Technische Universität München

Fakultät für Maschinenwesen

Lehrstuhl für Windenergie

High-fidelity Simulation for Wind Turbine Wake Interaction Research

Jiangang Wang

Ph.D. Thesis

Submittal Date: 25.06.2018

Technische Universität München

Fakultät für Maschinenwesen

Lehrstuhl für Windenergie

High-fidelity Simulation for Wind Turbine Wake Interaction Research

Jiangang Wang

Vollständiger Abdruck der von der Fakultät für Maschinenwesen der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr.-Ing. Manfred Hajek
Prüfer der Dissertation:	Prof. Dr. Carlo L. Bottasso
	Prof. Dr. Martin Kühn, Carl von Ossietzky Universität Oldenburg

Die Dissertation wurde am 25.06.2018 bei der Technischen Universität München eingereicht und durch die Fakultät für Maschinenwesen am 01.10.2018 angenommen.

Technische Universität München
Faculty of Mechanical Engineering
Fakultät für Maschinenwesen
Wind Energy Institute
Lehrstuhl für Windenergie
Boltzmannstraße 15
D-85748 Garching bei München
Germany
Tel.: +49 (0) 89 / 289 – 16681
Fax.: +49 (0) 89 / 289 – 16611
Email: info@wind.tum.de
Web: www.wind.mw.tum.de

SUMMARY

This Ph.D. thesis focuses on the numerical method, applications and validation of a recently developed high-fidelity simulation framework, which is developed aiming to build a digital copy of bottom fixed and floating scaled wind turbines in a boundary layer wind tunnel.

The description of wind farm and wake control strategies is introduced first. The methodologies used for investigating wake interaction in a wind farm are reviewed next. Then the review focuses on the high-fidelity computational fluid dynamics (CFD) framework and role of CFD for the wake interaction research. Following this, the author summarizes the experimental research activities performed in the wind energy institute (WEI) in Technische Universität München (TUM) and highlights that this Ph.D. study is mainly motivated by the needs to build a digital copy of the scaled wind turbine facility.

Next, a detailed description of the developed simulation framework is introduced, which contains the characteristics of the SOWFA (original Simulator fOr Wind Farm Applications) framework and the essential numerical methods implemented in the framework. For instance, the immersed boundary modeled nacelle and tower, the blade multi-airfoil table polar, the integral velocity sampling method, the scale-adaptive simulation (SAS) model and the floating offshore wind turbine solver. Moreover, suitable numerical schemes and linear algebraic solvers are investigated to improve the computational efficiency of the framework.

To verify the simulation framework, the author first compared the large-eddy simulation (LES) results with experimental measurements performed in a boundary layer wind tunnel with bottom fixed scaled wind turbine models, including the rotor and wake quantities. The study contains normal operating conditions, as well as wake manipulation by derating, yaw misalignment and cyclic pitching of the blades. Results indicate a good overall matching of simulations with experiments. Low turbulence test cases appear to be more challenging than moderate turbulence ones, due to the need of having a denser grid to limit the numerical diffusion.

Secondly, the SAS model is introduced. The motivation for the use of SAS is its significantly reduced computational cost with respect to LES, made possible by the use of less dense grids. Results indicate that the two turbulence models yield in general results that are very similar, in terms of rotor-integral quantities and wake behavior. Similar to LES, the matching is less satisfactory in very low turbulence inflows, due to a lack of resolution of the blade tip vortices caused by the coarser grids. Given that the computational cost is about one order of magnitude smaller, SAS is found to be an interesting alternative to LES for repetitive runs where one can sacrifice a bit of accuracy for a substantially reduced computational burden.

The developed framework is used for the research of wake redirection methods. Two control strategies are investigated: yaw misalignment and cyclic pitch control. First, analytical formulas are derived for both methods, with the goal of providing a simple physical interpretation of the behavior of the two methods. Next, more realistic results are obtained by numerical simulations and by experiments. Comparing the analytical, numerical and experimental models allows for a comparison of the results and a better understanding of the two wake redirection methods.

Apart from the bottom fixed wind turbine solver, the floating wind turbine solver is also compared with the experiments. Data from a scaled floating offshore wind turbine (FOWT) experiment was used to verify the simulation solver. The computational setup of a simulation strictly adheres to the experiment. The same process of turbulence generation was realized in a digital model, using an ad-hoc precursor simulation. Results show that FOWT pitching introduces a vertical wake meandering, which affects the downstream machine. Both simulation and experimental results clearly indicate this effect and show the compensation performed by a closed-loop controller to keep a steady power output.

To this end, the comparison concludes that the developed high-fidelity simulation framework is a viable way of simulating scaled wind turbine experiments and can be used from wake steering research. An outlook, in a continuation of this Ph.D. study, is included in the very end of this thesis.

Contents

1 Introduction 7

1.1 Background . . . 7

1.1.1 Wind farm and wake control . . . 7

1.1.2 Research activities . . . 8

1.1.3 High-fidelity simulation framework . . . 8

1.2 Motivations . . . 11

1.3 Goals . . . 12

1.4 Approaches . . . 12

1.5 Structure of the thesis . . . 16

2 Simulation Framework and Numerical Methods 17

2.1 Original SOWFA framework . . . 17

2.2 Framework development . . . 18

2.2.1 Nacelle and tower . . . 18

2.2.2 Multi-airfoil table . . . 21

2.2.3 Numerical scheme . . . 21

2.2.4 Linear algebraic solver . . . 28

2.2.5 Pressure-velocity coupling . . . 30

2.2.6 Runtime performance optimization . . . 31

2.2.7 Integral velocity sampling . . . 35

2.2.8 Sub-grid scale model . . . 36

2.2.9 Floating solver . . . 40

3 Computational and Experimental setups for Framework analysis 45

3.1 Experimental facility . . . 45

3.1.1 Boundary layer wind tunnel . . . 45

3.1.2 Wind turbine model . . . 45

3.1.3 Floating platform . . . 47

3.1.4 Inflow measurement . . . 47

3.1.5 Wake measurement . . . 49

3.2 Precursor simulation . . . 50

3.3 Successor simulation . . . 54

3.3.1 Bottom fixed wind turbine . . . 54

3.3.2 Floating offshore wind turbine . . . 58

4 Large-eddy Simulation Framework 61

4.1 Baseline simulation and parameter tuning . . . 61

4.2 Low turbulence inflow simulation . . . 65

4.2.1 Power derating . . . 65

4.2.2 Wake steering by yaw misalignment . . . 65
4.2.3 Wake-enhanced recovery by cyclic pitch control . . . 66
4.3 Moderate turbulence inflow simulation . . . 67
4.4 Wake redirection methods . . . 70
4.4.1 Analytical Model . . . 70
4.4.2 Model verification by LES framework . . . 74
4.4.3 Wake characteristics . . . 76
4.4.4 Wake steering . . . 77
4.4.5 Global power capture . . . 81

5 Scale-adaptive Simulation Framework 85
5.1 Flow quantities . . . 85
5.2 Parameter tuning . . . 85
5.3 Single-turbine baseline case . . . 86
5.4 Single-turbine yaw-misaligned case . . . 89
5.5 Two-aligned-turbine baseline case . . . 92
5.6 Turbulent cases . . . 93
5.6.1 Single-turbine moderate-turbulence case . . . 93
5.6.2 Single-turbine high-turbulence case . . . 95
5.7 Three aligned turbines . . . 96

6 Floating Offshore Wind Turbine Framework 99
6.1 FOWT experiment . . . 99
6.1.1 Floating platform . . . 99
6.1.2 Prescribed pitching motion . . . 100
6.1.3 Inflow conditions . . . 100
6.1.4 Data analysis . . . 101
6.2 Comparison for FOWT framework . . . 107
6.2.1 Flow properties . . . 107
6.2.2 Rotor thrust . . . 107

7 Conclusions and outlook 111
7.1 Conclusions . . . 111
7.2 Outlook . . . 114

8 Dissemination 117
8.1 Journal papers . . . 117
8.2 Peer-reviewed conference papers . . . 117

Bibliography 119

Acknowledgments 131

List of Figures

1.1 Photographs of Horns Rev 2 taken by Bel Air Aviation Denmark in Jan. 2016 at 12:45 UTC 7

2.1 Left: XZ plane of CAD nacelle and tower, right: XZ plane of immersed boundary modelled nacelle and tower. Blue color represents the solid domain. Red color is the fluid domain. White color stands for the fluid-solid interface. 20

2.2 Notation for a one dimensional grid system (captured from [80]) 24

2.3 Comparison of numerical (center differencing scheme and upwind scheme) and analytical solutions for a 1D connvection-diffsuion equation (captured from [80]) 24

2.4 Red curve: linear speedup. Blue curve: speedup of the developed high-fidelity simulation solver. 36

2.5 Flow chart for floating-offshore wind turbine simulation framework 42

3.1 Picture for the sectional view of the boundary layer wind tunnel at Politecnico di Milano (POLIMI) [57]. 46

3.2 Top left: XZ plane of floating platform equiped with two wind turbine models, top right: YZ plane, moddile: XY plane, bottom: 3D platform model. 48

3.3 Inflow velocity profile (facing upstream) as measured by two scanning LiDARs at a 3D distance upstream of the wind turbine model. 49

3.4 Left: hot wire probe and its automated traversing system; right: PIV automated traversing system 49

3.5 Layout of the precursor (left) and successor (right) computational domains. . . . 50

3.6 Normalized time-averaged streamwise velocity $\langle u_x \rangle$ (top row) and turbulence intensity $\sigma/\langle u_x \rangle$ (bottom row), 1.5D downstream of the rotor. Left column: hub-height horizontal profile; right column: hub-centered vertical profile. Numerical results: red + symbols; experimental measurements: black ∘ symbols. 52

3.7 Turbulent kinetic energy spectrum $E(f)$ at hub height 1.5D upstream of the rotor, for the experiment (top-left, black line) and simulation (top-right, red line). Autocorrelation $r(\tau)$ at hub height 1.5D upstream of the rotor (bottom-left) and 0.25D to its left, looking downstream (bottom-right). 53

3.8 Experimental layout and computational domain. 56

3.9 CFD computational domain. Left: lateral view; right: streamwise view. 59

4.1 Streamwise velocity contours for the constant Smagorinsky model and PIV experimental measurements, on a plane 0.56D downstream of the rotor. Black arrows indicate the cross-wind velocity component at a number of sampling points. . . 62

4.2 Profiles of normalized time-averaged streamwise velocity $\langle u_x \rangle / U_0$ at hub height XY plane (top row), at XZ plane through the tower (middle row), and turbulence intensity $\sigma / \langle u_x \rangle$ (bottom row) at hub height XY plane, at the downstream positions 0.56D, 3D, 4D, 7D, 8D, 10D and 11D (red + symbols: constant Smagorinsky model with nacelle and tower included (marked as CS w/- NT); black −−: constant Smagorinsky model without nacelle and tower included (marked as CS w/o NT); blue × symbols: Lagrangian dynamic Smagorinsky model with nacelle and tower included (marked as LDS w/- NT); black ∘ symbols: experimental results). 64
4.3 Normalized time-averaged streamwise velocity $\langle u_x \rangle / U_0$ profiles at 100%, 97.5%, 95% and 92.5% power settings, measured at hub height and 4D downstream of the rotor (red + symbols: LES; black ∘ symbols: experimental results). 66
4.4 Normalized time-averaged streamwise velocity profiles at hub height for different yaw misalignments, 4D downstream of the rotor (red + symbols: LES; black ∘ symbols: experimental results). 66
4.5 Streamwise velocity contour plots for the PIV measurements (top row) and LES model (bottom row), measured 0.56D (left) and 6D (right) downstream of the rotor. Black arrows indicate the cross-wind velocity component at a number of sampling points. 68
4.6 Normalized time-averaged streamwise velocity $\langle u_x \rangle / U_0$ (top row) and turbulence intensity $\sigma / \langle u_x \rangle$ (bottom row) profiles at hub height (red + symbols: LES; black ∘ symbols: experimental results). 69
4.7 Instantaneous streamwise speed component u_x / U_0 (top row), vorticity $\langle \nabla \times \mathbf{u} \rangle$ (central row), and turbulence intensity $\sigma / \langle u_x \rangle$ (bottom row). At left, moderate turbulence case; at right: low turbulence case. 70
4.8 Rotor disk velocities & Coordinate system & Wake deflection and downstream position 71
4.9 Left: yaw and C_yPC configurations & Coordinate systems, right: wind turbine configuration & rotor loading & coordinate systems & Sign conventions 72
4.10 Airfoil velocity triangles & Local blade forces 72
4.11 7 deg CyPC & 20 deg YM streamwise velocity at hub height horizontal plane. From case 1 to 6 are: YM without NT with uniform inflow; YM with NT with uniform inflow; YM with NT with turbulent inflow; CyPC without NT with uniform inflow, CyPC with NT with uniform inflow; CyPC with NT with turbulent inflow. Column left and right are respectively the time averaged and instantaneous U_X. 78
4.12 7 deg CyPC & 20 deg YM U_X at 1D, 3D and 5D (from column 1 to 3, left to right) downstream position. From case 1 to 6 are: YM without NT with uniform inflow; YM with NT with uniform inflow; YM with NT with turbulent inflow; CyPC without NT with uniform inflow, CyPC with NT with uniform inflow; CyPC with NT with turbulent inflow. 79
4.13 Wake displacement for YM vs. yaw misalignment φ 80
4.14 Downstream longitudinal flow velocity U_X for the CyPC 7 deg and YM 20 deg cases. 82
4.15 total power change vs. downstream position X/D for the CyPC 7 deg and YM 20 deg cases. 82

5.1 Vorticity iso-surface plots ($|\nabla \times \mathbf{u}| = 40$ 1/s) for LES, SAS and SST (from left to right). 86
5.2 Normalized time-averaged stream-wise velocity (top) and turbulence intensity (bottom) profiles at hub height and at several downstream positions. Experiment: black ∘ symbols; LES: red + symbols; SAS: blue × symbols; SST: black dashed line. 87

5.3 Normalized time-averaged stream-wise velocity (top) and turbulence intensity (bottom) profiles at hub height and 4D downstream, for various yaw misalignment angles. Experiment: black ∘ symbols; LES: red + symbols; SAS: blue × symbols. 90
5.4 From top to bottom, vorticity magnitude $|\nabla \times \mathbf{u}|$ and turbulence intensity $\sigma / \langle u_x \rangle$ on a hub-height horizontal plane, and turbulence intensity on a vertical plane (looking downstream) 4D behind of the rotor. Black arrows indicate the cross-flow velocity vector at a number of sampling points. 91
5.5 Normalized time-averaged stream-wise velocity (top) and turbulence intensity (bottom) profiles at hub height and at several downstream positions. Experiment: black ∘ symbols; LES: red + symbols; SAS: blue × symbols; SST: black dashed line. 93
5.6 Normalized time-averaged stream-wise velocity (top) and turbulence intensity (bottom) profiles at hub height and at several positions. Experiment: black ∘ symbols; LES: red + symbols; SAS: blue × symbols. 94
5.7 Normalized time-averaged streamwise velocity and turbulence intensity profiles at hub height at downstream positions: -1.5D, 1.4D, 1.7D, 2D, 8D, 3D, 4D, 6D and 9D. blue x symbols: SAS; red crosses: LES; black circles: experimental results. . . 95
5.8 Normalized time-averaged stream-wise velocity (top) and turbulence intensity (bottom) profiles at hub height and at several positions. Experiment: black ∘ symbols; LES: red + symbols; SAS: blue × symbols. 97
5.9 From top to bottom, normalized instantaneous streamwise velocity u_x / U_0, 60 s time-averaged streamwise velocity $\langle u_x \rangle / U_0$, vorticity magnitude $\|\nabla \times \mathbf{u}\|$ and turbulence intensity $\sigma / \langle u_x \rangle$, for both LES and SAS along a streamwise vertical plane across the hub. 98

6.1 1 DOF motion (oscillation along y_f) of the floating platform. 100
6.2 Platform pitch, induced horizontal wind speed and upflow angle χ plotted for a single period. 101
6.3 Blade collective pitch (top left, black circles indicate the time instants when the machine enters into region II), rotor speed (top right), tower base fore-aft bending moment (bottom left) and rotor thrust measured with the pitching platform (bottom right) 102
6.4 Blade collective pitch (left), rotor speed (right) measured with the fixed platform 102
6.5 $C_P(\lambda, \beta_B)$ curves of the G1 model together with the region III regulation trajectory and the three expected power coefficients (red crosses) for λ_{min}, λ_{avrg} and λ_{max} due to the platform-induced longitudinal speed 103
6.6 Frequency spectra of the collective pitch for the upstream and downstream wind turbine 104
6.7 $C_T(\lambda, \beta_B)$ curves of the G1 model together with the region III regulation trajectory and the three expected thrust coefficients (red crosses) for λ_{min}, λ_{avrg} and λ_{max} due to the platform-induced longitudinal speed 104
6.8 Frequency spectra of the rotor thrust for the upstream and downstream wind turbine 105
6.9 Variation (plotted for one period) of the speed that impinges the downstream rotor (orange line). The figure also reports the variation of the speed in the wake shed by the upstream wind turbine (blue line) and the platform-induced longitudinal speed variation (red line) 106

6.10 Collective blade pitch β_B of the downstream wind turbine plotted together with the platform pitch angle β_P 106
6.11 Normalized streamwise velocity (left) and vorticity (right) snapshots on the $x_f z_f$ plane through the turbine hubs. top: fixed platform case; center and bottom: pitching case at the maximum and minimum pitch angle. 108
6.12 Sketch of the model geometry for the calculation of thrust from tower base bending moment. 109
6.13 Comparison of numerical and experimental results for the downstream machine: blade pitch angle $\beta_{B,1}$ (top) and rotor thrust F_T (bottom). 110

List of Tables

3.1 Main characteristics of G1-type wind turbine model. 46

3.2 Specifications for 5MW NREL wind turbine model 47

3.3 Linear algebraic solvers used for the precursor and the wind turbine/wake simulations (CG = Conjugate gradient; GAMG = geometric-algebraic multi-grid; DIC = diagonal incomplete Cholesky; GS = Gauss-Seidel; DILU = diagonal incomplete LU factorization; NOC = non-orthogonal corrector). 55

4.1 Analytical and CFD rotor in-plane force F_H and F_G (in kN), CFD rotor nodding moment M_y and yawing moment M_z (in kNm). 76

