 and 83, respectively. Because ΔU_ϕ are phase and measurement location specific, the temporal window used to inspect the flow field for specific turbine motion and position impacts the location and magnitude of the low-high momentum velocity pockets generated by the oscillating rotor. These pockets represent large scale oscillations of the streamwise velocity with a 2D (x and z planes) spatial structure, which propagate in the turbine wake like a spatio-temporal wave. Therefore, depending on the measurement location and turbine phase, different wave amplitudes will be recorded. The lengthscale of the wave, is estimated from figures 81 and 83 as $L \approx 3 - 4D$, consistent with a temporal scale of 0.33s (dictated by 3HZ oscillations) and an average convection velocity of $0.5 \div 0.7U_{hub}$, which is consistent with the velocity deficit in the turbine wake. The vertical profiles shown in figure 82 highlight the influence of the coupled motion for heave and pitch at $x/D = 5$. Figure 82 (a) gives the results of ΔU_ϕ for the pitch only

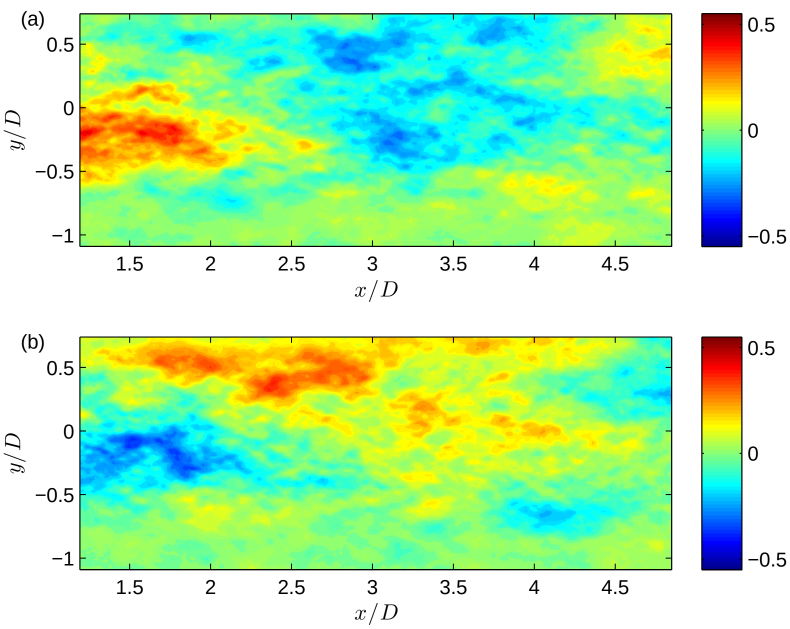


Figure 79: Contours of phase average streamwise velocity difference, $\Delta U_\phi(x, y)/u_{rms}$, for RAO case with pitch conditioning at $z = z_{hub}$ plane, (a) positive phase, (b) negative phase. ΔU_ϕ is normalized by $u_{rms}(x = 5D, y = 0, z = z_{hub})$ of the static turbine.

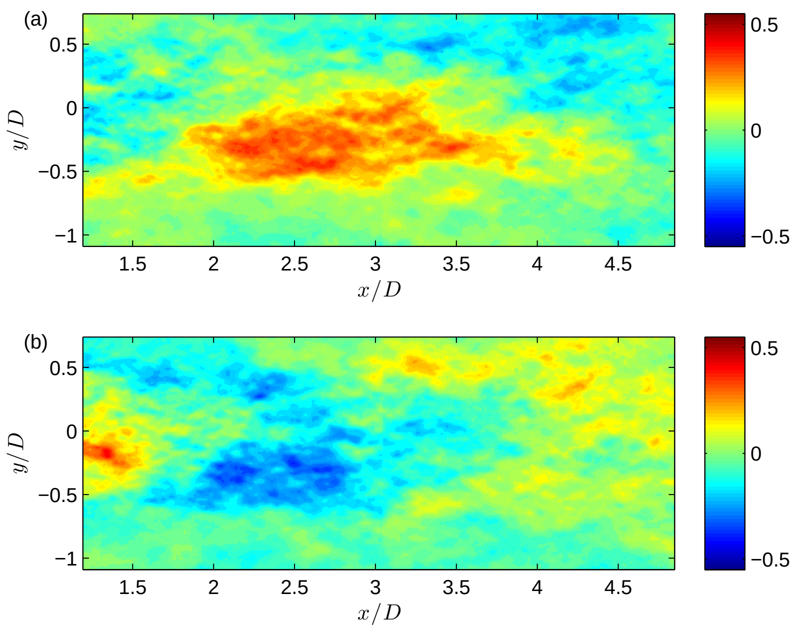


Figure 80: Contours of phase average streamwise velocity difference, $\Delta U_h(x, y)/u_{rms}$, for RAO case with heave conditioning at $z = z_{hub}$ plane, (a) positive phase, (b) negative phase. ΔU_h is normalized by $u_{rms}(x = 5D, y = 0, z = z_{hub})$ of the static turbine.

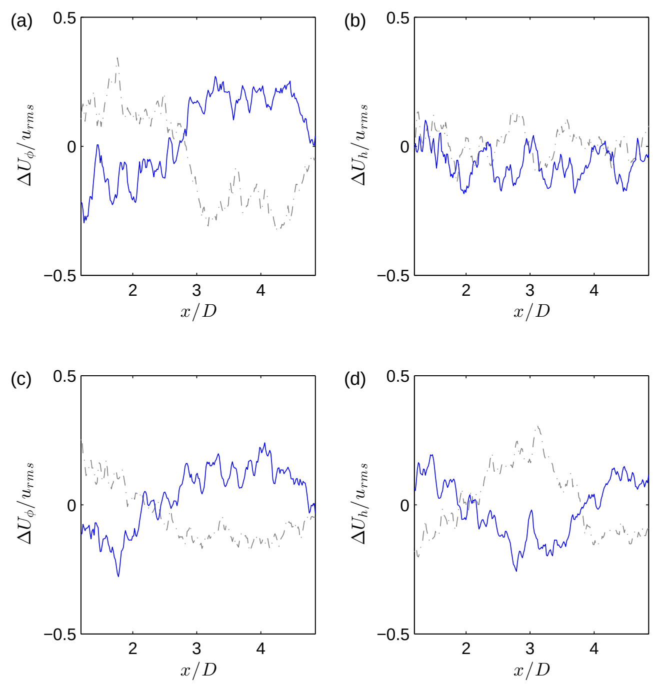


Figure 81: Phase average streamwise velocity difference, streamwise profile at $z = z_{hub}$ and $y = 0$, $\Delta U_\xi(x)/u_{rms}$ for (a) pitch only $\xi = \phi$, (b) heave only $\xi = h$, (c) RAO $\xi = \phi$, and (d) RAO $\xi = h$. ΔU_ξ is normalized by $u_{rms}(x = 5D, y = 0, z = z_{hub})$ of the static turbine. Solid blue line represents positive phase, dash-dot gray line negative phase.

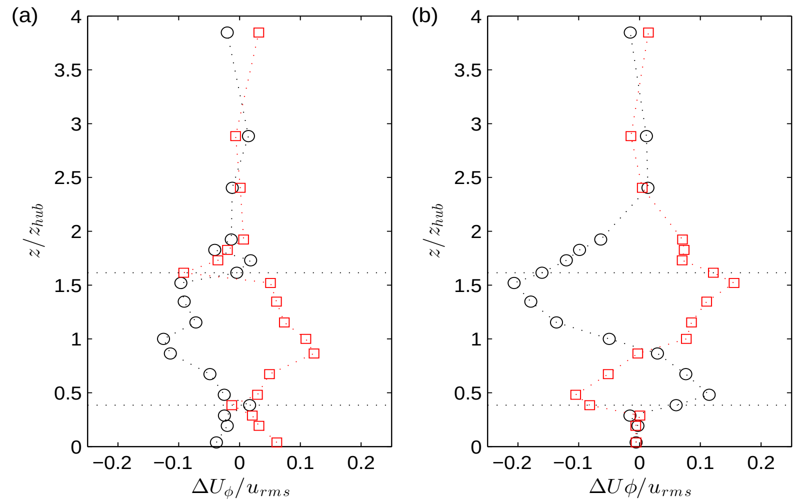


Figure 82: Phase average streamwise velocity difference, vertical profile at $x/D = 5$ and $y = 0$, $\Delta U_\phi(z)/u_{rms}$ for (a) pitch only and (b) RAO (pitch conditioning). ΔU_ϕ is normalized by $u_{rms}(x = 5D, y = 0, z = z_{hub})$ of the static turbine. ∘: positive phase, □: negative phase

case. The largest variation in velocity between the two phases is observed near z_{hub}. In contrast, the RAO case shown in figure 82 (b) has minimal velocity variations near z_{hub} and largest values near the rotor tips, $z_{bottom-tip}$ and mostly $z_{top-tip}$, suggesting a potential amplification of pitch-induced oscillation by the heaving motion near the tips.

As such organized, large-scale velocity fluctuation, with a significant amplitude, is observed in the phase averaged statistics, it is legitimate to ask why a similarly appreciable increase in u_{rms} was missing in the moving turbine wake statistics, as compared to the static turbine. The phase-average difference approach is thus applied to the u_{rms} velocity using a similar definition, for each case

$$\Delta u_{rms,\xi} = u_{rms,\xi} - u_{rms}, \tag{90}$$

where $u_{rms,\xi}$ is the phase specific rms velocity associated with the negative phase and positive phase defined above, and u_{rms} is the unconditioned r.m.s. velocity. $\Delta u_{rms,\xi}$ is also normalized by the $u_{rms}(x = 5D, y = 0, z = z_{hub})$ of the static turbine in the corresponding, frictional torque or optimal tip speed ratio, operating conditions.

The phase specific $\Delta u_{rms,\phi}$ profiles are given in figures 84 and 85. As was observed in the profiles of ΔU_ϕ, opposite values of $\Delta u_{rms,\phi}$ are associated with the positive and negative phases in both the vertical and streamwise profiles. These fluctuations in $\Delta u_{rms,\phi}$ result in a canceling effect on the overall unconditioned u_{rms} and accounts for the minimal differences between the static and oscillating turbine u_{rms} profiles. Comparing the positive phase value of ΔU_ϕ in figure 82 (b) and the positive phase value of $\Delta u_{rms,\phi}$ in figure 84 (b), a clear trend can be observed. Negative values of

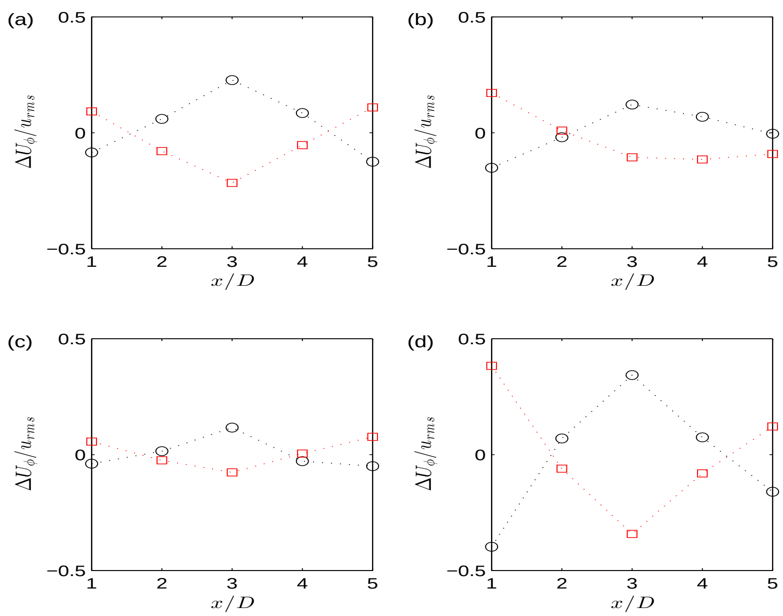


Figure 83: Phase average streamwise velocity difference, streamwise profiles at $y = 0$. $\Delta U_\phi(x)/u_{rms}$ for (a) pitch only $z = z_{hub}$, (b) pitch only $z = z_{top-tip}$, (c) RAO $z = z_{hub}$, and (d) RAO $z = z_{top-tip}$. ΔU_ϕ is normalized by $u_{rms}(x = 5D, y = 0, z = z_{hub})$ of the static turbine. ○: positive phase, □: negative phase

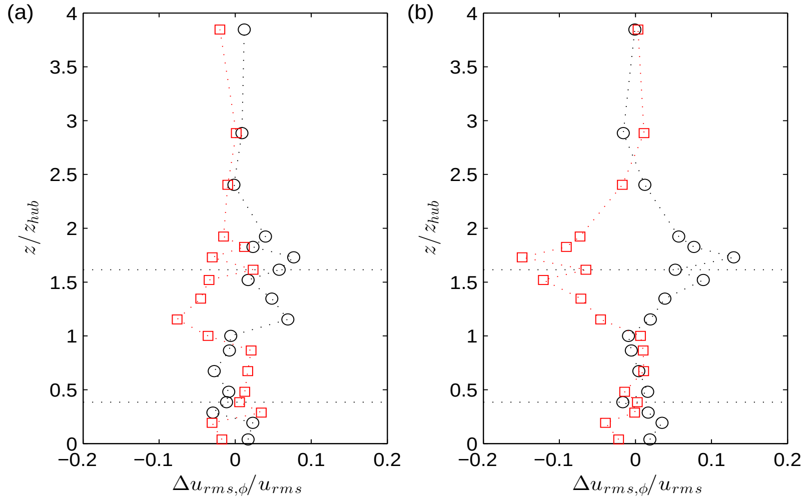


Figure 84: $\Delta u_{rms,\phi}$ vertical profile at $x/D = 5$ and $y = 0$, (a) pitch only and (b) RAO (pitch conditioning). $\Delta u_{rms,\phi}$ is normalized by $u_{rms}(x = 5D, y = 0, z = z_{hub})$ of the static turbine. ○: positive phase, □: negative phase

ΔU_ϕ correspond to a positive $\Delta u_{rms,\phi}$, most notably shown near the top-tip location. Likewise, positive values of ΔU_ϕ are associated with a negative $\Delta u_{rms,\phi}$, or a decrease in r.m.s. velocity. This same trend is observed in the negative phase statistics as well as in the streamwise profiles shown in figures 83 and 85. In summary, when the turbine is pitching into the incoming wind the mean velocity in the wake decreases while the turbulence increases, viceversa, when the turbine is pitching with the wind, the mean velocity increases and the turbulence decreases. The period of this wave depends on the turbine motion periods, while its lengthscale depends on a convection velocity comparable with the peak velocity deficit at hub height.

6.9 Summary and Conclusions

The general dynamics of offshore, floating wind turbines have been explored through a quasi-coupled wind wave experimental investigation. In the first phase of this work, the motion of an offshore floating wind turbine under varying sinusoidal wave forcing has been defined by two degrees of freedom, heave and pitch, employing a barge-type platform and a turbine model in the SAFL wave testing facility; in a second phase, specific turbine-platform motions from the wave experiment were scaled and replicated in the SAFL wind tunnel using an actuated miniature wind turbine model immersed in a fully developed turbulent boundary layer. The effect of wind on the floating platform dynamics has been implemented in the wave experiment by imposing a constant, or variable, thrust at the turbine hub by controlling the angular velocity of the rotor with opportunely pitched blades. The effect of the waves on the spatial evolution of the turbine wake, was implemented in the wind tunnel by heaving and

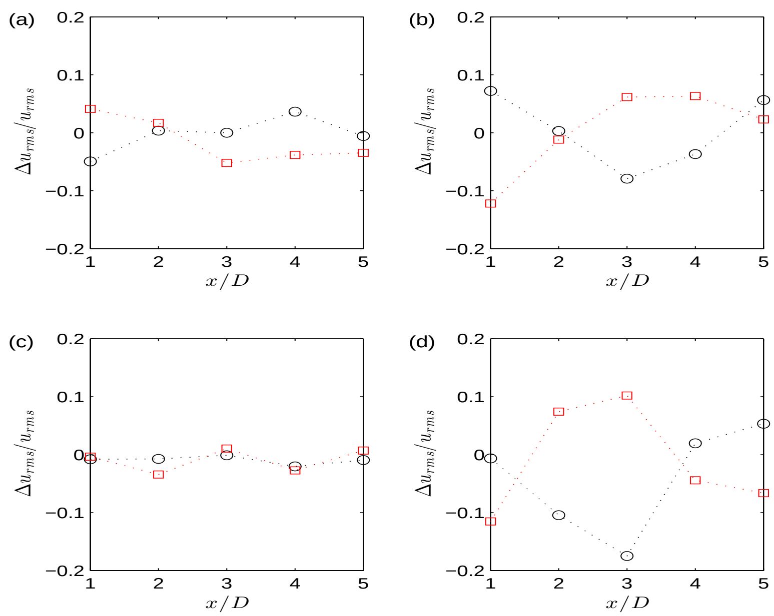


Figure 85: $\Delta u_{rms,\phi}$ streamwise profiles at $y = 0$ for (a) pitch only $z = z_{hub}$, (b) pitch only $z = z_{top-tip}$, (c) RAO $z = z_{hub}$, and (d) RAO $z = z_{top-tip}$. $\Delta u_{rms,\phi}$ is normalized by $u_{rms}(x = 5D, y = 0, z = z_{hub})$ of the static turbine. $\circ$: positive phase, $\square$: negative phase

pitching the miniature turbine model. The novelty of this approach is highlighted in the following scaling arguments, allowing the investigation of a utility scale turbine under realistic sea waves and wind conditions. The important dimensionless numbers are:

- The critical Strouhal number for turbine wake meandering, in the literature, is in the range $St = fD/U_{hub} = 0.2 \div 0.3$. Choosing a critical value in the St range, e.g. $C_{St} = 0.23$, allows to derive a wind gust frequency $f_g = \frac{U_{hub} C_{St}}{D}$, where U_{hub} is the mean velocity at hub height and D is the rotor diameter.

- The Froude number for floating structure in wave environments is defined as $Fr = C/\sqrt{(gL)}$, where L is a characteristic length scale, chosen here as the platform barge diameter, g is the acceleration of gravity and C is the wave celerity; for deep water waves, mostly investigated here, $C = gT/2\pi$ depending only on the wave period T. By fixing the prototype platform and the significant wave period from sea buoy data, the Froude number is defined and the wave frequency can be then derived as $f_w = \frac{g^{1/2}}{Fr\ 2\pi\ L^{1/2}}$.

The wave tank and wind tunnel experiments are designed to reproduce key physical mechanisms at different but interconnected scales: waves must have a large enough amplitude to avoid surface tension effects, while genuine boundary layer turbulence requires a long enough fetch to develop a realistic mean shear before impacting the turbine model. These constraining factors, including the requirements to have statistically representative wind and wave conditions, were accounted for in the experimental and scaling strategies described in the following:

1. The offshore floating wind turbine is a 13.2MW concept designed by SNL under significant wave conditions recorded from a buoy offshore the Oregon coast.

2. Prototype to wave basin scaling adopts a geometric scale factor of 1:100 and Froude number similarity. Geometry and mass distributions of the SNL turbine were used to scale the moment of inertia of the tested platform-turbine system, ensuring that the stabilizing (buoyancy) and destabilizing (gravity) moments were consistent with the scaled wave amplitudes and periods.

3. Incident wave characteristics at different locations with respect to the platform, and platform kinematics were all measured simultaneously to generate a response amplitude operator (RAO) for heave and pitch motions, representing the response of the system under waves of different period. Results are consistent with free decay test for initial, given, pitch and/or heave displacements.

4. Wind data from the offshore site were Froude number scaled to provide, at the laboratory scale, a comparable destabilizing moment due to the thrust force

applied on the nacelle. The thrust force was then varied periodically to simulate the effect of wind gusts. In particular, to simulate the effect of upwind floating turbines in an offshore power plant, the critical Strouhal number for wake meandering was used to derive the period of the thrust force (simulated wind gust) oscillations.

5. The recorded heave and pitch motions of the platform in the wave tank were reproduced in the wind tunnel by actuating a miniature turbine model. Heave amplitudes were scaled geometrically, while pitch angles were not changed. The rotor oscillation period and the hub velocity were chosen to keep the ratio between the wake meandering period and the incident wave period invariant.

6. The geometrical scaling ratio in the wind tunnel was chosen to keep the miniature turbine in the 25% of the boundary layer height, as typically occurring for utility-scale turbines in the atmospheric boundary layer. As an added benefit of using a miniature turbine model, the turbine wake spatial evolution up to 6D was accessible to hotwire and PIV measuring techniques.

A number of limitations in this experimental approach must be acknowledged: 1) In the wave testing facility the turbine has never been exposed to a realistic wave spectrum, but only to single runs of sinusoidal waves with different periods and amplitudes. 2) The wind and the wave propagation directions were always assumed to be aligned, which may not be the case in many geophysical conditions. 3) The baseline turbulent conditions in the wind tunnel does not reflect the wave induced roughness, as the wind tunnel floor was not actuated or wavy, but hydraulically smooth. 4) The miniature turbine has several limitations on its own; specifically, low power coefficient ($C_p = 0.27$), no operating control aside of the frictional vs optimal tip speed ratio, fixed pitch blade, and relatively large nacelle as compared to the rotor diameter ($\sim 9\%$). We also acknowledge that the vastity of the parameter space (wave amplitude, period, wind magnitude and wind gust intensity) was only minimally explored in both phases of the experiments; however the control and monitoring of the boundary conditions, the quality of the measurements and the experimental and scaling strategy developed here allow new experiments and numerical simulations to fill the parameter space. The benchmark dataset presented here is being used to validate numerical simulations of fluid structure interaction [127] to assess predictive capabilities for offshore wind energy.

The major results of this work are summarized in the following points:

- The effect of a steady wind on the kinematic response of the floating turbine system is dynamically negligible, with the peaks at the heave and pitch natural frequencies unchanged. The only appreciable difference is on the shift of the angular displacement range due to a change in the equilibrium position (vertical turbine without wind vs. inclined turbine under steady wind), which can be easily mitigated. Under gusty wind conditions, i) the interaction between wind and wave forcing increases the range of pitch angles, depending on the respective amplitudes, i.e. wave height and wind gust intensity; ii) the local platform dynamics is controlled by the continuously varying phase marking a constructive/destructive interference between the two forcing mechanisms.

- Mean and fluctuating velocities in the wake of the oscillating turbine only show weak differences, mostly concentrated near the top tip region, with respect to the land based static turbine. However, this is the result of opposing trends in the mean and fluctuating velocities based on turbine motion phases. Large wake fluctuations created by the turbine pitching motion are described by a cycle of low and high uniform momentum regions, scaling with the rotor and traveling in the wind direction as a two dimensional wave. The wavelength is derived as a function of the the platform pitch period T and a convection velocity comparable to the maximum velocity deficit at hub height $U_{hubwake} \simeq 0.6 \div 0.7 U_{hub}$, likely depending on the tip speed ratio and turbine operating regime. The amplitude of this wave at a distance $x/D = 5$ was estimated as 30-40% of the r.m.s. velocity in the same wake location of the static turbine case. While such a velocity fluctuation appears to be smaller than the turbulent fluctuation of the incoming flow for land base turbine, we stress that the structure and the spatial coherence of such fluctuation is relevant. The high and low velocity regions, periodically generated by the oscillating rotor, scale with the rotor diameter in all three spatial coordinates, so they do represent a large scale motion that will be perceived by downwind turbines. This phase locked variation of the mean flow induce an opposite phase variation of the turbulent intensity. Specifically, when the turbine pitches into the wind, the velocity deficit in the wake increase (low momentum pocket) but the r.m.s. velocity increase (high turbulence); viceversa when the turbine pitches with the wind the mean velocity in the wake increases (high momentum pocket) and the turbulence decreases. Both contributions average out if no phase averages are computed.

- The analysis of decoupled heave and pitch motions suggest that, for the wake flow, heave contribution is only weakly relevant near the blade tips, in particular the top tip, while pitch contribution is mostly responsible for the large scale flow variability in the wake.

- The coupling between the wave and wind dynamic forcing destabilizing floating turbines is established by recognizing wake meandering as a the major source of wind variability in an offshore wind power plant. As wake meandering oscillations govern the far wake ($x/D > 2$) of a turbine, they represent the typical incoming wind condition for downwind turbines, (typically located at $x/D > 5$). Hence in order to investigate the combined effect of wind gusts and waves, it is suggested here to keep the ratio between the two time scales (or frequencies f_g and f_w) invariant while reproducing field scale observations by numerical simulation. We can define an offshore wind energy number, as a governing parameter in offshore wind turbine siting, $OW = \frac{f_g}{f_w} = \frac{g^{1/2} D}{U_{hub} C_{St} 2\pi L^{1/2}}$ (function of the wavelength L , hub velocity U_{hub} , rotor diameter D, and critical Strouhal number C_{St}). OW defines whether the wind and waves will have a separated or similar periodic forcing, the latter enhancing constructive interference, e.g. both contributing to destabilize the platform for more stretched periods of time. Note that OW only accounts for the forcing, without including the natural frequency of the oscillating system, which is marking the peak response in the RAO diagrams. Obviously the worst conditions for operating the offshore wind power plant occurs when the three frequencies are equal $f_w = f_g = f_N$ and both wake meandering and waves are contributing to force the oscillating system at resonant conditions, with potentially catastrophic consequences.

Based on the above results, we can attempt to provide some guidelines on the design strategy for floating turbine siting in an offshore wind power plant: the power production is expected to fluctuate with the platform pitch, but to be, on average, aligned with the prediction based on the mean velocity at hub height, thus consistent with resource assessment practice for land based turbine. However, in terms of unsteady loads and turbine component lifetime, floating turbines present new challenges. Assuming that the wake of an upwind turbine evolves into the inflow condition for a downwind unit, we predict that floating turbines within the array will be exposed to large scale flow variability imposed by the pitching of upwind rotors, which will result in unsteady loads propagating from the blades, to the low and high speed shafts, turbine support tower, and platform. As opposed to the case of random wind gusts, however, these unsteady loads are periodic and predictable, so an opportune control algorithm can be designed to act on variable torque and individual blade pitch, based on the real-time input of platform pitch sensors.

The relevance of an offshore-specific wind turbine control is further motivated by the increased pitch experienced by the floating turbine system under variable wind. We acknowledge that the imposed fluctuations in the thrust force, tested during wave tank phase 1 experiments, and corresponding to specific wave gust intensities, were not strictly based on field measurements. However, from Phase 2 experiments in the wind tunnel, we quantified the largest pitch-induced velocity difference in the range of $\pm 40\%$ $u_{rms} \approx \pm 4\%$ U_{hub}. Therefore, the observed increased variability in platform pitch, under $\pm 10, 20\%$ oscillations of the thrust force is likely overestimating the induced gusts expected in the field, unless the collective behavior of multiple, pitching, upwind turbines could result in the amplification of the single wake flow variability. In the absence of a pitch mitigation strategy, the cyclic low/high momentum regions induced by upwind pitching rotors may impose an oscillating thrust force on downwind turbines, resulting in increased pitch, potentially augmented by in-phase wave forcing, and/or by approaching the platform natural frequency. Such an unfortunate sequence of superimposed effects might increase significantly both structural loads, large scale variability of the wake flow, and the pitch response. Both above arguments suggest that pitch mitigation strategy and offshore specific control algorithms should be implemented before utility scale deployment.

6.10 Publications and Presentations

6.10.1 Publications

Feist C. Aerodynamic wake study: oscillating model wind turbine within a turbulent boundary layer. Master Thesis. 2015.

C. Feist; A. Calderer, K. Ruehl; F. Sotiropoulos; and M. Guala. Platform kinematics and wake evolution of a floating wind turbine: a quasi coupled wind-wave experimental study. Ready for submission in the Journal of Fluid Mechanics.

6.10.2 Presentations

Feist, F.; Ruehl, K.; Guala, M. Scaling and kinematics of a floating wind turbine under ocean waves and variable thrust: an experimental study. 66th Annual Meeting of the APS Division of Fluid Dynamics 58 (18). 2013.

Feist C. Aerodynamic wake study: oscillating model wind turbine within a turbulent boundary layer. Master Thesis Defense, June 2015.

7 Public Release of the Code with Documentation

The code has been released under the GNU General Public License (GPL 2.0) in the following Github repository:

https://github.com/SAFL-CFD-Lab/VFS-Wind

The released version of the code is capable of simulating:

- Wind turbines either in geometry-resolving mode or parameterized with a range of actuator models;
- Complex terrain topographies for site-specific wind farm simulations;
- Atmospheric turbulent wind conditions;
- Floating offshore wind turbines via a two-phase flow formulation accounting for broadband ocean waves and 6 degrees of freedom (DOF) FSI.

In addition to the code, the repository contains a detailed users manual (include in the end of the report) with all the information necessary for start using the code. In the manual, first, a description is given in how to install the libraries required for the code and how to compile and execute the code. Then, instructions are given on how to post-process and visualize the resulting data. Then, all the elements (input variables, input mesh files, and the control options files) for preparing a simulation test case are introduced. We also give an overview of the structure of the code, with explanation of the main subroutines, and a code flow chart to understand the steps followed by the code upon execution. Finally we present a set of 10 case studies designed to facilitate the learning of the various aspects of the code in a comprehensive manner.

The case studies include:

1. 3D Sloshing in a Rectangular Tank
2. 2D VIV of an Elastically Mounted Rigid Cylinder
3. 2D Falling Cylinder (Prescribed Motion)
4. 3D Heave Decay Test of a Circular Cylinder
5. 2D Monochromatic Waves
6. 3D Directional Waves
7. Floating Platform Interacting with Waves
8. Clipper Wind Turbine
9. Model Wind Turbine Case
10. Channel Flow

The files of the test case studies are included in the Github repository.

7.1 Project team

Fotis Sotiropoulos James L. Record Professor, Director of the St. Anthony Falls Laboratory. Principal investigator of the project.

Antoni Calderer PhD student in the Saint Anthony Falls Laboratory. Mr. Calderer worked on the turbine geometry resolving model (subtask 1.1) and the process of preparing the code for release.

D. Todd Griffith Principal Member of the Technical Staff in the Wind and Water Power Technologies Department at Sandia National Laboratories.

Thomas Herges Post-doctoral researcher at Sandia National Laboratories. Dr. Herges assisted in the preparation of the code for release. Supervised the process of preparing the code for release.

Ann Dallman Post-doctoral researcher at Sandia National Laboratories. Dr. Dallman assisted in the preparation of the code for release.

Kelley Ruehl Ocean Renewable Energy Engineer at Sandia National Labs. Mrs. Ruehl is the coordinator from Sandia National Labs, providing her expertise in ocean engineering to the project, both in the numerical and experimental side. Performed simulations using engineering level models.

References

[1] X. Yang, F. Sotiropoulos, R. J. Conzemius, J. N. Wachtler, and M. B. Strong, "Large-eddy simulation of turbulent flow past wind turbines/farms: the virtual wind simulator (vwis)," *Wind Energy*, 2014.

[2] S. Kang, X. Yang, and F. Sotiropoulos, "On the onset of wake meandering for an axial flow turbine in a turbulent open channel flow," *Journal of Fluid Mechanics*, vol. 744, pp. 376–403, 2014.

[3] J.-Y. Yang, S.-C. Yang, Y.-N. Chen, and C.-A. Hsu, "Implicit weighted {ENO} schemes for the three-dimensional incompressible navierstokes equations," *Journal of Computational Physics*, vol. 146, no. 1, pp. 464 – 487, 1998.

[4] K. H. Lee, *Responses of floating wind turbines to wind and wave excitation*. Thesis, Massachusetts Institute of Technology, 2005. Thesis (S.M.)–Massachusetts Institute of Technology, Dept. of Ocean Engineering, 2005.

[5] E. N. Wayman, *Coupled dynamics and economic analysis of floating wind turbine systems*. PhD thesis, Massachusetts Institute of Technology, 2006.

[6] F. Sandner, D. Schlipf, D. Matha, R. Seifried, and P. W. Cheng, "Reduced nonlinear model of a spar-mounted floating wind turbine," 2012.

[7] F. G. Nielsen, T. D. Hanson, and B. Skaare, "Integrated dynamic analysis of floating offshore wind turbines," in *25th International Conference on Offshore Mechanics and Arctic Engineering*, pp. 671–679, American Society of Mechanical Engineers, 2006.

[8] T. Zambrano, T. MacCready, T. Kiceniuk Jr, D. G. Roddier, and C. A. Cermelli, "Dynamic modeling of deepwater offshore wind turbine structures in gulf of mexico storm condition," in *Proceedings of OMAE2006*, (Hamburg, Germany), 2006.

[9] J. M. Jonkman, "Dynamics of offshore floating wind turbinesmodel development and verification," *Wind Energy*, vol. 12, no. 5, pp. 459–492, 2009.

[10] K. Kleefsman, G. Fekken, A. Veldman, B. Iwanowski, and B. Buchner, "A volume-of-fluid based simulation method for wave impact problems," *Journal of Computational Physics*, vol. 206, pp. 363–393, June 2005.

[11] S. Tanaka and K. Kashiyama, "ALE finite element method for FSI problems with free surface using mesh re-generation method based on background mesh," *International Journal of Computational Fluid Dynamics*, vol. 20, pp. 229–236, Mar. 2006.

[12] K.-J. Paik, P. M. Carrica, D. Lee, and K. Maki, "Strongly coupled fluid–structure interaction method for structural loads on surface ships," *Ocean Engineering*, vol. 36, no. 17, pp. 1346–1357, 2009.

[13] J. Sanders, J. E. Dolbow, P. J. Mucha, and T. A. Laursen, "A new method for simulating rigid body motion in incompressible two-phase flow," *International Journal for Numerical Methods in Fluids*, vol. 67, no. 6, pp. 713–732, 2010.

[14] L. Shen and E.-S. Chan, "Numerical simulation of fluidstructure interaction using a combined volume of fluid and immersed boundary method," *Ocean Engineering*, vol. 35, pp. 939–952, June 2008.

[15] S. Kang and F. Sotiropoulos, "Numerical modeling of 3D turbulent free surface flow in natural waterways," *Advances in Water Resources*, vol. 40, pp. 23–36, May 2012.

[16] S. Kang, A. Lightbody, C. Hill, and F. Sotiropoulos, "High-resolution numerical simulation of turbulence in natural waterways," *Advances in Water Resources*, vol. 34, pp. 98–113, Jan. 2011.

[17] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, "A dynamic subgridscale eddy viscosity model," *Physics of Fluids A: Fluid Dynamics*, vol. 3, pp. 1760–1765, July 1991.

[18] S. J. Osher and R. P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces.* Springer, 2003 ed., Oct. 2002.

[19] A. Iafrati and E. F. Campana, "Free-surface fluctuations behind microbreakers: spacetime behaviour and subsurface flow field," *Journal of Fluid Mechanics*, vol. 529, pp. 311–347, 2005.

[20] A. Iafrati, "Numerical study of the effects of the breaking intensity on wave breaking flows," *Journal of Fluid Mechanics*, vol. 622, p. 371, Feb. 2009.

[21] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *J. Comput. Phys.*, vol. 79, pp. 12–49, Nov. 1988.

[22] M. Sussman and E. Fatemi, "An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow," *SIAM J. Sci. Comput.*, vol. 20, pp. 1165–1191, Feb. 1999.

[23] I. Borazjani, L. Ge, and F. Sotiropoulos, "Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies," *Journal of Computational Physics*, vol. 227, pp. 7587–7620, Aug. 2008.

[24] I. Borazjani, *Numerical Simulations of Fluid-Structure Interaction Problems in Biological Flows.* PhD thesis, University of Minnesota, 2008.

[25] A. Gilmanov and F. Sotiropoulos, "A hybrid cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies," *Journal of Computational Physics*, vol. 207, pp. 457–492, Aug. 2005.

[26] L. Ge and F. Sotiropoulos, "A numerical method for solving the 3D unsteady incompressible NavierStokes equations in curvilinear domains with complex immersed boundaries," *Journal of Computational Physics*, vol. 225, pp. 1782–1809, Aug. 2007.

[27] I. Borazjani and F. Sotiropoulos, "Vortex induced vibrations of two cylinders in tandem arrangement in the proximity-wake interference region," *Journal of fluid mechanics*, vol. 621, pp. 321–364, 2009. PMID: 19693281.

[28] W. Cabot and P. Moin, "Approximate wall boundary conditions in the large-eddy simulation of high reynolds number flow," *Flow, Turbulence and Combustion*, vol. 63, no. 1-4, pp. 269–291, 2000.

[29] M. Wang and P. Moin, "Dynamic wall modeling for large-eddy simulation of complex turbulent flows," *Physics of Fluids*, vol. 14, no. 7, 2002.

[30] J.-I. Choi, R. C. Oberoi, J. R. Edwards, and J. A. Rosati, "An immersed boundary method for complex incompressible flows," *Journal of Computational Physics*, vol. 224, pp. 757–784, June 2007.

[31] E. M. Yettou, A. Desrochers, and Y. Champoux, "Experimental study on the water impact of a symmetrical wedge," *Fluid dynamics research*, vol. 38, no. 1, pp. 47–66, 2006.

[32] A. Iafrati, A. Babanin, and M. Onorato, "Modulational instability, wave breaking, and formation of large-scale dipoles in the atmosphere," *Physical Review Letters*, vol. 110, no. 18, 2013.

[33] J. Hunt, A. Wray, and P. Moin, "Eddies, stream, and convergence zones in turbulent flows," *Center For Turbulence Research*, vol. Report CTR-S88, 1988.

[34] A. ANSYS, "Version 15.0; ansys," *Inc.: Canonsburg, PA, USA November*, 2013.

[35] K. Ruehl, C. Michelen, S. Kanner, M. Lawson, and Y.-H. Yu, "Preliminary verification and validation of wec-sim, an open-source wave energy converter design tool," in *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*, pp. V09BT09A040–V09BT09A040, American Society of Mechanical Engineers, 2014.

[36] M. Lawson, Y.-H. Yu, R. Kelley, C. Michelen, *et al.*, "Development and demonstration of the wec-sim wave energy converter simulation tool," 2014.

[37] C. Feist, A. Calderer, K. Ruehl, F. Sotiropoulos, and M. Guala, "Platform kinematics and wake evolution of a floating wind turbine: a quasi coupled wind-wave experimental study," *Journal of Fluid Mechanics, Submitted*, 2015.

[38] "Wec-sim web page." http://wec-sim.github.io/WEC-Sim/, 2015. Accessed: 2015-10-22.

[39] K. H. Jung, K. A. Chang, and H. J. Jo, "Viscous effect on the roll motion of a rectangular structure," *Journal of engineering mechanics*, vol. 132, no. 2, pp. 190–200, 2006.

[40] A. Calderer, S. Kang, and F. Sotiropoulos, "Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures," *Journal of Computational Physics*, vol. 277, pp. 201–227, 2014.

[41] "National data buouy center," 2015.

[42] D. T. Griffith and T. D. Ashwill, "The sandia 100-meter all-glass baseline wind turbine blade: Snl100-00," *Sandia National Laboratories, Albuquerque, Report No. SAND2011-3779*, 2011.

[43] A. Robertson, J. Jonkman, M. Masciola, H. Song, A. Goupee, A. Coulling, and C. Luan, "Definition of the semisubmersible floating system for phase ii of oc4," *Offshore Code Comparison Collaboration Continuation (OC4) for IEA Task*, vol. 30, 2012.

[44] C. Jensen, L. Damkilde, and R. R. Pedersen, "Numerical simulation of gyroscopic effects in ansys," *Aalborg University, Denmark*, 2011.

[45] D. Yang and L. Shen, "Simulation of viscous flows with undulatory boundaries. part i: Basic solver," *Journal of Computational Physics*, vol. 230, pp. 5488–5509, June 2011.

[46] D. Yang and L. Shen, "Simulation of viscous flows with undulatory boundaries: Part II. coupling with other solvers for two-fluid computations," *Journal of Computational Physics*, vol. 230, pp. 5510–5531, June 2011.

[47] X. Guo and L. Shen, "On the generation and maintenance of waves and turbulence in simulations of free-surface turbulence," *Journal of Computational Physics*, vol. 228, pp. 7313–7332, Oct. 2009.

[48] X. Yang, X. Zhang, Z. Li, and G.-W. He, "A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations," *Journal of Computational Physics*, vol. 228, no. 20, pp. 7821–7836, 2009.

[49] Z. Du and M. S. Selig, "A 3-d stall-delay model for horizontal axis wind turbine performance prediction," *AIAA Paper*, vol. 21, 1998.

[50] W. Z. Shen, R. Mikkelsen, J. N. Sørensen, and C. Bak, "Tip loss corrections for wind turbine computations," *Wind Energy*, vol. 8, no. 4, pp. 457–475, 2005.

[51] W. Z. Shen, J. N. Sørensen, and R. Mikkelsen, "Tip loss correction for actuator/navier–stokes computations," *Journal of Solar Energy Engineering*, vol. 127, no. 2, pp. 209–213, 2005.

[52] L. P. Chamorro and F. Porté-Agel, "A wind-tunnel investigation of wind-turbine wakes: boundary-layer turbulence effects," *Boundary-layer meteorology*, vol. 132, no. 1, pp. 129–149, 2009.

[53] Y.-T. Wu and F. Porté-Agel, "Large-eddy simulation of wind-turbine wakes: evaluation of turbine parametrisations," *Boundary-layer meteorology*, vol. 138, no. 3, pp. 345–366, 2011.

[54] L. P. Chamorro and F. Porté-Agel, "Turbulent flow inside and above a wind farm: a wind-tunnel study," *Energies*, vol. 4, no. 11, pp. 1916–1936, 2011.

[55] M. Wang and P. Moin, "Dynamic wall modeling for large-eddy simulation of complex turbulent flows," *Physics of Fluids (1994-present)*, vol. 14, no. 7, pp. 2043–2051, 2002.

[56] S. Kang and F. Sotiropoulos, "Flow phenomena and mechanisms in a field-scale experimental meandering channel with a pool-riffle sequence: Insights gained via numerical simulation," *Journal of Geophysical Research: Earth Surface (2003–2012)*, vol. 116, no. F3, 2011.

[57] X. Yang, S. Kang, and F. Sotiropoulos, "Computational study and modeling of turbine spacing effects in infinite aligned wind farms," *Physics of Fluids (1994-present)*, vol. 24, no. 11, p. 115107, 2012.

[58] Y.-T. Wu and F. Porté-Agel, "Simulation of turbulent flow inside and above wind farms: Model validation and layout effects," *Boundary-layer meteorology*, vol. 146, no. 2, pp. 181–205, 2013.

[59] L. Chamorro, C. Hill, S. Morton, C. Ellis, R. Arndt, and F. Sotiropoulos, "On the interaction between a turbulent open channel flow and an axial-flow turbine," *Journal of Fluid Mechanics*, vol. 716, pp. 658–670, 2013.

[60] L. P. Chamorro, D. R. Troolin, S.-J. Lee, R. Arndt, and F. Sotiropoulos, "Three-dimensional flow visualization in the wake of a miniature axial-flow hydrokinetic turbine," *Experiments in fluids*, vol. 54, no. 2, pp. 1–12, 2013.

[61] I. Rodriguez, R. Borell, O. Lehmkuhl, C. D. Perez Segarra, and A. Oliva, "Direct numerical simulation of the flow over a sphere at re= 3700," *Journal of Fluid Mechanics*, vol. 679, pp. 263–287, 2011.

[62] J. Y. Tu and L. Fuchs, "Overlapping grids and multigrid methods for three-dimensional unsteady flow calculations in ic engines," *International Journal for Numerical Methods in Fluids*, vol. 15, no. 6, pp. 693–714, 1992.

[63] S. Caruso, J. Ferziger, and J. Oliger, "Adaptive grid techniques for elliptic fluid-flow problems," *AIAA-86-0498*, 1986.

[64] P. Coelho, J. C. F. Pereira, and M. G. Carvalho, "Calculation of laminar recirculating flows using a local non-staggered grid refinement system," *International Journal for Numerical Methods in Fluids*, vol. 12, no. 6, pp. 535–557, 1991.

[65] J. S. Anagnostopoulos, "Discretization of transport equations on 2D Cartesian unstructured grids using data from remote cells for the convection terms," *International Journal for Numerical Methods in Fluids*, vol. 42, no. 3, pp. 297–321, 2003.

[66] G. Papadakis and G. Bergeles, "A local grid refinement method for three-dimensional turbulent recirculating flows," *International Journal for Numerical Methods in Fluids*, vol. 31, no. 7, pp. 1157–1172, 1999.

[67] A. Nezis, D. Angelidis, V. Assimakopoulos, and G. Bergeles, "On the wind flow patterns under neutral and unstable conditions in an urban area," *International Journal of Environment and Pollution*, vol. 47, no. 1-4, pp. 257–267, 2011.

[68] D. Quinlan, "AMR++: A design for parallel object-oriented adaptive mesh refinement," in *Structured Adaptive Mesh Refinement (SAMR) Grid Methods* (S. Baden, N. Chrisochoides, D. Gannon, and M. Norman, eds.), vol. 117 of *The IMA Volumes in Mathematics and its Applications*, pp. 53–58, Springer New York, 2000.

[69] A. M. Wissink, R. D. Hornung, S. R. Kohn, S. S. Smith, and N. Elliott, "Large scale parallel structured amr calculations using the SAMRAI framework," in *Proceedings of the 2001 ACM/IEEE Conference on Supercomputing*, Supercomputing '01, (New York, NY, USA), pp. 6–6, ACM, 2001.

[70] M. Berger and P. Colella, "Local adaptive mesh refinement for shock hydrodynamics," *Journal of Computational Physics*, vol. 82, no. 1, pp. 64–84, 1989.

[71] M. J. Berger and J. Oliger, "Adaptive mesh refinement for hyperbolic partial differential equations," *Journal of Computational Physics*, vol. 53, no. 3, pp. 484–512, 1984.

[72] M. Berger and I. Rigoutsos, "An algorithm for point clustering and grid generation," *IEEE Transcactions on Systems, Man and Cybernetics*, vol. 53, pp. 1278–1286, 1991.

[73] P. MacNeice, K. M. Olson, C. Mobarry, R. de Fainchtein, and C. Packer, "PARAMESH: A parallel adaptive mesh refinement community toolkit," *Computer Physics Communications*, vol. 126, no. 3, pp. 330–354, 2000.

[74] X. Feng, S. Zhang, C. Xiang, L. Yang, C. Jiang, and S. T. Wu, "A hybrid solar wind model of the CESE+HLL method with a Yin-Yang overset grid and an AMR grid," *The Astrophysical Journal*, vol. 734, no. 1, p. 50, 2011.

[75] M. Vanella, P. Rabenold, and E. Balaras, "A direct-forcing embedded-boundary method with adaptive mesh refinement for fluid-structure interaction problems," *Journal of Computational Physics*, vol. 229, no. 18, pp. 6427–6449, 2010.

[76] F. Zahle and N. N. Sørensen, "On the influence of far-wake resolution on wind turbine flow simulations," *Journal of Physics: Conference Series*, vol. 75, no. 1, p. 012042, 2007.

[77] J. A. Michelsen, "Block structured multigrid solution of 2d and 3d elliptic pdes," tech. rep., Technical University of Denmark, 1994.

[78] T. Tran, D. Kim, and J. Song, "Computational fluid dynamic analysis of a floating offshore wind turbine experiencing platform pitching motion," *Energies*, vol. 7, no. 8, pp. 5011–5026, 2014.

[79] L. A. Martnez-Tossas, M. J. Churchfield, and S. Leonardi, "Large eddy simulations of the flow past wind turbines: actuator line and disk modeling," *Wind Energy*, , doi:10.1002/we.1747, 2014.

[80] J. A. Jasak H., Tukovic Z., "Openfoam: A c++ library for complex physics simulations," in *International Workshop on Coupled Methods in Numerical Dynamics IUC, Dubrovnik, Croatia*, 2007.

[81] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, and H. Zhang., "Petsc webpage."

[82] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, "Efficient management of parallelism in object oriented numerical software libraries," in *Modern Software Tools in Scientific Computing* (E. Arge, A. M. Bruaset, and H. P. Langtangen, eds.), pp. 163–202, Birkhäuser Press, 1997.

[83] J. Smagorinsky, "General Circulation Experiments with the Primitive Equations," *Monthly Weather Review*, vol. 91, p. 99, 1963.

[84] X. Yang, F. Sotiropoulos, R. J. Conzemius, J. N. Wachtler, and M. B. Strong, "Large-eddy simulation of turbulent flow past wind turbines/farms: the virtual wind simulator (vwis)," *Wind Energy*, 2014, DOI: 10.1002/we.1802.

[85] M. Wang and P. Moin, "Dynamic wall modeling for large-eddy simulation of complex turbulent flows," *Physics of Fluids*, vol. 14, no. 7, 2002.

[86] A. Gilmanov and F. Sotiropoulos, "A hybrid cartesian/immersed boundary method for simulating flows with 3d, geometrically complex, moving bodies," *Journal of Computational Physics*, vol. 207, no. 2, pp. 457 – 492, 2005.

[87] L. Ge and F. Sotiropoulos, "A numerical method for solving the 3d unsteady incompressible navier-stokes equations in curvilinear domains with complex immersed boundaries," *Journal of Computational Physics*, vol. 225, no. 2, pp. 1782–1809, 2007.

[88] J. R. Koseff and R. L. Street, "The Lid-Driven Cavity Flow: A Synthesis of Qualitative and Quantitative Observations," *Journal of Fluids Engineering-transactions of The Asme*, vol. 106, 1984.

[89] B. Jiang, T. Lin, and L. A. Povinelli, "Large-scale computation of incompressible viscous flow by least-squares finite element method," *Computer Methods in Applied Mechanics and Engineering*, vol. 114, no. 34, pp. 213 – 231, 1994.

[90] F. Shoichi, T. Masahisa, and F. Yasuji, "Extension to three-dimensional problems of the upwind finite element scheme based on the choice of up- and downwind points," *Computer Methods in Applied Mechanics and Engineering*, vol. 112, no. 14, pp. 109 – 131, 1994.

[91] C. J. Ho and F. H. Lin, "Numerical simulation of three-dimensional incompressible flow by a new formulation," *International Journal for Numerical Methods in Fluids*, vol. 23, no. 10, pp. 1073–1084, 1996.

[92] D. DeZeeuw and K. Powell, "Euler calculations of axisymmetric under-expanded jets by an adaptive refinement method," *AIAA Paper 92-0321, 1992.*

[93] L. Chamorro and F. Port-Agel, "A wind-tunnel investigation of wind-turbine wakes: Boundary-layer turbulence effects," *Boundary-Layer Meteorology*, vol. 132, no. 1, pp. 129–149, 2009.

[94] Y.-T. Wu and F. Port-Agel, "Simulation of turbulent flow inside and above wind farms: Model validation and layout effects," *Boundary-Layer Meteorology*, vol. 146, no. 2, pp. 181–205, 2013.

[95] R. Sheldahl, F. Sotiropoulos, R. J. Conzemius, J. N. Wachtler, and M. B. Strong, "Aerodynamic characteristics of seven airfoil sections through 180 degrees angle of attack for use in aerodynamic analysis of vertical axis wind turbines," SAND80-2114, Sandia National Laboratories, Albuquerque, New Mexico, March 1981.

[96] U. M. Yang *et al.*, "Boomeramg: a parallel algebraic multigrid solver and preconditioner," *Applied Numerical Mathematics*, vol. 41, no. 1, pp. 155–177, 2002.

[97] L. Vermeer, J. N. Sørensen, and A. Crespo, "Wind turbine wake aerodynamics," *Progress in aerospace sciences*, vol. 39, no. 6, pp. 467–510, 2003.

[98] J. Sørensen, "Aerodynamic aspects of wind energy conversion," *Annu. Rev. Fluid Mech.*, vol. 43, pp. 427–448, 2011.

[99] R. Cal, J. Lebrón, L. Castillo, H. Kang, and C. Meneveau, "Experimental study of the horizontally averaged flow structure in a model wind-turbine array boundary layer," *Journal of Renewable and Sustainable Energy*, vol. 2, no. 1, p. 013106, 2010.

[100] L. Chamorro, R. Arndt, and F. Sotiropoulos, "Turbulent flow properties around a staggered wind farm," *Boundary-layer meteorology*, vol. 141, no. 3, pp. 349–367, 2011.

[101] L. Chamorro, M. Guala, R. Arndt, and F. Sotiropoulos, "On the evolution of turbulent scales in the wake of a wind turbine model," *Journal of Turbulence*, no. 13, 2012.