List of Algorithms

2.1 Preconditioned conjugate gradient method 28
2.2 No fill-in diagonal matrix 30

ALM	Actuator line method
a	Axial induction factor
α	Angle of attack
α_l	Wake skew angle
α_{SAS}	Closure coefficient in ω equation
$\alpha_{CG}^{(n)}$	Coefficient for calculating the steepest decent at n_{th} timestep
a_c	cos coefficient for the axial induction factor
a'_c	cos coefficient for the tangential induction factor
a_s	sin coefficient for the axial induction factor
a'_s	sin coefficient for the tangential induction factor
a_0	Constant for the axial induction factor
a'_0	Constant for the tangential induction factor
a'	tangential induction factor
a_C, a_F	Linearization coefficients for the linearzied governing equation
$a_{i,f}$	Acceleration component of a mass point at fluid reference frame
a_{ij}	Components of matrix A
A	Rotor swept area
$\mathbf{A}$	Sparse matrix for the coefficients of a linearized governing equation
bi-CG	Bi-conjugate gradient
$\mathbf{b}$	Constant vector in a linear equation system
β_B	Blade pitch angle
β_m	Tunable constant for Gamma differencing scheme
β_{SAS}	Closure coefficient in ω equation
β_P	Platform pitch angle
$\beta^{(n)}$	Conjugate residual of conjugate gradient algorithm
b_c	Linearization coefficients for the linearzied governing equation
B	Blade number
CD	Central differencing
CFD	Computational fluid dynamics
CG	Conjugate gradient
CPU	Central processing unit
CS	Constant Smagroinsky
CTP	Compact task placement
CyPC	Cyclic pitch control
C	Chord length
c	Phase angle for platform pitching motion
χ	Inflow angle
$C_{l\alpha}$	Slope for the linear regime of the lift coefficient
C_D	Drag coefficient
$\hat{C}_D$	Coefficient for $F_{\mathrm{H,YM}}$
C_L	Lift coefficient
$\hat{C}_L$	Coefficient for $F_{\mathrm{H,CyPC}}$
C_k^0	Nominal coefficient
C_μ	Closure coefficient in k equation
C_o	Model parameter for the von Kármán length scale
C_s	Smagorinsky constant
C_0, C_1, C_2, C_3 and C_4	Coefficients for computing weighted least squares on the extended stencil of an immersed boundary surface

DIC-GS	Gauss-Seidel smoothing with diagonal incomplete Cholesky factorization
DILU	Diagonal incomplete-LU factorization
d	Smear function
$\mathbf{d}$	A direction points from one cell center to a neighboring cell center
$\mathbf{d}^{(n)}$	A direction points from one cell center to a neighboring cell center at $n_t h$ timestep
δ	Dirac delta function
d_{ij}	components of matrix $\mathbf{D}^*$
D	Rotor diameter
$\mathtt{D}$	Drag
D	Rotor diameter
Δ_g	Grid size
Δ_x	Distance for x direction
$\Delta x_f, \Delta y_f, \Delta z_f$	Cell size measured along the fluid coordinates
ϵ	Gaussian width
η	Non-dimensional blade spanwise location
e_{ij}	error for Germano identity
$E(f)$	Turbulent kinetic energy spectrum
FAST	NREL aero-servo-elastic simulation tool
FOWT	Floating offshore wind turbine
FPTP	Fully populated task placement
f	Portion of a task (job), which can be parallelized)
$\mathbf{f}$	Projected body force on each grid cell
$\mathbf{f}_f$	Aerodynamic force in the fluid model reference frame
$\mathbf{f}_s$	Aerodynamic force in the structural model reference frame
$\mathbf{f}_{i,max}$	The largest projected blade force point i
$\mathbf{F}$	Body force for each actuator line element
F_r	Radial force at a generic blade section
F_t	Tangential force at a generic blade section
F_C	Centrifugal force
F_G	Vertical force on rotor disk
$F_{\mathrm{G,CyPC}}$	Vertical force for CyPC
$F_{\mathrm{G,YM}}$	Vertical force for YM
F_H	Lateral force on rotor disk
$F_{\mathrm{H,CyPC}}$	Lateral force for CyPC
$F_{\mathrm{H,YM}}$	Lateral force for YM
F_{SAS}	Scaling parameter for SAS term
F_T	Rotor thrust
$F_{T,r}$	Unmodified rated rotor thrust
F_1	Blending function for the SST model
GAMG	Gemetric-algebraic multi-grid
GCC	GNU compiler collection
G1	scaled wind turbine model with diameter equal to 1.1 m
G2	scaled wind turbine model with diameter equal to 2 m
g	Gravity
$g(\mathbf{r})$	Projected force function
γ	Phase angle
Γ	Unknown vector (flow quantities)

$\Gamma_i^{(k+1)}$	i_{th} flow quantity element at (k+1) coarsening mesh level
Γ_b	Immersed boundary condition
H_{hs}	Distance between hub and strain gauge
k	Turbulence kinetic energy
IB	Immersed boundary
$\mathbf{I}$	Identity matrix
J	Cost function
k	Turbulence kinetic energy
LDS	Lagrangian averaging dynamic Smagroinsky
LES	Large-eddy simulation
LiDAR	Light Detection and Ranging
LHS	Right hand side
L	Lift
L	Characteristics length (Sampled length)
L_{cs}	Length between center of mass and strain gauge o_s)
ζ_{ij}	Germano identity
λ	Tip speed ratio
$\mathbf{L}$	Lower part of a sparse matrix A)
$\mathbf{L}^*$	Lower no-fill-in sparse matrix)
Λ	Operator representing the discretized Navier-Stokes equations
L_{vk}	von Kármán length scale
MPI	Message passing interface
m	Mass
$\dot{m}_i$	Mass flow at $i_t h$ cell face
μ	Dynamic viscosity
M_{ij}	A matrix used in the LDS model
M_y	Tower base fore-aft bending moment
n	number of parallelized processors
N	Available experimental observations
NOC	non-orthogonal corrector
NREL	National renewable energy laboratory
NT	Nacelle and tower
$\mathbf{n}$	Vector of assumed singular-value decomposition
ν	Kinematic viscosity
ν_t	Eddy viscosity
OpenFOAM	C++ toolbox called Open source Field Operation And Manipulation
ω	Specific rate of dissipation
o_m	Center of mass
o_s	Origin of structural reference frame
o_f	Origin of fluid reference frame
Ω	Rotor speed
$\dot{\Omega}$	Angular acceleration
$\Omega_{\mathbf{a}}$	Angular velocity vector
Ω_{CV}	Volume of a grid cell
Ω_r	Unmodified rated rotor speed
Pe	Péclet numbers
PISO	Pressure-Implicit with Splitting of Operators
PIV	Particle Image Velocimetry

p	Pressure
$\widehat{p}$	Modified pressure variable
ϕ_i	Field variable at $i_t h$ cell face
ϕ_I	Field variable at $i_t h$ cell center
$\widehat{\phi}_i$	Normalized variable at $i_t h$ cell face
$\widehat{\phi}_I$	Normalized variable at $i_t h$ cell center
ϕ_I^n	Flux at $i_t h$ cell center at $n_t h$ timestep
ψ	Blade azimuth position
p_f	Partialization factor
p'	Pressure variable containing the vertical pressure gradient
P	Rotor power
PLC	Programmable logic controller
P_u	Upstream rotor power
$\mathbf{P}$	Preconditioning matrix
Pe_L	Péclet number measured based on a sampled length
Φ_i	Blade azimuth angle
$\Psi(\mathbf{x}, t)$	Integral for fluid-particle trajectory (pathline)
P_k	turbulence kinetic energy production term
Q	Source term of the governing equation
Q_{SAS}	Source term for SAS model
RAM	Random accessing memory
RANS	Reynolds-averaged Navier Stokes
RMS	Root mean square
$r(\tau)$	Auto-correlation
r	Discrepancy between power and thrust coefficients
$\mathbf{r}$	Distance vector between a CFD cell center and an actuator line element
$\mathbf{r}^{(n)}$	Conjugate residual vector of a linear equation system
ρ	Density
ρ_0	Reference density
R	Rotor radius
$\mathbf{R}(\mathbf{t})$	Time-dependent direction cosine rotation matrix
Re	Reynolds number
SAS	Scale-adaptive simulation
SOWFA	A NREL tool for simulating coupled dynamic response of wind turbines
SPMD	Single program multiple data parallelism
SFS	Sub-filter scale
SGS	Sub-grid scale
SST	Shear-stress transport
σ	Rotor solidity
σ_k	Closure coefficient in k equation
σ_ω, $\sigma_{\omega 2}$	Closure coefficient in ω equation
$\sigma/\langle u_x \rangle$	Turbulence intensity
S	Surface of a control volume
$\mathbf{S}$	Rate-of-strain tensor
$\lvert S \rvert$	Magnitude of a rate-of-strain tensor
S_{Amdahl}	Parallel Speedup (Amdahl's law)
t	Time
τ	Temporal variable

τ_{kk}	Trace of stress tensor
τ^{D}_{ij}	The deviatoric part of a stress tensor at scale Δ
τ^{r}_{ij}	Reynolds stress tensor
$\tau_{\mathbf{SGS}}$	A sub-grid scale stress tensor
$\tau^{\mathbf{D}}_{\mathbf{SGS}}$	The deviatoric part of a sub-grid scale stress tensor
θ_c	1P blade pitch amplitude
θ_i	Blade pitch angle
θ_{to}	Angle between the tower axis and the line connecting o_m and o_s
θ_0	Collective blade pitch constant
θ_P	Platform pitch angle
θ_{P0}	Initial platform pitch angle
t	Time
T	Temperature
T_{int}	Integral time scale
T^{D}_{ij}	The deviatoric part of a stress tensor at scale 2Δ
T_r	Integral time scale
URANS	Unsteady Reynolds-averaged Navier Stokes
u	Unspecified velocity
$\mathbf{u}$	Velocity vector
Υ_i	Platform rotation axis
$\mathbf{u}_s$	Sampled velocity vector
$\tilde{\mathbf{u}}$	Filtered velocity vector
$\tilde{\mathbf{u}}_s$	Filtered sampled velocity vector
u_i	Velocity vector
$\langle u_x \rangle$	Normalized time-averaged streamwise velocity
$\mathbf{U}$	Upper part of a sparse matrix A)
U_∞	Inflow velocity
U_x	Streamwise velocity for the rotor disk plane coordinate system
U_P	Blade local axial velocity
U_T	Blade local tangential velocity
U_w	Downstream velocity
U_X	Streamwise velocity for the global coordinate system
U_0	Free stream velocity
$\mathbf{U}^*$	Upper no-fill-in sparse matrix)
$\underline{U}^*$	Solution of the velocity and pressure field without considering the effects of the immersed boundary
φ	Yaw misalignment
φ_y	Platform angle between the tower axis and the line connecting o_s and o_f
V	Volume
V_c	Control volume
V_y	Lateral velocity for the rotor disk plane coordinate system
V_Y	Lateral velocity for the global coordinate system
w/o NT	Without nacelle and tower
w/- NT	With nacelle and tower
WT-Perf	Simulation tool similar to FAST
$W(t-t')$	Weighting function defining the extent background along a pathline
xyz	Rotor stationary coordinate system
ξ	$1-\cos\theta_P(t)$

(x_f, y_f, z_f)	Fluid domain coordinate frame
(x_s, y_s, z_s)	Structural domain coordinate frame
$x'y'z'$	Blade coordinate system
$\mathbf{x}_f$	Position vector of a point in fluid model reference frame
$\mathbf{x}_k$	Specified volume mesh location (vector)
$\mathbf{x}_{ib}$	An immersed boundary point (vector)
x_i	Location of $i_t h$ cell face
$\mathbf{x}_s$	Position vector of a point in structural model reference frame
x_I	Location of $I_t h$ cell center
XYZ	Global fixed coordinate system
X_c	Center of a sampled circle in SYZ frame
YM	Yaw misalignment
y^+	Dimensionless wall distance
$\mathbf{z}(t')$	Trajectory for earlier times $(t' > t)$ at lagrangian frame of reference
ζ	Tuning parameter for wake expansion
$\bar{(\cdot)}$	Grid LES filter at scale Δ
$\widehat{(\cdot)}$	Test LES filter at scale 2Δ (subscript $\widehat{(\cdot)}$
$\langle\cdot\rangle$	Spatial averaging operator

Chapter 1

Introduction

1.1 Background

1.1.1 Wind farm and wake control

Wind farms are collections of wind turbines, often operating in close proximity to one another. During operation, several major phenomena take place within a wind farm: there is an interaction between the atmospheric boundary layer and each individual wind turbine (as well as a more global interaction between the boundary layer and the whole wind farm), and an interaction among upstream and downstream wind turbines through their wakes. The wake produced by the upstream wind turbines, coupled with the atmospheric flow, has a significant influence on the performance of downstream operating machines. In fact, the wake-affected inflow results in lower power output and increased loading for the downstream wind turbines when compared to an isolated condition.

Fig. 1.1 shows the photographs of Horns Rev2 taken by Bel Air Aviation Demark from the helicopter [1]. The advected cold-water fog from the shallow layer is exposed to the humid warm air mass and interacts with the wind farm. Some wake from the upstream wind turbines in the same figure also covers or partially covers the downstream wind turbines and results in significant power and loads variations.

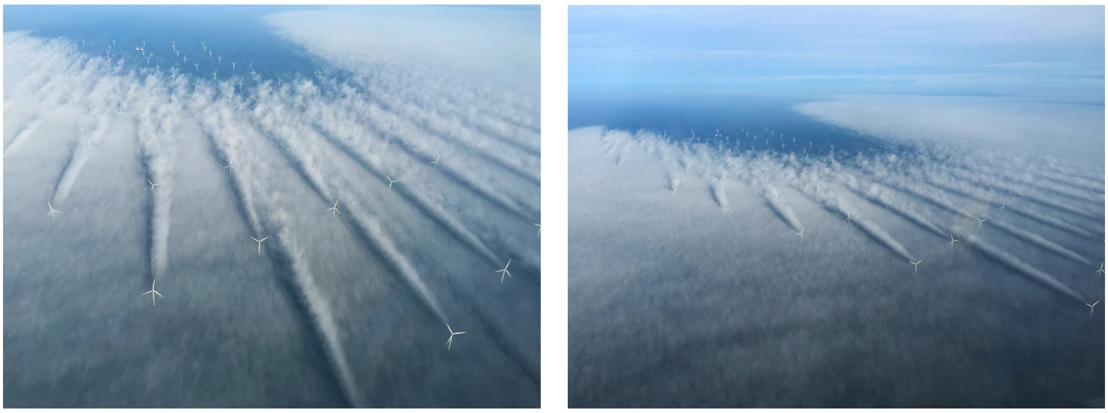

Figure 1.1: Photographs of Horns Rev 2 taken by Bel Air Aviation Denmark in Jan. 2016 at 12:45 UTC

Wake control strategies are currently being investigated to optimize the operation of wind farms, including power de-rating, wake deflection, and enhanced wake recovery [2–4]. They are generally proposed to either introduce the wake steering or produce higher wake recovery so as to enable an increase in the power capture for the downstream machines. A well-designed wind plant control system might also increase the global power capture of a wind farm and allows for the blade load mitigation. To properly implement such control strategies into the modern wind farm, one needs to first obtain enough knowledge regarding the wind farm flow characteristics. The simulation and experimental research on the wind farm allow the wake interaction to be observed, the physics behind the phenomena to be understood and the reduced order model to be formulated. The developed model may eventually be used as the kernel for model-based wind plant control.

1.1.2 Research activities

The current research in this field is very active, covering high-fidelity numerical simulations, scaled experiments in the wind tunnel, direct measurements in the field, reduced-order models, and control methods. In the past decades, the meteorological measurement data collected from the Horns Rev and Middelgrunden offshore wind farm has been investigated systematically [5–7]. Moreover, scaled wind farm experiments were conducted in the wind tunnel to study the wake deficit and its impact on the downstream wind turbines [8–10]. Those test campaigns are often regarded as the benchmark and have been actively used to validate several engineering and CFD wake models, in terms of velocity profile, higher-order flow quantities and power capture [11–15]. Those wake models, including many other that have been developed, can be grouped on the basis of their fidelity to reality. The steady-state kinematic wake models of [16] and [17] were the first proposed analytical formulations, while Larsen et al. [18] later derived a more sophisticated dynamic wake meandering model. In the following, an improvement to simulation fidelity has been obtained thanks to CFD. Carcangiu et al. [19] used the Reynolds averaged Navier Stokes (RANS) turbulence model to simulate the near wake performance, while Stovall et al. [20] used the same turbulence model to simulate the wind turbine cluster condition and verify the RANS model by the LES model. Results indicated that RANS is not sufficiently accurate, as it typically overestimates numerical diffusion.

1.1.3 High-fidelity simulation framework

With the significant increase in computational capability in recent years, the large-eddy simulation is commonly adopted in the wind farm research community [14, 21, 22], as it shows a much higher accuracy, as well as the capability of offering improved insight into the flow characteristics within a wind farm. In general, these simulation frameworks use the actuator disk, actuator disk rotation or actuator line model to model the rotor blades [23–27] and Boussinesq approximation [28] to account for the buoyancy effect.

The application of wind farm simulations is quite extensive. Since the LES model ensures the fidelity level of simulations and CFD solvers often provide complete knowledge of flowfield, researchers can use LES framework to obtain flow properties where the experimental facilities are difficult to measure. This data helps explain complex phenomena in the wind farm. Moreover, a well-validated simulation framework can also be used to perform pure numerical research. Such research activities substantially reduce expenses and the simulations produced can be highly trusted. For instance, several researchers [3, 15, 17] used the framework to investigate different wake control strategies, which are all based on pure numerical investigations. In addition, since some of the simulation frameworks are numerically efficient and allow for significant computational hours, a diurnal cycle of a wind farm in an atmospheric boundary layer was also investigated [29, 30].

The short literature review represents the current state of the wind farm research and, in particular, the high-fidelity simulation framework. There are still some open areas to be investigated. Such research can potentially improve simulation accuracy, reduce computational cost and extend simulation capability.

1. Nacelle and tower modeling

   Unlike the near wake simulation, which typically uses the body-conformed mesh for the blade, nacelle, and tower to investigate the structure loading and near wake characteristics [31], the wind farm simulation uses the actuator line model to model the wind turbine geometry [23,32–34]. Generally, the nacelle and tower are not considered in the wind farm simulation, as the signature of these two components is smeared and can be neglected at the far wake. However, even though the presence of the nacelle and tower has negligible impact on the velocity field, it drastically changes second order quantities, such as vorticity and turbulence intensity. Moreover, if the wind farm is designed to be more compact, the reduced wind turbine spacing results in the insufficient flow mixing. In this sense, the nacelle and tower should still be modeled in the wind farm simulation.

   Wu et al. [27] use the actuator line method (only the drag coefficient is included) to model the nacelle and tower. The tower shedding and corresponding rotor-tower interaction are completely missing in the simulation. The actuator line modeled nacelle and tower may also require additional parameter tuning, which is often undesirable for a reliable simulation framework. On the other hand, using the body-conformed mesh to model the nacelle and tower will certainly introduce huge computational burdens. Therefore, a new numerical approach should be used in the framework to increase the fidelity of the modeled geometry and maintain the computational cost.