[102] C. Meneveau, "The top-down model of wind-farm boundary-layers and its applications," *J. Turbul.*, vol. 13, p. N7, 2012.

[103] K. Howard, J. Hu, L. Chamorro, and M. Guala, "Characterizing the response of a wind turbine model under complex inflow conditions," *Wind Energy*, vol. 18, no. 4, p. 729743, 2015.

[104] P. Hancock and F. Pascheke, "wind-tunnel simulation of the wake-flow of a large wind-turbine in a stable boundary-layer: part 2, the wake-flow," *Boundary-Layer Meteorol.*, vol. 151, pp. 3–21, 2014.

[105] T. Mikkelsen, N. Angelou, K. Hansen, M. Sjöholm, M. Harris, C. Slinger, P. Hadley, R. Scullion, G. Ellis, and G. Vives, "A spinner-integrated wind lidar for enhanced wind turbine control," *Wind Energy*, vol. 16, pp. 625–643, 2013.

[106] R. Wagner, T. Pedersen, M. Courtney, I. Antoniou, S. Davoust, and R. Rivera, "Power curve measurement with a nacelle mounted lidar," *Wind Energy*, vol. 17(9), pp. 1441–1453, 2014.

[107] M. Rhodes and J. Lundquist, "he effect of wind-turbine wakes on summertime us midwest atmospheric wind profiles as observed with ground-based doppler lidar," *Boundary-Layer Meteorol.*, vol. 149, pp. 85–103, 2013.

[108] J. Hong, M. Toloui, L. Chamorro, M. Guala, K. Howard, S. Riley, J. Tucker, and F. Sotiropoulos, "Natural snowfall reveals large-scale flow structures in the wake of a 2.5-mw wind turbine," *Nature communications*, vol. 5, 2014.

[109] L. Ivanova and N. E.D., "Numerical simulation of wind-farm influence on wind flow," *Wind Energy*, vol. 24, pp. 257–269, 2000.

[110] M. Calaf, C. Meneveau, and M. Parlange, "Large eddy simulation study of a fully developed thermal wind-turbine array boundary-layer," *Ercoftac SER*, vol. 15, pp. 239–244, 2011.

[111] Y. T. Wu and P.-A. F., "On the statistics of wake meandering: an experimental study," *Boundary-Layer Meteorol.*, vol. 146(2), pp. 181–205, 2013.

[112] X. Yang, S. Kang, and F. Sotiropoulos, "Computational study and modeling of turbine spacing effects in infinite aligned wind farms," *Physics of Fluids (1994-present)*, vol. 24, no. 11, p. 115107, 2012.

[113] G. V. Iungo, F. Viola, S. Camarri, F. Porté-Agel, and F. Gallaire, "Linear stability analysis of wind turbine wakes performed on wind tunnel measurements," *Journal of Fluid Mechanics*, vol. 737, pp. 499–526, 2013.

[114] F. Viola, G. Iungo, , S. Camarri, F. Porté-Agel, and F. Gallaire, "Prediction of the hub vortex instability in a wind turbine wake: stability analysis with eddy-viscosity models calibrated on wind tunnel data," *Journal of Fluid Mechanics*, vol. 750, p. R1, 2014.

[115] V. Okulov, I. Naumov, R. Mikkelsen, I. Kabardin, and J. Sørensen, "A regular strouhal number for large-scale instability in the far wake of a rotor," *Journal of Fluid Mechanics*, vol. 747, pp. 369–380, 2014.

[116] V. L. Okulov and J. N. Sørensen, "Stability of helical tip vortices in a rotor far wake," *Journal of Fluid Mechanics*, vol. 576, p. 125, 2007.

[117] K. Howard, A. Singh, F. Sotiropoulos, and M. Guala, "On the statistics of wake meandering: an experimental study," *Phys. of fluids*, 2015.

[118] D. Yang, C. Meneveau, and L. Shen, "Effect of downwind swells on offshore wind energy harvesting–a large-eddy simulation study," *Renewable Energy*, 2014.

[119] D. Griffith and P. Richards, "The snl100-03 blade: Design studies with flatback airfoils for the sandia 100-meter blade," *Sandia National Laboratories Technical Report*, pp. SAND2014–18129., 2014.

[120] S. Chakrabarti, *Handbook of Offshore Engineering (2-volume set)*. Elsevier, 2005.

[121] C. P. Butterfield, W. Musial, J. Jonkman, P. Sclavounos, and L. Wayman, *Engineering challenges for floating offshore wind turbines*. National Renewable Energy Laboratory, 2007.

[122] E. Wayman, P. Sclavounos, S. Butterfield, J. Jonkman, W. Musial, *et al.*, "Coupled dynamic modeling of floating wind turbine systems," in *Offshore technology conference*, vol. 139, Houston, 2006.

[123] W. Turbines, "Part 1: Design requirements, iec 61400-1," 2005.

[124] A. Singh, K. Howard, and M. Guala, "On the homogenization of turbulent flow structures in the wake of a model wind turbine," *Physics of Fluids (1994-present)*, vol. 26, no. 2, p. 025103, 2014.

[125] C. K. Jung, K. H., , and H. Jo, "Viscous effect on the roll motion of a rectangular structure," *Journal of engineering mechanics*, vol. 132, no. 2, pp. 190–200, 2006.

[126] D. Medici and P. Alfredsson, "Measurements on a wind turbine wake: 3d effects and bluff body vortex shedding," *Wind Energy*, vol. 9, no. 3, pp. 219–236, 2006.

[127] A. Calderer, C. Feist, K. Ruehl, M. Guala, and F. Sotiropoulos, "Experimental validation of a fluid structuire interaction model for simulating offshoire floating wind turbines," *Journal of Fluids and Structures (submitted)*, 2016.

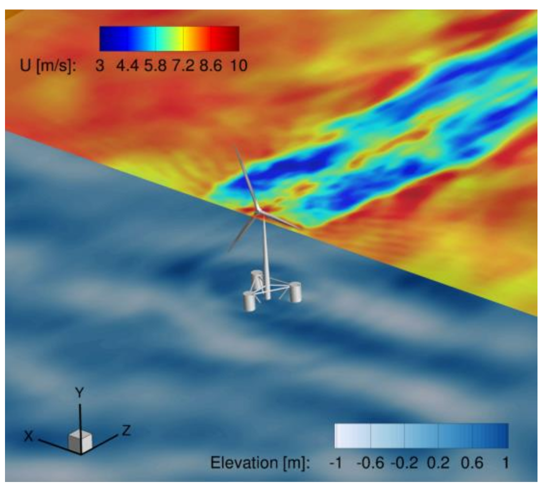
Appendix A: User Manual

VFS

User Manual

Wind Version 1.0

October 30, 2015



University of Minnesota Driven to Discover℠

Sandia National Laboratories

Acknowledgements

VFS is the result of many years of research work by several graduate students, post-docs, and research associates that have been involved in the Computational Hydrodynamics and Biofluids Laboratory directed by professor Fotis Sotiropoulos. The preparation of the present manual has been supported by the U.S. Department of Energy (DE-EE 0005482) and the University of Minnesota Initiative for Renewable Energy and the Environment (RM-0005-12).

Current Code Developers

Dennis Angelidis

Antoni Calderer

Anvar Gilmanov

Ali Khosronejad

Trung Le

Xiaolei Yang

Former Code Developers

Iman Borazjani

Liang Ge

Seokkoo Kang

Contributors to this manual

Antoni Calderer (Saint Anthony Falls Lab.)

Ann Dallman (Sandia National Laboratories)

D. Todd Griffith (Sandia National Laboratories)

Thomas Herges (Sandia National Laboratories)

Kelley Ruehl (Sandia National Laboratories)

License

Contents

Chapter 1

Introduction

Virtual Flow Simulator (VFS) is a three-dimensional (3D) incompressible Navier-Stokes solver based on the Curvilinear Immersed Boundary (CURVIB) method developed by Ge and Sotiropulos [1]. The CURVIB is a sharp interface type of immersed boundary (IB) method that enables the simulation of fluid flows with the presence of geometrically complex moving bodies. In IB approaches, the structural body mesh is superposed on the underlying Eulerian fluid mesh that is kept fixed. This approach circumvents the limitation of classical body fitted methods, in which the fluid mesh adapts to the body and thus limited to relatively simple geometries and small amplitude motions.

A particularity of the CURVIB method with respect to most sharp interface IB methods is that it is formulated in generalized curvilinear coordinates. This feature allows application of a body-fitted approach for the simpler boundaries while maintaining the ability to incorporate complex and moving geometries. For instance, in wind energy applications, one could take advantage of this feature when simulating the site specific geometry of a wind farm. The fluid mesh can follow the actual topography of the terrain while treating the turbines as immersed bodies.

VFS also integrates a two-phase flow solver module based on the level set method that allows simulation of coupled free surface flows with water waves, winds, and floating structures [2, 3]. This module was designed to simulate offshore floating wind turbines considering the non-linear effects of the free surface with a two-phase flow solver, the coupled 6 degrees of freedom (DoF) dynamics of the floating structure, and the ability to incorporate turbulent wind and wave conditions representative of realistic offshore environments.

The CURVIB method has been applied to a broad range of applications, such as cardiovascular flows [4, 5, 6], river bed morphodynamics [7, 8, 9], and wind and hydro-kinetic turbine simulations [10, 11]. For wind energy applications, a turbine can be resolved by immersing the geometry with the IB method or by using one of the different rotor parametrization models implemented.

The current version of the manual is for the VFS-Wind version of VFS. This version of the code includes the base solver and all the necessary libraries for simulating land based and offshore wind farms. The parts that are not included in this version are the modules for sediment transport, bubbly flows, and elastic body deformations.

The organization of this user manual is as follows: in Chapter 2, the main governing equations and numerical methods employed by VFS are briefly described. In Chapter 3, the user is introduced to the basic steps to start using VFS. In Chapter 4, the source code organization is introduced with a description of the main functions in each module. In Chapter 5, the code input parameters are described. Finally, in Chapter 6, a series of application cases are documented.

Chapter 2

Overview of the Numerical Algorithms

2.1 The Flow Solver

The code solves the spatially-filtered Navier-Stokes equations governing incompressible flows of two immiscible fluids. The equations adopt a two-fluid formulation based on the level set method and are expressed in generalized curvilinear coordinates as follows $(i, j, k, l = 1, 2, 3)$:

$$J\frac{\partial U^i}{\partial \xi^i} = 0, \tag{2.1}$$

$$\begin{aligned}\frac{1}{J}\frac{\partial U^j}{\partial t} &= \frac{\xi_l^i}{J}\left(-\frac{\partial}{\partial \xi_j}\left(U^j u_l\right) + \frac{1}{\rho\left(\phi\right)Re}\frac{\partial}{\partial \xi^j}\left(\mu\left(\phi\right)\frac{\xi_l^j \xi_l^k}{J}\frac{\partial u_l}{\partial \xi^k}\right) - \right. \\ &\left. -\frac{1}{\rho\left(\phi\right)}\frac{\partial}{\partial \xi^j}\left(\frac{\xi_l^j p}{J}\right) - \frac{1}{\rho\left(\phi\right)}\frac{\partial \tau_{lj}}{\partial \xi^j} - \frac{\kappa}{\rho(\phi)We^2}\frac{\partial h(\phi)}{\partial x_j} + \frac{\delta_{i2}}{Fr^2}\right),\end{aligned} \tag{2.2}$$

where ϕ is the level set function defined below, ξ^i are the curvilinear components, ξ_l^i are the transformation metrics, J is the Jacobian of the transformation, U^i are the contravariant volume fluxes, u_i are the Cartesian velocity components, ρ is the density, μ is the dynamic viscosity, p is the pressure, τ_{lj} is the sub-grid scale (SGS) tensor, κ is the curvature of the interface, δ_{ij} is the Kronecker delta, h is the smoothed Heaviside function, and Re, Fr, and We are the dimensionless Reynolds, Froude, and Weber numbers, respectively, which can be defined as:

$$Re = \frac{UL\rho_{water}}{\mu_{water}}, Fr = \frac{U}{\sqrt{gL}}, We = U\sqrt{\frac{\rho_{water}L}{\sigma}} \tag{2.3}$$

where U and L are the characteristic velocity and linear dimension, ρ_{water} and μ_{water}, the density and dynamic viscosity of the water phase, g the gravitational acceleration, and σ the surface tension.

The level set function ϕ is a signed distance function, adopting positive values on the water phase and negative values on the air phase. The density and viscosity are taken to be constant within each phase, and transition smoothly across the interface, which is smeared over a distance 2ϵ, as follows:

$$\rho\left(\phi\right)=\rho_{air}+\left(\rho_{water}-\rho_{air}\right)h\left(\phi\right), \tag{2.4}$$

$$\mu\left(\phi\right)=\mu_{air}+\left(\mu_{water}-\mu_{air}\right)h\left(\phi\right), \tag{2.5}$$

where $h(\phi)$ is the smoothed Heaviside function given in [12] and defined as:

$$h(\phi)=\begin{cases} 0 & \phi<-\epsilon, \\ \frac{1}{2}+\frac{\phi}{2\epsilon}+\frac{1}{2\pi}\sin(\frac{\pi\phi}{\epsilon}) & -\epsilon\leq\phi\leq\epsilon, \\ 1 & \epsilon<\phi. \end{cases} \tag{2.6}$$

Note that using the above equations, one can recover the single fluid formulation, by setting the density and viscosity to a constant value throughout the whole computational domain.

The free surface interface, given by the zero level of the distance function ϕ, can be modeled by solving the following level set equation proposed by Osher and Sethian [13]:

$$\frac{1}{J}\frac{\partial\phi}{\partial t}+U^j\frac{\partial\phi}{\partial\xi^j}=0. \tag{2.7}$$

After solving the above level set advection equation, the distance function no longer maintains a unit gradient $|\nabla\phi|=1$, which is a requirement to ensure conservation of mass between the two phases. To remedy this situation, the code solves the mass conserving re-initialization equation proposed by Sussman and Fatemi [14]. A detailed description of the method in the context of curvilinear coordinates can be found in Kang and Sotiropoulos [15].

The flow equations (2.1) and (2.2) are solved using the fractional step method of Ge and Sotiropoulos [1]. The momentum equations are discretized in space and time with a second-order central differencing scheme for the viscous, pressure gradient, and SGS terms, a second-order central differencing or a third-order WENO scheme for the advective terms, and the second-order Crank-Nicholson method for time advancement as follows:

$$\frac{1}{J}\frac{\mathbf{U}^*-\mathbf{U^n}}{\triangle t}=\mathbf{P}(p^n,\phi^n)+\frac{1}{2}(\mathbf{F}(\mathbf{U}^*,\mathbf{u}^*,\phi^{n+1})+(\mathbf{F}(\mathbf{U^n},\mathbf{u^n},\phi^n)), \tag{2.8}$$

where n indicates the previous time step, $\triangle t$ the time step size, $\mathbf{F}$ the right hand side of Eq.(2.2) excluding the pressure term, and $\mathbf{P}$ the pressure term. The continuity condition, discretized with three-point central differencing scheme, is enforced in the second stage of the fractional step method with the following pressure Poisson equation:

$$-J\frac{\partial}{\partial\xi^i}\left(\frac{1}{\rho(\phi)}\frac{\xi_l^i}{J}\frac{\partial}{\partial\xi^j}\left(\frac{\xi_l^j\Pi}{J}\right)\right)=\frac{1}{\triangle t}J\frac{\partial U^{j,*}}{\partial\xi^j}, \tag{2.9}$$

where Π is the pressure correction, used as follows, to update the pressure and the velocity field resulting from the first step of the fractional step method:

$$p^{n+1} = p^n + \Pi, \tag{2.10}$$

$$U^{i,n+1} = U^{i,*} - J\Delta t \frac{1}{\rho(\phi)} \frac{\xi_l^i}{J} \frac{\partial}{\partial \xi^j} \left(\frac{\xi_l^j \Pi}{J} \right), \tag{2.11}$$

The momentum equations are solved using an efficient matrix-free Newton-Krylov solver, and the Poisson equation with a generalized minimal residual (GMRES) method preconditioned with multi-grid.

The level set equations (2.7) are discretized with a third-order WENO scheme in space, and second-order Runge-Kutta in time. The re-initialization step uses a second-order ENO scheme. A detailed description of the method can be found in [15].

2.2 The CURVIB Method

The code can simulate flows around geometrically complex moving bodies with the sharp interface CURVIB method developed by Ge and Sotiropoulos [1]. The method has been thoroughly validated in many applications, such as fluid-structure interaction (FSI) problems [16, 17], river bed morphodynamics [7, 8, 9], and cardiovascular flows [4, 5, 6].

In the CURVIB method, the bodies are represented by an unstructured triangular mesh that is superposed on the background curvilinear or Cartesian fluid grid. First the nodes of the computational domain are classified depending on their location with respect to the position of the body. The nodes that fall inside the body are considered structural nodes and are blanked out from the computational domain. The nodes that are located in the fluid but in the immediate vicinity of the structure are denoted as IB nodes, where the boundary condition of the velocity field is reconstructed. The remaining nodes are the fluid nodes where the governing equations are solved.

The velocity reconstruction is performed in the wall normal direction with either linear or quadratic interpolation in the case of low Reynolds number flows when the IB nodes are located in the viscous sub-layer. While, the velocity reconstruction uses the wall models described by [18, 19, 20] in high Reynolds number flows when the grid resolution is not sufficient to accurately resolve the viscous sub-layer.

The distance function ϕ also needs to be reconstructed at the body-fluid interface. This is done by setting gradient $\Delta\phi$ to be zero at the cell faces that are located between the fluid and IB nodes as described in [15].

2.3 The Structural Solver and the Fluid-Structure Interaction Algorithm

The original FSI algorithm implementation for single phase flows is described in Borazjani, Ge, and Sotiropoulos [16], and the extension to two-phase free surface flows in Calderer, Kang, and Sotiropoulos [2].

The code solves the rigid body equations of motion (EoM) in 6 DoF, which can be written in Lagragian form and in principle axis as follows (i=1,2,..6),

$$M\frac{\partial^2 Y^i}{\partial t^2} + C\frac{\partial Y^i}{\partial t} + KY^i = F_f^i + F_e^i \tag{2.12}$$

where Y^i represents the coordinates of the Lagrangian vector describing the motion of the structure. For the translational DoFs, Y^i are the Cartesian components of the body position, M is the mass matrix, C is the damping coefficients matrix, K is the spring stiffness coefficient matrix, F_f^i are the forces exerted by the fluid, and F_e^i are the components of the external force vector. For the rotational DoFs, Y^i are relative angle components of the body, M represents the moment of inertia, and F_f^i and F_e^i are the moments around the rotation axis, induced by the fluid and by the external forces, respectively.

The forces and moments that the fluid exerts on the rigid body are computed by integrating the pressure and the viscous stresses along the surface Γ of the body as follows

$$F_f = \int_\Gamma -pn d\Gamma + \int_\Gamma \tau_{ij} n_j d\Gamma \tag{2.13}$$

$$M_f = \int_\Gamma -\epsilon_{ijk} r_j p n_k d\Gamma + \int_\Gamma \epsilon_{ijk} r_j \tau_{kl} n_l d\Gamma \tag{2.14}$$

where p denotes the pressure, τ the viscous stress, ϵ_{ijk} the permutation symbol, r the position vector, and n the normal vector.

The EoM (Eqs. 2.12) are coupled with the fluid domain equations through a partitioned FSI approach. The time integration can be done explicitly with loose coupling (LC-FSI), or implicitly with strong coupling (SC-FSI). The Aitken acceleration technique of [21] allows for significant reduction in the number of sub-iterations when the SC-FSI algorithm is used. A detailed description of both time-integration algorithms is given in [16].

2.4 Turbine Parameterizations

The actuator disk and actuator line models implemented in the code for parameterizing turbine rotors are given, respectively, in Yang, Kang, and Sotiropoulos [10] and in Yang et al. [22]. The basic idea of these models is to extract from the flow field the kinetic energy that is estimated to be equivalent to that from a turbine rotor, without the need to resolve the flow around its geometry. To introduce such kinetic energy reduction into the flow, a sink term, affecting the fluid nodes located at the vicinity of the turbine, is considered in the right hand side of the momentum equations.

2.4.1 Actuator Disk Model

In the actuator disk model, the turbine rotor is represented by a circular disk that is discretized with an unstructured triangular mesh. The body force of the disk per unit area is the following

$$F_{AD} = -\frac{F_T}{\pi D^2/4}, \tag{2.15}$$

where D is the rotor diameter and F_T is the thrust force computed as

$$F_T = \frac{1}{2}\rho C_T \frac{\pi}{4} D^2 U_\infty^2, \tag{2.16}$$

where U_∞ is the turbine incoming velocity, $C_T = 4a(1-a)$ is the thrust coefficient taken from the one-dimensional momentum theory, and a is the induction factor. The incoming velocity U_∞ is also computed from the one-dimensional momentum theory as

$$U_\infty = \frac{u_d}{1-a}, \tag{2.17}$$

where u_d is the disk-averaged stream-wise velocity computed as

$$u_d = \frac{4}{\pi D^2} \sum_{N_t} u(X) A(X), \tag{2.18}$$

where N_t is the number of triangular elements composing the disk mesh, $A(X)$ is the area of each element, and $u(X)$ is the velocity at the element centers. The fluid velocity at the disk ($u(X)$) requires interpolation from the velocity values at the surrounding fluid mesh points, as the nodes from the fluid and disk meshes do not necessarily coincide. If we consider X to be the coordinates of the actuator disk nodes and x the coordinates of the fluid mesh nodes, the interpolation, which uses a discrete delta function, reads as follows

$$u(X) = \sum_{N_D} u(x)\delta_h(x-X)V(x), \tag{2.19}$$

where δ_h is a 3D discrete delta function, $V(x)$ is the volume of the corresponding fluid cell, and N_D is the number of fluid cells involved in the interpolation.

Finally, the body force F_{AD}, which is computed at the disk mesh nodes, needs to be distributed over the fluid cells located in the immediate vicinity using the following expression:

$$f_{AD}(x) = \sum_{N_D} F_{AD}(X)\delta_h(x-X)A(x). \tag{2.20}$$

2.4.2 Actuator Line Model

In the actuator line method, each blade of the rotor is modeled with a straight line, divided in several elements along the radial direction. In each of the elements, the lift (L) and drag (D) forces are computed using the following expressions:

$$L = \frac{1}{2}\rho C_L C V_{rel}^2, \tag{2.21}$$

$$D = \frac{1}{2}\rho C_D C V_{rel}^2, \tag{2.22}$$

where C_L, C_D are the lift and drag coefficients, respectively, taken from tabulated two-dimensional (2D) airfoil profile data, C is the chord length, and V_{ref} is the incoming reference velocity computed as

$$V_{rel} = (u_z, u_\theta - \Omega r) \tag{2.23}$$

where u_z and $u_\theta - \Omega r$ are the components of the velocity in the axial and azimuthal directions, respectively, Ω the angular velocity of the rotor, and r the distance to the center of the rotor.

To compute the reference velocity at the line elements, similarly to the actuator disk method, the velocity at the fluid mesh is transferred to the line elements using a discrete delta function as given by equation (2.19). Once the lift and drag forces are computed at each of the line elements, the distributed body force in the fluid mesh can be computed using the following equation:

$$f_{AL}(x) = \sum_{N_L} F(X)\delta_h(x - X)A(x). \tag{2.24}$$

where N_L is the number of segments composing one of the actuator lines, $F(X)$ is the projection of L and D, expressed in actuator line local coordinates, into Cartesian coordinates.

2.5 Large-Eddy Simulation

The description of the Large-eddy simulation (LES) model implemented in the code is extensively described in Kang et al. [23]. The sub-grid stress term in the right hand side of the momentum equation resulting from the the filtering operation is modeled with the Smagorinsky SGS model of [24] which reads as follows

$$\tau_{ij} - \frac{1}{3}\tau_{kk}\delta_{ij} = -2\mu_t \overline{S}_{ij}, \tag{2.25}$$

where μ_t is the eddy viscosity, the overline indicates the grid filtering operation, and $\overline{S_{ij}}$ is the large-scale strain-rate tensor. The eddy viscosity can be written as

$$\mu_t = C_s \Delta^2 |\overline{S}|, \tag{2.26}$$

where Δ is the filter width taken from the box filter, $|\overline{S}| = (2\overline{S}_{ij}\overline{S}_{ij})^{1/2}$ is the magnitude of strain-rate tensor, and C_s is the Smagorinsky constant computed dynamically with the Smagorinsky model of Germano et al. [25].

2.6 Wave Generation

The code uses an internal generation method as described in [3]. As illustrated in Figure 2.1, a surface force is applied at the so called source region, generating waves that propagate symmetrically to both stream-wise directions. A sponge layer method is used at the lateral boundaries to dissipate the waves and prevent reflections. The area between the source region and the sponge layer can be used to study body-wave interactions. Both the sponge layer force and the wave generation force are introduced in the code using a source term in the right hand side of the momentum equations.

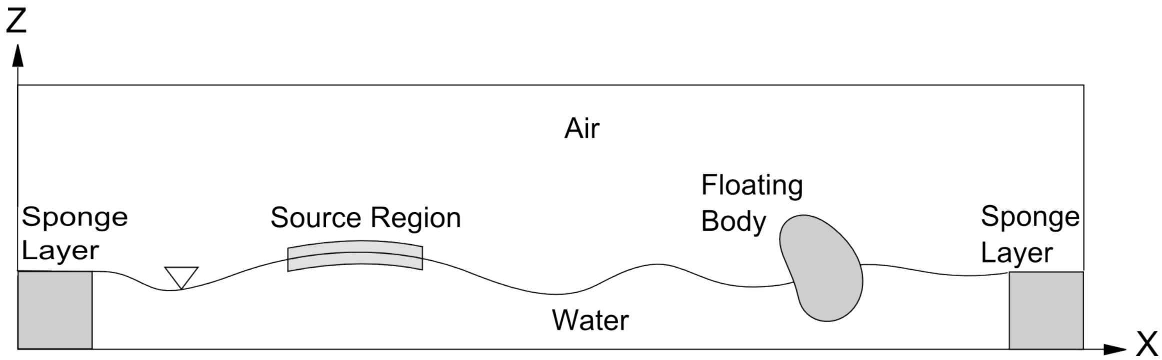


Figure 2.1: Schematic description of the wave generation method using the free surface forcing method [3].

To generate the following surface elevation,

$$\eta(x, y, t) = A\cos(k_x x + k_y y - \omega t + \theta), \tag{2.27}$$

where k_x and k_y are the components of the wavenumber vector K, θ is the wave phase, A is the wave amplitude, and ω the wave frequency, the forcing term reads as follows

$$S_i(x, y, t) = n_i(\phi)P_0\delta(x, \epsilon_x)\delta(\phi, \epsilon_\phi)sin(\omega t - k_y y - \theta), \tag{2.28}$$

where P_0 is a coefficient that depends on the wave and fluid characteristics,

$$P_0 = A\frac{g^2}{\omega^2}\frac{\epsilon}{f(\epsilon_x, k_x)}\frac{2\rho_w}{\rho_a + \rho_w}\frac{k_x}{\left(k_x^2 + k_y^2\right)^{1/2}}, \tag{2.29}$$

δ is a distribution function defined as:

$$\delta(\alpha, \beta) = \begin{cases} \frac{1}{2\beta}\left[1 + \cos\left(\frac{\pi\alpha}{\beta}\right)\right] & \text{if } -\beta < t < \beta \\ 0 & \text{otherwise.} \end{cases}, \tag{2.30}$$

and $f(\epsilon_x, k_x)$ is

$$f(\epsilon_x, k_x) = \frac{\pi^2}{k_x\left(\pi^2 - \epsilon_x^2 k_x^2\right)}\sin(k_x\epsilon_x). \tag{2.31}$$

With the present forcing method, wave fields with multiple components, such as a broadband wave spectrum, can be incorporated into the fluid domain. This method enables to simulate the interaction of floating structures with complex wave fields. The wave fields can either be originated in a far-field precursor simulation or taken from theoretical or measured data.

The sponge layer method for dissipating the waves at the boundaries reads as follows:

$$S_i(x,y,t) = -\left[\mu C_0 u_i + \rho C_1 u_i \left|u_i\right|\right] \frac{\exp\left[\left(\frac{x_s - x}{x_s}\right)^{n_s}\right] - 1}{\exp(1) - 1} \text{ for } (x_0 - x_s) \leq x \leq x_0, \tag{2.32}$$

where x_0 denotes the starting coordinate of the source region, x_s is the length of the source region, and C_0, C_1, and n_s are coefficients to be determined empirically. In [3], n_s is 10, C_0 is 200000, and C_1 is 1.0.

Chapter 3

Getting Started

In this chapter, we describe how to install the libraries required for VFS to work and how to run a simulation case.

3.1 Installing PETSC and Required Libraries

VFS is implemented in C and is parallelised using the Message Passing Interface (MPI). The Portable, Extensible Toolkit for Scientific Computation (PETSC) libraries are used for the code organization and to facilitate its parallel implementation. Also we use the library HYPRE for solving the Poisson equation. Before the code can be compiled, the following libraries need to be properly installed.

- PETSC version 3.1
- Blas and Lapack
- openmpi
- HYPRE

Note that when installing the PETSC libraries there is the option to automatically install the other required libraries in the case that they are not already in the computer. The PETSC web page [26] (http://www.mcs.anl.gov/petsc/documentation/installation.html) gives a detailed description on how to install all of these libraries. We briefly outline the steps for installing PETSC in the command line of a linux machine in the case that none of the aforementioned libraries have been previously installed:

1. Create a directory where to download the PETSC source files:
   mkdir source
2. Create the installation directory:
   mkdir system

3. Download the PETSC source code from the PETSC server in the source folder:
   wget http://ftp.mcs.anl.gov/pub/petsc/release-snapshots/petsc-3.1-p6.tar.gz

4. Unzip the PETSC source code in the source folder:
   tar xvfz petsc-3.1-p6.tar.gz

5. Execute the PETSC configuration script:
   config/configure.py --with-cc=mpicc --with-cxx=mpicxx --with-fc=mpif90 --download-f-blas-lapack=1 --download-mpich=1 --download-hypre=1 --prefix=installation_folder --with-shared=0 --with-debugging=0

6. If the previous action executes successfully, PETSC will print on the screen the subsequent steps.

   Note that PETSC can be installed with the debugging option either active or inactive. It is recommended that PETSC is installed without debugging because it will be used in production mode. The PETSC installation with debugging may be needed for developing purpose but it compromises the speed of the code.

3.2 Compiling the Code

To compile the code and generate an executable file we include a file named "makefile". This file is general and can work for any linux-based computer without being modified. It basically links all the source code files (*.c, *.h) and the necessary libraries (PETSC, HYPRE, etc). Given that in every computer the compiler libraries are located in different directories, the user has to create a new text file with the name: "makefile.local". This file is read by makefile and should contain the following user dependent information:

```
MPICXX       = mpicxx
HOME     = /Your_Home_Folder/
ACML     = /Your_ACML_Folder/
PETSC    = /Your_PETSC_Folder/
HYPRE    = /Your_HYPRE_Folder/
USE_TECPLOT=1 ##1 if tecplot is intalled , otherwise set to 0
TEC360HOME=/
```

If all libraries have been successfully installed and properly referenced in the file "makefile.local" the code should compile by typing the following command in the linux shell:

make

Alternatively, one can add the option -j to increase the computation speed by taking advantage of the several processors as follows:

make -j

Either of the last two commands will generate the executable source file "**vwis**".

In addition to the executable file for running the code, it is also necessary to compile the executable file for post-processing the resulting output data. The post-processing file can generate result files in both Tecplot format (plt) and Totalview format (vtk). The advantage of Totalview over Tecplot is that it is open source and can be download from the Totalview webpage.

3.2.1 Compiling the Post-Processing File for Totalview Only

This may be useful when Tecplot 360 is not installed in the computer. For creating the post-processing file which is only suitable for generating Totalview output files, first set the tecplot option in the file "makfile.local" to inactive as follows "USE_TECPLOT=0".

Then type the following command in the linux shell:

make data

3.2.2 Compiling the Post-Processing File for Tecplot and Totalview

If this is the case, in the previously described file "makfile.local", the tecplot option has to be active as follows "USE_TECPLOT=1".

In addition, one needs to download the library file "libtecio.a" from the tecplot library webpage (http://www.tecplot.com/downloads/tecio-library/) and add it to the code directory.

The post-processing file named "**data**" should then be created with the following command:

make data

3.3 Running VFS

3.3.1 File Structure Overview

All the files that are necessary for running VFS should be located in a user defined folder. The required input files are the following:

grid.dat or xyz.dat The structured mesh for the fluid domain.

bcs.dat The option file for setting the BCs of the fluid boundaries.

vwis Executable file obtained upon code compilation.

Submission script The linux script for submitting the job in a linux based cluster.

control.dat The file containing most of the control options.

ibmdata00, ibmdata01, etc. The mesh files for the immersed bodies, if any.

data The post-processing executable file

The fluid grid file, the immersed bodies grid files, the boundary conditions file and the control file are described extensively in Section 4.2.

The remainder of this chapter describes the compiling process for obtaining the executable vwis file, and the submission script for running the code.

3.3.2 Execute the Code

Each cluster may have different job resource manager systems although the most common is PBS. If it is not PBS, your system manager may provide instructions about the resource manager in use. There is ample documentation online as well.

In the case that your system uses the PBS system, the user can submit a job in the cue with the example script shown below:

```
#!/bin/bash
### Job name
### Mail to user
#PBS -k o
#PBS -l nodes=1:ppn=16,walltime=4:00:00
#PBS -j oe

cd $PBS_O_WORKDIR

/openmpi_directory/mpirun --bind-to-core  ./vwis>& err
```

In the script above, the job will use 1 node of 16 cpus per node (ppn). The job maximum duration will be 4 hours (walltime). The name of the file executed is "vwis", and the on screen information will be stored in "err" file.

The command for submitting this script is

qsub script_name.sh

One can check the status of the job,

showq

To finalize the job type

qdel job_id

3.3.3 Simulation Outputs

vfieldX.dat, ufieldX.dat, pfieldX.dat, nvfieldX.dat, lfieldX.dat, cs_X.dat

These are binary files containing the flow variables at the whole computational domain at a given time step indicated by "X". These files will be exported at every "tio" times steps, where "tio" is a control option. The content in each of these files is summarized in Table 3.1.

Table 3.1: Description of the instantaneous output results

File Name	Containing variable	Description of the Variable
vfield'x'.dat	Ucont	Contravarian velocity components (fluxes)
ufield'x'.dat	Ucat	Cartesian velocity components
pfield'x'.dat	P	Pressure field
nvfield'x'.dat	Nvert	If IB is used, it indicates the classification of nodes. 3 is structure node, 1 is IB node, and 0 is fluid node.
lfield'x'.dat	level	The distance function used in the levelset method to track the interface
cs_'x'.dat	Cs	The eddy viscosity coefficient when using LES.

Converge_dU

This text file contains the following information:

- The first number displayed in each of the lines is the time step number.
- Second column is the algorithm that the line refers to (momenum, poisson, levelset, IBMSERA0)
  - momentum: Momentum equation solver.
  - poisson: Poisson solver for the second step of the fractional step method.
  - levelset: If using the level set method it refers to the Reinitialization equation.
  - IBMSERA0: Refers to the searching algorithm for node classification when at least one immersed body is present.
- The third column is the computational time in seconds that it takes the algorithm to complete.
- The fourth column, if any, is the convergence of the corresponding solver. In the case of the Poisson solver the convergence is the maximum divergence and is indicated with "Maxdiv=".

Kinetic_Energy.dat

This file exports a text file with two columns. The first column displays the time, and the second column the total kinetic energy of the whole computational domain calculated as follows

$$KE = \sum_{i=0}^{N_i} \sum_{j=0}^{N_j} \sum_{k=0}^{N_k} u_{ijk}^2 + v_{ijk}^2 + w_{ijk}^2, \tag{3.1}$$

where N_i,N_j, and N_k is the number of grid nodes in the i, j, and k directions, respectively, and u_{ijk}, v_{ijk}, and w_{ijk}, are the Cartesian velocity components computed at the cell centers. The function **KE_Output** is responsible for creating this file.

FSI_position00, FSI_position01, etc

This is a text file containing information about the immersed body location, velocity and forces for the linear degrees of freedom in the x, y, and z direction. It consists of 10 columns as follows:

$$ti, Y_x, \dot{Y}_x, F_x, Y_y, \dot{Y}_y, F_y, Y_z, \dot{Y}_z, F_z, \tag{3.2}$$

where ti is the time step number, Y is the body position with respect to its initial position, $\dot{Y}$ is the body velocity, and F the force that the fluid imparts to the immersed body. If multiple immersed bodies are used, the code will print a file for each of the bodies, labeled with the body number.

FSI_Angle00, FSI_Angle01, etc.

This output file is similar to the FSI_position file, but for the rotational degrees of freedom. It contains information about the immersed body rotation angle and angular velocity of the body. It consists of 7 columns as follows:

$$ti, \Theta_x, \dot{\Theta}_x, \Theta_y, \dot{\Theta}_y, \Theta_z, \dot{\Theta}_z, \tag{3.3}$$

where ti is the time step number, Θ is the body rotation with respect to its starting position and $\dot{\Theta}$ is the body angular velocity. If multiple immersed bodies are used, the code will print a file for each of the bodies, labeled with the body number.

Force_Coeff_SI00

This text file contains information about the forces that the fluid imparts to the immersed body in the three linear degrees of freedom, x, y, and z. The file has 10 columns with the following data:

$$ti, F_x, F_y, F_z, Cp_x, Cp_y, Cp_z, Ap_x, Ap_y, Ap_z, \tag{3.4}$$

where ti is the time step number, F denotes the fluid force applied to the body, Cp the normalized force coefficient, and Ap the area of the body projected in the corresponding direction which has been used for computing Cp. Again, a different file is exported for each additional immersed body.

Momt_Coeff_SI00

This file is equivalent to Force_Coeff_S00 but in the rotational degrees of freedom. The information exported is the following:

$$ti, M_x, M_y, M_z, \tag{3.5}$$

where ti is the time step number and M denotes the moments applied to the structure.

surface000_00_nf.dat, surface000_01_nf.dat, ...

These are tecplot ASCII files containing the surface of the corresponding IB body at the time step indicated at the first number of the file name. It basically contains the x, y, and z coordinates of the nodal points of the triangular mesh.

sloshing.dat

Data file with information about the free surface elevation in the center of the tank for the sloshing case. The first column is the simulation time [sec], the second is the computed surface elevation [m] at the tank center, the third is the expected theoretical solution at the tank center, and the final column is the error.

3.3.4 Post-Processing

Once VFS reaches a time step at which data are output (multiple of "tio", or output time step), the output file can be post-processed. Post-processing consists of converting the output files (ufieldxx.dat, vfieldxx.dat, pfieldxx.dat, nvfieldxx.dat, lfieldxx.dat, lfieldxx.dat, etc), which are in binary form, to a format which is readable for a visualization software such as TecPlot360 or Totalview.

Create plt Files for Tecplot360

To post-process the data, the executable "data" should be used with the following command:

mpiexec ./data -tis 0 -tie 50000 -ts 100

where -tis is the start timestep, -tie is the end timestep and -ts is the time step interval at which the files were generated.

The above step will generate n number of result files (for each timestep) compatible with tecplot with names similar to: Resultxx.plt where xx indicates the timestep. The .plt file can be opened in tecplot and worked upon.

The following webpage contains several tutorials in flash video on how to use tecplot: http://www.genias-graphics.de/cms/tp-360-tutorials.html

Create vtk Files for Totalview

If the user wants to visualize the files using Totalview, the option -vtk 1 should be included as follows
mpiexec ./data -tis 0 -tie 50000 -ts 100 -vtk 1

Averaged results

By default, the plt and vtk files contain instantaneous data results even if the code has performed averaging of the results (-averaging set to 1, 2, or 3 at the time that the code is executed). To include the averaged results in the post-processed file the option -avg needs to be activated when executing the file "data". The option -avg can be set to 1, 2, and 3, depending on the amount of information to be included in the post-processed file, having an impact on the overall file size. If -avg is set to 1, the post-processed file contains only averaged velocity and turbulence intensities (U, V, W, uu, vv, ww, wv, vw, uw); if it is set to 2, the post-processed file contains the same as the case for -avg 2 plus the averaged pressure and pressure fluctuations; if it is set to 3, the file contains the same variables as in -avg 2 plus averaged vorticity. Note that for post-processing the data using the options -avg 2 and 3, the code should have been executed using the option -averaging with a value equal or larger than the -avg value. An example on how to process averaged results is as follows:
mpiexec ./data -tis 0 -tie 50000 -ts 100 -avg 1

3.3.5 The Post-Processed File

Instantaneous Results

The variables of the post-processed results file, regardless of being Tecplot or Totalview formatted, are summarized in Table 3.2.

Table 3.2: Description of the instantaneous output results in the postprocessed file.

Variable	Description
X, Y, Z	Coordinates of the fluid grid
U, V, W	Velocity components at the grid cell centers
UU,	Velocity magnitude
P	Pressure field
Nvert	If IB is used, it indicates the classification of nodes. 3 is structure node, 1 is IB node, and 0 is fluid node.
level	The distance function used in the levelset method to track the interface

Averaged Results

The variables of the post-processed results file, regardless of being Tecplot or Totalview formatted, are summarized in Table 3.3.

Table 3.3: Description of the time averaged output results in the post-processed file.

Variable	**Description**
X, Y, Z	Coordinates of the fluid grid
U, V, W	Averaged velocity components at the grid cell centers
uu, vv, ww, uv, uw, vw	Turbulence intensities
P	Averaged pressure field
Nvert	If the IB method is used, it indicates the classification of nodes. 3 is structure node, 1 is IB node, and 0 is fluid node.
level	The distance function used in the levelset method to track the interface

Chapter 4

Code Input Parameters

4.1 Units

In terms of the units that the code uses, the user needs to differentiate between two cases, when the level set method is active (levelset option set to 1) and when it is not active (levelset option set to 0). In the case that the level set method is not active, the Navier-Stokes equations are solved in the dimensionless form. In this case, it is recommended that the reference length of the domain and the reference velocity of the flow are both equal to one.

In the case that the level set method is in use, the equations solved have real dimensions, and the units are in MKS (meters-kilograms-seconds system).

4.2 VFS Input Files

The VFS code requires several input files that have to be located by default at the cases directory. The input files are the following:

control.dat

bcs.dat

grid.dat

ibmdata00, ibmdata01, ibmdata02, etc. (if IB method is in use)

4.2.1 The control.dat File

The file control.dat is a text file that is read by the code upon initiation. It contains most of the input variables for the different modules of the code. The order in which the different options are included in the control file is not relevant, however it is recommended to group the options in categories as proposed below.

The control options start with a dash “-” symbol. If for any case the same option is introduced twice the code will take the value introduced latest in the file. If a control option wants to be kept in the control file for later use it can be commented by introducing a “!” sign in the start of the line.

Time Related Options

dt (double)

Time step size for the solution of the Navier-Stokes equations. The CFL number has to be smaller than 1 although values less than 0.5 are recommended. When using the level set method, the CFL number is more restrictive and in that case the CFL should be lower than 0.05.

tio (int)

The code exports the complete flow field data every time step that is a multiple of the value of this parameter.

totalsteps (int)

Total number of time steps to run before ending the simulation.

rstart (int)

This option is for restarting a simulation. The value given to this parameter is the time step number for which the simulation will restart. This option can only be used if results from a previous simulation are present in the folder. Note that even if the value is set to zero, the simulation will restart from a previous run, in that case, from a zero time step.

rs_fsi (int 0, 1)

This parameter can be used when restarting a simulation (rstart active) and FSI module is in use. If rs_fsi is set to 1, the code will read the file FSI_DATA corresponding to time step rstart. This file contains information regarding the body motion such as body position, velocity, forces, etc.

delete (int 0, 1)

If this option is set to 1, the code keeps in the hard drive only the result files from the two last exported time step, deleting the files from previously exported time steps.

averaging (int 0, 1, 2, 3)

If this parameter is set to 1, 2, or 3, the code performs time averaging of the flow field. Time averaging is typically started when the flow field is fully developed which occurs when the kinetic energy of the flow is stabilized. Therefore, averaging is a two stage process. In the first stage, the simulation is started with the averaging option set to zero and advanced to a point in which the flow is developed. In the second stage, the flow results from stage 1 are used to restart the simulation and start the averaging. The first step to do in stage 2 is to rename the flow field files to be used from stage 1 (ufieldXXXXXX.dat, vfieldXXXXXX.dat, ...) to the file name corresponding to time step 0 files (ufield000000.dat, vfield000000.dat, ...). Then, the simulation can be restarted by activating the averaging option and setting the rstart option to 0 (restart from time step 0). The reason that the files need to be renamed is because of the way that the code performs averaging. The code uses the current time step for dividing the velocity sum and obtaining the averaged results. So if averaging is started at a non-zero time step, the number of velocity summands will not correspond to the current time step number and the average will not be correct. As long as the averaging has been started at time step zero, there is no problem with restarting the simulation during stage 2. The different values for this parameter, 1, 2, and 3, refer to the amount of information that is averaged and exported to the results files. If average is set to 1, only averaged velocities (U, V, and W) and turbulence intensities (uu, vv, ww, uw, vw, uv) are processed. If the parameter is set to 2, in addition to the averaged velocities and turbulence intensities, the averaged pressure and pressure fluctuations are computed. Finally, if the parameter is set to 3, the processed variables include the ones from option 2 plus the averaged vorticity vector. As already indicated in section 3.3.4, to post-process the results and include the averaged results to the output file, the option "avg" needs to be activated. The value given to "avg" should be lower or equal to the value given to the option "averaging".

Options for Boundary Contitions

inlet (int)

The inlet option defines the inflow profile type when the inlet plane is set to inflow mode (see bcs.dat description). It also sets the velocity initial conditions to the one corresponding to the inlet profile.

1: Uniform inflow with velocity value determined by the option flux.

13: This option is used for performing channel flow simulation. It sets the velocity initial condition to follow a log law. The domain height (channel half height) has to be 1.

100: Imports inflow from external file. The external files must have a cross plane grid geometry equivalent to the one of the current simulation grid.

perturb (int 0, 1)

If a non-zero initial condition is given, this option perturbs the initial velocity by adding random velocity values which are proportional to the local streamwise velocity component.

wallfunction (int)

Use wall model at the walls of the immersed bodies. It basically interpolates the velocity at the IB nodes using a wallfunction.

ii_periodic, jj_periodic, kk_periodic (int 0,1)

Consider periodic boundary conditions in the corresponding direction. When this option is chosen in the control file, the corresponding boundaries in bcs.dat need to be set to any non-defined value such as 100.

flux (double)

0: Sets the velocity at the inlet boundary to 1.

Non-zero value = Sets the flux at the inlet boundary. The flux is defined as the bulk inlet velocity divided by the area of the inlet cross section. The units are m^3/s or non-dimensional depending on whether level set is used.

Level Set Method Options

levelset (int 0,1)

Activates the levelset method. If used, the solved Navier-Stokes equations are in dimensional form.

dthick (double)

If using the level set method, this parameter defines half the thickness of the air/water interface. The fluid properties adopt their corresponding value in each phase, and vary smoothly across this interface. Typical values adopted by "dthick" are on the order of 2 times the vertical grid spacing. Larger values may be necessary in extreme cases.

sloshing (int 0, 1, 2)

Sets the initial condition of the sloshing problem in a tank and exports to a file (sloshing.dat) the free surface elevation at the center of the tank.

1: Sets the initial condition for the 2D sloshing problem in a tank.

2: Sets the initial condition for the 3D sloshing problem in a tank.

0: Sloshing problem is not considered.

level_in (int 0, 1, 2)

Flat initial free surface with elevation defined by level_in_height.

1: The free surface normal is in the z direction.

2: The free surface normal is in the y direction.

level_in_height (double)

When the level_in option is active this parameter determines the free surface vertical coordinate which is uniform.

fix_level (int 0,1)

1: The free surface is considered but kept fixed. In this case, the level set equation is not solved.

fix_outlet, fix_inlet (int 0,1)

When using inlet and outlet boundary conditions, activating any of these parameters will keep constant the free surface elevation at the corresponding boundary.

levelset_it (int)

Number of times to solve the reinitialitzation equation for mass conservation. Higher number may be useful in cases involving high curvature free surface patterns.

levelset_tau (double)

Parameter to define the pseudo-time step size used in the reinitialization equation. The pseudo time step is levelset_tau times the minimum grid spacing.

rho0, rho1 (double)

Density of the water and the air respectively.

mu0, mu1 (double)

Dynamic viscosity of the water and the air respectively.

stension (int 0,1)

If active it considers the surface tension at the free surface interface.

Modelling Options and Solver Parameters

les (int 0,1,2)

Activating the LES model.

1: Smagorinsky - Lilly model.

2: Dynamic Smagorinsky model (recommended).

imp (int 1,2,3,4)

Type of solver for the momentum equation. The only value supported is 4 which corresponds to the Implicit solver. Other values correspond to obsolete approaches and are not guaranteed to work.

imp_tol (double)

Tolerance for the momentum equation. A value smaller than 1.0×10^{-5} would be recommended.

poisson (int -1,0,1)

Selection of the Poisson solver. The only value supported is 1, other values correspond to obsolete approaches and are not guaranteed to work.

poisson_it (int)

Maximum number of iteration for solving the Poisson equation. If the tolerance set by the option poisson_tol is reached the Poisson solver is completed before reaching poisson_it iterations.

poisson_tol (double)

Tolerance for the maximum divergence of the flow.

ren (double)

This parameter defines the Reynolds Number in the case of non-dimensional simulations. When using the level set method, the equations solved have dimensions and "ren" is not in use.

inv (int 0,1)

1: Neglects the viscous terms in the RHS of the momentum equation, and thus the flow is considered inviscid.

Immersed Boundary Method Options

imm (int 0,1)

Activate the immersed boundary method. The code will expect a structural mesh (ibmdata00).

body (int)

If using the immersed boundary method, this parameter determines the number of bodies considered. There must be the same number of IB meshes (e.g., ibmdata00, ibmdata01,...,ibmdataXX, where XX is the number of bodies).

thin (int)

Option for simulating bodies with very sharp geometries where the resolution is not sufficient to resolve the depth.

x_c, y_c, z_c (double)

Initial translation of the immersed body position.

angle_x0, angle_y0, angle_z0 (double)

Initial rotation of the immersed body position with respect to the center of rotation defined by x_r, y_r, and z_r. Note that the initial rotation is applied after the translation for which x_r, y_r, and z_r are defined.

Fluid-Structure Interaction Options

fsi (int 0,1)

1: Activates the ability to move the structure in a single translational DoF. Select the desired DoF by setting one of the following options to 1: dgf_ax, dgf_ay, or dgf_az.

forced_motion (int 0, 1)

When "fsi" is active, the parameter controls whether to use prescribed motion or FSI motion.

0: The motion of the structure is computed in a coupled manner with the flow.

1: The motion of the structure is prescribed. Both the position of the structure in time as well as the velocity in time are specified in the function Forced_Motion which is located in the code source file fsi_move.c. In this function the motion is defined with an analytic expression. If a user needs to set a specific motion which can be defined through a mathematical expression it needs to be implemented by editing this function. Obviously, if the code is edited it also needs to be recompiled.

rfsi (int 0,1)

1: Activates the ability to move the structure in a single rotational DoF. Select the desired rotational DoF by setting rotdir.

rotdir (int 1,2,3)

When rfsi is active, the parameter selects the axis of rotation.

1: Rotation along the x axis.

2: Rotation along the y axis.

3: Rotation along the z axis.

fsi_6dof (int 0,1)

1: Activates the ability to move the structure in up to six DoF. Each of the six DoF can be activated independently of each other with the following control options: dgf_x, dgf_y, dgf_z, dgf_ax, dgf_ay, dgf_az, (described below). Note that any combination of the six DoF is valid regardless of the translational DoF or rotational DoF. One can obtain the same results as using the "fsi" option or the "rfsi" option by activating only one of the 6 DoF.

dgf_ax, dgf_ay, dgf_az (int 0, 1)

In the case of a single translational DoF (fsi 1), the desired translational DoF is specified by setting one of these to 1.

When using fsi_6dof multiple DoF can be activated.

dgf_x, dgf_y, dgf_z (int 0, 1)

In the case of a single rotational DoF (rfsi 1), the desired rotational DoF is specified by setting one of these to 1.

str (int 0, 1)

0: When either fsi or rfsi are active, the parameter uses the loose coupling algorithm.