2. Sub-grid scale model

   The LES approach is widely accepted as a turbulence model for a wind farm simulation. However, its computational burden is significanth and hinders its applicability for ongoing wind plant control research. In fact, dense meshes are typically necessary for resolving wakes and their interactions with the downstream machines, while several runs of lengthy physical time durations are often necessary to test the relevant scenarios. On the other hand, the traditional RANS and RANS SST (shear stress transport) approaches are unable to model the wind turbine wake and wake overlapping effects properly. Stovall et al. [20] used the RANS model to run wind turbine cluster simulations and verified the results by means of the large-eddy simulation. The verification indicates that the RANS model overestimates the numerical diffusion and concludes that the accuracy of the RANS model is insufficient for the wind farm simulation. Therefore a new turbulence model needs to be proposed in order to reduce the computational costs and maintain the level of accuracy.

3. Offshore floating application

   Offshore wind energy research is attracting an increasing level of interest in deep-water applications [35]. Floating offshore wind turbine, as a technology primly designed for deep-water application, currently represents one of the most significant engineering challenges in the wind energy field. Matha et al. [36–39] have presented an exemplary methodology for the integrated design concept of a floating wind turbine platform and systematically compared the industrial design methodologies for a bottom-fixed wind turbine and a floating wind turbine. Rockel et al. [40] investigated the platform pitching of a floating wind turbine system and concluded that the platform pitching induces an additional vertical displacement of velocity components. INNWIND, a project focuses on the research of innovative offshore wind turbines built at 50 m water depths and beyond to reduce the levelized cost of energy [41–44].

Among the many different research aspects of floating offshore technology, understanding the wake interaction of multiple floating wind turbines under complex motions is a particularly challenging task, which requires both suitable high-fidelity numerical models and relevant experimental observations. A wind tunnel experimental campaign [45] was recently conducted using a scaled model of a floating platform concept, developed by the Korea Research Institute of Ships & Ocean Engineering (KRISO) [46]. Next, the author extended the simulation capability to the FOWT simulation so as to build a digital copy of this floating platform facility.

4. Turbulence inflow generation

   There are several ways of generating the scaled atmospheric boundary layer (ABL) in the wind tunnel [47–51]. These approaches, in general, are classified into two categories. The first approach is to use the coarse wire mesh and honeycomb flow straightener to enable the turbulence control and sheared boundary layer [51], while the second one is to use turbulence-generating spires [52]. The latter is adopted in all the experimental work used in this thesis.

   In order to reproduce the same scaled ABL inflow map in the experiment, the simulation framework needs to either model the velocity shear and turbulence properties or mesh the turbulence-generating spires. Most of the research focuses on the first approach [14, 53]. Porté-Agel et al. [54] imposed a wall model to generate an atmospheric boundary layer. Troldborg et al. [55, 56] use volume force formulation to generate velocity shear and turbulence. Neither algorithm requires significant manual tunings and both are computationally inexpensive. On the other hand, although the body-conformed mesh approach requires sophisticated mesh generation and is time-consuming, it is a tuning free approach and ensures the fidelity of the simulation.

5. Advanced actuator line method

   Most of the LES frameworks currently available, which employ the actuator line method to represent the lifting surfaces such as wind turbine blades, require proper tuning of the projection width,ϵ, of a Gaussian kernel [24, 26]. However, the research conducted so far does not generally discuss in detail how ϵ should be tuned, depending on the simulated wake control strategies. In fact, ϵ should be kept constant for the same wind turbine model at different yaw misalignment angles, power deratings, collective/cyclic pitching motions, and different turbulence environments. This is the only way in which ϵ can be regarded as a value independent of operating conditions. In addition, provided that the validation work on the framework shows controversial results, a more advanced actuator line modeling technique has to be investigated, as the case-dependent ϵ is not a reliable concept for a wind plant control-oriented simulation framework.

Apart from the aforementioned potential research areas, a detailed and systematic validation of a simulation framework with respect to experimental data is generally missing. Very few frameworks have undergone careful validation processes. Such validation usually requires the comparison of rotor-integral quantities (power capture and rotor thrust), inflow map and wake profiles at a minimum of 3 downstream positions covering from the near wake to the far wake region. These wake profiles are sampled by means of scattered velocity points in the computational domain. Each velocity point can also be used to reconstruct the turbulence intensity and turbulence kinetic energy. Moreover, the velocity points with high sampling frequencies and longtime histories (or a slice of an instantaneous velocity field) can be employed for the velocity spectra analysis.

Moreover, a comprehensive validation of the adopted simulation framework, applied to the investigation of different wake control strategies, is also missing. A framework without the proper validation may easily mislead the wake control research, because the quantities, such as wake recovery and wake displacement, play an important role in the power estimation of downstream wind turbines. If these quantities are estimated based on incorrect data, this may result in very different conclusions. Therefore, it is crucial to have the framework carefully validated or at least systematically compared with the measurement data so that it can be used as a reliable simulation tool for wake control research.

1.2 Motivations

WEI has developed a scaled experimental facility for testing wind turbines and wind farms in a boundary layer wind tunnel. This unique facility enables experiments to be conducted in aeroservoelasticity, as well as the study of wakes, machine-to-machine interactions, and wind plant control for power maximization and load mitigation. The facility is highly instrumented, allowing for the collection of a wide range of high-quality data, both regarding the flow conditions and the response of the machines. Such data can be used to validate the performance of control strategies and computational tools. The experimental setup is highly flexible, supporting different machine configurations and operational scenarios, as well as the testing of different control algorithms used on board the individual wind turbines or the whole wind farm. In early 2016, WEI conducted the first-ever experimental demonstration of closed-loop wake-redirection control. In this wind tunnel test, a cluster of three wind turbines was operated in a coordinated manner, leading to an improvement in their overall power output of 15%. This successful experiment led to WEI being awarded the 2016 Bayerischer Energiepreis, in recognition of its groundbreaking work in wind plant control. Another highlight of the year was the wind tunnel testing of a wind turbine cluster made of four closely-spaced wind turbines mounted on a floating platform, for deep offshore installations. The tests validated advanced control laws for the operation of the floating cluster, including extreme sea and wind conditions [57–62].

A parallel ongoing research effort by the author aims to develop a digital copy of this experimental facility, including the wind turbines and the wind tunnel itself. This digital copy is established based on a high-fidelity CFD framework (section 1.1). Both the onshore and offshore configuration shall be included in the digital copy. This digital copy has to be validated first, ensuring that the experimental facility is properly reproduced. Once the digital copy is validated, one can use it to sample flow properties at points that are difficult for the measuring equipment to reach. The sampled data also helps to explain complex phenomena in the interacting wake. Moreover, a well-validated simulation framework allows for pure numerical researches, which save huge investments on the test campaign and, at the same time, guarantees simulation accuracy.

1.3 Goals

This Ph.D. thesis focuses on the numerical methods, validation, and applications of a recently-developed high-fidelity simulation framework, which is developed with the aim of building a digital copy of an onshore and offshore scaled wind turbine cluster facility.

In detail, state-of-the-art numerical methods are used to model wind tunnel-generated flows (different turbulence levels), wind turbine models (including blades, nacelle, tower), induced shed wakes interacting with the other machines, and the test section of a wind tunnel. The algorithm used also aim to reduce computational costs and the dependency of results to the tuning parameters. Moreover, the developed framework tries to solve, or at least partially solve, the remaining open research areas addressed in section 1.1.3. Indeed, an immersed boundary modeled nacelle and tower and a torque and pitch controller are included in a wind turbine model; a low-cost scale-adaptive approach is used as a sub-grid scale (SGS) model to significantly reduce computational costs; a floating algorithm is implemented to enable FOWT simulations; a body-conformed mesh approach is used for the precursor domain to generate a high-quality turbulent flow; an integral approach is used for velocity sampling of actuator line elements; a multi-airfoil table was identified for wind turbine blades. The data obtained from previous wind tunnel experiments is employed to compare with the framework developed. The comparison contains single and multiple wind turbine configurations operated at different wake control strategies. Finally, the developed framework is used for pure numerical research, which is about the wake redirection control by interacting wind turbines.

1.4 Approaches

In this section, the logic of the author's Ph.D. research will be introduced. The author will focus on how to identify, decompose, and eventually solve the research issues. A clear and logical approach to tackling each problem faced during the Ph.D. research is the key to the success of the whole Ph.D. study.

1. As addressed in section 1.3, the goal of this thesis is to focus on the development and application of a high-fidelity simulation framework for wind turbine wake control research. This framework is compared with the test campaigns conducted in a boundary layer wind tunnel. Thus, the framework is used primarily simulating a scaled wind turbine facility. The first step is to, if possible, find the currently available framework, which can be used directly for the scaled wind turbine simulation, or modified with minimum coding efforts.

2. The framework adopted is SOWFA from National renewable energy laboratory (NREL) [63]. SOWFA is a set of CFD solvers, boundary conditions, and turbine models based on the OpenFOAM (Open source Field Operation And Manipulation) open source CFD platform. It includes a version of the actuator line model coupled with FAST [64]. This tool allows users to investigate the wind turbine and wind farm performance under the full range of atmospheric conditions and in terrain [21, 63]. As a wind farm simulation tool developed based on a standard CFD solver in OpenFOAM repository, SOWFA also obtains all the simulation capabilities from OpenFOAM.

   OpenFOAM is a C++ object-oriented library for the development of customized solvers, and pre/post-processing utilities for the solution of continuum mechanics, including the computational fluid dynamics (CFD) [65]. OpenFOAM uses finite volume based discretization with friendly syntaxes for partial differential equations and enables the unstructured polyhedral grid capabilities. These are important characteristics for a wind farm simulation:

(a) Instead of meshing structures and fully resolving all details in a computational domain, the wind farm simulation usually requires wind turbine components to be modeled and wall models to be included in order for computational costs to be substantially reduced. OpenFOAM provides user-friendly high-level programming, which helps the developer to couple an existing solver with external simulation tools so as to model the essentials properly. The easy manipulation of mathematical forms also helps to implement additional source terms into the governing equations, which allows body forces to be imposed on the domain artificially.

(b) The OpenFOAM meshing utility allows for an easy and fast build of a local refined computational domain. This capability is crucial for minimizing the computational cost, as a big domain is generally required for both full-scale and scaled wind farm simulation to eliminate the blockage effect and adopt suitable momentum and heat exchanges between a wind turbine cluster and a boundary layer. The constant mesh resolution of the entire domain creates significant computational costs, without improving the simulation accuracy any further. In fact, the mesh resolution only needs to be sufficiently high within the passage covering the wind turbine and wake region. Therefore, the local refinement capability is actively used for wind farm simulations.

(c) The program scalability is another important characteristic of the wind farm simulation. Simulating large computational domain usually requires a correspondingly large number of parallelized processors to reduce the time consumption. The solver should perform in such a way that minimizes the portion of serial calculation [66], and ensures that processor inter-communication loss cannot increase exponentially with an increasing number of parallelized processors. For instance, a scaled wind turbine cluster simulation usually uses up to 616 processors and is composed of 80 million cells for a formal production run (including 6 wind turbine models). The parallel efficiency of OpenFOAM allows the solver to work efficiently within this range.

3. SOWFA is generally considered a framework used for full-scale wind farm application. Martínez-Tossas et al. [26] used the SOWFA model to directly simulate the scaled wind turbine model operated in a wind tunnel, and compared the simulation results with the wind tunnel tests. However, this comparison did not show decent agreement in terms of the wake profiles. Moreover, the comparison did not cover the range from the low turbulence to high turbulence inflow condition, as well as the wake overlapping effect, which is clearly insufficient to draw general conclusions. Therefore, it is essential to perform a more comprehensive comparison for the original SOWFA framework in order to:

   (a) Quantify the level of simulation accuracy using the original SOWFA.

   (b) Identify the source of errors for each simulated quantities, e.g. velocity, pressure, temperature, etc.

   (c) Assess the time consumption for each production run.

4. Based on the assessment of the original SOWFA, there are two issues that need to be fixed in the first instance, namely the nacelle and tower modeling and the multi-airfoil table identification.

    (a) The nacelle and tower should be modeled to account for the signature of these two components in the flowfield. The nacelle and tower introduce the periodic oscillation of power capture and rotor thrust. More importantly, the presence of nacelle and tower produces unsymmetrical wake characteristics. The effects not only change the near wake behavior but also influence the far wake region, where downstream wind turbines are located. In addition, the overlapping of turbine wakes generates a more complex wake profile and has to be modeled correctly in a wake interaction simulation.

    (b) A multi-airfoil table needs to be used for a scaled wind turbine simulation. The actuator line method is used for the modeling of wind turbine blades. This method requires the lift and drag coefficients (C_L and C_D) to be predefined for each blade section. The scaled wind turbine facility usually operates at low Reynolds number (Re). At this range, C_L and C_D are not constant and are dependent on the change of Re. Therefore a single airfoil table is not valid at low Re. Instead, a Reynolds number-dependent multi-airfoil table should be included in the simulation model. An interpolation algorithm should also be implemented for the simulation framework to linearly interpolate C_L and C_D according to the variation of the Reynolds number.

    Several state-of-art numerical techniques were investigated before the immersed boundary method was adopted for the nacelle and tower model. The multi-airfoil table is used as an input for the blade actuator line model and the linear interpolation is implemented into the framework. The inclusion of these two modeling techniques significantly improves the simulation accuracy.

5. The next research step is to further improve the simulation framework and perform the corresponding comparison:

    (a) The actuator line method uses three steps to achieve blade modeling, namely velocity sampling, force calculation (calculated by FAST) and the force projection method. Now, a more advanced actuator line based blade modeling approach is investigated to reduce the dependency of ϵ on the rotor power capture.

        i. In the original framework, velocities are sampled from single points in the domain. The new approach, on the other hand, uses the integral velocity method, which is the sum of a velocity field weighted by a force projection function [67]. So, instead of sampling the velocity from a point, the new method takes into account a set of the velocity points in a space. In fact, the pointwise approach uses a variation of Gaussian width to reshape the Gaussian curve, in turn changing the peak value and eventually influencing the calculated aerodynamic power, while the integral approach uses a weighted average that mitigates the reshaping effect. In this way, the dependency of ϵ to the rotor power is reduced.

        ii. In the original framework, a single ϵ value is used along the blade in spanwise, chordwise and thickness direction. A new approach is to project the body force according to the blade geometory [68]. This new method is proposed to provide a more precise interpretation of the blade geometry to project forces and improve the modeling of blade tip vortices. Unfortunately, based on the tests, this geometric projection approach cannot offer either the improved wake profile or the blade tip vortices.

(b) A body-conformed meshing approach is used to mesh the turbulence-generating spire. The meshed domain is used to generate a proper atmospheric boundary layer for wake control simulations. As described in section 1.1.3, the conformed mesh approach ensures simulation accuracy. Moreover, in general, the ABL inflow map only needs to be simulated and stored once. The same inflow map is used in different test campaigns. Therefore, even though each turbulence-generating simulation is computationally expensive, the total number of production runs are negligible.

(c) A RANS-based turbulence model, scale-adaptive simulation (SAS), was proposed by Mentor et al. [69]. The formulation is derived from a $k-\omega$ SST model. It introduces the von Kármán length scale to measure the local flow inhomogeneities so that the local eddy viscosity can be properly resolved. Therefore, compared with the SST model, the SAS model is capable of solving the numerical dissipation more accurately and results in proper wake estimation. The extra costs of calculation for the von Kármán length are also negligible. More importantly, the SAS model is allowed to use the same grid resolution as the SST model, which means that its computational cost is also maintained at the level of the RANS model, and is significantly lower than the LES model.

(d) A numerical approach to enabling the floating wind turbine simulation is investigated. In this work, the immersed boundary method is used to model geometries (nacelle, tower, floating platform) and the assumption of the rigid body motion is valid. Several geometry motion techniques have been proposed to achieve this motion capability, which includes the offset mesh, and deformable mesh [70]. Those approaches usually require high computational costs and a complex code development. However, the motion algorithm for the immersed boundary method can be significantly simplified, as it can use the non-topological change. This approach dynamically allocates the geometry profile index to the mesh domain according to the motion of geometry, rather than creating new cells to adopt the moved geometry. So, no mesh regeneration is required and the computational cost is, therefore, drastically reduced.

6. Finally, all of the numerical techniques used in the simulation tool developed have been investigated, the framework can be considered as a reliable tool and can be used for numerical researches. The numerical research in this thesis focuses on the wake redirection study, as this is primarily of interest to the wind farm research community. There are typically two ways of redirecting the turbine wake, yaw misalignment (YM), and cyclic pitch control (CyPC). The author aims to formulate the wake lateral displacement and quantify the wake recovery of the two wake redirection methods.

   (a) A three-pronged approach is used to investigate the lateral displacement of YM and CyPC, namely simple analytical equations, simulation, and experiment. The analytical equations are derived from the stream-tube theory and the momentum conservation. They are used to provide the physical interpretation of the two wake redirection methods. The simulation and experiment, on the other hand, provide the cross-verification of the analytical model.

(b) Apart from the verification of the analytical model, the simulation framework is used to quantify the wake recovery and the global power capture of interacting wind turbines. Because it offers complete flow properties in a three-dimensional domain, one can easily sample all the essential information and use it for the investigation. In particular, since the simulation results are capable of providing 2D wake profiles, they allow the non-uniformity of a wake to be observed. This kind of 2D wake, on the other hand, cannot be obtained by the simple analytical model and is difficult to measure in the experiment.

1.5 Structure of the thesis

In chapter 1, the description of the wind farm associated with wake control strategies is introduced first (section 1.1.1). Next, the methodologies used for research activities to investigate the wake interaction in wind farms are reviewed (section 1.1.2). Then, the review focuses on the high-fidelity CFD framework and the role of CFD used for the wake interaction research (section 1.1.3). Following this, the thesis describes several open research areas to be investigated, which become part of the motivations for the development of a framework. Then, in section 1.2, the author summarizes the research activities performed in WEI and lists all the motivations for this Ph.D. thesis. The final targets of this Ph.D. thesis are highlighted in section 1.3, followed by the approaches used in section 1.4 to achieve the aim of this thesis.

Next, a detailed description of the simulation framework is introduced in chapter 2, which includes the characteristics of the original SOWFA framework (section 2.1) and the essential numerical methods used to improve the framework (section 2.2). These numerical methods contain the immersed boundary-modeled nacelle and tower (section 2.2.1), the blade multi-airfoil table polar (section 2.2.2), the integral velocity sampling method (section 2.2.7), sub-grid scale model (section 2.2.8), and the floating wind turbine solver (section 2.2.9). In addition, from section 2.2.3 to 2.2.6 describe the numerical schemes and linear solver used in the simulation framework so as to improve the computational efficiency. The experimental and computational setup for the framework analysis are reported in chapter 3 including turbulent inflow generation and wind turbine simulation, while the experimental setup is presented in section 3.1.

Results are presented and discussed in chapter 4, 5 and 6. The first two chapters describe the the bottom fixed wind turbine configuration and include a single wind turbine aligned and misaligned with respect to a low turbulence inflow, and in moderate and high turbulence conditions. A case consisting of three aligned wind turbines is used to assess the method in wake-overlapping conditions. In chapter 4, the simulation model is also used for the investigation of two wake redirection methods. Chapter 6 represents the the floating wind turbine simulation. The vertical wake interaction effect is clearly observed and analyzed in section 6.2.2. Finally, the conclusions and outlook are drawn in chapter 7.

CHAPTER 2

Simulation Framework and Numerical Methods

A detailed description of the developed simulation framework is introduced in this chapter, which contains the characteristics of the original SOWFA and the essential numerical methods implemented in the framework. For instance, the immersed boundary modeled nacelle and tower, the blade multi-airfoil table polar, the integral velocity sampling method, the scale-adaptive simulation (SAS) model and the floating wind turbine solver. In addition, suitable numerical schemes and linear algebraic solvers are investigated to improve the computational efficiency.

2.1 Original SOWFA framework

The high-fidelity simulation framework for the wake control simulation is developed within SOWFA NREL [63]. SOWFA (Simulator fOr Wind Farm Applications) is a set of CFD solvers, boundary conditions, and turbine models based on the OpenFOAM open source CFD platform [65]. It includes an actuator line + FAST wind farm simulation solver [64], which is derived from an incompressible solver, BuoyantBoussinesqPimpleFoam [1], in the OpenFOAM repository. This tool allows users to investigate the wind turbine and the wind farm performance under the full range of atmospheric conditions and in terrain [21, 63].

The incompressible Navier-Stokes equations used in SOWFA are spatially filtered to derive the formulation of the large-eddy simulation. By neglecting the Coriolis force due to the relatively small simulation scale, the vertical heat transport and the source terms used for the boundary layer wall model, the filtered governing equations used in SOWFA is formulated as

$$\nabla \cdot \tilde{\mathbf{u}} = 0, \tag{2.1a}$$

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} + (\tilde{\mathbf{u}} \cdot \nabla)\tilde{\mathbf{u}} = -\nabla \widehat{p} - \frac{1}{\rho}\nabla p_0(x,y) + \nu\nabla^2\tilde{\mathbf{u}} - \nabla \cdot \tau^{\mathbf{D}}_{\mathbf{SGS}} + \frac{\mathbf{f}}{\rho}, \tag{2.1b}$$

where $\tilde{\mathbf{u}}$ is the flitered velocity vector, or so called the resolved-scale velocity vector. $\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}'$, where $\mathbf{u}'$ is the sub-filter scale (SFS) velocity vector [71]. Since the hexahedral cell is mainly used in the computational setup of this thesis, the grid cell is often regarded as the size of a filter. So the sub-filter scale is equalvant to the sub-grid scale (SGS) in this thesis. $(\tilde{\mathbf{u}} \cdot \nabla)\tilde{\mathbf{u}}$ is the convection term in the filtered governing equation, which is often considered as an untrival term to be discrtized. The scheme used for discretizing this term will be discussed later in this chapter. $\nabla \widehat{p} = \nabla \left(\frac{\tilde{p}'}{\rho_0} + \frac{1}{3}\tau_{kk,SGS} \right)$ is the gradient of a modified pressure variable lumped with the

[1] To be more specific, only the Pressure-Implicit with Splitting of Operators (PISO) algorithm is used for SOWFA + FAST coupled simulations. The SIMPLE algorithm is never activated in PIMPLE loops, since the timestep is always prescribed and fixed within the simulation.

trace of a stress tensor, τ_{kk}. $\tilde{p}'$ is the pressure variable containing the vertical pressure gradient and ρ_0 is the reference desnity. $\frac{1}{\rho_0}\nabla p_0(x,y)$ is the background driving pressure gradient. ν is the kinematic viscosity, $\nu\nabla^2\tilde{\mathbf{u}} = \nu\nabla\cdot(\nabla\tilde{\mathbf{u}}+\nabla\tilde{\mathbf{u}}^T)$ is the viscous induced stress. $\tau^{\mathbf{D}}_{\mathbf{SGS}} = \tau^{D}_{ij,SGS} = \tau_{ij,SGS} - \frac{1}{3}\tau_{kk,SGS}\delta_{ij}$ is the deviatoric part of a stress tensor. $\tau_{kk,SGS}$ is the trace of a stress tensor lumped with the modified pressure variable and δ_{ij} is the Kronecker delta. In fact, compared to the contribution of $\nu\nabla^2\tilde{\mathbf{u}}$ in a wind farm simulation, the divergence of the stress tensor, $\nabla\cdot\tau^{\mathbf{D}}_{\mathbf{SGS}}$, is significantly higher due to the high Reynolds number [21]. $\overrightarrow{\mathbf{f}}$ is the body force vectors exerted by the actuator line model.

The author tried to use the SOWFA framework to directly simulate the scaled wind turbine simulation. However, SOWFA is generally considered as a framework used for the full-scale wind farm application. The framework has to be extensively developed to adopt the application of the scaled turbine simulation.

2.2 Framework development

Before the development of the framework, an assessment of SOWFA was performed using the experiments conducted in the scaled wind turbine facility to examine the simulation accuracy and identify where the sources of error may come from. The comparison regarding the original SOWFA framework with the experiments has shown that two issues need to be primly fixed so as to satisfy basic requirements of the simulation accuracy. Nacelles and towers should be included wind turbine models in the framework to account for unsymmetrical wake characteristics. A Reynolds dependent airfoil table should be used for the simulation model to adopt characteristics of an airfoil at low Reynolds numbers.

2.2.1 Nacelle and tower

In this section, two approaches are introduced to model the nacelle and tower (NT) of a wind turbine model. The first approach uses a line of drag elements to represent the two components, while the second approach employs the immersed boundary method to model the nacelle and tower.

Actuator line method

The actuator line (AL) method is attractive because, compared to geometry-resolved simulations, the generated mesh is much simpler and the computational cost is also significantly lower. The AL method models the nacelle and tower into lines of drag elements, instead of resolving the full geometry.

$$\overrightarrow{\mathbf{f}} = \mathbf{F}\cdot g(\mathbf{r}), \tag{2.2}$$

where $\mathbf{F}$ is the body force for each actuator line element. Since no lift force is generated by the nacelle and tower, only the drag is included in the $\mathbf{F}$ calculation. $\mathbf{r}$ is the distance vector between a CFD cell center and actuator line elements, while $g(\mathbf{r})$ is the Gaussian projection function. The isotropic projection method is used for $g(\mathbf{r})$ and reads

$$g(\mathbf{r}) = \frac{1}{\epsilon^3\pi^{3/2}} exp\left[-(\frac{|\mathbf{r}|}{\epsilon})^2\right] \tag{2.3}$$

where ϵ is the Gaussian width, a tuning parameter determining the distribution of a body force projection. Churchfield et al. [72] have investigated the actuator line modeled nacelle and tower for wind farm simulations. Moreover, Steven et al [73] introduced an improved version of actuator line modeled nacelle tower, which is capable of modeling flow oscillation effects. Yang and Sotiropoulos [74] proposed an actuator surface model to represent nacelle and tower. Nonetheless, the parameter tuning is often undesired for a high-fidelity simulation framework. More importantly, the AL approach is not capable of modeling the tower shedding and the corresponding rotor-tower interaction. Therefore, a new approach is proposed, which uses the immersed boundary (IB) method to model the nacelle and tower.

Immersed boundary method

An IB method is employed to model the turbine geometry, as this approach does not require a complex body-conformed mesh, which usually implies a tedious process and significant effort in the present structured-grid context [75]. Furthermore, the immersed boundary method can usually achieve an adequate local resolution with minimum overall mesh density, and eliminate the risk of obtaining deteriorated local cells. Santoni et al. [76] have successfully used the immersed boundary method to model the effect of nacelle and tower on the flow past a testing wind turbine.

The adopted IB approach uses a direct imposition of the boundary conditions (discrete forcing approach). To introduce the characteristics of this method and explain its advantages over the alternative indirect imposition method, the principles of both are briefly reviewed next. The present formulation is derived based on Mittal et al. [75] and Bandringa's work [77]. The incompressible governing equations are reformulated based on equation 2.1 and given as

$$\nabla \cdot \tilde{\mathbf{u}} = 0, \tag{2.4a}$$

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} + (\tilde{\mathbf{u}} \cdot \nabla)\tilde{\mathbf{u}} + \frac{1}{\rho}\nabla p - \nu \nabla^2 \tilde{\mathbf{u}} + \nabla \cdot \tau^{\mathbf{D}}_{\mathbf{SGS}} = 0, \tag{2.4b}$$

where $\tilde{\mathbf{u}}$ is the fluid velocity, p the pressure (Boussinesq buoyant effect is neglected to simplify the problem), ρ the density and Nu the kinematic viscosity. The governing equations are first discretized without considering the immersed boundary and are written as

$$[\Lambda]\{\underline{U}^*\} = 0, \tag{2.5}$$

where $\underline{U}^* = (\tilde{\mathbf{u}}, p)$ specifically indicates the solution of the velocity and pressure field without considering the effects of the immersed boundary, while L is the operator representing the discretized Navier-Stokes equations. Next, the immersed boundary condition (Γ_b) is imposed on the flow-field $\underline{U} = \underline{U_{\Gamma_b}}$, resulting into Eq. (2.5) being reformulated as

$$[\Lambda + (I - \Lambda)\delta(|\mathbf{x}_k - \mathbf{x}_{ib}|)]\{\underline{U}\} = \{\underline{U_{\Gamma_b}}\}d(|\mathbf{x}_k - \mathbf{x}_{ib}|). \tag{2.6}$$

Here $[I]$ is the identity matrix and δ is the Dirac delta function, which allows for the formula to switch between Eq. (2.5) and $\underline{U} = \underline{U_{\Gamma_b}}$. However, the cell center of a specified volume mesh location (vector), noted $\mathbf{x}_k$, in general does not coincide with the immersed boundary point, $\mathbf{x}_{ib}$. The two numerical methods — direct and indirect — differ in the way they address this problem. The indirect method tries to replace δ with a smear function d, which writes

$$[\Lambda]\{\underline{U}\} = \{\underline{U_{\Gamma_b}} - \underline{U}^*\}d(|\mathbf{x}_k - \mathbf{x}_{ib}|) + [\Lambda]\{\underline{U}^*\}d(|\mathbf{x}_k - \mathbf{x}_{ib}|). \tag{2.7}$$

The forcing term resulting from this smooth distribution implies that sharp edges of the geometry cannot be exactly represented. Therefore, the forcing term will smear in space, which is an undesirable effect when one wants to resolve the surface boundary layer of the geometry at high Reynolds numbers.

On the other hand, the direct imposition approach preserves the sharpness of the geometry. No smeared force will be generated in the neighboring cells, but instead an interpolation technique is used to treat the IB cell center using its neighboring cells. The current solver uses a two-dimensional polynomial function with second order accuracy [78]. The Dirichlet immersed boundary condition imposed at location k is formulated as

$$\underline{U_k} = \underline{U_{ib}} + C_0(x_k - x_{ib}) + C_1(y_k - y_{ib}) + C_2(x_k - x_{ib})(y_k - y_{ib}) + C_3(x_k - x_{ib})^2 + C_4(y_k - y_{ib})^2, \quad (2.8)$$

where all the coefficients C_0, C_1, C_2, C_3 and C_4 are computed using weighted least squares on the extended stencil [78]. x_k and x_{ib} represent x coordinate (scalar) of the specified points. With this interpolation technique, the direct imposition method is able to locate the boundary surface, identify each individual solid and fluid cell, and assign proper boundary conditions to the fluid-solid interface. This way, the sharpness of the geometry surface is preserved and no extra stability constraints are introduced. Fig. 2.1 shows the CAD and immersed boundary modeled nacelle and tower of a wind turbine model. The fluid-solid interface is clearly identified in the domain. With the sharp representation of the IB, one can directly impose boundary conditions and a wall model on the surface boundary, which typically results in an improved quality of the solution for high Reynolds number viscous flows.

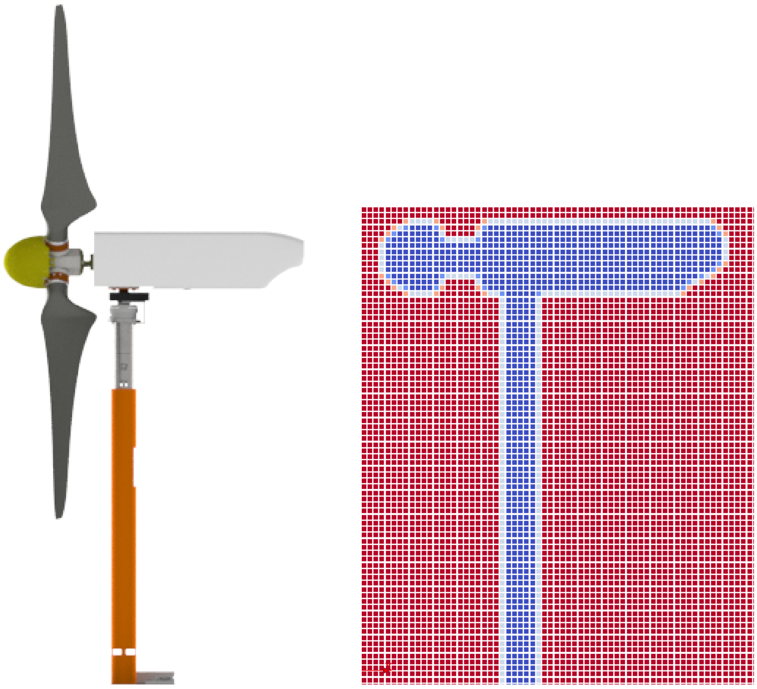

Figure 2.1: Left: XZ plane of CAD nacelle and tower, right: XZ plane of immersed boundary modelled nacelle and tower. Blue color represents the solid domain. Red color is the fluid domain. White color stands for the fluid-solid interface.

Since the immersed boundary class was implemented and deeply embedded in Foam-extend-4.0, the entire SOWFA simulation tool has to be moved from OpenFOAM 2.X to the Foam-extend platform.

2.2.2 Multi-airfoil table

In section 1.4, the author has addressed that the airfoil coefficients (C_L and C_D) are dependent on the change of Reynolds number at relatively low Re. Jacobs and Sherman [79] have conducted wind tunnel tests and investigated the low Reynolds number effects on lift and drag coefficient. Their results for Re over 1.0×10^5 showed the minor change of lift coefficient C_L, whereas C_L changes nonlinearly and cannot be neglected if Re is lower than 5.0×10^4. So, the lift coefficient is dependent on the change of the low Reynolds number. Based on this phenomenon, a single airfoil polar table, with lift and drag coefficient versus angle of attack, is not sufficient to represent C_L and C_D at low Re. Therefore, multiple airfoil tables at several Reynolds numbers associated with linear interpolation are introduced in this thesis to estimate the values more accurately.

The current G1 simulation model uses an airfoil table with C_L and C_D values measured at 4 Reynolds numbers (i.e. Re=50000, 60000, 75000, 90000). The airfoil C_L and C_D values are computed at each instant of time based on the local angle of attack (α) by linearly interpolating the tabulated values. In this way, the actuator line inputs take into account the Reynolds number effects, which allows the wind turbine simulation to operate at a wider Re range. It is an important improvement as the large yaw misalignment and power the derating simulations often change the operating Re and therefore changes the C_L and C_D values.

After solving the Reynolds number issue, The research efforts will try to improve the quality of the airfoil tabulated data by introducing new identification techniques [61] in the future, as these clearly have a crucial importance in a LES-lifting-like formulation.

By including the immersed boundary modeled nacelle & tower and the multi-airfoil table, the framework satisfies the basic requirements of the simulation accuracy. Next step, a high-resolution bounded scheme is used for the convection term to balance the numerical stability and numerical diffusion. Preconditioned linear algebraic solvers are investigated to ramp up the simulation speed. In addition to the solver, the domain decomposition, and code scalability is investigated to further improve the runtime performance. The PISO algorithm coupled with the Rhie-Chow interpolation is also used to segregatedly solve a sets of density-variant governing equations.

2.2.3 Numerical scheme

The discretization process mainly consists of modeling a geometric domain, discretizing a domain into the structure or unstructured mesh and discretizing governing equations [80]. The domain geometry and the mesh generation will be described in chapter 3. The discretization of the governing equations is categorized to the spatial discretization and temporal discretization. The discretization methodologies used in the simulation framework will be introduced in this section.

Spatial discretization

The spectral-pseudo and energy-conserving discretization CFD method are implemented to ensure the zero numerical disspiation [54, 81], while the finite volume method doesn't allow the dissipation free condition. Therefore, it is essential to investigate a proper discretization approach to minimize the numerical dissipation. Indeed, the scheme should also ensure the numerical stability and minimize the numerical dispersion. There are a number of numerical methods used to stabilize the simulation by introducing the negative feedback to the numerical oscillation. However, many of them are associated with the strong numerical dissipation. So, a feasible discretization scheme should balance the numerical stability [2] and the numerical dissipation to ensure the optimal performance of simulations.

This section first introduces the finite volume formulation and explains the root cause of the numerical dispersion and stability issues. Then, it explains why the upwind scheme, as an effective numerical approach to stabilizing the directional flow, causes strong numerical dissipation. To this end, a new convection discretization scheme is proposed to properly balance the numerical stability and numerical dissipation.

1. Finite volume method formulation

   Regarding a steady state condition, terms discretized in the governing equations are the diffusion term, the convection term and the source term. The finite volume method uses the integral form to formulate these terms. By neglecting the temporal discretization and SGS stress, the integral form of equation 2.1 over a finite volume is formulated as

   $$\oint_{\partial V_c} (\tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}}) \cdot d\mathbf{S} = \oint_{\partial V_c} (\nu \nabla^2 \tilde{\mathbf{u}}) \cdot d\mathbf{S} + \oint_{V_c} Q^{\phi} dV, \tag{2.9}$$

   where subscripts $\oint_{\partial V_c}$ is the surface integral over a discretized control volume, $\mathbf{u}$ is interpreted as a unity of flux across cell surfaces. The term on the left hand side (LHS) is the convection term, while the first term on the right hand side (RHS) is the diffusion term. Q is the source term, including the pressure and external body forces. Since the operation of Q is not subjected to the flux transport on cell faces, it does not require Gaussian theorem to convert from the volume integral to the surface integral. Next step, the equation is linear and initially writes

   $$\sum_{f\sim nb(C)} (\tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}} - \nu \nabla^2 \tilde{\mathbf{u}}) \cdot d\mathbf{S} = QV_C, \tag{2.10}$$

   where $\sum_{f\sim nb(C)}$ is the sum of momentums across cell surfaces. The number of integration points on each side (face) of a cell determines the order of integration accuracy [80]. OpenFOAM uses the second order accurate integration approach, since only one integration point is considered on each side of a cell. The final form of the linear momentum equation is formulated as

   $$a_C \tilde{\mathbf{u}} + \sum_{f\sim nb(C)} (a_F) \tilde{\mathbf{u}}_F = b_C, \tag{2.11}$$

   where a_C, a_F and b_C are the linearization coefficients. The linear convection and diffusion term is programmed into the OpenFOAM environment and is described in [80]. This thesis, on the other hand, focuses on the numerical methodologies used to stabilize simulations, as well as constrain the numerical diffusion. In general, the convergence issue is triggered by the poor mesh quality, the unfriendly boundary condition and the flow misalignment due to the grid orientation.