1: When either fsi or rfsi are active, the parameter uses the strong coupling algorithm.

red_vel, damp, mu_s (double)

Parameters to be used in the test case "VIV of an elastically mounted rigid cylinder".

body_mass (double)

Mass of the structure.

body_inertia_x, body_inertia_y, body_inertia_z (double)

Moment of inertia with respect to the center of rotation defined by x_r, y_r, and z_r.

body_alpha_rot_x, body_alpha_rot_y, body_alpha_rot_z (double)

Damping coefficient for the rotational DoFs.

body_alpha_lin_x, body_alpha_lin_y, body_alpha_lin_z (double)

Damping coefficient for the translational DoFs.

body_beta_rot_x, body_beta_rot_y, body_beta_rot_z (double)

Elastic spring constant for the rotational DoFs.

body_beta_lin_x, body_beta_lin_y, body_beta_lin_z (double)

Elastic spring constant for the translational DoFs.

x_r, y_r, z_r (double)

Center of rotation in the rotational DoFs.

fall_cyll_case (int 0, 1)

Option for the falling cylinder test case.

Wave Generation Options

wave_momentum_source (int 0, 1, 2)

When using the level set method, the parameter activates the wave generation module, based on the method of Guo and Shen (2009).

0: Waves are not generated.

1: Waves are read from an external file.

2: A single monochromatic wave is generated.

wave_angle_single (double)

In the case that wave_momentum_source is equal to 2, the parameter sets the wave initial phase.

wave_K_single (double)

In the case that wave_momentum_source is equal to 2, the parameter sets the wave number.

wave_depth (double)

In the case that wave_momentum_source is active, the parameter sets the water depth. This parameter is used in the dispersion relation to compute the wave frequency.

wave_a_single (double)

In the case that wave_momentum_source is equal to 2, the parameter sets the wave amplitude.

wave_ti_start (int)

Time step to start applying the wave generation method.

wave_skip (int)

This option is used in the case that wave_momentum_source is equal to 1 (waves are imported from external files). The external data file to be imported, which has been generated using an external code, may have a time step size not equivalent to the time step of the present code simulation. Usually the time step size in the wave data is significantly larger than that from the present simulation ($\Delta t_{wave-data} = \Delta t_{simulation} \times b$, where b is an integer). By setting wave_skip to b, the code will import a new wave data file every wave_skip time steps, and since wave_skip=b, the time of the simulation will match the time of the wave data.

wind_skip (int)

This option is equivalent to wave_skip but for importing the inlet wind profile when the option air_flow_levelset is equal to 2.

wave_start_read (int)

This parameter is useful for restarting the simulation in the case that wave_momentum_source is equal to 1 (waves are imported from external files). The code will import the wave file corresponding to the wave time step wave_start_read, instead of importing the starting wave data file (WAVE_info000000.dat).

wind_start_read (int)

This parameter is useful for restarting the simulation in the case that air_flow_levelset is equal to 2 (inlet wind profile is imported from external files). The code will import the wind data file corresponding to the wind time step wind_start_read, instead of importing the first wind data file (WAVE_wind000000.dat), .

wave_recicle (int)

In the case that wave_momentum_source is equal to 1 (waves are imported from external files) and the wave time step reaches this number, the wave time step is recycled to 0, which means that the code will import the wave data corresponding to time step 0 (WAVE_info000000.dat).

wind_recicle (int)

In the case that air_flow_levelset is equal to 2 (wind is imported from external files) and the wave time step reaches this number, the wind data time step is recycled to 0, which means that the code will import the wind data corresponding to time step 0 (WAVE_wind000000.dat).

wave_sponge_layer (int 0, 1, 2)

1: Sponge layer method is only applied at the x boundaries.

2: Sponge layer method is applied at the four lateral boundaries.

wave_sponge_xs (double)

Length of the sponge layer applied at the x boundaries.

wave_sponge_x01 (double)

Starting x coordinate of the first sponge layer applied at the x boundary.

wave_sponge_x02 (double)

Starting x coordinate of the second sponge layer applied at the x boundary.

wave_sponge_ys (double)

Length of the sponge layer applied at the y boundaries.

wave_sponge_y01 (double)

Starting y coordinate of the first sponge layer applied at the y boundary.

wave_sponge_y02 (double)

Starting y coordinate of the second sponge layer applied at the y boundary.

Turbine Parameterization Options

rotor_modeled (int 0, 1, 2, ..., 6)

Activate the turbine modeling option with the following parameterization approach:

1: Actuator disk model using the induction factor as input parameter.

2: Option for development purpose. This option is currently obsolete.

3: Actuator line model.

4: Actuator disk model using thrust coefficient as a input parameter.

5: Actuator surface model (under development).

6: Actuator line model with an additional actuator line for computing the reference velocity.

turbine (int)

Number of wind/hydro-kinetic turbines to be modeled.

reflength (double)

Reference length of the turbine. The code will divide the imported turbine diameter (from the mesh file and Turbine.inp) by this amount.

rotatewt (int 1,2,3)

Direction to which the turbines point to.

1: i direction

2: j direction

3: k direction

r_nacelle (double)

This parameter represent the radius of the turbine nacelle. The code will ignore the rotor effect within this radius.

num_foiltype (int)

Number of foil types used along the turbine blade. VFS requires a description file named FOIL00, FOIL01, ..., for each foil type as described in Section $4.2.6.

num_blade (int)

Number of blades in the turbine rotor.

refvel_wt, refvel_cfd (double)

These parameters do not have any effect in the simulation, and are only for normalizing the output profiles. One can set them to 1 and normalize when plotting the data.

loc_refvel (int)

Distance upstream of the turbine in rotor diameters where the turbine incoming velocity or reference velocity is computed.

deltafunction (int)

Type of smoothing function within which the pressure due to the rotor is applied.

halfwidth_dfunc (double)

Half the distance for which the turbine effect is smoothed. The value is expressed in number of grid nodes.

4.2.2 The bcs.dat File

The bcs.dat file is another text file with information about the boundary conditions of the fluid domain boundaries. Lets denote the six boundaries as Imin, Imax, Jmin, Jmax, Kmin, and Kmax, corresponding to the starting and ending boundary in the i, j and k directions, respectively.

The format of the bcs.dat file is a single line with the 6 integers corresponding to each of the boundaries. This number can adopt the following values:

Table 4.1: Options for the bcs.dat file

Boundary condition type	**Value**
Slip Wall	10
No slip wall	1
No slip with wall modelling, smooth wall	-1
No slip with wall modelling, rough wall	-2
[1]Periodic boundary conditions	100
[1]Inlet	5
Outlet	4

The bcs.dat file has the following aspect: Imin-value Imax-value Jmin-value Jmax-value Kmin-value Kmax-value

Example. Simulation case with slip wall at the Imin, Imax, Jmin, Jmax boundaries, and inlet and outlet along the k direction:
10 10 10 10 5 4

[1]Require additional information in the control file. Further details can be found in the corresponding section.

4.2.3 The Grid File

The grid file (grid.dat) is formatted with the standard PLOT3D. The file can be imported in binary form or in ASCII form. For importing the grid in binary form the option binary in the control.dat has to be set to 1, otherwise the code expects the ASCII form.

Any grid generator software that is able to export PLOT3D grids may be suitable for VFS. However, Pointwise is recommended.

When creating the mesh, the user needs to pay attention to the orientation of two different sets of coordinate systems. The Cartesian components which are indicated in Figure 4.1 with x, y, and z, and the curvilinear components which are attached to the mesh and are referred to as i, j, and k. Both coordinate systems should be right-hand oriented.

The recommended axis combination between Cartesian components and grid coordinates is depicted in Figure 4.1.

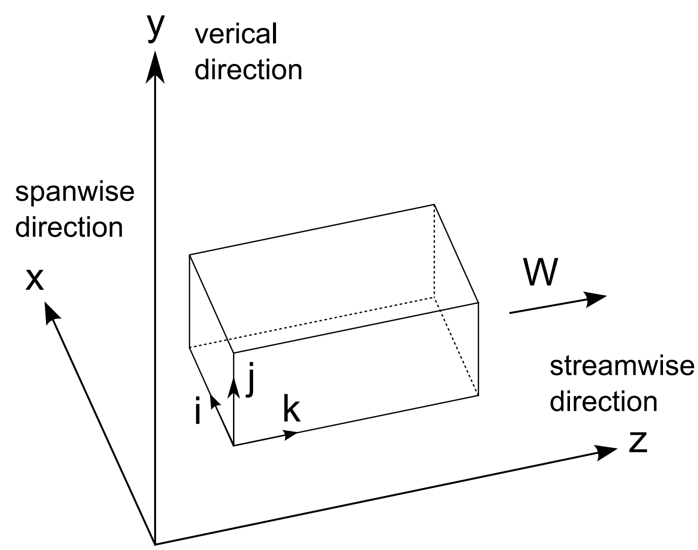


Figure 4.1: Axis orientation

A third format type that VFS can handle is "SEGMENT". This format is suitable only for cases where the fluid mesh is Cartesian, as the only information that the mesh file stores are the grid points of the three axis. An obvious advantage of this approach is that the mesh file size is much smaller than the "PLOT3D" formatted files.

To use "SEGMENT" format, the grid file should be named "xyz.dat" and the option "xyz" in the control file should be set to 1. In the first three lines, the "xyz.dat" file contains the number of grid nodes of the mesh for each of the three axis N_x, N_y, and N_z. The values are followed by the three coordinate of the points in the X axis, then the coordinates of the points in the Y axis, and finally the coordinates of the points in the Z axis as follows:

X_{x1} Y_{x1} Z_{x1}
X_{x2} Y_{x2} Z_{x2}

...
X_{xN_x} Y_{xN_x} Z_{xX_x}
X_{y1} Y_{y1} Z_{y1}
X_{y2} Y_{y2} Z_{y2}
...
X_{yN_y} Y_{yN_y} Z_{yX_y}
X_{z1} Y_{z1} Z_{z1}
X_{z2} Y_{z2} Z_{z2}
...
X_{zN_z} Y_{zN_z} Z_{zX_z}

4.2.4 The Immersed Boundary Grid File

The immersed boundary method allows one or more immersed bodies to be incorporated into the computational domain. If more than one body is considered, by default, each body has its own body mesh and its name is ibmdata00 for the first body, ibmdata01 for the second body, etc.

The body mesh is an unstructured surface mesh with triangular nodes. The format is the standard UCD. When generating an immersed boundary mesh one needs to consider the following:

- The normal direction of the triangular elements must point towards the flow.
- In general, a triangular mesh with triangles of similar sizes as the fluid background mesh is recommended. If the immersed boundary is a flat wall, the triangular mesh may be coarser than the fluid mesh without loss of accuracy.

4.2.5 The Wave Data External File

The wave generation module allows the incorporation of broadband wave fields with large number of frequency components (wave_momentum_source set to 1). The origin of the wave data can be from a precursor simulation (far-field simulation) using an external wave code or from real measurements. A broadband wave field composed of $NZMOD/2 \times (NYMOD+1)$ wave frequencies, where $NZMOD/2$ is the number of frequencies in the Z direction and $2 \times (NYMOD+1)$ is the number of frequencies in the X direction) can be given by the following expression

$$\eta(z,x) = \sum_{k_z=1}^{k_z=NZMOD/2} \sum_{k_x=-NXMOD/2}^{k_x=NXMOD/2} a_{k_z,k_x} cos(k_z*PEZ*z+k_x*PEX*x+\theta_{k_z,k_x}) \quad (4.1)$$

where η is the free surface elevation, k_z and k_x indicate the directional wave numbers, a is the wave amplitude, and (θ) is the wave phase. PEZ and PEX are coefficients to scale the wave numbers, which are usually set to 1.

The code will read the wave data file named WAVE_infoXXXXXX.dat, where XXXXXX represents the time step of the wave data. The first line of the wave data file contains the time of the wave, NZMOD, NXMOD, PEZ, and PEX. The second line is where the actual wave data starts. The wave file is as follows:

$time_{wave}\ NZMOD\ NXMOD\ PEZ\ PEX$
$a_{k_z=1,k_x=0}\ \theta_{k_z=1,k_x=0}$
$a_{k_z=2,k_x=0}\ \theta_{k_z=2,k_x=0}$
...
$a_{k_z=NZMOD/2,k_x=0}\ \theta_{k_z=NZMOD/2,k_x=0}$
$a_{k_z=1,k_x=1}\ \theta_{k_z=1,k_x=1}$
$a_{k_z=2,k_x=1}\ \theta_{k_z=2,k_x=1}$
...
$a_{k_z=NZMOD/2,k_x=1}\ \theta_{k_z=NZMOD/2,k_x=1}$
$a_{k_z=1,k_x=-1}\ \theta_{k_z=1,k_x=-1}$
$a_{k_z=2,k_x=-1}\ \theta_{k_z=2,k_x=-1}$
...
$a_{k_z=NZMOD/2,k_x=-1}\ \theta_{k_z=NZMOD/2,k_x=-1}$
$a_{k_z=1,k_x=2}\ \theta_{k_z=1,k_x=2}$
$a_{k_z=2,k_x=2}\ \theta_{k_z=2,k_x=2}$
...
$a_{k_z=NZMOD/2,k_x=2}\ \theta_{k_z=NZMOD/2,k_x=2}$
$a_{k_z=1,k_x=-2}\ \theta_{k_z=1,k_x=-2}$
$a_{k_z=2,k_x=-2}\ \theta_{k_z=2,k_x=-2}$
...
$a_{k_z=NZMOD/2,k_x=-2}\ \theta_{k_z=NZMOD/2,k_x=-2}$
...
$a_{k_z=1,k_x=NXMOD/2}\ \theta_{k_z=1,k_x=NXMOD/2}$
$a_{k_z=2,k_x=NXMOD/2}\ \theta_{k_z=2,k_x=NXMOD/2}$
...
$a_{k_z=NZMOD/2,k_x=NXMOD/2}\ \theta_{k_z=NZMOD/2,k_x=NXMOD/2}$
$a_{k_z=1,k_x=-NXMOD/2}\ \theta_{k_z=1,k_x=-NXMOD/2}$
$a_{k_z=2,k_x=-NXMOD/2}\ \theta_{k_z=2,k_x=-NXMOD/2}$
...
$a_{k_z=NZMOD/2,k_x=-NXMOD/2}\ \theta_{k_z=NZMOD/2,k_x=-NXMOD/2}$

As discussed in the control option for the wave module, the wave time step size may no be equal to the time step of the present code simulation. Usually the time step in the wave data is significantly larger than that from the present simulation ($\Delta t_{wave-data} = \Delta t_{simulation} \times b$, where b is an integer). By setting wave_skip to b, the code will import a new wave data file every wave_skip time steps, and since wave_skip=b, the time of the simulation will match the time of the wave data.

If the wave frequencies do not vary in time, one could use a single wave data file by setting the option wave_skip to a very large value.

The wave module also allows the wind field associated with a wave field pre-computed simulation to be incorporated by setting air_flow_levelset to 2. In such a case, the code will read the wave data file named WAVE_windXXXXXX.dat, with XXXXXX representing the time step of the wind data. The first line of the wind data file contains the time of the far-field simulation, the number of grid points in the vertical direction NY, and the number of grid points in the horizontal direction NX. The actual wave data starts in line two. The overall structure of the file is as follows:

$time_{far-field}$ NY NX
$X_{0,0}$ $Y_{0,0}$ $Z_{0,0}$ $U_{0,0}$ $V_{0,0}$ $W_{0,0}$
$X_{0,1}$ $Y_{0,1}$ $Z_{0,1}$ $U_{0,1}$ $V_{0,1}$ $W_{0,1}$
...
$X_{0,NX}$ $Y_{0,NX}$ $Z_{0,NX}$ $U_{0,NX}$ $V_{0,NX}$ $W_{0,NX}$
$X_{1,0}$ $Y_{1,0}$ $Z_{1,0}$ $U_{1,0}$ $V_{1,0}$ $W_{1,0}$
$X_{1,1}$ $Y_{1,1}$ $Z_{1,1}$ $U_{1,1}$ $V_{1,1}$ $W_{1,1}$
...
$X_{1,NX}$ $Y_{1,NX}$ $Z_{1,NX}$ $U_{1,NX}$ $V_{1,NX}$ $W_{1,NX}$
...
$X_{NY,0}$ $Y_{NY,0}$ $Z_{NY,0}$ $U_{NY,0}$ $V_{NY,0}$ $W_{NY,0}$
$X_{NY,1}$ $Y_{NY,1}$ $Z_{NY,1}$ $U_{NY,1}$ $V_{NY,1}$ $W_{NY,1}$
...
$X_{NY,NX}$ $Y_{NY,NX}$ $Z_{NY,NX}$ $U_{NY,NX}$ $V_{NY,NX}$ $W_{NY,NX}$

4.2.6 The Files Required for the Turbine Rotor Model

The Turbine.inp Control File

The Turbine.inp file is a text file containing input parameters used by the rotor model. The file has two lines, the first line is ignored by VFS and only used for informative purposes by listing the input variable names. The second line is the control value corresponding to the variable listed in line 1 as shown in the example below.

```
nx_tb-ny_tb-nz_tb-x_c-y_c-z_c-indf_axis-Tipspeedratio-J_rot ...
0.0 0.0 1.0 0.0 0.104 0.256 0.36 4.5 14380000 ...
```

nx_tb, ny_tb, nz_tb (double)

Normal direction of the turbine rotor plane.

x_c, y_c, z_c (double)

The turbine rotor initial translation.

indf_axis (double)

Induction factor when using the actuator disk model (rotor_model=1).

Tipspeedratio (double)

Tip speed ratio when using the actuator line model (rotor_model=3,6). The tip-speed ratio (TSR) can adopt negative values which indicate that the turbine is rotating counterclockwise with respect to the stream-wise axis.

J_rotation (double)

Rotor moment of inertia used only when the option "turbinetorquecontrol" is active.

r_rotor (double)

Radius of the turbine rotor.

CP_max (double)

Maximum power coefficient of the turbine; used only when the option "turbinetorquecontrol" is active.

TSR_max (double)

Refers to the maximum TSR of the turbine; used only when the option "turbinetorquecontrol" is active.

angvel_fixed (double)

Rotor rotational speed when the option "fixturbineangvel" is active. The variable angvel_fixed can adopt negative values which indicate that the turbine is rotating counterclockwise with respect to the stream-wise axis.

Torque_generator (double)

Turbine torque, used only when the option "turbinetorquecontrol" is active.

pitch (double)

Pitch angle of the turbine blades when using actuator line or actuator surface models.

CT (double)

Thrust coefficient used with rotor_modelled 4.

The acldata000 Mesh File

The acldata000 file is an ASCII data file containing the mesh of the turbine model. When using the actuator line model, the file consists of n segments, where n is the number of rotor blades, as shown in Figure 4.2(a). The ASCII data file uses the SEGMENT format.

In the case of the actuator disc models, the mesh is a UCD formatted unstructured triangular mesh, and the rotor is represented with a circle as shown in Figure 4.2(b).

The turbine center, o, of this mesh could be located directly at the actual position of the turbine, or alternatively, centered at the origin and later translated with the control options x_c, y_c, and z_c defined in the rotor model control file “turbine.inp”.

In the actuator line model the coordinate attached to the segment i has to point towards the tip of the blade. In the actuator disk model, the direction normal to the rotor has to point towards the direction of the flow.

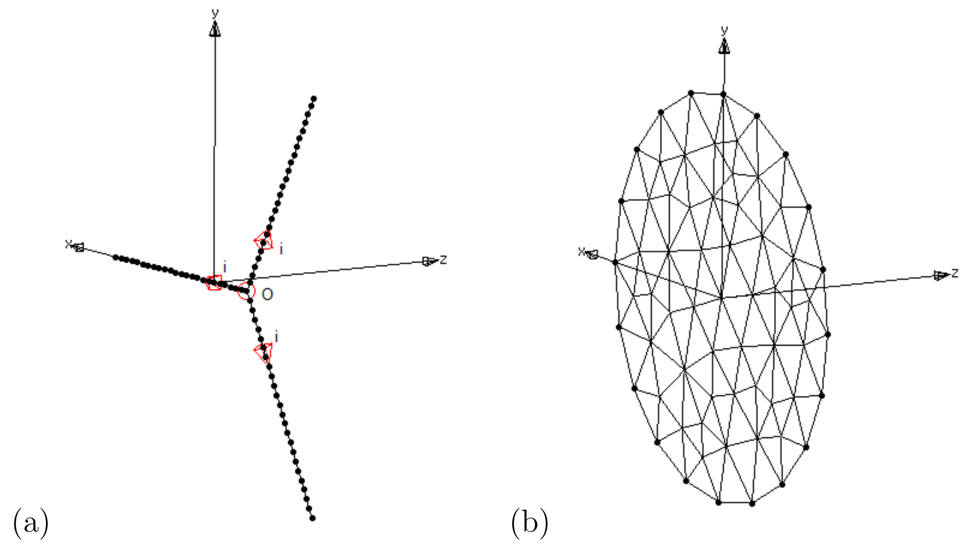


Figure 4.2: Representation of the “acldata000” mesh used to represent the blades in the (a) actuator line model and (b) actuator disk model.

The Urefdata000 Mesh File

The Urefdata000 file is a UCD formatted triangular mesh equivalent to the actuator disk acldata000 file. The purpose of this file is to compute the inflow reference velocity required for both the actuator disk and actuator line models.

The disk dimensions and normal direction in Urefdata000 have to match the dimensions of the turbine rotor defined in “turbine.inp”.

The code positions the disk upstream of the actual turbine location. At every time step, the velocity of the flow is transferred to each triangular element of this mesh. By adding the velocity at all triangular elements and dividing by the surface of the disk, the code computes the turbine reference velocity (disk average velocity). As an example, when using the actuator line, the reference velocity is used for determining the TSR of the simulation.

The CD00, CD01, CD02, ..., and CL00, CL01, CL02, ... Files

These files contain the lift and drag coefficients at each profile along the turbine blades when using the actuator line model (rotor_model 3 or 6). The first two lines in the file are descriptive and the third line defines the number of data points in the file. Starting at line 4, the angle of attack (column 1), in degrees, and Lift/Drag coefficient (column 2) are listed as shown in the example below.

```
Airfoil type: DU97-W-300
Drag coefficients
31
7.50925697358677e-001    1.41002221673661e-002
2.25610960256727e+000    2.10614663046161e-002
4.32615403604048e+000    2.79782769686497e-002
6.20902246358924e+000    2.78301653912614e-002
...
8.98528141199704e+001    2.13806467538879e+000
```

The FOIL00, FOIL01, FOIL02, ... Files

These files contain the angle of attack and chord length for each profile used along the turbine blades when using the actuator line model (rotor_model 3 or 6). The first two lines in the file are descriptive and the third line defines the number of data points in the file. Starting at line 4, the distance from the blade section to the turbine hub in non-dimensional units or in meters (column 1), the blade section angle of attack in degrees (column 2), and the profile chord length in non-dimensional units or meters (column 3) are listed as shown in the example below.

```
Turbine type: Clipper 2.5 MW
Airfoil type: Circular
7
0.000 9.5 2.400
2.800 9.5 2.400
3.825 9.5 2.385
4.950 9.5 2.259
6.950 9.5 2.338
8.950 9.5 2.339
10.800 9.5 2.848
```

Chapter 5

Library Structure

5.1 The Source Code Files

The source code is structured in several files with extension ".c" and one file with extension ".h". The header file (variables.h) is included at the beginning of any other ".c" file and contains all the function prototypes, global variable definitions, and structure definitions. The ".c" files contain the subroutines which are generally grouped by code module. A brief description of the ".c" files is presented as follows:

main.c
: Main code file where the code is initialized and finalized

bcs.c
: Subroutines for specifying boundary conditions

compgeom.c, ibm.c, ibm_io.c, variables.c
: Subroutines for the IB method

fsi.c, fsi_move.c
: Subroutines for the FSI module

level.c, distance.c
: Subroutines for simulating two-phase free surface flows with the levelset method

wave.c
: Subroutines for the wave module (also to simulate wind over waves, in which the wind is imported from a far-field simulation)

data.c
: Subroutines for post-processing and visualizing the results

wallfunction.c
: Subroutines for the wall modeling

rotor_model.c
Contains all the subroutines that are necessary for simulating a wind turbine or a hydro-kinetic turbine using the actuator disk or actuator line models

les.c
Subroutines for the turbulent models

solvers.c, implicitsolver.c, momentum.c, poisson.c, poisson_hypre.c, rhs.c, rhs2.c, timeadvancing.c, timeadvancing1.c
Contains all the subroutines used by the flow solver, including the momentum and the Poisson equations

init.c
Subroutines for initializing the code variables

metrics.c
The subroutines for computing the grid Jacobian and metrics

5.2 The Flow Solver

To describe the basic elements of the flow solver we present a code flow chart of VFS, which displays the order in which the relevant functions of the code are called. This code flow chart corresponds to the most simple case that VFS can simulate and no additional module is considered. An example would be to perform Direct Numerical Simulation of the channel flow case.

As in any "c" code, the so called main function is the entry point or where the software starts the execution. In the code flow chart presented below the functions, emphasized in bold, are indented such that the functions from a lower level are called by the function of the above level.

- **main** (pre-processing)
  In the first part of this main function, the code pre-processing is performed as follows:
  - **MG_Initial**
    Reads the structured grid file (grid.dat), partitions the domain within the cpus, allocates memory for the main variables in a partitioned form. Also reads the boundary conditions file (bcs.dat).
    * **FormMetrics**
      Computes the metrics and Jacobians of the transformation given by equation (2.2).
  - **Calc_Inlet_Area**
    Computes the inlet area corresponding to section k=0. The code was designed such that the stream-wise direction is k.

- **SetInitialGuessToOne**
  Sets the initial velocity of the whole computational domain at time=0 to a specific profile defined by the variable inlet.
- **Contra2Cart**
  Using the Cartesian velocity components at the cell centers, the contravariant velocity components at the cell faces are calculated through interpolation.

- **main** (time iteration)
  At this point of the main function, the time-stepping loop starts.