[2] In [80], the numerical stability is strictly distinguished with the numerical dispersion. However, the author addresses that the numerical boundedness and numerical dispersion (oscillation/wiggle/overshoot) are the subsets of the numerical stability. If a simulation is confirmed to be numerically unstable, all those effects will certainly happen and result in catastrophe of the simulation. On the other hand, the existence of those effects might not necessarily trigger an unstable simulation.

There are a few indicators used to measure the mesh quality. Apart from the mesh hanging and overlapping, which certainly costs the simulation crash, the most crucial indicators are the grid aspect ratio and the grid orthogonality. The two indicators are determined by the mesh skewness. The majority portion of wind turbine simulation domain is composed of purely cubic hexahedral mesh, while the non-orthogonal grid only exists at the surface of local mesh refinements [82]. The surface grid is also not severally skewed. The maximum skewness measured in the computational domain is typically very low (approximately equal to 0.33). Therefore, the convergence issue is not induced by the poor quality.

On the other hand, the imposing of an immersed boundary condition does introduce severe convergence issues. Fig. 2.1 shows an in-penetrated surface (nacelle and tower) imposed to the fluid domain. A closed surface occupied by a fluid domain introduces the local flow acceleration (Results in high local velocity) due to the geometry local blockage effect. In addition, the immersed boundary modeled nacelle and tower in this thesis is not associated with the local mesh refinement. Combing the two effects (the increased local velocity and the fixed mesh resolution), the presence of geometry introduces a significant increase in the Peclet number (Pe_L). Given the condition that the second order central differencing scheme is used for the convection term, the simulation will suffer from the simulation instability. In detail, the high Pe_L produces the local numerical wiggles, which in other words, the numerical dispersion [83]. The decelerated flow at the upstream of the geometry triggers the so-called convective stability criteria, which induces the positive feedback of numerical oscillations and therefore results in the simulation catastrophe.

2. Numerical stability

The presence of immersed boundary modeled geometries gives rise to the simulation instability, which is primly important for a reliable simulation framework. Therefore, it is essential to perform a quantitative analysis to understand the root cause of the simulation instability. Based on the understanding of the numerical stability, new numerical schemes can be proposed to improve the simulation stability.

A simple 1D convection-diffusion analysis is performed to first investigate the numerical dispersion. Fig. 2.2 shows schematically the notation of a one-dimensional grid. The flow direction points to the east. The capital letter stands for cell centers, while the lowercase represents cell faces. The quantity ϕ_i in momentum equation 2.10 is the flow velocity.

By neglecting source terms, the analytical and numerical solution of the steady state equation 2.10 are derived and write

$$\frac{\phi_C - \phi_W}{\phi_E - \phi_W} = \frac{e^{Pe_L \frac{x_C - x_W}{L}} - 1}{e^{Pe_L} - 1}, \tag{2.12a}$$

$$\frac{\phi_C - \phi_W}{\phi_E - \phi_W} = \frac{1}{2}\left(1 - \frac{Pe_L}{2}\right), \tag{2.12b}$$

where $Pe_L = \frac{\rho \tilde{\mathbf{u}} L}{\nu}$, $L = x_E - x_W$. Equation 2.12a and Equation 2.12b represent the analytical and numerical solution respectively. The numerical solution uses the center differencing scheme, a second order accurate scheme. If a uniform grid is assumed ($\delta x_w = \Delta x_C$), the value of flux on cell face e is the mean of the fluxes at two neighboring

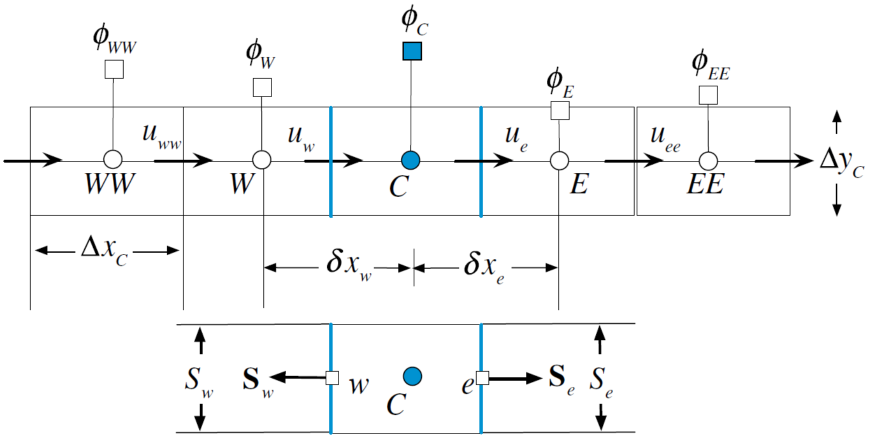

Figure 2.2: Notation for a one dimensional grid system (captured from [80])

cell centers, $\phi_e = \frac{\phi_C + \phi_E}{2}$. The form is equivalent to the Taylor expansion of a point by neglecting higher order terms. It implies that the discretized momentum equation is not capable of providing the same result as the analytical solution. Fig. 2.3 shows curves for the analytical and numerical solutions. The analytical curve approaches asymptotically to the values of 0 and 1, while the central differencing curve is unbounded. Therefore, Pe_L has to be limited to a small range. Otherwise, the non-negligible error will eventually produce the large numerical oscillation and destabilize the simulation. The researches [80, 84] addressed that the error of numerical solution is defined to be negligible if the magnitude of Pe_L is smaller than 2.

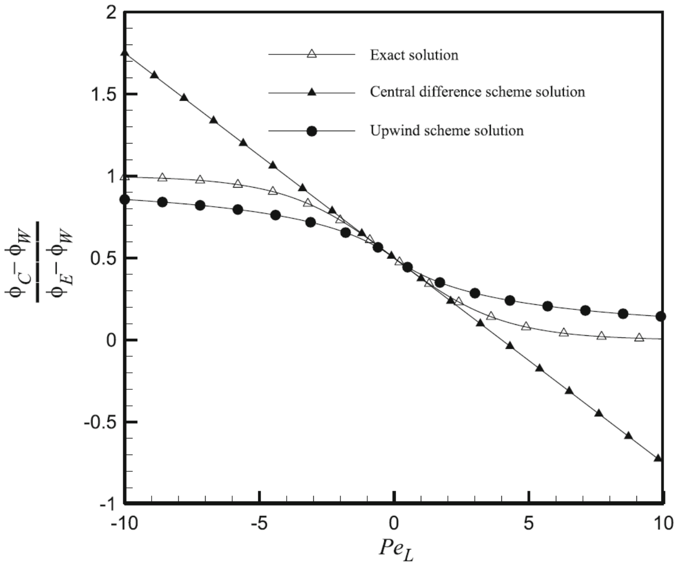

Figure 2.3: Comparison of numerical (center differencing scheme and upwind scheme) and analytical solutions for a 1D connvection-diffsuion equation (captured from [80])

The diffusion term is equally influenced by the upstream and downstream fluxes. The center differencing scheme, a scheme assigns equal weight to the surrounding cells, is suitable for the diffusion term. However, such scheme is not a good approximation for the convection term, as it is not capable of describing the directional preference of the convection. Therefore, if the flow is under a highly directional flow condition (i.e. gusts), the center differencing scheme is not a proper scheme and has to be replaced by a more advanced discretization scheme.

Apart from the numerical dispersion, the numerical stability regarding the convective term has to be investigated, since the center differencing scheme has no inherent convective stability. Leonard [85, 86] employed a one dimensional unsteady convection-diffusion equation with constant velocity to analyze the convective stability. The unsteady term is added to the 1D form of equation 2.9 and writes

$$\frac{\partial \tilde{\mathbf{u}}}{\partial t} = -\tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}} + \nu \nabla^2 \tilde{\mathbf{u}} + Q^{\phi}, \tag{2.13}$$

where the left-hand side (LHS) of the equation is the unsteady term, which, in other words, is the rate change of flux normal to a unit surface, $\tilde{\mathbf{u}}$[3]. Everything else is moved to the right-hand side (RHS) of the equation. The derivative of RHS, $\frac{\partial(RHS)}{\partial\phi_C}$, is the change of net influx across the cell faces and the source terms stored at the cell centers with respect to the change of flux ϕ_C. The Such derivative has to ensure stability criteria and satisfy

$$\frac{\partial(RHS)}{\partial\phi_C} < 0, \tag{2.14}$$

where the inequity presents that the change of net influx has negative feedback to the change of the ϕ_C. The change of the discretized diffusion term in Eq: 2.14 using the center differencing scheme is derived as

$$\frac{\partial(RHS_{CD}^{Diff})}{\partial\phi_C} = -2\nu\frac{\Delta y}{\delta x}, \tag{2.15}$$

where CD means the center differencing scheme. The whole term is strictly negative and satisfies the stability criteria. The change of the distretized convective term in equation 2.14 writes

$$\frac{\partial(RHS_{CD}^{Conv})}{\partial\phi_C} = -2\left(\dot{m}_e + \dot{m}_w\right), \tag{2.16}$$

where $\dot{m}_e$ and $\dot{m}_w$ are not exactly identical to each other for an unsteady flow. If the flow is decelerated, the term becomes positive and is prohibitive to the stability of the simulation.

The 1 dimensional analysis concludes that the center differencing scheme is feasible for the diffusion term, as it ensures the numerical stability. However, a different scheme has to be investigated for the upwind scheme.

3. Upwind scheme

   The Gamma scheme [87, 88] is employed in this thesis as a discretization scheme for the convection term. It essentially contains the upwind scheme to preserve the numerical stability. So, before studying the Gamma scheme, the formulations of upwind scheme is introduced first. The upwind scheme is able to effectively reduce the numerical dispersion and ensure the simulation stability. A simple explanation is that the upwind scheme

[3] $\tilde{\mathbf{u}}$ in this thesis is equivalent to quantity ϕ in this thesis.

provides the directional preference and is therefore compatible with the convection term.

Instead of linearly interpolating the flux values from the cell faces to the centers, the upwind scheme bounds the flux from the cell faces to the upstream cell centers. This way, the directional preference is guaranteed. The upwind scheme writes

$$\phi_e = \begin{cases} \phi_C & if \;\; u_e > 0 \\ \phi_E & if \;\; u_e < 0 \end{cases} \tag{2.17}$$

By substituting the interpolating scheme from the center differencing scheme to the upwind scheme, the analytical solution is formulated as

$$\frac{\phi_C - \phi_W}{\phi_E - \phi_W} = \frac{2 + || - Pe_L, 0||}{4 + |Pe_L|}. \tag{2.18}$$

The upwind curve in Fig. 2.3 shows a similar asymptotic behavior as the analytical solution, indicating that the solution is strictly bounded. Moreover, the numerical stability of upwind scheme is also investigated. The change of the discretized convective term with respect to the change of flux ϕ_C in equation refeq:RHS1 writes

$$\frac{\partial(RHS^{Conv}_{upwind})}{\partial\phi_C} = -||\dot{m}_e, 0|| - ||\dot{m}_w, 0||, \tag{2.19}$$

where $\frac{\partial(RHS^{Conv}_{upwind})}{\partial\phi_C}$ is strictly negative and therefore eliminates the stability issues.

On the other hand, the upwind scheme is less accurate than the center differencing scheme at low Pe_L numbers. Because the upwind scheme is a typical first-order accurate scheme, while the center differencing scheme is second order accurate. The order of accuracy significantly influences the simulation performance. The low order scheme often leads to the strong numerical diffusion (or truncation error), which is certainly undesirable for the wake estimation of wind farm simulations.

4. Gamma differencing scheme

   The concept of the center differencing and upwind scheme indicates a tradeoff between the numerical stability and simulation accuracy (diffusion). Therefore, a more advanced scheme is investigated, aiming to balance the two effects. The Gamma scheme, a bounded high resolution scheme categorized to the family of normalized variable formulation [89–91], is employed as the discretization scheme for the convection term in this thesis. In essence, the proposed scheme is second-order accurate and can be extensively used for the arbitrarily unstructured mesh [92, 93].

   The normalized variable formulation (NVF) is a framework to develop the high resolution schemes. NVF The developed high resolution schemes generally satisfies the convection boundedness criterion (CBC) [89]. NVF locally normalizes the fluxes based on the cell center fluxes at upwind, downwind and far upwind cell. The operator for orthogonal and non-orthogonal (or arbitrarily unstructured mesh) grid write

$$\widehat{\phi} = \frac{\phi - \phi_U}{\phi_D - \phi_U} = 1 - \frac{\phi_D - \phi_C}{2(\nabla\phi)_C \cdot \mathbf{d}}, \tag{2.20}$$

where $\widehat{\phi}$ is the normalized field variable. The expression $(\nabla\phi)_C \cdot \mathbf{d}$ uses the gradient across the face to formulate $\widehat{\phi}$. This way, the flux at cell center U is not explicitly used [87] and, therefore, allows for the arbitrarily unstructured mesh formulation. The face flux $\widehat{\phi}_f$ can be interpolated as a function of $\widehat{\phi}_C$ and is formulated as

$$\widehat{\phi}_f = f\left(\widehat{\phi}_C\right). \tag{2.21}$$

$\widehat{\phi}_C$ is a good indicator of the smoothness. For instance, a local gradient jump can be expressed by $\widehat{\phi}_C$ approaching to 0. The Gamma differencing scheme is given as

$$\widehat{\phi}_f = \begin{cases} \widehat{\phi}_c \left[1 + \frac{1}{2\beta_m}(1 - \widehat{\phi}_c)\right] & 0 \le \widehat{\phi}_c \le \beta_m \\ \frac{1}{2}\widehat{\phi}_c + \frac{1}{2} & \beta_m \le \widehat{\phi}_c \le 1 \\ \widehat{\phi}_c & \widehat{\phi}_c < 0 \,||\, \widehat{\phi}_c > 1 \end{cases} \tag{2.22}$$

where β_m is the tuning parameter. At the range of $\widehat{\phi}_c < 0 \,||\, \widehat{\phi}_c > 1$, the CBC criterion enforces the formulation to use upwind scheme to guarantee the boundedness. The formulation uses central differencing scheme at the range of $\beta_m \le \widehat{\phi}_c \le 1$. Moreover, there is a smooth switching at the point $\widehat{\phi}_c = 1$, which is desirable for a well designed NVF scheme. On the other hand, if the central differencing scheme extends directly to $\widehat{\phi}_c = 0$, the scheme will suffer from an abrupt change from the central differencing scheme to upwind scheme. The Gamma scheme provides a unique switching at the point $\widehat{\phi}_c = 0$ to allow a smooth transition between the upwind scheme and central differencing scheme. The formulation is also independent of the Courant number, so the solution is not determined by the size of the time step or the relaxation factor used for the convergence [87].

The Gamma differencing scheme uses the upwind scheme to guarantee the boundedness at the range of $\widehat{\phi}_c < 0 \,||\, \widehat{\phi}_c > 1$, where the velocity gradient is significantly large. However, the high numerical dissipation results in the low simulation accuracy. So the Peclet number should still be limited to a small value ($\approx$2) so as to prevent $\widehat{\phi}_c$ from entering into large velocity gradient regimes. Within the range of $0 \le \widehat{\phi}_c \le 1$, The Gamma scheme employes the tuning parameter β_m to smooth the transition between the upwind scheme and central differencing scheme. Indeed, larger β_m indicates smoother switching, which, however, reduce the range of the central differencing scheme and, therefore, reduces the simulation accuracy. The range of β_m is suggested to be within 1/10 and 1/2.

Temporal discretization

The backward Euler method, as a conditionally stable second order implicit scheme, is used for the temporal discretization. The discretization of the unsteady term is similar to that of the convective term. The backward scheme can be regarded as an upwind scheme for the spatial discretization. The second order backward Euler scheme is formulated as

$$\frac{\partial \phi_C^n}{\partial t} = \frac{1}{\Delta t}\left(\frac{3}{2}\phi_C^n - 2\phi_C^{n-1} + \frac{1}{2}\phi_C^{n-2}\right) \tag{2.23}$$

where the time derivative of ϕ_C at n timestep is determined by ϕ_C at $n-1$ and $n-2$ timestep. Such formulation ensures the second order time integration accuracy and implicit characteristics.

2.2.4 Linear algebraic solver

In addition to the numerical schemes, the author investigated the performance of state-of-art linear algebraic solvers in the simulation framework [82], as they substantially influence the simulation efficiency. In fact, the most time consuming part of a simulation is the operation of sparse linear algebraic kernels. Therefore, how to efficiently precondition the sparse matrix and iteratively solve the preconditioned matrix is primely important for runtime performance of the solver.

An actuator line based wind turbine simulation, in general, allows for faster convergence criteria. The computational domain, in this thesis, uses high-quality mesh (99% cubic cells) and small simulation time-steps (Courant number = 0.3), which are all suitable for the quick convergence. Moreover, the simulation does not show any significant flow separation or adverse pressure gradient. All these facts clearly indicate that it is not essential to employ time-consuming pre-conditioner algorithms and non-orthogonal mesh correctors to solve the resulting sparse matrices. In fact, the runtime performance is primely important for the wind farm simulation. Therefore, faster linear algebraic solvers are investigated and used in this simulation framework.

The conjugate and bi-conjugate gradient method are eventually adopted to solve the governing equations. The geometric agglomerated algebraic multigrid method is used as a preconditioner for symmetric matrices, while the diagonal incomplete upper and lower (LU) preconditioner is used for asymmetric matrices.