  - **Flow_Solver**
    This function solves for the velocity and pressure fields to advance to time step ti+1.
    * **Calc_Minimum_dt**
      Calculates and prints the minimum time step size (dt) required such that the CFL number is equal to 1.
    * **Pressure_Gradient**
      Reads the pressure field and computes the pressure gradient.
    * **Formfunction_2**
      Forms the right hand side of the momentum equation.
    * **Implicit_MatrixFree**
      Solves the momentum equation.
    * **PoissonSolver_Hypre**
      Solves the Poisson equation in the second step of the fractional step method to obtain the pressure correction.
    * **UpdatePressure**
      The pressure correction is applied to obtain the pressure field.
    * **Projection**
      Corrects the velocity to make it divergence free.
    * **IB_BC**
      Sets most of the boundary conditions. Note, however, that other functions such as "Implicit_MatrixFree" and "Contratocart" also deal with a part of the boundary conditions.
    * **Divergence**
      Checks and prints the maximum divergence to the output file Convergence_du.
    * **Contra2Cart**
      Using the Cartesian velocity components at the cell centers, the contravariant velocity components at the cell faces are calculated through interpolation.

- * **Calc_ShearStress**
  Computes and outputs the shear stress.
- * **KE_Output**
  Exports the total kinetic energy of the whole computational domain to the file Kinetic_Energy.dat.
- * **Ucont_P_Binary_Output**
  Writes the flow field results to files provided that the time step is a multiple of the control option "tiout".

- End of time-stepping loop
  - **MG_Finalize**
    This function is called right before ending the code to di-allocate all the memory created during the execution of the code.

5.3 Code Modules

In the present section the main functions used by the different modules of the code are reviewed. All modules follow a common structural pattern. First, a group of functions are called for pre-processesing purposes, then, a second set of functions are called with the purpose of advancing the solution in time.

- **Subroutines for pre-processing**. Upon initiation of the program and before starting the time iteration, a set of functions are called to: (1) import the module specific input files (if any); and (2) initialize and allocate memory for the necessary variables. This process happens only once in the beginning of the main function located in the file "main.c".

- **Subroutines for time advancing**. After the initial pre-processing part is completed, the code is ready to start advancing in time. Then a second set of functions are used to compute, at every time step, the necessary elements involved in the corresponding module. This part is generally executed from the function Flow_Solver located in "solvers.c".

5.3.1 The Level Set Method Module

- **Subroutines for pre-processing**. In this module, the pre-processing basically consists of initializing the level set main variables and setting the free-surface initial condition.
  - **MG_Initial** Initializes the levelset variables. The levelset main variable is named "level[k][j][i]", which is defined as the distance from the current cell center to the closest point of the free surface interface. It adopts a positive value in the water phase and negative value in the air phase. The free surface interface coincides with the zero level.

  - **Levelset_Function_IC** Sets the initial position of the free-surface.

- **Subroutines for time advancing**. Here the time-advancing involves solving an advection equation to find the new location of the free surface interface and a mass conserving reinitialization to ensure that the mass within the two phases is conserved.

  - **Advect_Levelset** Solves the levelset advection equation.
    - **Levelset_Advect_RHS** Forms the right hand side terms of the advection equation.
  - **Reinit_Levelset** Solves the mass conserving reinitizlization equation.
    - **Init_Levelset_Vectors** Creates temporary arrays for solving the reinitialization equation.
    - **Solve_Reinit_explicit** Solves the equation in an explicit form.
      - **Distance_Function_RHS** Forms the right hand side terms of the reinitialization equation.
    - **Destroy_Levelset_Vectors** Deletes the temporary arrays.
  - **Compute_Density** Updates the density and viscosity values of the fluid with the values corresponding to the new location of the free surface. The function executes the functions Advect_Levelset and Reinit_Levelsetfree, which update the free surface location.
  - **Compute_Surface_Tension** Applies the surface tension at the free surface.
  - **Levelset_BC** Sets the boundary conditions of the free surface. The function is called both before and after solving the advective and the reinitialization equations.
  - **Calc_free_surface_location** Exports the free surface elevation to the external file FreeSurfaceElev_XXXXXX.dat (XXXXXX refers to the time step) at every "tiout" time steps.

5.3.2 The Large-Eddy Simulation (LES) Method Module

- **Subroutines for pre-processing**. In this module the pre-processing basically consists of initializing the LES main variables.

  - **MG_Initial** Initializes the LES model main variables.

- **Subroutines for time advancing**. Here the time-advancing involves computing the new eddy viscosity, which is added to the diffusion term of the momentum equation.

- **Compute_Smagorinsky_Constant_1** Computes the Smagorinsky constant C_s.
- **Compute_eddy_viscosity_LES** Computes the eddy viscosity μ_t by applying equation (2.26).
- **Formfunction_2** Adds the eddy viscosity term to the right hand side of the momentum equation.

5.3.3 The Immersed Boundary (IB) Method Module

- **Subroutines for pre-processing**. In this module the pre-processing consists of initializing the IB method variables and importing the IB mesh.

  - **main** Initializes the primary variables for the IB method.
  - **ibm_read_ucd** Reads and imports the body triangular mesh (ibmdata00, ibmdata01, ...).
  - **ibm_search_advanced** Performs a classification of the fluid nodes depending on its position with respect to the structure. This classification is stored in the variable "nvert". If nvert is 0 the node belongs to the fluid domain and the equations are solved; if nvert is 3, the node belongs inside the structural domain and the node is blanked from the computational domain; if nvert is 1, the node is an IB node, which belongs in the fluid domain but is located at the immediate vicinity of the structure. IB nodes are where the velocity boundary condition of the body are specified.
  - **ibm_interpolation_advanced** Computes the velocity boundary conditions at the IB nodes. This computation can be done using linear interpolation or using a wall model.
    - **noslip** Applies the no-slip-wall boundary condition using linear interpolation.
    - **freeslip** Applies the slip-wall boundary condition using linear interpolation. Used when the inviscid option is active.
    - **wall_function** Applies a wall model assuming a smooth wall. Used when the option wallfunction is active.
    - **wall_function_roughness** Applies a wall model assuming a rough wall. Used when the wallfunction option is active and rough_set is specified.

- **Subroutines for time advancing**. The time-advancing part depends on whether the body is moving or not. While the velocity boundary condition at the IB nodes has to be recomputed at every time step, the classifications of nodes has to be recomputed only if the body is moving.

- **ibm_search_advanced** This function does not need to be called if the body is not moving. Otherwise, this function needs to be called at every time step, if the body is moving, to update the node classification once the body position has been updated.
- **ibm_interpolation_advanced** The velocity at the IB nodes has to be updated at every time step.

5.3.4 The Fluid-Structure Interaction (FSI) Algorithm Module

- **Subroutines for pre-processing**. In this module the pre-processing consists of initializing the FSI variables and applying an initial motion to the body.

  - **FsiInitialize** Initializes the variables for the body motion; either the motion is prescribed or determined using FSI.
  - **FSI_DATA_Input** Reads the external file "DATA_FSIXXXXXX_YY.dat". (XXXXXX refers to the time step and YY to the body number). This process is necessary when the simulation is restarted. The option rstart_fsi needs to be active.
  - **Elmt_Move_FSI_TRANS** This function applies a linear translation to the body mesh in a single DoF. The function is called when the single translational DoF module is in use. In the pre-processing, the function is used to apply an initial translation to the body either when starting the simulation or when restarting.
  - **Elmt_Move_FSI_ROT** This function applies a rotation to the body mesh in a single DoF. The function is called when the single rotational DoF module is in use. In pre-processing, the function is used to apply an initial rotation to the body, either when starting the simulation or when restarting.
    - **rotate_xyz** This function applies a rotation to a given point with respect to a center of rotation in a given direction.
  - **Elmt_Move_FSI_ROT_TRANS** This function applies the structural motion in any of the six DoF to the body mesh. The function is called when the six DoF module is in use. During pre-processing, the function is used to apply an initial motion to the body either when starting the simulation or when restarting.
    - **rotate_xyz6dof** This function applies a rotation to a given point with respect to a center of rotation in the three axial directions.

- **Subroutines for time advancing**. The time-advancing part depends on whether the body is moving or not. As already discussed for the IB method module, the velocity boundary condition at the IB nodes has to be recomputed at every time step, and the classifications of nodes has to be recomputed only if the body is moving.

- **Struc_Solver** This function computes and updates the new position of the body. The function is called within the main function at every time step.
  - **Calc_forces_SI** Computes the force and moments that the fluid imparts to the body.
  - **Calc_forces_SI_levelset** Computes the force and moments that the fluid imparts to the body. It replaces **Calc_forces_SI** when the level set method is in use.
  - **Forced_Motion** Computes the position and velocity of the structure using the prescribed motion mode. Both the position and the velocity are specified through an analytic expression. Needs to be followed by a call to either the function **Elmt_Move_FSI_TRANS** or **Elmt_Move_FSI_ROT_TRANS**.
  - **Calc_FSI_pos_SC** Solves the EoM in a single translational DoF. Needs to be followed by a call to **Elmt_Move_FSI_TRANS**.
  - **Calc_FSI_pos_SC_levelset** Solves the EoM in a single translational DoF when the levelset method is active. Needs to be followed by a call to **Elmt_Move_FSI_TRANS**.
  - **Calc_FSI_pos_6dof_levelset** Solves the six DoF EoM. Needs to be followed by a call to **Elmt_Move_FSI_ROT_TRANS**
  - **Calc_FSI_Ang** Solves the EoM in a single rotational DoF. Needs to be followed by a call to **Elmt_Move_FSI_ROT**.
  - **Forced_Rotation** Computes the rotation and angular velocity of the structure using the prescribed motion mode through an analytic expression. Needs to be followed by a call to **Elmt_Move_FSI_ROT**.
  - Note that after the motion has been applied to the body mesh, the function **ibm_search_advanced** needs to be applied to update the fluid mesh node classification.
- **FSI_DATA_Output** At every "tiout" time step, it exports the body motion information in the file "DATA_FSIXXXXXX_YY.dat". (XXXXXX refers to the time step and YY to the body number).

5.3.5 The Wave Generation Module

- **Subroutines for pre-processing**. In this module the pre-processing consists of initializing the variables for the wave generation method and for specifying the inlet wind from the far-field precursor simulation.

- **Initialize_wave** Initializes the variables for the wave generation method.
- **Initialize_wind** Initializes the variables for importing the wind field from the far field precursor simulation.

- **Subroutines for time advancing**. In this module the code needs to import the wave data and, if necessary, the wind data, at the time steps for which it needs to be updated (every wave_skip and wind_skip time steps, respectively). Then the imported wave/wind information is applied.
  - **WAVE_DATA_input** Sets the water wave field information (amplitude, frequencies, direction angle, ...) to be simulated.
    If the option wave_momentum_source is 1, the function imports the wave information from an external file.
    If wave_momentum_source is equal to 2, the function uses the information given in the control file.
  - **WAVE_Formfuction2** Adds the pressure force in the right hand side of the momentum equation in the form of a source term.
  - **WAVE_SL_Formfuction2** Applies the sponge layer method at the side wall boundaries that are specified in the control file.
  - **WIND_DATA_input** Reads the external file containing information of the wind field of the far-field simulation to be applied at the inlet of the present simulation.
    * **WIND_vel_interpolate** Function to perform bi-linear interpolation to obtain the velocity values at the present fluid mesh which generally differs from the far-field fluid mesh.

5.3.6 The Rotor Turbine Modeling Module

Actuator Disk Model

The actuator disk model is activated by setting rotor_modeled to 1 (the model input is the induction factor) or to 4 (the input is the thrust coefficient).

- **Subroutines for pre-processing**. In the turbine modelling module the pre-processing subroutines import the turbine model input file, and initialize the corresponding variables, allocating memory if necessary. Again, this process happens only once in the beginning of the main function located in the file "main.c".
  - **main** Initializes variables and imports the turbine control file "Turbine.inp".
    * **disk_read_ucd** Imports the disk mesh. The function is called first to import the actual turbine mesh, named acddata000, and then to import the disk mesh for the reference length, named Urefdata000.

    * **Pre_process** This functions searches the fluid cells that are at the vicinity of the disk mesh and it is called every time step provided that the disk changes its position.

- **Subroutines for time advancing**. After the previous part is completed and the code starts advancing in time, the code computes the necessary elements involved in the turbine models at every time step, such as the interaction forces between the fluid and the turbine rotor or updates the new position of the rotor. These subroutines are called in the function Flow_Solver located in "solvers.c".

  - **Uref_ACL**
    Calculates the reference velocity (U_ref). This value corresponds to the space averaged velocity along a disk of the same diameter as the rotor and located some distance upstream of the turbine. The value is multiplied by the disk normal that points downstream.
  - **Calc_U_lagr**
    Interpolates the velocity from the fluid mesh to the Lagrangian points at the rotor model mesh.
  - **Calc_F_lagr**
    Computes the actuator line forces at each element of the Lagrangian mesh.
  - **Calc_forces_rotor**
    Computes the overall turbine forces.
  - **Calc_F_eul**
    Transfers the forces from the Lagrangian mesh to the fluid mesh.

Actuator Line Model

The actuator line model is activated by setting rotor_modeled to 3 (the reference velocity is computed within a disk located upstream of the turbine) or to 6 (the reference velocity is computed within a line mesh instead of a disk).

- **Subroutines for pre-processing**. Equivalent to the actuator disk model with the difference that the turbine blades are represented with a one-dimensional mesh and the blade profile information is required.

  - **main** Initializes variables and imports the turbine control file "Turbine.inp".
    * **ACL_read_ucd** Imports the actuator line mesh file named "acldata000".
    * **disk_read_ucd** Imports the disk mesh file for computing the reference velocity named "Urefdata000".
    * **Pre_process** This function searches the fluid cells that are at the vicinity of the actuator line mesh or the reference velocity disk/-line mesh. The function is called every time that the disk/line mesh changes its position.

    * **airfoil_ACL** Imports the airfoil information.

- **Subroutines for time advancing**.
  - **Uref_ACL**
    Calculates the reference velocity (U_ref) for the actuator line model. This value corresponds to the space averaged velocity along a disk of the same diameter as the rotor and located some distance upstream of the turbine. The value is multiplied by the disk normal that points downstream.
  - **Calc_turbineangvel**
    Calculates the rotational velocity of the turbine based on the U_ref velocity value.
  - **rotor_Rot**
    Applies a rotation to the turbine equivalent to the rotation velocity times the time step dt.
  - **Pre_process**
    Updates the new location of the turbine.
  - **refAL_Rot**
    Applies a constant rotation to the reference line located upstream of the turbine.
  - **rotor_Rot_6dof_fsi**
    If the 6 DoF FSI module is active, this function applies the same motion to the actuator line as was applied to the floating platform.
  - **Calc_U_lagr**
    Interpolates the velocity from the fluid mesh to the Lagrangian points at the rotor model mesh.
  - **Calc_F_lagr_ACL**
    Computes the actuator line forces at each element of the Lagrangian mesh.
  - **Calc_forces_ACL**
    Computes the overall turbine forces.
  - **Calc_F_eul**
    Transfers the forces from the Lagrangian mesh to the fluid mesh.

Chapter 6

Applications

6.1 3D Sloshing in a Rectangular Tank

6.1.1 Case Definition

This test aims to validate the implementation of the level set method which is used in the code to track the motion of the free surface. The test consists of a 3D sloshing of liquid in a tank with the dimension of $L \times L$ and a mean flow depth of D (see Figure 6.1). The initial free surface elevation is of Gaussian shape and is given by

$$\varrho(x, z) = D + \eta_0(x, z), \tag{6.1}$$

where

$$\eta_0(x, z) = H_0 \exp\left\{\left(x - \frac{L}{2}\right)^2 + \left(z - \frac{L}{2}\right)^2\right\}, \tag{6.2}$$

H_0 is the initial hump height, and κ is the peak enhancement factor.

In the computation, the following parameters are used: $L = 20$, $\kappa = 0.25$, and $H0 = 0.1$. The free-slip boundary conditions are applied at all boundaries. This condition is signified by the value 10 in bcs.dat. The number of grid points in the x, y, and z directions are 201, 41, and 201, respectively. Uniform grid spacing of $\Delta x = \Delta z = 0.1$ is employed in the x and z directions, while stretched grid is used in the y direction. The initial hump height (0.1 m) is resolved by approximately five vertical grid nodes. The time step used for the computation is $\Delta t = 0.001s$ and the value of ϵ (free surface thickness) is set to $0.03m$. The solution of the free surface elevation at the center of the tank is given in Figure 6.3.

Further details about the simulation as well as the analytical solution of the problem can be found in Kang and Sotiropoulos [15].

6.1.2 Main Parameters

The main parameters in the control file for setting this test case are listed in Table 6.1.2.

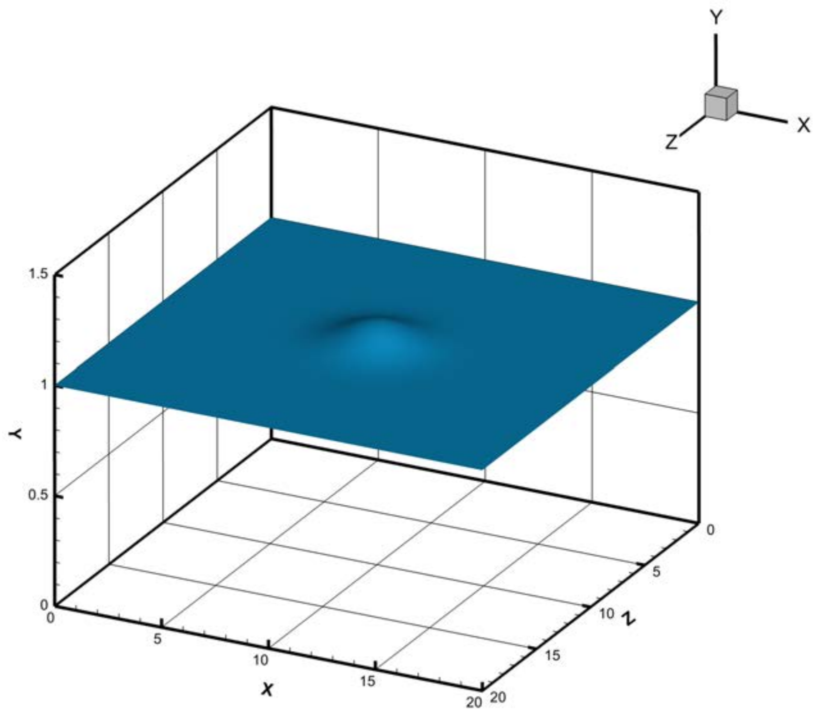


Figure 6.1: Schematic description of the domain and the initial condition of the free surface.

Table 6.1: Parameters in control.dat file for the sloshing case.

Parameter	Option in control file	Value
Time step size	dt [sec]	0.001
Read xyz.dat type mesh	xyz	1
Activate level set method	levelset	1
Set density of water and air	rho0, rho1 [kg/m3]	1000, 1
The viscosity is neglected	inv	1
Set gravity in y direction	gy [m2/s]	-9.81
Activate 3D Sloshing test case option	sloshing	2
Initialize flat free surface at elevation set by level_in_height	level_in	2
Set level set interface thickness	dthick [m]	0.03
Number of level set reinitializations	levelset_it	15
Reinitialization time step size	levelset_tau [sec]	0.05

By activating the sloshing option, two actions are implemented in the code. First, the initial condition of the distance function, and thus the free surface elevation, is set to the one corresponding to the present case. Then, at every time step after the flow solution is updated, it computes the analytical solution of the free surface position at the tank center and exports it along with the computed solution in a file named sloshing.dat described below.

Also note that because the level set method is active, the equations are solved with dimensions.

6.1.3 Validation

The output value for comparison in this test case is the time evolution of the free surface elevation at the tank center. This value can be checked at the post-processed data file containing the whole domain solution, however, this information is only available for the few exported time steps defined by the input option -tio. Another approach that is more convenient for evaluating the surface elevation in the tank center is by checking the output data file (sloshing.dat). This file exports information at every time step and consists of an ASCII data file with four columns. The first column is the simulation time [sec], the second is the computed surface elevation [m] at the tank center, the third is the expected theoretical solution also at the tank center, and the last is the error. If the purpose of running this test case is for validation only, it is not necessary to post-process any flow-field data. The surface elevation at the tank center is plotted in Figure 6.3.

Although Figure 6.3 shows the solution for 60000 time steps (60s), for validation purpose, it is not necessary to run the simulation for that long. With the proposed time step size, between 6000 and 8000 iterations (equivalent to 6 to 8 sec) should be sufficient to demonstrate that the solution is valid. As a reminder, the total number of time steps for which the code runs is specified at the control.dat file with the variable "totalsteps".

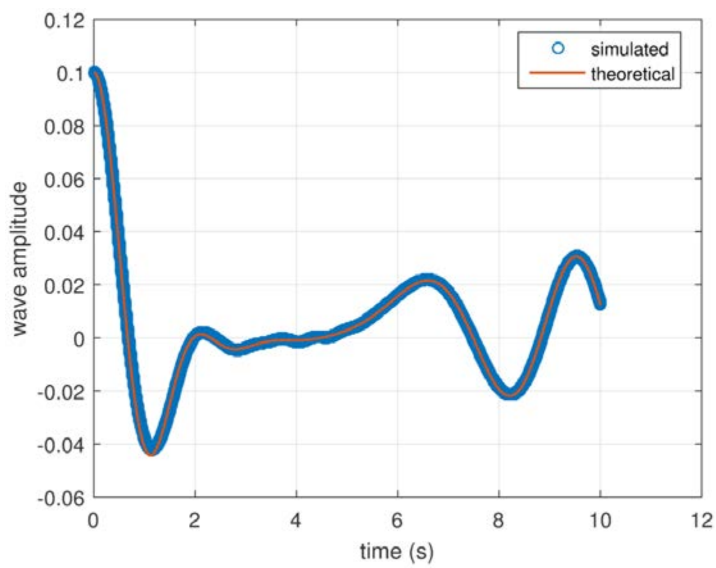


Figure 6.2: Comparison of the computed and analytic free surface elevation at the center of the tank. (symbol: computed solution, solid line: analytical solution).

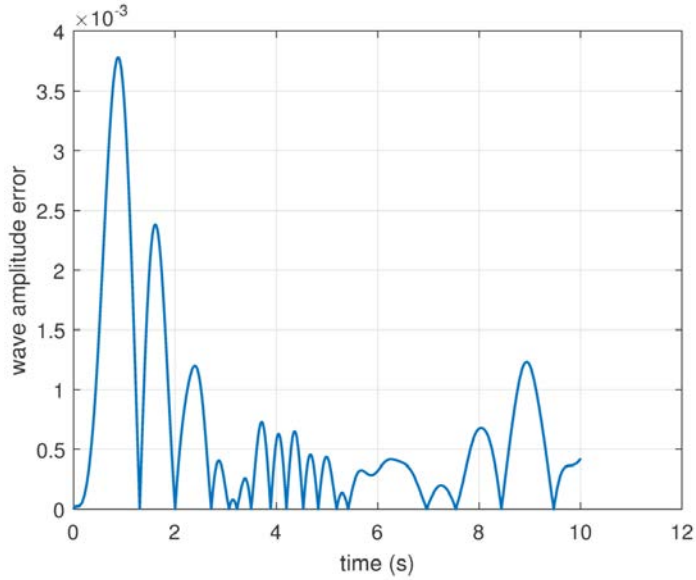


Figure 6.3: Error in the computed free surface elevation.

6.2 2D VIV of an Elastically Mounted Rigid Cylinder

6.2.1 Case Definition

Vortex induced vibration (VIV) of an elastically mounted rigid cylinder is a well-known benchmark case for validating FSI codes. The schematic description of the problem is shown in Figure 6.4. The problem consists of a 2D rigid cylinder of diameter D that is elastically mounted in a uniform flow of velocity U. The cylinder is allowed to move in the direction perpendicular to the flow with one degree of freedom. Additional details can be found in Borazjani, Ge, and Sotiropoulos [16].

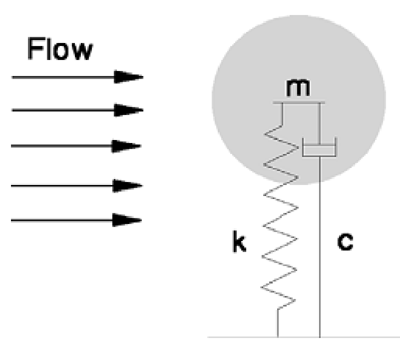


Figure 6.4: Schematic description of the elastically mounted rigid cylinder in the free stream. This figure was reproduced from [16].

A rectangular computational domain with dimensions $32D \times 16D$ is considered.

The cylinder is initially positioned at $8D$ from the inlet and centered. The boundary conditions for the side walls are slip wall (defined by 10 in bcs.dat), uniform flow is prescribed at the inlet (5 in bcs.dat), and a convective boundary condition is applied at the outlet (4 in bcs.dat). A non-uniform grid is used with 281×241 nodes in the streamwise and span-wise directions, respectively. The code requires at least 5 grid nodes (current test case employs 6) in the span-wise direction to carry out a two-dimensional simulation since the code is fully three-dimensional in addition to slip-wall boundary conditions along the span-wise walls. A square box with constant grid spacing of $0.02D$ and 50×50 nodes centered on the cylinder is used. Outside of the box the grid is gradually stretched towards the boundaries.

6.2.2 Main Parameters

The main parameters in the control file for setting this case are listed in Table 6.2. See Borazjani, Ge, and Sotiropoulos [16] for the definition and details of the parameters. In contrast to the previous case the level set method is not employed and the solved equations are all non-dimensional.

Table 6.2: Parameters in control.dat file for the VIV case.

Parameter	**Option in control file**	**Value**
Time step size	dt	0.01
Reynolds number ($RE = U * D/)$	ren	150
Reduced velocity of 6	red_vel	1.0472
Raduced mass of 2	mu_s	0.25
Cylinder damping	damp	0.0
Activate IB method	imm	1
Use of one body	body	1
Activate FSI module	fsi	1
Activate proper degree of freedom	dgf_y	1
Apply an initial translation of the cylinder to the actual position. The cylinder mesh was initially defined at the origin and needs to be translated to the desired location.	y_c, z_c	8.0, 8.0

6.2.3 Validation

The VIV case can be validated by comparing the position of the cylinder in time. Similar to the previous case, there is no need to post-process the flow field data in order to get the cylinder position. The cylinder position is provided at every time step in the FSI_position00 file. The FSI_position00 file is a text file containing information about the immersed body location, velocity, and forces for the linear degrees of freedom in the x, y, and z direction. The file consists of 10 columns as follows:

$$ti, Y_x, \dot{Y}_x, F_x, Y_y, \dot{Y}_y, F_y, Y_z, \dot{Y}_z, F_z, \tag{6.3}$$

where ti is the time step number, Y is the body position with respect to its initial position Y/D, $\dot{Y}$ is the body velocity, and F is the force that the fluid imparts to the immersed body. For this test case, 6000 time steps should be sufficient.