Conjugate gradient solver

The conjugate gradient (CG) method is proposed to iteratively solve a linear algebraic system of equations [94, 95]. The algorithm allows for the stable and fast convergence rate [96], which is formulated as a set of equations and writes [80]

Algorithm 2.1: Preconditioned conjugate gradient method

```
Result: $\Gamma^{(n+1)}$
Initialize $\mathbf{d}^{(0)} = \mathbf{P}^{-1}\mathbf{r}^{(0)}$ ,  $\mathbf{r}^{(0)} = \mathbf{b} - \mathbf{A}\Gamma^{(0)}$;
for $n = 1 : k$ do
    if $r^{(n+1)} - r^{(n)} \geq Tor$ then
        $\alpha_{CG}^{(n)} = \frac{\mathbf{d}^{(n)T}\mathbf{P}^{-1}\mathbf{r}^{(n)}}{\mathbf{d}^{(n)T}\mathbf{A}\mathbf{d}^{(n)}}$;
        $\Gamma^{(n+1)} = \Gamma^{(n)} + \alpha_{CG}^{(n)}\mathbf{d}^{(n)}$;
        $\mathbf{r}^{(n+1)} = \mathbf{r}^{(n)} + \alpha_{CG}^{(n)}\mathbf{A}\mathbf{d}^{(n)}$;
    else
        Return $\Gamma^{(n+1)}$;
    end
    $\beta^{(n+1)} = \frac{\mathbf{r}^{(n+1)T}\mathbf{P}^{-1}\mathbf{r}^{(n+1)}}{\mathbf{r}^{(n)T}\mathbf{P}^{-1}\mathbf{r}^{(n)}}$;
    $\mathbf{d}^{(n+1)} = \mathbf{P}^{-1}\mathbf{r}^{(n+1)} + \beta^{(n+1)}\mathbf{d}^{(n)}$;
end
```

where $\mathbf{d}^{(n)}$ is a searched direction at n_{th} timestep. $\mathbf{d}^{(n)}$ has to be A-orthgonal to $\mathbf{d}^{(m)}$ and results in $\mathbf{d}^{(n)T}\mathbf{A}\mathbf{d}^{(n)} = 0$. $\mathbf{P}$ is a preconditioning matrix. The preconditioning operation allows for a faster convergence rate. $\mathbf{b}$ is a constant vector in a linear equation system, $\mathbf{A}\Gamma = \mathbf{b}$. The conjugate gradient method is associated with a symmetric $\mathbf{A}$ matrix, while the bi-conjugate gradient (BiCG) method allows for an asymmetric matrix. Γ is an unknown vector (flow quantities), representing p and $\tilde{\mathbf{u}}$ (or T) for the symmetric and asymmetric $\mathbf{A}$ respectively. α_{CG} and β are respectively the coefficients for calculating the steepest descent and the conjugate residual [80]. $\mathbf{r}^{(n)}$ is a conjugate residual vector of the linear equation system. The iteration loop terminates when $\mathbf{r}^{(n)}$ satisfies the convergence criterion.

Geometric agglomerated algebraic multigrid preconditioner

The multigrid method is proposed by Fedorenko et. al. [97] (geometric multigrid) and Poussian et al. [98, 99] (algebraic multigrid). The method is derived to increase the convergence rate for large algebraic systems by eliminating the low-frequency error [100]. The high-resolution mesh with significantly refined grids is not capable of sensing low-frequency oscillations, which results in the degradation of the convergence rate [80]. The multigrid method, on the other hand, uses the coarse grid to transform the low-frequency error into higher frequency error [80]. The grid coarsening is achieved by the direct grid agglomeration [101–103]. This approach is named as the algebraic multigrid method (AMG). The traversal of grid hierarchies from the fine grid to the coarser grid follows [80]:

1. Apply a restriction procedure.
2. Apply a setup or update of linear equations at coarse grids.
3. Apply smoother iterations.

Similarly, the traversal of grid hierarchies from coarse grids to fine grids follows [80]:

1. Apply a prolongation procedure.
2. Apply a correction of flow quantities at finer grids.
3. Apply smoother iterations.

In foam-extend-4.0, AMG is replaced by the geometric agglomerated algebraic multigrid (GAMG) method. GAMG uses the face area pair instead of the algebraic pair as an agglomerator. This face area pair agglomerator is claimed to outperform AMG method in most of the tested simulation setups [95].

According to the work of Imano [104] and Tatebe [105], the GAMG method performs more efficiently as a preconditioner for the CG solver than acting directly as an iterative solver in an highly parallelized simulation (large number of parallelized processors). Therefore, the GAMG method is used as a conjugate gradient preconditioner in the present simulation framework. The residuals agglomerated at each adhered coarsening mesh level is formulated as

$$\Gamma_i^{'(k)} = \Gamma_I^{(k+1)} - \Gamma_i^{(k)} \tag{2.24}$$

where $\Gamma_I^{(k+1)}$ is the I_{th} flow quantity element at (k+1) coarsening mesh level, while $\Gamma_i^{(k)}$ is the i_{th} element parent to I_{th} element at adhered finer (k) mesh level. So, $\Gamma_i^{'(k)}$ is the residual of k_{th} mesh level. Such residual is equivalent to the residual of Γ at n_{th} iteration of a CG iteration loop. Equation 5 in algorithm 2.1 can be reformulated as

$$\Gamma^{(n+1)} = \Gamma^{(n)} + \mathbf{P}^{-1}\mathbf{r}^{(n)} \tag{2.25}$$

where $\mathbf{P}^{-1}\mathbf{r}^{(n)} = \Gamma_i^{'(k+1)}$. $\mathbf{P}^{-1}\mathbf{r}^{(n)}$ can be substituted into the conjugate gradient algorithm 2.1. This way, the concept is achieved in terms of using the GAMG method as a preconditioner for the conjugate gradient solver.

Diagonal incomplete-LU preconditioner

The diagonal incomplete-lower-upper (DiLU) decomposition method [106] is used to construct the preconditioner **P** in a bi-conjugate gradient solver, since it is capable of efficiently decomposing the asymmetric matrices. The diagonal incomplete-LU factorization writes [80]

$$\mathbf{P} = (\mathbf{D}^* + \mathbf{L})\mathbf{D}^{*-1}(\mathbf{D}^* + \mathbf{U}), \tag{2.26}$$

where $\mathbf{L}^*$ and $\mathbf{U}^*$ are the lower and upper no fill-in matrix respectively. The no fill-in matrix is a matrix dropping nonzero elements in the original **L** and **U**, where zero elements appear at the same positions in matrix **A**. This way, instead of creating a precise **L** and **U**, the incomplete LU-factorization creates two matrices with the same number of non-zero elements as **A**. Therefore, no extra memory is required to store additional non-zero elements in factorized upper/lower matrices and the conservation of sparsity is secured. $\mathbf{D}^*$ is the no fill-in diagonal matrix and is calculated as

Algorithm 2.2: No fill-in diagonal matrix

```
Result: $\mathbf{D}^*$
for $i = 1 : N$ do
    $d_{ii} = a_{ii}$;
end
for $i = 1 : N$ do
    for $j = (i+1) : N$ && $a_{ij} \neq 0$ && $a_{ji} \neq 0$ do
        $d_{jj} = d_{jj} - \frac{a_{ji}}{d_{ii}} \times a_{ij}$;
    end
end
```

where d_{ij} and a_{ij} are the components in matrix $\mathbf{D}^*$ and **A** respectively.

2.2.5 Pressure-velocity coupling

The discretization scheme and linear algebraic solver, as part of the developed simulation framework, were introduced in section 2.2.3 and section 2.2.4 to solve each governing equation. Now a set of governing equations are used to solve the transient velocity, pressure, and temperature field in an incompressible fluid flow domain. OpenFOAM uses a segregated approach to sequentially solve the governing equations [80] for the incompressible flow problem. In this circumstance, algorithms are required to numerically couple the velocity and pressure field.

The PISO algorithm is proposed to handle the velocity-pressure coupling [107]. Indeed, since the simulation framework is developed based on buoyantBoussinesqPimpleFoam, the temperature transport equation and the gravity force term are also coupled by adjusting the Boussinesq approximation [28]. The PISO algorithm also allows for non-iteration loops and large simulation time steps for an unsteady simulation. Very few steps of corrections are necessary to achieve the desired accuracy. Moreover, a non-orthogonal correction loop is embedded in the PISO algorithm to recursively solve (or correct) the pressure flux equation to account for non-orthogonal grid elements [82]. The number of iterations is fixed a priori and set equal to 1 and 0 for the precursor and wind turbine/wake simulation respectively. Indeed, the lack of non-orthogonal correction on one side does not effectively influence the simulation convergence, since only 1% of the elements are non-orthogonal, while, on the other, enables to lower the computational cost by 10%.

OpenFOAM uses collocated grid to store the velocity, pressure, and temperature. The collocated grid often introduces the checkerboard problem, which is a sign of the pressure and velocity decoupling. The Rhie-Chow interpolation is proposed to solve the numerical issue by artificially establishing a pseudo-momentum equation at a cell face with its coefficients interpolated from the momentum equations at cell centroids. In this way, a small stencil is created by coupling the velocity and pressure field in a staggered grid arrangement [80].

2.2.6 Runtime performance optimization

The Runtime performance is associated with a numerical solver simulating a specified computational domain. So the optimization of the simulation runtime contains two factors, the decomposition of a domain and the scalability of a solver. In the section of domain decomposition, the kernel of parallelism for OpenFOAM is briefly introduced first, followed by the description of a customized decomposition method. In the section of scalability, the author first explains the parallelism in a solver and introduces a classic way of measuring the fraction of parallelism. Then he describes other factors, which might influence the parallel efficiency, including hardware and software aspects. Following this, the author describes the selection of compilers and module environments in an high performance cluster(HPC). The tuning of Message Passing Interface (MPI) parameters is also introduced. To this end, the final runtime performance of the developed solver is presented.

Domain decomposition

OpenFOAM uses domain decomposition to enable the process-level parallelism. The one-to-one mapping is used between the computing processors and the subdomains, which allows the single program multiple data parallelisms (SPMD) [108]. OpenFOAM employs the zero-halo layer as the communication scheme between each subdomain [88]. The overlapping elements are created by duplicating the cells attached to the processor boundaries. This way, the processor boundaries are considered as internal edges. The zero-halo scheme is integrated into the Pstream module, which is a class of OpenFOAM independent of linear solvers and scheme categories.

Apart from the principles for the parallelization of OpenFOAM, a well scheduled decomposed domain is also the key to balancing processor (task) workloads, minimizing processor boundary communications and eventually reducing simulation runtime performances. The Scotch decomposition method, supporting static load partitioning algorithms, is mainly used in this thesis [109, 110]. Scotch decomposition requires no geometric inputs from the user and is proposed to minimize the number of processor boundaries [82]. OpenFOAM has the Scotch decomposition embedded in its platform so that this method can be directly used. However, the Scotch code has to be modified to adopt the numerical issues of IB method. Due to the current limitations of the IB parallelization algorithm, decomposed domains are adjusted to ensure that immersed boundaries do not span multiple processors. A subroutine is implemented in the Scotch code to adjust the processor boundaries. The input entry writes

```
void Foam::scotchDecomp::modifyDecomposition.
```

The custimized input file reads

```
tumScotchCoeffs
{
    delta           0.001;
    k               1;   // number of boxes
    vertex01        (29.50  −0.94  0.00);
    vertex02        (30.80  −0.66  0.67);
},
```

where the $vertex$01 and $vertex$02 are two vertices of a cube diagonal. This way, the size and position of a processor boundary are properly defined.

Meanwhile, the research also indicates that the Scotch method is not capable of handling a large number of decomposed domains (>1000), which frequently costs the simulation hanging issue. It is probably due to the latency of the MPI interface communication somewhere on the processor boundaries. Therefore, a geometry-based decomposition method is proposed to deal with the computational domain with large numbers of processors. Instead of minimizing the processor boundaries, the geometry-based method tries to allocate an equal number of cells assigned to each processor partition. This way, the runtime performance is degraded, but the simulation stability is guaranteed.

Scalability

1. Amdahol's law

   To obtain a higher parallel efficiency, the fraction of computation that can be parallelized needs to be maximized. Amdahl's law is a formula used to measure the parallelization of a code by estimating the speedup of a simulation [111, 112]. S_{Amdahl} is the speedup of an executed task and is defined as

   $$S_{Amdahl} = \frac{1}{(1-f)+\frac{f}{n}} \tag{2.27}$$

   where f is a portion of a task, which can be parallelized, while n is the number of parallelized processors. If n increases infinitely, S_{Amdahl} will approach to the maximum speedup [113]. Moreover, if f rises up to the unity (a fully parallelized solver), the speedup of S_{Amdahl} will approach to the theoretical speedup curve and is therefore linearly proportional to the number of processors n.

   However, apart from the fraction of the parallelism, the scalability, in reality, is also influenced by other factors, such as the memory bandwidth, interconnect bandwidth, L3 cache, compiler, etc. Due to the nature of the CFD simulation, OpenFOAM requires a significant amount of processor boundary communication to guarantee consistent results across the domains. This means that even though the software is in theory infinitely parallelizable ($f = 1$), the extra processor boundary increases the communication overhead, which reduces the speedup. Additionally, as the physical size of a cluster is expended, system bottlenecks, such as the network latency, can prevent the runtime optimization from additional computing units [114]. Therefore, the linear scaling law is only a theoretical curve, which is certainly not achievable [115].

2. Hardware

   The supercomputing cluster used in this thesis is located in Leibniz-Rechenzentrum Zentrum, Garching bei München, Germany. The cluster uses the Intel Xeon 28-way Haswell-E5-2697 v3 with infini-band FDR14 as its Central processing unit (CPU) chip, which is the "fourth-generation core", successor to the Ivy Bridge microarchitecture [116, 117]. The CPU clock rate is 2.6 GHz and the rate for turbo-boost is 3.6GHz. The L3 Cache contains 35 MB of memory. The RAM uses DDR4-2133. The interface on the motherboard uses QPI (up to 9.6 GT/s), DMI 2.0, PCIe 3.0. The socket uses LGA 2011-3 [118].

Schröder [119] and Keough [114] have investigated the influence of hardware on OpenFOAM and have drawn the following conclusions:

(a) Turbo boost is a way to increase the clock rate. However, CPU frequency shows a limited impact on the runtime performance, if the frequency is higher than 2.5 GHz. Indeed, the longtime overclocking also damages the cluster and should not be intensively used.

(b) Large L3 cache improves the simulation speed. So, instead of using the fully populated task placement (FPTP) (assign all available processors in a node to a simulation), the compact task placement (CTP) is more favorable, since it offers more L3 cache for each processor.

(c) CTP configuration also prevents a memory channel from reaching its maximum capacity. Xeon E5-2697 has 4 memory channels and 28 processors (threads) for data transfers and computations respectively. It means that multiple processors share a single memory channel in a CPU unit. If the FPTP configuration is used, the OpenFOAM simulation can easily suffer from the limited memory bandwidth, which in other words, the overhead of the memory channel. Although the idling process of each floating point operation (due to the saturation of memory channel is) typically very short, the accumulated effects cost an extra of simulation runtime. On the other hand, CTP offers more memory bandwidth for each processor and, in principle, allows for faster simulation speed. Indeed, the tests have shown that CTP increases the simulation speedup up to around 300 parallelized processors. Beyond this range, the interconnect bandwidth dominates the runtime performance.

(d) Hyper-threading a capability to create virtual threads, which is designed to maximize the CPU utilization. However, similar to the use of FPTP, the memory bandwidth hinders the efficient use of CPU.

(e) Interconnect bandwidth, or in other words, MPI communication overhead, blocks the parallel efficiency for large parallelization numbers. However, the limitation of the interconnect bandwidth is not only a hardware issue but is also related the compiling and MPI parameter tuning.

To sum up, it is recommended to use CTP to allow more L3 cache and memory bandwidth. However, the influence of the memory bandwidth on the simulation speedup can be neglected for highly parallelized (< 300 processors) simulation runs. At this range of scales, the interconnect bandwidth of the hardware is largely responsible for the scalability of OpenFOAM.

3. Software

Foam-extend-4.0, a fork of OpenFOAM opensource library for CFD, is used as a solver platform in this thesis. The runtime performance of the framework is dependent on the solver platform because the platform contains the essential low-level codes, which significantly influences the computational efficiency. Tests have shown that the runtime performance of a solver (BuoyantBoussinesqPisoFoam), based on foam-extend-4.0, is similar to the same solver implemented in OpenFOAM 2.2.2 environment.

4. Compilers and module enviroment

   The software needs to be properly compiled to the hardware to achieve the optimal performance. To this aim, the compilers and other essential libraries require being carefully selected. Wyldckat [120] recommended the use of GNU Compiler Collection (GCC) compiler plus the OpenMPI library for the parallelization, whereas Intel offers a C++ proprietary compiler (ICC) and MPI library (IMPI) aimed to optimize the performance of Intel-based hardware. Since the Intel Xeon CPU chips are used in this thesis, it is necessary to try to compile Foam-extend-4.0 also using the Intel software. The Author investigated the best combination of C++ compiler and MPI library to compile Foam-extend-4.0. The results show that GCC compiler coupled with IMPI outperforms the other ways of coupling by nearly 20%.

   Apart from the compiler and MPI library, the compiling of CFD + aero-servo-elastic solver also requires other dependencies and libraries. The list of loaded module environment (compilers and libraries) reads

   ```
   1) admin/1.0   2) tempdir/1.0   3) intel/16.0   4) mkl/11.3
   5) lrz/default   6) gcc/6   7) mpi.intel/2017_gcc.
   ```

   The profiling of the compiled Foam-extend-4.0 requires significant manual work and is not included in this thesis. Massimiliano [115] systematically documented the OpenFOAM profiling and identified the bottleneck of its scalability. He found that the scalar product computed within the loop of the conjugate gradient algorithm (see section 2.2.4) requires intensive MPI AllReduced operations, which act as communication barriers. Indeed, AllReduced operation accounts for over 50% of MPI operations [108] and therefore significantly hinders the scalability. To minimize the AllReduced operation, Massimiliano suggested adding an additional cache-blocking mechanism to the storage format of sparse matrices. He also addressed the necessities of implementing a multi-threaded layer to AllReduced operation to reduce the communication time. On the other hand, Schröder [119] suggested Using binominal gather + scatter algorithm. These three recommended ways of improving the scalability are still under the investigation and were not implemented in the current framework.

5. Performence tuning

   There are few parameters designed for MPI tuning:

   (a) The process binding determines the processor and socket binding for a parallelized simulation. The default MPI setting assigns each operation to cores in a way that the workload on a system is balanced. The workload approach allows each floating point operation to be dynamically assigned to different cores, attempting to improve the overall parallel efficiency. This way, however, is limited by the interconnect memory bandwidth. Because each operation of an OpenFOAM simulation needs to read and write data on a local memory storage (RAM). The operation works more efficiently if the assigned RAM is attached to the same core running that operation. If no binding is enforced in the simulation, it is most likely that the assigned RAM is not attached to the computing cores. The large data transfer across the sockets will certainly cause the saturation of the interconnect memory bandwidth. Therefore, the reassigned operation results in the degradation of the runtime performance. In particular, since the OpenFOAM simulation is classified to the memory incentive simulation, the program will frequently rewrite the data (data swap) and could have a more detrimental effect on the runtime performance. As a result, "bind-to-core", the approach to binding the RAM attached to computing cores (the operation is also assigned to a fixed computing core), confirms to be the optimal binding strategy. Although the workload is not perfectly balanced for each operation of the simulation, the "bind-to-core" approach allows for faster runtime performance.

(b) The process binding is coupled with the distribution method to distribute the operation to each motherboard. MPI distributes operations consecutively by cores, sockets or nodes. "by-core" distributes operations within each socket. "by-node" distributes operation across the motherboard, which is constrained by the network connection. The investigation shows that "by-socket" is performed optimally among three options. This method is also proposed to make full use of the memory bandwidth under CTP configuration. If only a portion of processors in a socket is used for the parallel simulation (a typical CTP configuration), the"by-socket" option allocate the processors in a way that the memory channel is evenly distributed.

The execution command contains two additional MPI flags and writes

```
mpiexec –np XXX –bind–to–core –bysocket @exectuable –parallel.
```

6. Overall performance

   Fig. 2.4 shows a speedup map of the developed solver used in this thesis. The blue curve is obtained by using the optimal module environment and performing the MPI tuning. As a result, the overall parallel efficiency is equivalent to 87%. This value implies that a large number of parallelized processors (>1000) may not be a cost-effective solution. According to the figure, the use of 1120 processors wastes 62% of the potential simulation speedup, while the use of 560 processors only wastes 40% of the potential speedup. More importantly, the value of speedup is only improved by 22%, when the number of parallelized processors is doubled from 560 processors to 1120 processors. Therefore, the decomposition of 560 processors is employed as a reference setup for the formal production runs.

2.2.7 Integral velocity sampling

The integral velocity sampling method is originally proposed to solve the problem of an ambiguous velocity sampling location [67]. In the original SOWFA, the velocity is sampled at the center or interpolated center of each actuator element (pointwise sampling). Such assumption is valid when the projected actuator forces are isotropic in space. The sampled velocity is, therefore, not influenced by the vortex-induced velocity. However, if the non-isotropic projection approach is used, the way of velocity sampling has to be changed. Moreover, the accuracy of pointwise sampling also depends on the velocity interpolation and mesh resolution, which has to be tuned by the parameters and is undesired for a reliable simulation framework.

To overcome such issues, Spalart [67] derived a method to determine the sampling velocity using an integral of velocity point in space weighted by a projected force function $g(\mathbf{r})$ (see section 2.2.1). The velocity sampling equation reads

$$\tilde{\mathbf{u}}_s = \int g(\mathbf{r})\tilde{\mathbf{u}}\mathbf{r}, \tag{2.28}$$

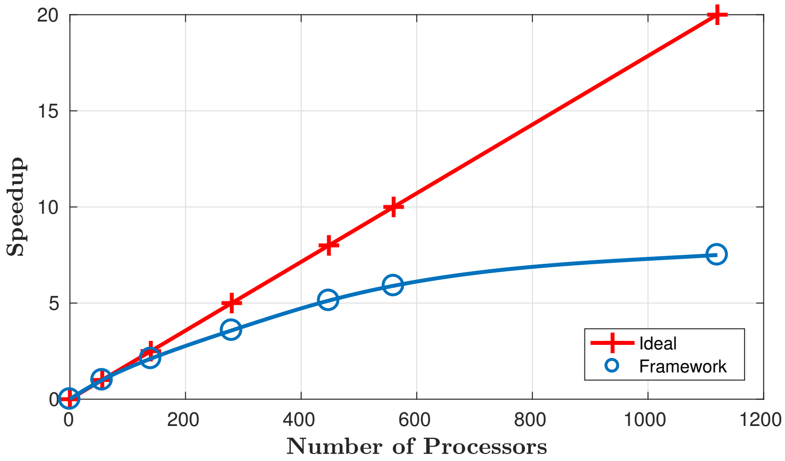

Figure 2.4: Red curve: linear speedup. Blue curve: speedup of the developed high-fidelity simulation solver.

where $\tilde{\mathbf{u}}_s$ is the sampled velocity. This way, the formulation is associated with $g(\overrightarrow{\mathbf{r}})$, which excludes the undesired influence of the bound vorticity induced by an non-isotropic force projection.

In addition, the integral velocity sampling method also reduces the dependency of velocity sampling on the tuning parameter ϵ. The pointwise sampling approach samples the velocity point at the center of an actuator line element, the same location where the maximal value of $\mathbf{F}$ (projected body force), $\mathbf{f}_{i,max}$, is located. So, the sampled velocity is sensitive to $\mathbf{f}_{i,max}$. Moreover, $\mathbf{f}_{i,max}$ depends on the projected Gaussian curve and ϵ determines the shape of the Gaussian curve (see section 2.2.1). Therefore, $\mathbf{f}_{i,max}$ and, consequently, the sampled velocity are sensitive to ϵ. On the other hand, the integral velocity sampling is subjected to a weighted average calculation, which considers $\mathbf{F}$ instead of $\mathbf{f}_{i,max}$. Since the value of $\mathbf{F}$ is not determined by the shape of the Gaussian curve, the integral sampling approach reduces the dependency of the velocity sampling on ϵ.

The sampled velocity is used to calculate power capture (P) and reads [121]

$$P = \frac{1}{4} RBC\tilde{\mathbf{u}}_s^3 \lambda (C_L \sin\chi - C_D \cos\chi), \tag{2.29}$$

where R is the rotor radius, B is the blade number, C is the chord length, λ is the tip speed ratio and chi is the flow angle. Since ϵ is loosely coupled with the integral velocity sampling and P is determined by the sampled velocity. Therefore, P has reduced the dependency on ϵ.

2.2.8 Sub-grid scale model

The concept of the sub-grid scale (SGS) model is to reduce the computational costs by modeling the small-scale (sub-grid scale) turbulence and resolving the large-scale turbulence [122]. Regarding the sub-grid scale turbulence, instead of a universally valid turbulence description, the SGS model uses an empirical form to model the deviatoric part of the fluid tensor term, $\tau_{\mathbf{SGS}}^{\mathbf{D}}$, at sub-grid scales in equation 2.1. Next step, a low-pass filter is proposed to identify the sub-grid scale turbulence by filtering the Naiver-Stokes equation on the temporal and spatial domain [123, 124]. In this thesis, the temporal and spatial filtering is regarded, respectively, as the discretized simulation timestep and the discretized computational mesh.

The computational mesh used in this thesis is mainly composed of the structural grid (>99%) with the perfectly cubic shape. Such mesh setup is compatible with the isochoric sub-grid scale model, which suggests that a simple top-hat spatial filter is sufficient for a good simulation accuracy. The LES filter size Δ is equal to $\Omega_{CV}^{1/3}$ (the grid limit of the domain), which indicates that the formulations used in this thesis regard the grid cell as the top-hat spatial filter. Therefore, no additional filtering process is required. The formulation also represents that the sub-filter scale (SFS) [125] is equivalent to the sub-grid scale and there is no resolved sub-filter scale appeared under the current mesh setup.

In this section, the three sub-grid scale models adopted in the simulation framework are introduced, namely the Constant Smagorinsky (CS) model, the dynamic Lagrangian Smagorinsky (LDS) model and the scale-adaptive simulation (SAS) model. The two Smagorinsky models formulate the C_S term, which is used to calculate the eddy viscosity and eventually the sub-grid scale tubulence. So, they are clearly classified to the sub-grid scale modeling, while the scale-adaptive simulation model is derived from the $k-\omega$ SST formulation. Indeed, the SAS model is supposed to be classified to RANS or URANS formulation. However, the SAS velocity field is not subjected to the temporal filtering process (or a steady state solution) and the top-hat (grid) spatial filter is also used for the SAS model. Therefore, instead of a URANS-type formulation, $\tilde{\mathbf{u}}$ for the SAS model is subjected to a typical LES filtering process. In this regard, the formulation can be considered as an LES dynamic sub-grid scale model, which uses the k and ω transport equation to solve the eddy viscosity field and use ν to model the sub-grid scale field. The SGS model, in general, exhibits an LES-like behavior at the resolved scales. Similar formulation is described in [69].

Constant Smagorinsky model

The sub-grid scale approach adopted in this paper is the functional model, which in other words is an eddy-viscosity ν_t model. ν_t is formulated as

$$\nu_t = (C_s \Delta_g)^2 |\mathbf{S}| \tag{2.30}$$

where C_s is the Smagorinsky constant [124–126], Δ_g is the grid size, $\mathbf{S} = \frac{1}{2}\left(\nabla\tilde{\mathbf{u}} + (\nabla\tilde{\mathbf{u}})^T\right)$ is the rate-of-strain tensor. The constant Smagorinsky model simply means a time and spatial invariant C_s coefficient pre-specified in the whole computational domain. After obtaining ν_t, $\tau_{\mathbf{SGS}}^{\mathbf{D}}$ can be calculated using $\tau_{\mathbf{SGS}}^{\mathbf{D}} = -2\nu_t S$,

Lagrangian averaging dynamic Smagorinsky model

Instead of prescribing a single Smagorinsky constant (C_s) in a flowfield, the transient local C_s values can be dynamically calculated based on the resolved turbulence scales. Ghaisis [127] investigated a dynamic SGS modeling of the buoyant flow simulation and showed the dependencies of a wind farm simulation to the dynamic SGS model.

Meneveau et al. [128] proposed a numerical approach to calculating the time-dependent local C_s values using Lagrangian fluid trajectories. This dynamic Smagorinsky model can potentially improve the wind farm simulation, as the flow in a wind farm is, in general, inhomogeneous flow. The inhomogeneous turbulence can be interpreted as the non-uniform C_s in the flowfield, because C_s represents the local numerical diffusion ν_t and ν_t mimics the local turbulence. Therefore, it is important to dynamically determine the local C_s values, so as to inject the proper numerical diffusion into the flowfield.

Prior to the Lagrangian averaging dynamic Smagorinsky model, the dynamic Smagorinsky model proposed by Germano et al. [129] is introduced. Two filters are employed for the dynamic Smagorinsky formulation, namely a grid LES filter at scale Δ (subscript $(\bar{\cdot})$, identical to $(\tilde{\cdot})$) and a test LES filter at scale 2Δ (subscript $(\hat{\cdot})$). The deviatoric part of a stress tensor at two scales writes

$$T_{ij}^{D} = \widehat{\overline{u_i u}_j} - \hat{\bar{u}}_i \hat{\bar{u}}_j \quad and \quad \tau_{ij}^{D} = \widehat{\overline{u_i u}}_j - \widehat{\overline{u}_i \overline{u}}_j. \tag{2.31}$$

The derived model samples the turbulence stress from a band of turbulence (from Δ to 2Δ) resolved at the smallest scales and use the information for the sub-grid scale modeling. The Germano identity is defined as

$$\zeta_{ij} = T_{ij}^{D} - \tau_{ij}^{D} = -2C_s^2 \Delta^2 |\overline{S}|\overline{S}_{ij} + 2C_s^2 (2\Delta)^2 |\hat{\overline{S}}|\hat{\overline{S}}_{ij}, \tag{2.32}$$

where $\overline{S}_{ij}$ and $\hat{\overline{S}}_{ij}$ are the resolved rate-of-strain tensor at the scale of Δ and 2Δ. The equation can be directly computed by the resolved velocity field, $\tilde{\mathbf{u}}$ and is, therefore, computationally inexpensive. However, C_s cannot be directly calculated, as the Germano identity is a tensorial equation. The system (an overdetermined system of five equations [128]) overdetermines the unknown (C_s). The error used for computing the Germano identity is formulated as

$$e_{ij} = \zeta_{ij} - 2\Delta^2 \left[\widehat{C_s^2 |\overline{S}|\overline{S}_{ij}} - 4C_s^2 |\hat{\overline{S}}|\hat{\overline{S}}_{ij} \right]. \tag{2.33}$$

Several attempts were proposed to minimize the error e_{ij}. Lilly [130] and Ghosal et al. [131] use a minimum least-square error method and reads

$$C_S^2 = \frac{\langle \zeta_{ij} M_{ij} \rangle}{\langle M_{ij} M_{ij} \rangle}, \tag{2.34}$$

where $\langle \cdot \rangle$ denotes the spatial average over directions of the statistical homogeneity. This avoids the negative eddy viscosity during a trasient simulation. The averaging operation also stabilizes C_s values under the assumption that C_s is invariant of scales [132]. M_{ij} reads

$$M_{ij} = 2\Delta^2 \left[\widehat{|\overline{S}|\overline{S}_{ij}} - 4|\hat{\overline{S}}|\hat{\overline{S}}_{ij} \right]. \tag{2.35}$$

The Lagrangian dynamic sub-grid scale model is proposed to improve the spatial averaging operation in equation 2.34. The calculation of C_s is formulated by the ratio of two pathline integrals and reads

$$C_s^2(x,t) = \frac{\Psi_{\zeta M}}{\Psi_{MM}}, \tag{2.36}$$

where Ψ_{LM} and Ψ_{MM} are formulated as

$$\Psi_{\zeta M}(\mathbf{x},t) = \int_{-\infty}^{t} \zeta_{ij} M_{ij}(\mathbf{z}(t'),t') W(t-t') dt', \tag{2.37a}$$

$$\Psi_{MM}(\mathbf{x},t) = \int_{-\infty}^{t} M_{ij} M_{ij}(\mathbf{z}(t'),t') W(t-t') dt', \tag{2.37b}$$

where $W(t-t') = T_{int}^{-1} e^{-(t-t')/T_{int}}$, a weighting function defining the extent background along a fluid-particle trajectory (pathline). This weighting function averages the trajectory with an exponentially decreasing memory. $\mathbf{z}(t')$ is the trajectory for earlier times ($t' < t$) at Lagrangian frame of reference. Meneveau et al. [128] averages $L_{ij}M_{ij}$ along the trajectory to compute the present value of C_s. To efficiently solve Ψ_{LM} and Ψ_{MM}, a set of relaxation-transport equations are formulated, and writes

$$\frac{D\Psi_{\zeta M}}{Dt} = \frac{\partial \Psi_{\zeta M}}{\partial t} + \tilde{\mathbf{u}} \cdot \nabla \Psi_{\zeta M} = \frac{1}{T_{int}}\left(\zeta_{ij}M_{ij} - \Psi_{\zeta M}\right), \tag{2.38a}$$

$$\frac{D\Psi_{MM}}{Dt} = \frac{\partial \Psi_{MM}}{\partial t} + \tilde{\mathbf{u}} \cdot \nabla \Psi_{MM} = \frac{1}{T_{int}}\left(M_{ij}M_{ij} - \Psi_{MM}\right), \tag{2.38b}$$

Compared to the directional spatial averaging method, the Lagrangian sub-grid scale model considers the turbulence development history and the flow structure, which offers more physical meanings into the time-averaging calculation. The formulations of Lagrangian dynamic (LD) model also shows that it is a tuning free SGS model, requiring no extra manual input. However, the LD model needs to solve two additional transport equations, which costs 20% more the time consumption.

Scale-adaptive simulation model

The derivation of the SAS model follows the work of Menter et al. [69, 133], Egorov et al. [134] and Lindblad [135], and it is formulated in incompressible form with minor modifications.

Similar to the k-ω SST model, the kinematic eddy viscosity is modeled as $\nu_t = k/\omega$. In the SAS model, however, an additional source term Q_{SAS} is introduced to improve ν_t. The k and ω transport equations of the SAS model read

$$\frac{\partial k}{\partial t} + \nabla \cdot (\tilde{\mathbf{u}} k) = P_k - c_\mu k \omega + \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_k}\right)\nabla k\right], \tag{2.39}$$

$$\begin{aligned} \frac{\partial \omega}{\partial t} + \nabla \cdot (\tilde{\mathbf{u}}\omega) &= \alpha_{SAS}\frac{\omega}{\rho k}P_k - \beta_{SAS}\omega^2 + Q_{\mathrm{SAS}} \\ &+ \nabla \cdot \left[\left(\nu + \frac{\nu_t}{\sigma_\omega}\right)\nabla \omega\right] + (1 - F_1)\frac{2}{\sigma_{\omega_2}}\frac{1}{\omega}\nabla k \cdot \nabla \omega, \end{aligned} \tag{2.40}$$

where c_μ and σ_k are the closure coefficients in the k equation, while α, β, σ_ω and σ_{ω_2} are the closure coefficients in the ω equation. F_1 is a blending function that transitions between the k-ω and k-ϵ modes. The dissipation terms $c_\mu k\omega$ and $\beta\omega^2$ are discretized in implicit form to improve convergence and stability [135]. The present implementation does not consider any prediction-correction iteration between the momentum and the two transport equations: in the k and ω equations, $\tilde{\mathbf{u}}$ is considered as a known velocity field resolved by the PISO scheme, solving the transport equations in segregated form [135]. Therefore, the implementation of SAS does not require a change in the PISO algorithm.

Through the boundary layer length scale, the blending term is in charge of shifting between boundary layer and free-stream type conditions, which is the main idea behind the SST approach. Such formulation, however, cannot detect local flow inhomogeneities and requires the modeling of an additional source term. The Q_{SAS} term originates from Rotta's k-kL model of the correlation-based length scale [134] , and it is formulated as

$$Q_{\mathrm{SAS}} = F_{\mathrm{SAS}} \cdot \max\left(\zeta_2 \kappa S^2 \left(\frac{L}{L_{vK}}\right)^n - C\frac{k}{\sigma_\Phi}\max\left(\frac{|\nabla\omega|^2}{\omega^2}, \frac{|\nabla k|^2}{k^2}\right), 0\right), \tag{2.41}$$

where parameters ζ_2, σ_Φ and C were obtained from experiments. F_{SAS} behaves as a scaling parameter that dictates the amount of numerical damping injected in the flowfield. The value of F_{SAS} requires specific calibration. L is the length scale of the modeled turbulence, and L_{vK} the von Kármán length scale. In order to preserve the SST characteristics of the formulation, the Q_{SAS} term is defined as a strictly positive term. Regarding the scaling ratio $(L/L_{vK})^n$, numerical experiments have shown that the linear length scale ratio (also used in [134]) leads to better stability and robustness of the formulation when compared to the quadratic form of [136, 137]. The choice $n = 1$ is therefore adopted in this work.

The von Kármán length scale L_{vK} is formulated based on Rotta's equation, and it writes

$$L_{vK} = \max\left(\frac{\kappa|\mathbf{S}|}{\left|\nabla^2\tilde{\mathbf{u}}\right|}, C_o\Omega_{CV}^{1/3}\right). \tag{2.42}$$

ensuring the scale-adaptive characteristics of the method [69]. In fact, L_{vK} reflects the size of resolved eddies in the flow, while the SST model only considers length-scales associated with the boundary layer thickness.

The second derivative of the velocity field $\nabla^2\tilde{\mathbf{u}}$ detects inhomogeneities in the resolved turbulence scales, and brings this information into the eddy viscosity term. As mentioned earlier, the SST model is only associated with the boundary layer length scale (i.e., the F_1 term), but it is not adjusted to the local flow characteristics [69, 138]. Therefore, when the SST model is used in free-stream high-Reynolds wake flows, it cannot adapt ω in the wake regime, often resulting in a highly diffusive behavior. By introducing $\nabla^2\tilde{\mathbf{u}}$, the improved formulation is capable of adjusting the eddy viscosity to better control the amount of numerical diffusion to the local characteristics of the flow. In addition, to avoid an insufficient damping at high wave numbers, a lower limit to L_{vK} is imposed, which is proportional to the model parameter C_o and the cell size $\Omega_{CV}^{1/3}$, Ω_{CV} being the cell volume.