In the test case folder, a successful run file name _FSI_position00_good_run is provided to quickly validate the case. Figure 6.5 provides a plot showing a comparison with the provided successful run file. The maximum amplitude of the cylinder is approximately 0.49.

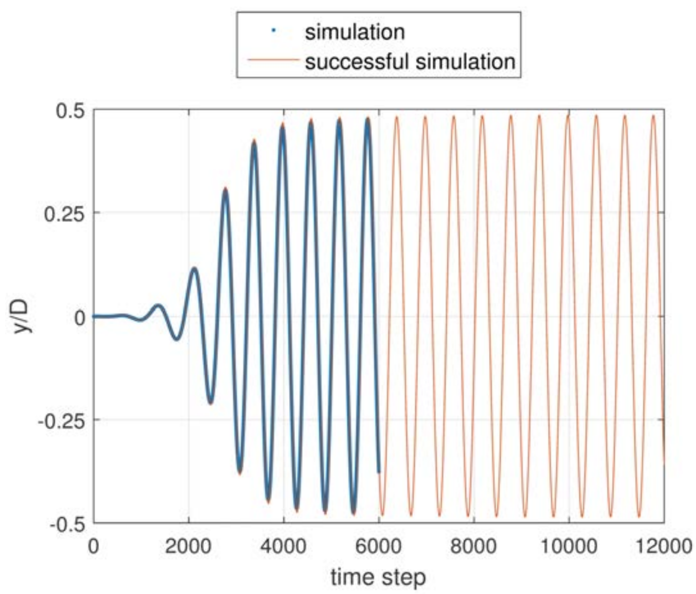


Figure 6.5: Displacement of the cylinder for a successful simulation.

6.3 2D Falling Cylinder (Prescribed Motion)

6.3.1 Case Definition

This test case is to validate the integration of the CURVIB and level set methods. The case involves a body moving across the air/water interface with prescribed motion. Note that the FSI algorithm is not used currently. This test case considers an infinitely long 2D cylinder of radius $R = 1m$ moving with constant downward speed in an infinitely wide domain and crossing the free surface from a gas to liquid phase.

The configuration provided currently is identical to that simulated by Yang and Stern [28] where an identical cylinder, initially positioned above the free surface at a distance $h = 1.25m$, moves downwards with constant velocity of $u = -1m/s$. The liquid and gas densities are $\rho_0 = 1kg/m^3$ and $\rho_1 = 1 \times 10^3 kg/m3$, respectively. Both the liquid and gas are considered inviscid, while the gravity is set to $gz = -1m/s^2$ and the time is normalized as $T = ut/h$.

The 2D computational domain is $40R \times 24R$ in the horizontal and vertical directions, respectively. A non-uniform grid consisting of an inner region, centered on the cylinder, within which the mesh is uniform and an outer region where the grid is gradually stretched. The inner region is the rectangular domain defined by $[-5, 5]$ and $[-4, 2.6]$ in the horizontal and vertical directions, respectively. Within this inner domain uniform grid spacing is employed along both directions, which is equal to $0.05R$. Outside of this inner domain the mesh is stretched gradually away from the cylinder using the hyperbolic stretching function with a stretching ratio kept below 1.05. The total number of nodes is $360 \times 255 \times 6$. A time step of $0.01s$ and a free surface thickness ϵ of $0.04m$ are used. For more detail about this test case refer to Calderer et al. [2]

6.3.2 Main Parameters

The main parameters in the control file for setting this case are listed in Table 6.3.

Table 6.3: Parameters in control.dat file for the 2D falling cylinder with prescibed motion test case.

Parameter	**Option in control file**	**Value**
Time step size	dt [s]	0.01
Activate level set method	levelset	1
Set density of liquid and gas	rho0, rho1 [kg/m3]	1, 0.001
Set gravity in z direction	gz [m/s2]	-1.0
Thickness of the free surface interface	dthick [m]	0.04
Initialize flat free surface at elevation set by level_in_height	level_in	1
Initial free surface elevation	level_in_height [m]	0.0
Set levelset interface thickness	levelset_it	10
Number of level set reinitializations	levelset_tau [s]	0.05
Activate IB method	imm	1
Use of one body	body	1
Activate fluid structure interaction module	fsi	1
Activate prescribed motion function	forced_motion	1
Activate Falling cylinder case. This option basically prescribes the velocity of the body to be constant and downward.	fall_cyll_case	1
Activate proper degree of freedom	dgf_ay	1
Initial translation of the ib at the right position	z_c	1.25

6.3.3 Validation

The falling cylinder case can be validated by observing the free surface elevation at four instances in time as shown in Figure 6.6. These plots are from the post processed cylinder position (Nv) and free surface (level = 0) at time step T = 125, 250, 375, and 500.

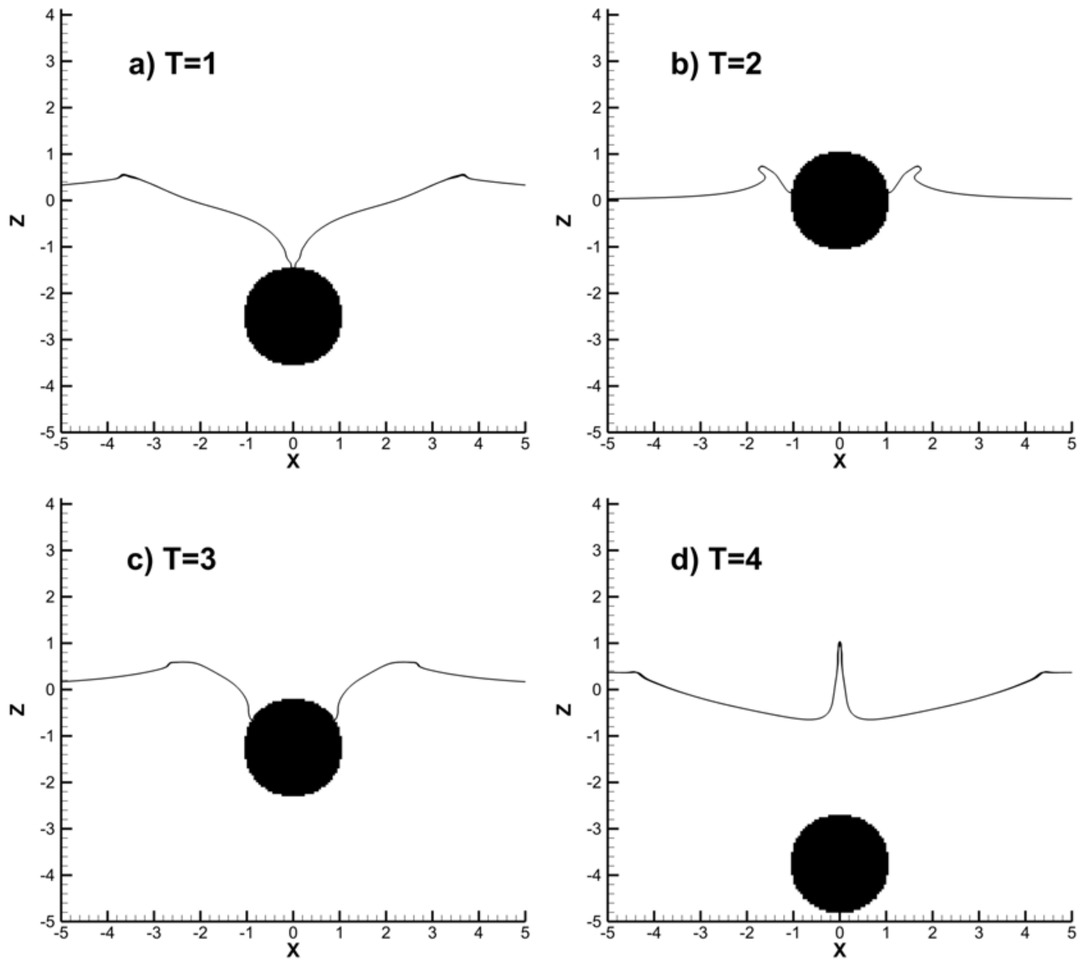


Figure 6.6: Water entry of a horizontal circular cylinder moving with prescribed velocity. Free surface position at different non-dimensional times T calculated by the present method.

6.4 3D Heave Decay Test of a Circular Cylinder

6.4.1 Case Definition

To validate the coupled FSI algorithm for simulating complex floating structures a free heave decay test of a horizontal cylinder is provided. This same test case was studied experimentally by Ito [29]. A horizontal circular cylinder of diameter $D = 0.1524m$ and density $\rho = 500kg/m^3$ is partially submerged with its center positioned $0.0254m$ above the free surface of a rectangular channel (see Figure 6.7 for a schematic representation).

The computational domain is a $27.4m$ long, $2.59m$ wide, and $1.22m$ deep channel with initially stagnant water. The cylinder movement is restricted to the vertical degree of freedom while allowed to oscillate freely. A non-uniform grid is employed with $436 \times 8 \times 261$ nodes in the horizontal, vertical, and span-wise directions, respectively.

The grid is uniform with spacing equal to $0.02D$ in a rectangular region centered around and containing the body defined by $[0.3, 0.3]$ in the horizontal direction and $[0.2, 0.2]$ in the vertical direction. The grid is stretched using a hyperbolic function, where the ratio never exceeds 1.05, in the domain outside of the uniform grid region. The interface thickness used is $0.065D$ and the simulation was carried out employing the loose coupling FSI algorithm and a time step size of $0.0005s$.

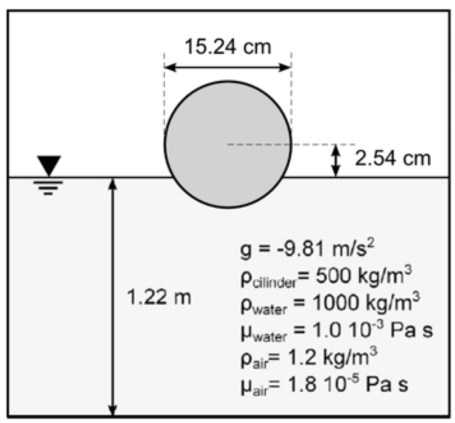


Figure 6.7: Schematic description of the cylinder case. This figure was reproduced from [2].

6.4.2 Main Parameters

The main parameters in the control file for setting this case are listed in table 6.4.

6.4.3 Validation

In the heave decay test case of a horizontal cylinder it is simplest to compare the vertical position of the cylinder in time as shown in Figure 6.8 using the FSI_position00 file's vertical position column as described in the VFS Manual Section $3.3.3 and in the test case of the 2D VIV of an elastically mounted rigid cylinder in section $6.2. An FSI_position00_good_run file is provided for comparison and validation.

Table 6.4: Parameters in control.dat file for the 3D heave decay of a circular cylinder test case.

Parameter	**Option in control file**	**Value**
Time step size	dt [s]	0.0005
Activate Dynamic Smagorinski LES model	les	2
Activate level set method	levelset	1
Set density of water and air	rho0, rho1 [kg/m3]	1000, 1.0
Set viscosity of water and air	mu0, mu1 [Pa s]	1e-3, 1.8e-5
Set gravity in z direction	gy [m/s2]	-9.81
Thickness of the free surface interface	dthick [m]	0.006
Initialize flat free surface at elevation set by level_in_height	level_in	2
Initial free surface elevation	level_in_height [m]	0.0
Set level set interface thickness	levelset_it	20
Number of level set reinitializations	levelset_tau [s]	0.01
Activate IB method	imm	1
Use of one body	body	1
Activate fluid structure interaction module	fsi	1
Activate proper degree of freedom	dgf_y	1
Mass of the cylinder	body_mass [kg]	23.63
Initial translation of the ib at the right position	y_c [m]	0.0254

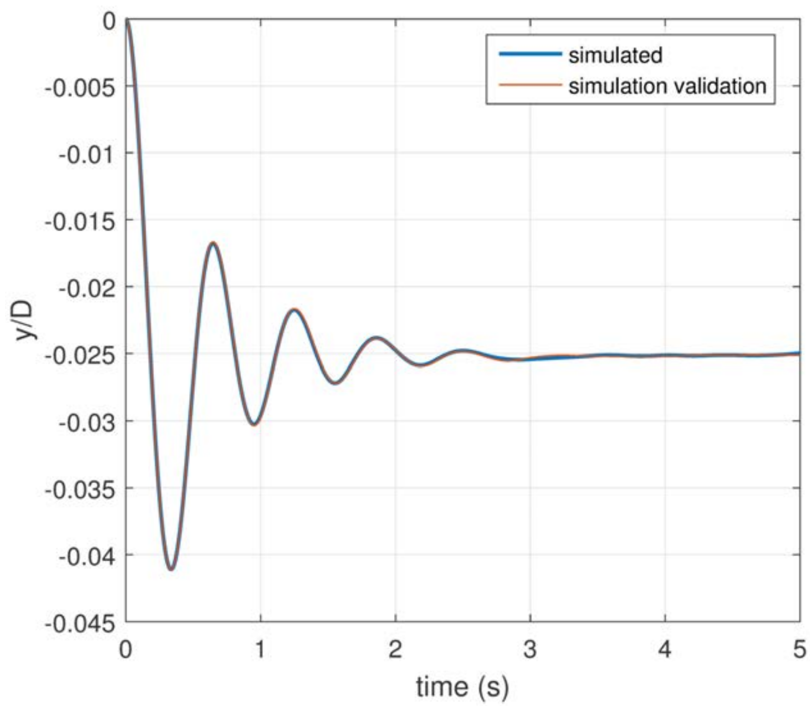


Figure 6.8: Normalized position of the cylinder in time computed with the present code.

6.5 2D Monochromatic Waves

6.5.1 Case Definition

The simulated generation and propagation of a progressive monochromatic wave in a 2D rectangular channel is used to validate the wave generation method of Guo and Shen [30] implemented in this code. A linear wave of amplitude $0.01m$ and wavelength of $1.2m$, for which the analytic solution is known from linear wave theory, is considered. A two-dimensional domain of length equal to $24m$, water depth of $2m$, air column above the water of $1m$, and gravity equal to $gy = -9.81m/s^2$ is simulated using a non-uniform grid size of 40×177 in the longitudinal and vertical direction, respectively. The grid is uniform in a rectangular region centered on $z = 0$ and spanning $24m$ along the z direction ($[12, 12]$), and containing the free surface along the vertical direction y ($[0.15, 0.15]$). Within this region the grid spacing is 0.06 and 0.005 in the horizontal and vertical directions, respectively, while outside of this region the grid spacing increases progressively with a stretching ratio limited to 1.05.

The time step of the simulation is $0.002s$ with the thickness of the interface set to $0.02m$. The source region is centered on the origin. Sponge layers with length equal to $3m$ are defined at the two ends of the computational domain and slip boundary conditions are implemented at the bottom and top boundaries (10 in bcs.dat).

Details of the method implementation are given in Calderer et al. [3]. The analytical solution is

$$\eta(z, t) = Acos(\omega t - kx), \tag{6.4}$$

where η is the surface elevation, A is the wave amplitude, $k = 2\pi/L$ is the wave number, d is the water depth, and ω is the angular frequency solved for with the dispersion relation as follows:

$$\omega = \sqrt{kgtanh(kd)}. \tag{6.5}$$

6.5.2 Main Parameters

The main parameters in the control file for setting this case are listed in Table 6.5.

6.5.3 Validation

The free surface elevation (height of level = 0 in the post-processed data files) and its comparison with the analytical solution are presented in Figure 6.12. Note that in the source region the computed results are not expected to follow the analytical free surface pattern.

Table 6.5: Parameters in control.dat file for the 2D monochromatic wave test case.

Parameter	**Option in control file**	**Value**
Time step size	dt [s]	0.002
Activate level set method	levelset	1
Set density of water and air	rho0, rho1 [kg/m3]	1000, 1.0
Set viscosity of water and air	mu0, mu1 [Pa s]	1e-3, 1.8e-5
Set gravity in z direction	gy [m/s2]	-9.81
Thickness of the free surface interface	dthick [m]	0.02
Initialize flat free surface at elevation set by level_in_height	level_in	1
Initial free surface elevation	level_in_height [m]	0.0
Set levelset interface thickness	levelset_it	15
Number of level set reinitializations	levelset_tau [s]	0.05
Activate the wave generation method for single wave frequency	wave_momentum_source	2
Time step for which wave generation method begins	wave_ti_start	1
Wave Direction. If 0rad waves travel in x direction	wave_angle_single [rad.]	0
Wavenumber	wave_K_single [1/m]	5.23599
Water depth	wave_depth [m]	2
Wave amplitude	wave_a_single [m]	0.01
Activate wave sponge layer method at the x boundaries for wave suppression.	wave_sponge_layer	1
Length of the sponge layer	wave_sponge_xs [m]	3
Starting coordinate of the first x sponge layer.	wave_sponge_x01 [m]	-12
Starting coordinate of the second sponge layer.	wave_sponge_x02 [m]	9

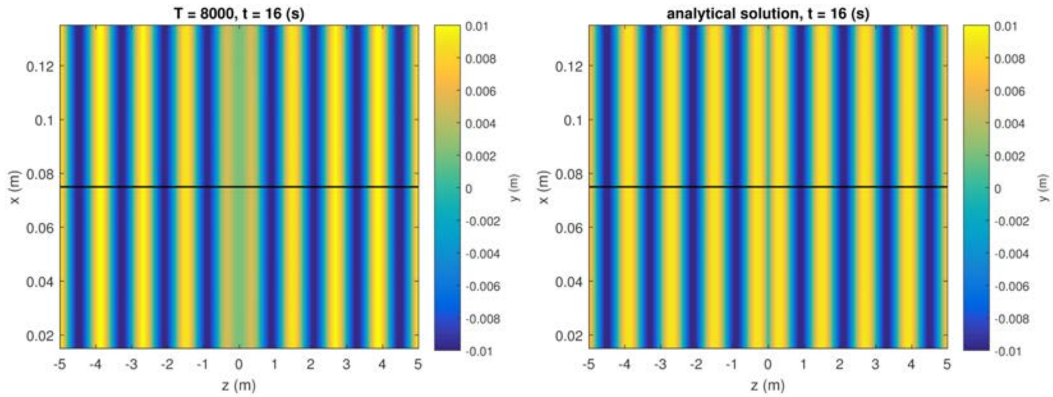


Figure 6.9: Generation of monochromatic waves. Contours of free surface elevation, computed (left) and analytical solution (right).

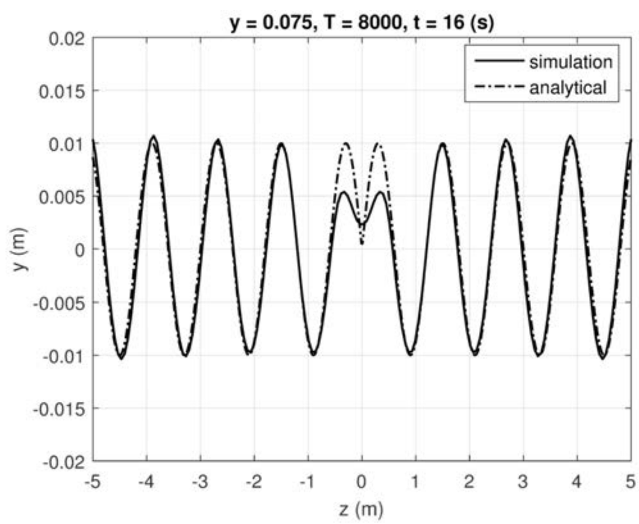


Figure 6.10: Generation of monochromatic waves. Computed and analytical free surface elevation.

6.6 3D Directional Waves

6.6.1 Case Definition

This test case validates the implemented forcing method for wave generation by generating a linear directional wave field in a 3D basin of constant depth. The wave amplitude is $A = 0.01m$, the wavelength is $L = 1.2m$, and the wave direction is $\beta = 25deg$. The domain length is $24m$ ($-12m \leq z \leq 12m$) in the longitudinal direction, $12m$ ($-6m \leq x \leq 6m$) in the span-wise direction, and the depth of water and air is $2m$ and $1m$, respectively. A non-uniform grid size of $121 \times 139 \times 201$ in the x, y, and z directions, respectively, is employed consisting of an inner rectangular region with uniform grid spacing and an outer region within which the mesh is gradually stretched towards the boundaries. The inner region ($-6m \leq z \leq 6m$, $-6m \leq x \leq 6m$, $-0.1m \leq y \leq 0.1m$), which contains the source region and part of the propagated waves, has a constant grid spacing of $0.1m$, $0.1m$, and $0.005m$, in the x, y and z directions, respectively. The source region is centered on $z = 0$ and spans the entire domain along the x direction. The time step is $0.005s$, the interface thickness is $0.02m$, and the gravity is $g = -9.81m/s^2$. The sponge layer method with length $3m$ is applied at the Z boundaries and periodic boundary conditions is applied at the X boundaries. Details of the method implementation are given in Calderer et al. [3]. The analytical solution is

$$\eta(z, x, t) = Acos(\omega t - k_x x - k_z z) \tag{6.6}$$

where η is the surface elevation, A is the wave amplitude, $k = 2\pi/L$ is the wave number, $k_x = kcos(\beta)$, $k_y = ksin(\beta)$, d is the water depth, and ω is the angular frequency solved for with the dispersion relation as follows

$$\omega = \sqrt{kgtanh(kd)}. \tag{6.7}$$

6.6.2 Main Parameters

The main parameters in the control file for setting this case are listed in Table 6.6.

6.6.3 Validation

The free surface elevation (height of level = 0) and its comparison with the analytical solution is presented in Figure **??**.

Table 6.6: Parameters in control.dat file for the 3D directional wave test case.

Parameter	**Option in control file**	**Value**
Time step size	dt [s]	0.002
Activate level set method	levelset	1
Set density of water and air	rho0, rho1 [kg/m3]	1000, 1.0
Set viscosity of water and air	mu0, mu1 [Pa s]	1e-3, 1.8e-5
Set gravity in z direction	gy [m/s2]	-9.81
Thickness of the free surface interface	dthick [m]	0.04
Initialize flat free surface at elevation set by level_in_height	level_in	2
Initial free surface elevation	level_in_height [m]	0.0
Set number of pseudo time steps to solve the reinitialization equation for the level set method	levelset_it	15
Number of level set reinitializations	levelset_tau [s]	0.05
Activate the wave generation method for single wave frequency	wave_momentum_source	2
Wave Direction. If set to $0rad$ waves travel in z direction	wave_angle_single [rad.]	0.5235988
Wavenumber	wave_K_single [1/m]	5.23599
Water depth	wave_depth [m]	2.0
Wave amplitude	wave_a_single [m]	0.01
Activate wave sponge layer method at the x boundaries for wave suppression.	wave_sponge_layer	1
Length of the sponge layer	wave_sponge_zs [m]	1.2
Starting coordinate of the first z sponge layer.	wave_sponge_z01 [m]	-12.0
Starting coordinate of the second sponge layer.	wave_sponge_z02 [m]	10.6

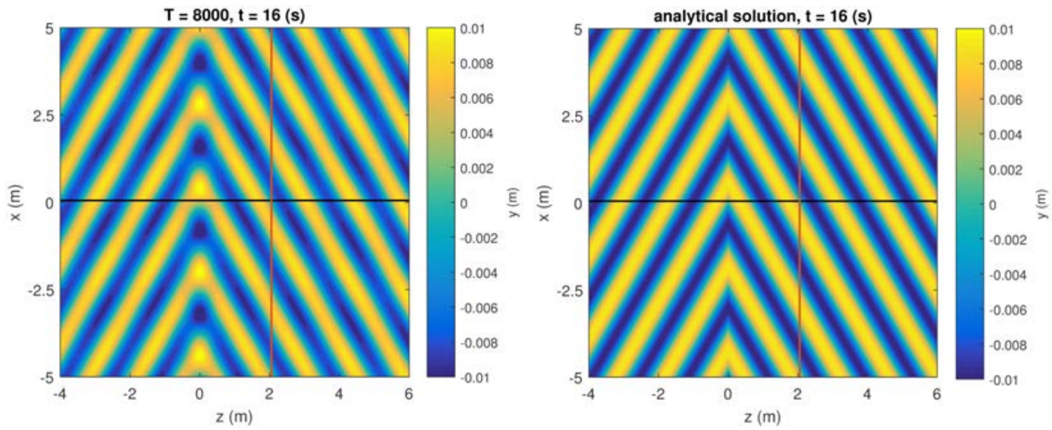


Figure 6.11: Generation of directional progressive waves in a 3D tank. Contours of computed (left) and analytical (right) free surface elevation.

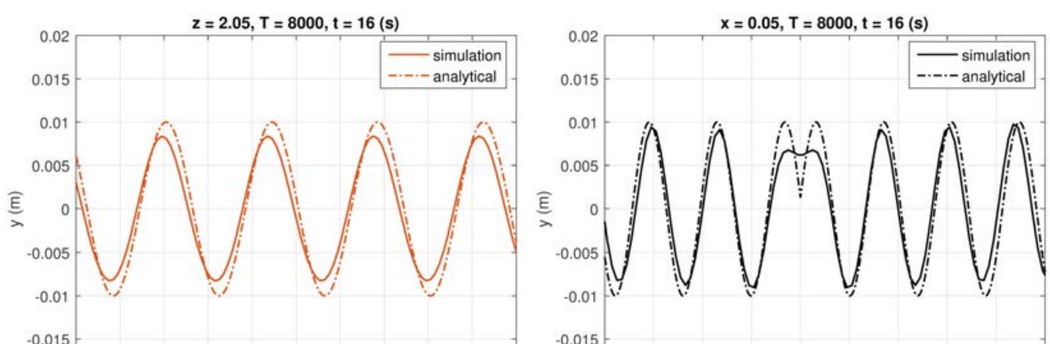


Figure 6.12: Generation of directional progressive waves in a 3D tank. Profiles of surface elevation.

6.7 Floating Platform Interacting with Waves

This test case involves a barge style floating platform model that is installed in the Saint Anthony Falls Laboratory main channel. The channel acts as a wave basin.

The floating platform has a cylindrical shape and is allowed to move in two degrees of freedom, pitch and heave. Rotations are in respect to the center of gravity of the structure. The platform has two small cylindrical masses located underneath which are also considered in this simulation.

In this particular case we study the response of the structure under a given incoming wave frequency. Monochromatic waves of amplitude $0.00075m$ and wave-number 0.8039 are generated at the source region located at $z = 0$. The platform is located at $z = 30m$ and oscillates as a response of the forces induced by waves both in the vertical direction Y and rotating along the X axis. Using the dispersion relation and considering that the water depth is $1.37m$ the wave period is $T = 2.5s$.

The value that we are interested for validating this simulation is the maximum amplitude of the oscillation in both the pitch and heave directions. These can be seen at the files FSI_angle00 and FSI_position00. Figure 6.13 shows the experimental results for several incoming wave frequencies known as Response amplitude operator (RAO). In Figure 6.13 the values are normalized by the incident wave height as follows

$$RAO_{heave} = Y_{max}/H_{wave} \tag{6.8}$$

and

$$RAO_{pitch} = \phi_{max}/H_{wave} \tag{6.9}$$

where Y_{max} is the maximum vertical displacement measured from peak to peak, H_{wave} is the incident wave height, and ϕ_{max} is the maximum rotation angle between two consecutive peaks.

Note that for this case the computed wave amplitude may be slightly inferior to that specified at the control file. This result is due to the fact the waves are in a shallow water regime. This fact will not alter the results as long as the actual simulated wave height is considered in the normalization.

6.8 Clipper Wind Turbine

6.8.1 Case Definition

This test case is for introducing and validating the actuator line model for turbine parameterization. The test case involves the simulation of the Clipper Liberty 2.5MW wind turbine which was built during the EOLOS project and is located in UMore Park, Rosemount, MN. The turbine has a diameter of $96m$ and a hub height of $80m$. Details can be found in [11]. For validation purpose we propose the case with uniform inflow which makes the case simple as it does not require any precursor simulation and the boundary conditions at the top wall, bottom wall and side walls are free slip. Two cases with different TSR, 5.0 and 8.0, are tested.

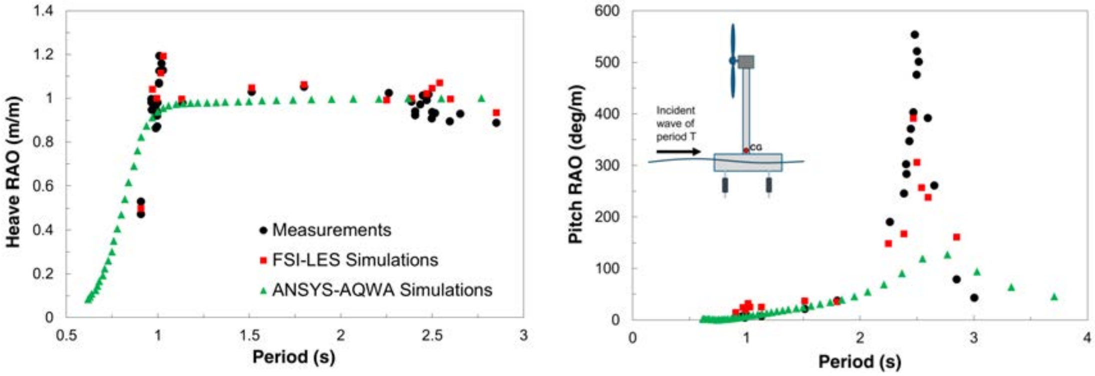


Figure 6.13: Response amplitude operator for the floating turbine.

The simulation is carried in a non-dimensional form being the reference length the turbine diameter. The grid dimensions are 2 units in the y and x directions, and 4 units in the z direction, which corresponds to a dimensional domain of 192m and 384m, respectively. The grid is uniform and the spacing is 0.05 units in all three directions which is equivalent to a grid size of $41 \times 41 \times 81$. Note that the grid file (grid.dat or xyz.dat) has already non-dimensionalized size. The simulation is set with a unit non-dimensional velocity equivalent to a real velocity of 8m/s.

In contrast to the fluid mesh, the actuator line mesh (acldata000) can be constructed with the real turbine dimenions ($96m$) and non-dimensionalized by setting the turbine reference length option "reflength_wt" in control.dat to 96.0. This will divide the actuator line mesh dimensions by "reflength_wt". Alternatively one could generate a rotor mesh already with the non-dimensionalised units and choose "reflength_wt" equals to 1.0.

6.8.2 Main Case Parameters

Since the main solver parameters have already been discussed in previous test cases, only the parameters related to the wind turbine rotor model will be addressed. When using the rotor model the control options for setting the case are located not only in control.c but also in Turbine.inp. The former parameters are summarized in Table 6.7, with the latter in Table 6.8. The parameters in Turbine.inp that are not discussed in the Table 6.8 are not used in the simulation.

This test case requires averaging, and therefore has to be executed in two stages as described in Section $4.2.1. In the first stage the flow is fully developed, while in the second stage the time averaging is performed.

For developing the flow field, it is sufficient to perform 5000 time steps. For time-averaging the flow field, 2000 time steps are enough. For time-averaging, set the option in the control file "averaging"" to 3, rstart to 0, and rename the result files from the last instantaneous time step (ufield005000.dat, vfield005000.dat, pfield005000.dat, nvfield005000.dat, and cs_field005000.dat) to the value corresponding to time zero (

Table 6.7: Parameters in control.dat file for the Clipper wind turbine.

Parameter	**Option in control file**	**Value**
Time step size	dt [s]	0.0025
Activate the actuator line model	rotor_modeled	6
Number if blades in the rotor	num_blade	3
Number of foil types along the blade	num_foiltype	5
Number of turbines	turbine	1
Reference length of the turbine	reflength_wt	96.
Nacelle diameter	r_nacelle	1.3
Reference velocity of the case. It is only used for dimensionalizing the turbine model output files	refvel_wt	8.0
Distance upstream of the turbine in rotor diameters where the turbine incoming velocity or reference velocity is computed	loc_refvel	0.5
Direction to which the turbine points to	rotatewt	1
Type of smoothing delta function	deltafunc	10
Delta function width in cell units	halfwidth_dfunc	2.0
Activate constant turbine rotation mode	fixturbineangvel	1

Table 6.8: Parameters in Turbine.inp file for the Clipper wind turbine.

Parameter	**Option in Turbine.inp file**	**Value**
Normal direction of the turbine rotor plane.	nx_tb, ny_tb, nz_tb	0.0 0.0 1.0
The turbine rotor initial translation.	x_c, y_c, z_c	0.0 0.0 0.0
Tip speed ration	Tipspeedratio	5
Radius of the rotor corresponding to the acldata000 mesh	r_rotor	48.0
Tip speed ratio when FixTipSpeedRatio option in control.dat is active, otherwise is not used	TSR_max	8.3
Rotor angular velocity when fixturbineangvel option in control.dat is active, otherwise is not used	angvel_fixed	10.0
Angle of pitch of the blades in degrees	pitch[0]	1.0

ufield000000.dat, vfield000000.dat, ...).

Also, when restarting the flow field for averaging, set rstart_turbinerotation as 1 and rename TurbineTorqueControl005001_000.dat as TurbineTorqueControl000001_000.dat.

6.8.3 Validation

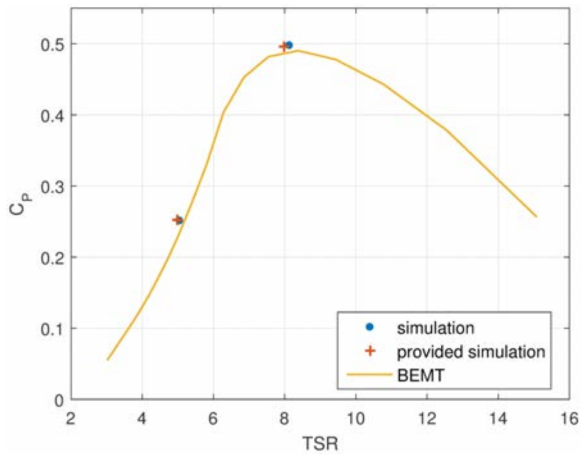


Figure 6.14: Cp coefficient of the Clipper turbine as a function of the TSR.

For validation the case the power coefficient (Cp) of the turbine is compared with the theoretical power coefficient obtained using blade element momentum theory.

A folder named "Tecplotfiles" is provided to assist in generating the Cp comparison. Executing the tecplot macro file ReadSaveCp.mcr a file CP.txt is created with the averaged Cp coefficient. The CP value for the TSR 5 and TSR 8 cases is 0.25 and 0.50, respectively, as shown in figure 6.14.

6.9 Model Wind Turbine Case

6.9.1 Case Definition

This test case is to further demonstrate the validity of the actuator line model by analyzing the turbulence statistics in the turbine wake. The test case consists of a miniature wind turbine simulation with geometry equivalent to the model turbine tested experimentally in the Saint Anthony Falls wind tunnel by [31]. The diameter of the turbine is $0.15m$ and the turbine hub height is $0.125m$. The rotor blade cross section is approximated as a NACA0012, although the real blade geometry is unknown. The tower is neglected and the nacelle is modeled by extending the actuator line effect to the center point of the rotor. The inflow velocity at hub height is $2.2m/s$ and the TSR is 4.1. Reynolds number based on rotor diameter D and the incident velocity at the hub height U_{hub} is 4.2×10^4. For more details about the case and the turbine geometry see [22].

The computational domain dimensions are $30D$ in the streamwise direction, $12D$ in the spanwise direction, and $3D$ in the vertical direction. The turbine is positioned $2D$ after the inlet plane and centered on the transverse direction. The grid is considered uniform along the spanwise and vertical directions. In the streamwise direction, the mesh is uniform from the inlet to 12D downstream and stretched towards the outlet. Grid size is $201 \times 121 \times 31$, which is equivalent to 10 grid points per turbine diameter.

The velocity profile specified at the inlet is from a pre-computed simulation consisting of a channel flow. An example of such channel flow simulation is provided in the test case in Section $6.10. The channel flow case presented in $6.10 is illustrative of how to prepare fully developed inlet flow conditions, although the case parameters may not exactly coincide with the precursor simulation used for the present simulation.

The bottom wall cannot be resolved with the current grid resolution and is treated with a wall model.

6.9.2 Main Case Parameters

Table 6.9: Parameters in control.dat file for the wind turbine model simulation.

Parameter	**Option in control file**	**Value**
Time step size	dt [s]	0.001
Activate the actuator line model	rotor_modeled	6
Number if blades in the rotor	num_blade	3
Number of foil types along the blade	num_foiltype	2
Number of turbines	turbine	1
Reference length of the turbine	reflength_wt	1.5
Nacelle diameter	r_nacelle	0.0
Reference velocity of the case. It is only used for dimensionalizing the turbine model output files	refvel_wt	8.0
Distance upstream of the turbine in rotor diameters where the turbine incoming velocity or reference velocity is computed	loc_refvel	1.0
Direction to which the turbine points to	rotatewt	1
Type of smoothing delta function	deltafunc	0
Delta function width in cell units	halfwidth_dfunc	2.0
Activate constant turbine rotation mode	fixturbineangvel	1

As in the previous test case, time averaging is required (see Section $4.2.1). For developing the flow field, it is sufficient to perform 5000 time steps. For time-averaging the flow field, 2000 time steps are sufficient. For time-averaging, set the option in the control file "averaging"" to 3, rstart to 0, and rename the result files from the last instantaneous time step (ufield005000.dat, vfield005000.dat, pfield005000.dat, nvfield005000.dat, and cs_field005000.dat) to the value corresponding to time zero (ufield000000.dat, vfield000000.dat, etc.).

Table 6.10: Parameters in Turbine.inp file for the wins turbine model simulation.

Parameter	**Option in Turbine.inp file**	**Value**
Normal direction of the turbine rotor plane.	nx_tb, ny_tb, nz_tb	0.0 0.0 1.0
The turbine rotor initial translation.	x_c, y_c, z_c	1.0 0.0833 0.2
Tip speed ration	Tipspeedratio	4.0
Radius of the rotor corresponding to the acldata000 mesh	r_rotor	0.025
Tip speed ratio when FixTipSpeedRatio option in control.dat is active, otherwise is not used	TSR_max	8.3
Angle of pitch of the blades in degrees	pitch[0]	1.0

Also, when restarting the flow field for averaging, set rstart_turbinerotation as 1 and rename TurbineTorqueControl005001_000.dat as TurbineTorqueControl000001_000.dat.

6.9.3 Validation

In this test case, vertical profiles of velocity and turbulence statistics at different downstream locations are compared with measured data from the experiment of [31]. To facilitate the creation of these figures a Tecplot macro file named "ExtractLines_3D.mcr", that automatically generates the comparison figures, is provided in the sub-folder "Tecplotfiles". The macro file calls the averaged results file "ResultsXXXXXX.plt" (XXXXXX refers to the time step), which may have to be edited based on the results available. As an example, figure 6.15 shows the vertical profiles of averaged velocity, Reynolds shear stresses, and turbulence intensity $5D$ behind the turbine.

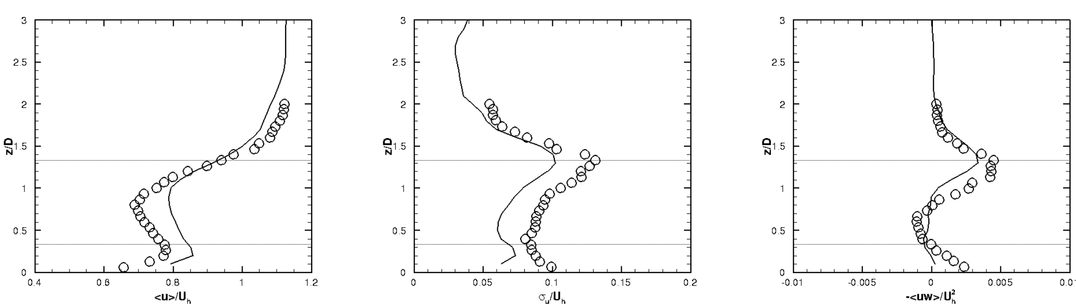


Figure 6.15: Vertical profiles of averaged stream-wise velocity (left), Reynolds stresses (center), and Turbulence intensity (right) at a distance $5D$ behind the turbine.

6.10 Channel Flow

6.10.1 Case Definition

This is the classical channel flow case as extensively described in [32]. A rectangular duct of height $H = 2h$ with uniform flow is considered. This test case can have a dual purpose: (1) to test the wall model at the bottom and/or top walls, or (2) as a precursor simulation to generate fully developed turbulent flow conditions to be fed at the inlet of other cases such as the wind turbine model case in section $6.9.

The domain is $1.2m$ in the spanwise direction (x), $0.4m$ in the vertical direction (y), and $2m$ in the streamwise direction. A uniform grid is used with size $121 \times 41 \times 61$. Note that the height of $0.4m$ corresponds to the channel half height and slip-wall boundary conditions are used at the top boundary. Fully developed turbulent boundary conditions can be achieved using periodic boundary conditions in the streamwise and spanwise directions and no slip wall boundary condition at the bottom boundaries.

The flow can be characterized with the Reynolds number defined as

$$Re = 2\delta U/\nu, \tag{6.10}$$

where δ is the channel half height and U is the bulk velocity. The flow case can also be defined with the friction Reynolds number defined as

$$Re_\tau = u_\tau \delta/\nu, \tag{6.11}$$

with $u_\tau = \sqrt{\tau_w/\rho}$, and τ_w the shear stress at the wall. The Reynolds number flow can be related to the friction Reynolds number with the following expression

$$Re_\tau \approx 0.09 Re^{.88}. \tag{6.12}$$

In the present simulation the friction Reynolds number is $Re_\tau = 3000$ which using (6.12) corresponds to a Reynolds number flow of $Re = 137900$. With the kinematic viscosity of air taken as $\nu = 1.6 \times 10^{-5}$, using equation (6.10) the flow bulk velocity is 2.7584.

6.10.2 Main Case Parameters

The main parameters used in the present simulation control file are summarized in Table 6.11.

6.10.3 Validation

This test case is validated by comparing the vertical time-averaged profiles of streamwise velocity and turbulence intensity. Time averaging is performed as described in Section $4.2.1 or as shown in previous test cases. The flow is first executed for about

Table 6.11: Parameters in control.dat file for the channel flow simulation.

Parameter	Option in control file	Value
Time step size	dt [s]	0.001
This parameter corresponds to $1/\nu$	ren	62500
Dynamic LES modelling active	les	2
Periodic boundary conditions in the streamwise and spanwise directions	kk_periodic, ii_periodic	1, 1
Initial condition set to uniform flow	inlet	1
The inlet flux to obtain a bulk velocity of 2.785. This takes in account the domain cross section area which is $0.48m^2$	flux	1.324
Introduces a random perturbation to the initial velocity field	perturb	1
Activate the wall model at the bottom boundary	viscosity_wallmodel	1
When active it exports to an external file the velocity field at the inlet plane at every time step. First it is set to 0 while the flow is developed. In the second stage the case is reinitialized with this option active.	save_inflow	0, 1
This parameter defines the number of times steps that are stored in an individual file	save_inflow_period	500
Folder where to store the exported velocity files. The code will create in the specified directory a folder named inflow.	path_inflow	"./"

4000 times steps to achieved developed conditions (check the file Kinetic_Energy.dat to ensure that the kinematic energy is stabilized). Then rename the flow field files to the corresponding to time step zero and restart the simulation with the time averaging option active. Also, when restarting, activate the option save_inflow to export the velocity field into the inlet.

A tecplot macro file is provided in the test case folder (ExtractLines_3D.mcr) which extracts vertical profiles of the data from the post-processed data file (Result010000-avg.plt). About 5000 times steps may be sufficient for time averaging although 10000 may provide smoother results. One can use the option ikavg equal to 1 to do space averaging in the streamwise and spanwise directions, when post-processing the average results in addition to the avg equals to 1 option.

Also note that the code output file provides the actual computed values of Re_{tau}

and u_τ. These values are printed at every time step and can be found, at the output file, by searching for the word "Bottom". The two values, are indicated as u* and Re*, respectively. However, the Re* value printed by the code for this case is not the real Re_{tau}. The code assumes that the vertical dimension is 1. So, to obtain the real value of Re_{tau}, the given Re* has to be multiplied by the channel half height δ, equals to 0.4 for this case. These two values are not exact and tend to oscillate in time. Errors of about 10% are within an admissible range.

The averaged stream-wise velocity profile can be compared with the logarithmic law of the wall (see figure 6.16) given by

$$u^+ = \frac{1}{\kappa}\ln(y^+) + 5.0 \tag{6.13}$$

where $\kappa \approx 0.41$ is the Von Karman constant, $u^+ = u/u_\tau$, and $y^+ = yu_\tau/\nu$.

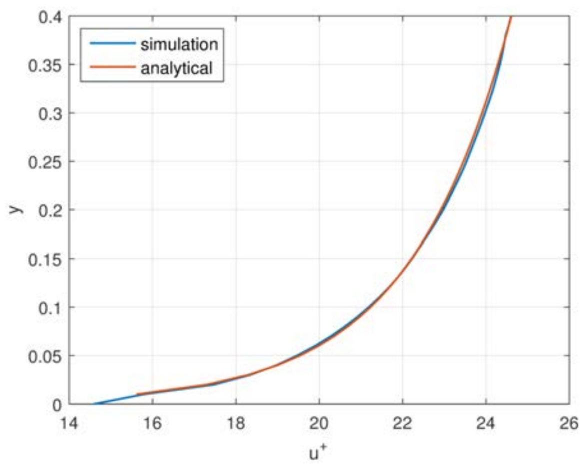


Figure 6.16: Vertical profile of averaged stream-wise velocity.

Bibliography

[1] L. Ge and F. Sotiropoulos, “A numerical method for solving the 3D unsteady incompressible NavierStokes equations in curvilinear domains with complex immersed boundaries,” *Journal of Computational Physics*, vol. 225, pp. 1782–1809, Aug. 2007.

[2] A. Calderer, S. Kang, and F. Sotiropoulos, “Level set immersed boundary method for coupled simulation of air/water interaction with complex floating structures,” *Journal of Computational Physics*, vol. 277, pp. 201–227, 2014.

[3] A. Calderer, X. Guo, L. Shen, and F. Sotiropoulos, “Coupled fluid-structure interaction simulation of floating offshore wind turbines and waves: a large eddy simulation approach,” in *Journal of Physics: Conference Series*, vol. 524, p. 012091, IOP Publishing, 2014.

[4] T. B. Le and F. Sotiropoulos, “Fluid–structure interaction of an aortic heart valve prosthesis driven by an animated anatomic left ventricle,” *Journal of computational physics*, vol. 244, pp. 41–62, 2013.

[5] T. B. Le, *A computational framework for simulating cardiovascular flows in patient-specific anatomies.* PhD thesis, UNIVERSITY OF MINNESOTA, 2011.

[6] I. Borazjani and F. Sotiropoulos, “The effect of implantation orientation of a bileaflet mechanical heart valve on kinematics and hemodynamics in an anatomic aorta,” *Journal of biomechanical engineering*, vol. 132, no. 11, p. 111005, 2010.

[7] A. Khosronejad, S. Kang, I. Borazjani, and F. Sotiropoulos, “Curvilinear immersed boundary method for simulating coupled flow and bed morphodynamic interactions due to sediment transport phenomena,” *Advances in water resources*, vol. 34, no. 7, pp. 829–843, 2011.

[8] A. Khosronejad, S. Kang, and F. Sotiropoulos, “Experimental and computational investigation of local scour around bridge piers,” *Advances in Water Resources*, vol. 37, pp. 73–85, 2012.

[9] A. Khosronejad, C. Hill, S. Kang, and F. Sotiropoulos, “Computational and experimental investigation of scour past laboratory models of stream restoration rock structures,” *Advances in Water Resources*, vol. 54, pp. 191–207, 2013.

[10] X. Yang, S. Kang, and F. Sotiropoulos, "Computational study and modeling of turbine spacing effects in infinite aligned wind farms," *Physics of Fluids (1994-present)*, vol. 24, no. 11, p. 115107, 2012.

[11] X. Yang, J. Annoni, P. Seiler, and F. Sotiropoulos, "Modeling the effect of control on the wake of a utility-scale turbine via large-eddy simulation," in *Journal of Physics: Conference Series*, vol. 524, p. 012180, IOP Publishing, 2014.

[12] S. J. Osher and R. P. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003 ed., Oct. 2002.

[13] S. Osher and J. A. Sethian, "Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations," *J. Comput. Phys.*, vol. 79, pp. 12–49, Nov. 1988.

[14] M. Sussman and E. Fatemi, "An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow," *SIAM J. Sci. Comput.*, vol. 20, pp. 1165–1191, Feb. 1999.

[15] S. Kang and F. Sotiropoulos, "Numerical modeling of 3D turbulent free surface flow in natural waterways," *Advances in Water Resources*, vol. 40, pp. 23–36, May 2012.

[16] I. Borazjani, L. Ge, and F. Sotiropoulos, "Curvilinear immersed boundary method for simulating fluid structure interaction with complex 3D rigid bodies," *Journal of Computational Physics*, vol. 227, pp. 7587–7620, Aug. 2008.

[17] I. Borazjani and F. Sotiropoulos, "Vortex induced vibrations of two cylinders in tandem arrangement in the proximity-wake interference region," *Journal of fluid mechanics*, vol. 621, pp. 321–364, 2009. PMID: 19693281.

[18] W. Cabot and P. Moin, "Approximate wall boundary conditions in the large-eddy simulation of high reynolds number flow," *Flow, Turbulence and Combustion*, vol. 63, no. 1-4, pp. 269–291, 2000.

[19] M. Wang and P. Moin, "Dynamic wall modeling for large-eddy simulation of complex turbulent flows," *Physics of Fluids*, vol. 14, no. 7, p. 2043, 2002.

[20] J.-I. Choi, R. C. Oberoi, J. R. Edwards, and J. A. Rosati, "An immersed boundary method for complex incompressible flows," *Journal of Computational Physics*, vol. 224, pp. 757–784, June 2007.

[21] B. M. Irons and R. C. Tuck, "A version of the aitken accelerator for computer iteration," *International Journal for Numerical Methods in Engineering*, vol. 1, no. 3, pp. 275–277, 1969.

[22] X. Yang, F. Sotiropoulos, R. J. Conzemius, J. N. Wachtler, and M. B. Strong, "Large-eddy simulation of turbulent flow past wind turbines/farms: the Virtual Wind Simulator (VWiS): LES of turbulent flow past wind turbines/farms: VWiS," *Wind Energy*, pp. n/a–n/a, Aug. 2014.

[23] S. Kang, A. Lightbody, C. Hill, and F. Sotiropoulos, "High-resolution numerical simulation of turbulence in natural waterways," *Advances in Water Resources*, vol. 34, pp. 98–113, Jan. 2011.

[24] J. Smagorinsky, "General circulation experiments with the primitive equations: I. the basic experiment*," *Monthly weather review*, vol. 91, no. 3, pp. 99–164, 1963.

[25] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, "A dynamic subgrid-scale eddy viscosity model," *Physics of Fluids A: Fluid Dynamics (1989-1993)*, vol. 3, no. 7, pp. 1760–1765, 1991.

[26] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, and H. Zhang, "Petsc web page," 2014. http://www.mcs.anl.gov/petsc.

[27] H. T. Ahn and Y. Kallinderis, "Strongly coupled flow/structure interactions with a geometrically conservative ALE scheme on general hybrid meshes," *Journal of Computational Physics*, vol. 219, pp. 671–696, Dec. 2006.

[28] J. Yang and F. Stern, "Sharp interface immersed-boundary/level-set method for wave-body interactions," *Journal of Computational Physics*, vol. 228, pp. 6590–6616, Sept. 2009.

[29] S. Ito, *Study of the transient heave oscillation of a floating cylinder*. PhD thesis, MIT, 1971.

[30] X. Guo and L. Shen, "On the generation and maintenance of waves and turbulence in simulations of free-surface turbulence," *Journal of Computational Physics*, vol. 228, pp. 7313–7332, Oct. 2009.

[31] L. P. Chamorro and F. Porté-Agel, "A wind-tunnel investigation of wind-turbine wakes: boundary-layer turbulence effects," *Boundary-layer meteorology*, vol. 132, no. 1, pp. 129–149, 2009.

[32] S. B. Pope, *Turbulent flows*. Cambridge university press, 2000.