2.2.9 Floating solver

Floating algorithm

A floating wind turbine algorithm is developed based on the current simulation framework to enable FOWT simulation, as already introduced in section 1.1.3. The development of the floating solver consists of moving wind turbine blades and moving nacelle & tower. It is trivial to achieve the floating capability for the turbine blades, since actuator line elements are used to model the blades. Because the location and orientation of the projected body forces from an actuator line element can be easily manipulated (translate, rotate or even bend) and there is no need for the regeneration of surrounding meshes. On the other hand, the nacelle and tower use the immersed boundary model to model their structure. Since the adopted immersed boundary method uses the discrete forcing approach, the immersed boundary is imposed after discretizing the mesh. The floating algorithm, therefore, involves local cell treatment and index allocation.

The implementation for actuator modeled blades is introduced first. To simplify the implementation, the FAST model is not actually moved within the CFD domain, but the relevant coupling quantities are transformed according to the rigid body motion induced by the platform. It should be noticed that this approach is limited to a rigid wind turbine model, and it would not be applicable to a flexible one, as the solver doesn't contain a fluid-structure coupling algorithm. Following this procedure, the components $\mathbf{x}_s$ of the position vector of a point in the structural model frame (subscript $(\cdot)_s$) are transformed into the components $\mathbf{x}_f$ in the fluid model reference frame (subscript $(\cdot)_f$) as $\mathbf{x}_s = \mathbf{R}\mathbf{x}_f$, where $\mathbf{R}(t)$ is the time-dependent direction cosine rotation matrix associated with the given prescribed motion [139], which writes

$$\mathbf{R}(t) = \begin{bmatrix} \cos\theta_P(t) + \Upsilon_x^2\xi & \Upsilon_x\Upsilon_y\xi - \Upsilon_z\sin\theta_P(t) & \Upsilon_x\Upsilon_z\xi + \Upsilon_y\sin\theta_P(t) \\ \Upsilon_x\Upsilon_y\xi + \Upsilon_z\sin\theta_P(t) & \cos\theta_P(t) + \Upsilon_y^2\xi & \Upsilon_y\Upsilon_z\xi - \Upsilon_x\sin\theta_P(t) \\ \Upsilon_x\Upsilon_z\xi - \Upsilon_y\sin\theta_P(t) & \Upsilon_y\Upsilon_z\xi + \Upsilon_x\sin\theta_P(t) & \cos\theta_P(t) + \Upsilon_z^2\xi \end{bmatrix}, \tag{2.43}$$

where ξ equals to $1-\cos\theta_P(t)$, $\theta_P(t)$ is the variation of the pitching angle for a floating platform, and unit vector $\Upsilon = [\Upsilon_x\Upsilon_y\Upsilon_z]$ is the platform rotation axis (at fluid model reference frame). The value $\theta_P(t)$ is scanned from a prescribed list, which can be a replication of the test campaign. The velocity vector components in the two frames are readily obtained as

$$\dot{\mathbf{x}}_s = \omega \times (\mathbf{R}\mathbf{x}_f) + \mathbf{R}\dot{\mathbf{x}}_f, \tag{2.44}$$

where $\omega(t)$ indicates the components of the angular velocity vector, and $(\dot{\cdot})$ indicates a derivative with respect to t. Therefore, given the velocity at a point in the CFD domain, the velocity seen by the lifting line on the FAST model can be readily computed. Once the corresponding aerodynamic force $\mathbf{f}_s$ at that lifting line station has been computed based on the stored lift and drag look up tables, that same force can be mapped back to the CFD frame by the inverse transformation, i.e. $\mathbf{f}_f = \mathbf{R}^T\mathbf{f}_s$.

Next, the moving algorithm used for the nacelle and tower is introduced. There are several geometry motion techniques proposed to achieve the moving capability for an immersed boundary modeled geometry, which includes the offset mesh, and deformable mesh [70, 140, 141]. Those approaches usually require high computational cost and complex coding efforts. However, the motion algorithm for the immersed boundary method can be significantly simplified, as it can use the non-topological change.

The immersed boundary subroutine applies a similar transformation as actuator elements in order to account for motion-induced effects. This approach dynamically allocates the geometry profile to the existing mesh cells within the domain according to the prescribed geometry motion, rather than creating new cells to accommodate the moved geometry. This way, only the geometry surfaces need to be dynamically identified because of the body motion, while the grid does not. No mesh regeneration is required. The computational cost is therefore drastically reduced.

Fig. 2.5 shows a flow chart for the developed FOWT framework. The data transfer interfaces between OpenFOAM and FAST are presented in the two dashed boxes. In essence, the transferred data consists of sampled velocities, projected forces and locations. The flow chart also shows the output files that the program writes, the prescribed motion script for the actuator line elements and the immersed boundary modeled geometries.

This procedure reconstructs the velocity at the lifting line and boundary surfaces, but clearly not the accelerations. Therefore, its use is limited to a rigid model and to cases where one is not interested in the complete loading of the structure, but only on the resulting aerodynamic forces. However, this is indeed the scope of the present work, and the use of this approach results in a much reduced computational cost than in the case of a more rigorous fluid-structure interaction formulation.

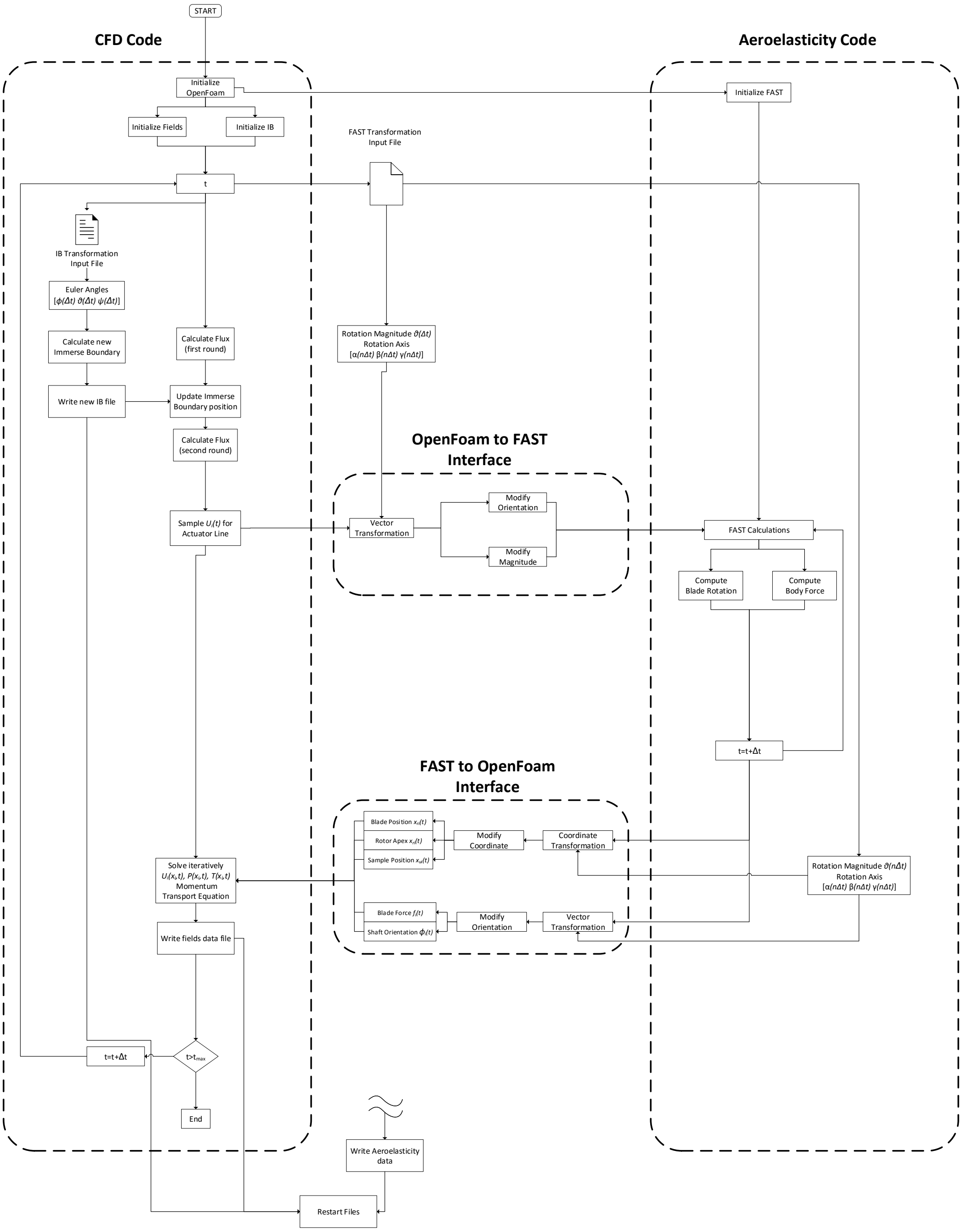
START
CFD Code
Aeroelasticity Code
Initialize OpenFoam
Initialize FAST
Initialize Fields
Initialize IB
FAST Transformation Input File
t
IB Transformation Input File
Euler Angles [$\phi(\Delta t)$ $\vartheta(\Delta t)$ $\psi(\Delta t)$]
Calculate new Immerse Boundary
Write new IB file
Calculate Flux (first round)
Update Immerse Boundary position
Calculate Flux (second round)
Rotation Magnitude $\vartheta(\Delta t)$ Rotation Axis [$\alpha(n\Delta t)$ $\beta(n\Delta t)$ $\gamma(n\Delta t)$]
OpenFoam to FAST Interface
Sample $U_i(t)$ for Actuator Line
Vector Transformation
Modify Orientation
Modify Magnitude
FAST Calculations
Compute Blade Rotation
Compute Body Force
t=t+Δt
FAST to OpenFoam Interface
Blade Position $x_{ri}(t)$
Rotor Apex $x_{ri}(t)$
Sample Position $x_{ui}(t)$
Modify Coordinate
Coordinate Transformation
Rotation Magnitude $\vartheta(n\Delta t)$ Rotation Axis [$\alpha(n\Delta t)$ $\beta(n\Delta t)$ $\gamma(n\Delta t)$]
Blade Force $f_i(t)$
Shaft Orientation $\phi_i(t)$
Modify Orientation
Vector Transformation
Solve iteratively $U_i(x_i,t)$, $P(x_i,t)$, $T(x_i,t)$ Momentum Transport Equation
Write fields data file
t=t+Δt
$t>t_{max}$
End
Write Aeroelasticity data
Restart Files

Figure 2.5: Flow chart for floating-offshore wind turbine simulation framework

Restart capability

The code implements a complete restart capability for all its coupled components, including CFD solver, structural dynamic model, immersed boundary formulation and wind turbine controllers. This is of great practical utility given the large computational cost of these simulations, as it allows for consecutive shorter runs. The ability to restart a simulation is obtained by saving to files all necessary internal information, so that a seamless solution is obtained, identical to the one that would have been computed without restarting.

While FAST already allows for restarts (The *Write Aeroelasticity data* in Fig. 2.5), other components of the code has to be modified. The OpenFOAM immersed boundary subroutine reads a CAD file and uses geometric information to identify the boundary surface in the CFD domain for each timestep [78]. If the geometry moves, the solver computes the associated displacement and creates a new CAD file, which therefore needs to be saved and read back for the restart run. The *Write new IB file* in Fig 2.5 is the IB file saved at the end of each simulation, recording the essential status of problem geometry. This IB file is used as an input geometry for the restart runs.

CHAPTER 3

Computational and Experimental setups for Framework analysis

This chapter introduces experimental and simulation setups for the framework Comparison. The experimental facility is introduced first, including a boundary layer wind tunnel, a scaled wind turbine model and a scaled floating wind turbine platform. A number of test campaigns have been conducted in this facility, which were eventually used for the comparison. The computational setup is introduced next. The computational domain is separated to two sections, namely the precursor domain and successor (wind turbine wake simulation) domain. The computational mesh and boundary condition of the simulations performed in this thesis is also described in this chapter.

3.1 Experimental facility

3.1.1 Boundary layer wind tunnel

The experiments were conducted in a boundary layer wind tunnel at Politecnico di Milano (POLIMI) [57]. The wind tunnel is a closed circuit facility vertically connected with two test rooms (see Fig. 3.1). The larger test room has the size of 36 m (length) × 13.84 m (width) × 3.84 m (height) and allows for boundary layer experiments. The inflow wind speed ranges from 0 to 15 m/s. The very long test room (36 m) is designed to perform a scaled fully-developed atmospheric boundary layer. A cooling system (heat exchanger) enables a consistent temperature environment in experiments.

Two arrays of 12-bladed electric fans generate the air inflow with an average turbulence intensity of 2%. The maximum allowed power output equals to 1.5 MW. The operating test facility is monitored by a programmable logic controller (PLC) and a computer network through wind speed, temperature, vibration, pressure, etc. The control and monitoring system ensures a stable operation of the wind tunnel.

3.1.2 Wind turbine model

G1 wind turbine model

Tests were performed with up to three scaled wind turbine models (marked as G1 model), whose rotor diameter (D) and rated rotor speed are equal to 1.1 m and 850 RPM, respectively [59]. The rotor is equipped with three blades, which is composed of a unidirectional carbon fiber covering a Rhoacell. The rotor uses two bearings to connect to the nacelle. The pitch motors are housed at blade roots and are equipped with a gearhead and an encoder. The entire nacelle can

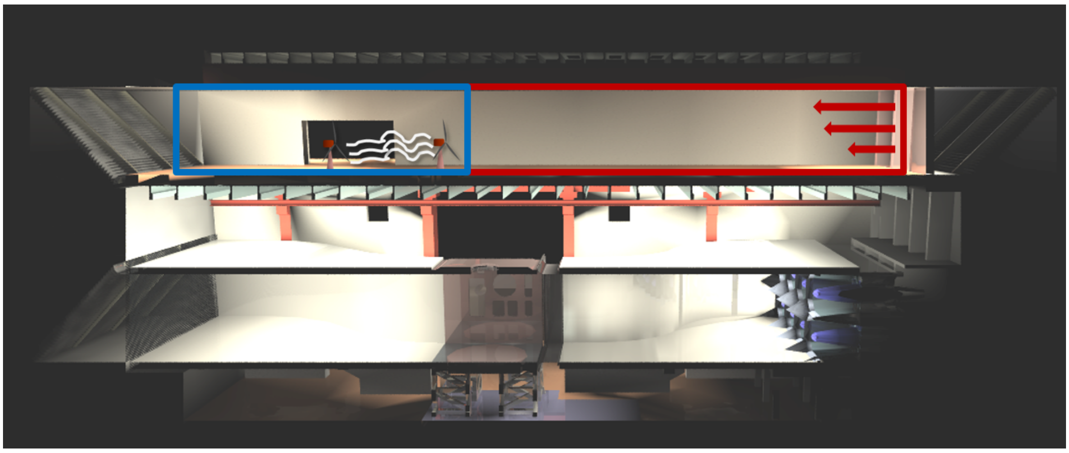

Figure 3.1: Picture for the sectional view of the boundary layer wind tunnel at Politecnico di Milano (POLIMI) [57].

be misaligned using the yaw motor and is fixed on the top of the tower. Some characteristics of the G1 model is illustrated in table 3.1.

Table 3.1: Main characteristics of G1-type wind turbine model.

Type	G1
Rotor orientation	Upwind
Rotor diameter	1.1 m
Rated speed	6.05 $m \cdot s^{-1}$
Rated angular speed	850 RPM
Rotating direction	Clockwise, facing downstream
Control	Variable speed, torque, pitch and yaw control

The low-Reynolds airfoil RG14 is used as the airfoil shape for G1 model [142]. The rotor aerodynamic performance was measured at different values of Re, tip speed ratio, and blade collective pitch. The measured maximum power coefficients equal to 0.42 at $TSR \approx 7.5$ and $\theta_P \approx -1$ deg.

The model, already used within other research projects [143–145], is characterized by a realistic aerodynamic performance at rotor levels and its wake is characterized by shape, deficit, and recovery that are in good accordance with those of full-scale machines. Moreover, the model features active individual pitch, torque and yaw control that, together with a comprehensive onboard sensorization of the machine (including measures of shaft and tower loads), enables the testing of modern control strategies similar to the ones described in [146] and references therein.

The G1 pitch and torque are managed by a closed-loop controller, implemented on a M1 Bachmann hard-real-time module. The controllers react to the turbulent inflow variations, as well as to the periodic relative-speed changes produced by the floating platform oscillations. The controller computes the desired blade pitch and torque demands, which are in turn sent to the actuator control boards. All wind turbine models use the same control algorithm that targets two main functions: power control and load reduction. In some wind turbine simulations, the same wind turbine torque and pitch controller used for G1 model is integrated into the solver to account for the dynamic characteristics of the wind turbine model [147].

Instead of prescribing the rotor speed and blade pitch angles, the controllers identify the values to adopt the balance between aerodynamic and generator torque.

5MW wind turbine model

5 MW NREL reference wind turbine is designed initially to support conceptual studies of the offshore wind technology [148]. This wind turbine is a three-bladed, upwind, variable-rotating-speed and blade-pitch-to-feather-controlled machine. Some characteristics of the 5MW wind turbine are illustrated in table 3.2.

Table 3.2: Specifications for 5MW NREL wind turbine model

WT Type	5 [MW] NREL
Rotor orientation	Upwind
Rotor diameter	126 m
Rated inflow speed	11.4 $m \cdot s^{-1}$
Rated rotating speed	12.1 RPM
Control	variable speed, pitch, yaw control

3.1.3 Floating platform

A wind tunnel experimental campaign [45] was conducted using a scaled model of a floating platform concept, developed by the Korea Research Institute of Ships & Ocean Engineering (KRISO) [46]. A square-shaped platform is equipped with four very closely spaced wind turbines located at its corners, allowing one degree of freedom (DoF) fore-aft pitching motion. Fig. 3.2 shows the XY plane, XZ plane, YZ plane and three-dimensional view of the floating platform CAD geometry.

3.1.4 Inflow measurement

Previous experimental testing in the POLIMI facility has revealed that the inflow generated by the wind tunnel is non-uniformly distributed, with variations in flow velocity of up to 6%. The inhomogeneity of the flow is non-negligible and it will influence the operation of wind turbine models and of the associated flowfield. To address this problem, instead of imposing a uniform inflow for the inlet boundary condition, a non-uniform inflow was used to more accurately represent the incoming flow generated in the wind tunnel. Properly capturing the inflow velocity requires a full cross-sectional wind map to be recorded upstream of the wind turbines, rather than sampling a single point (or several scattered points in space) at hub height. The measured experimental wind map can then be implemented as a inlet boundary condition to the CFD simulations.

In order to obtain a reliable high resolution wind map, two synchronized scanning LiDARs were employed. LiDARs are instruments equipped with a steering laser beam that fires rapid pulses and, by measuring the frequency shift of the backscattered light from the aerosols present in the air, are able to record the line-of-sight component of the wind speed. In this thesis, LiDAR data was obtained from the collaborative efforts of ForWind-Oldenburg, TUM, Technical University of Denmark and POLIMI [60]. Fig. 3.3 shows a time-averaged velocity scanned inflow map. The figure clearly shows the lack of uniformity of the inflow, probably due

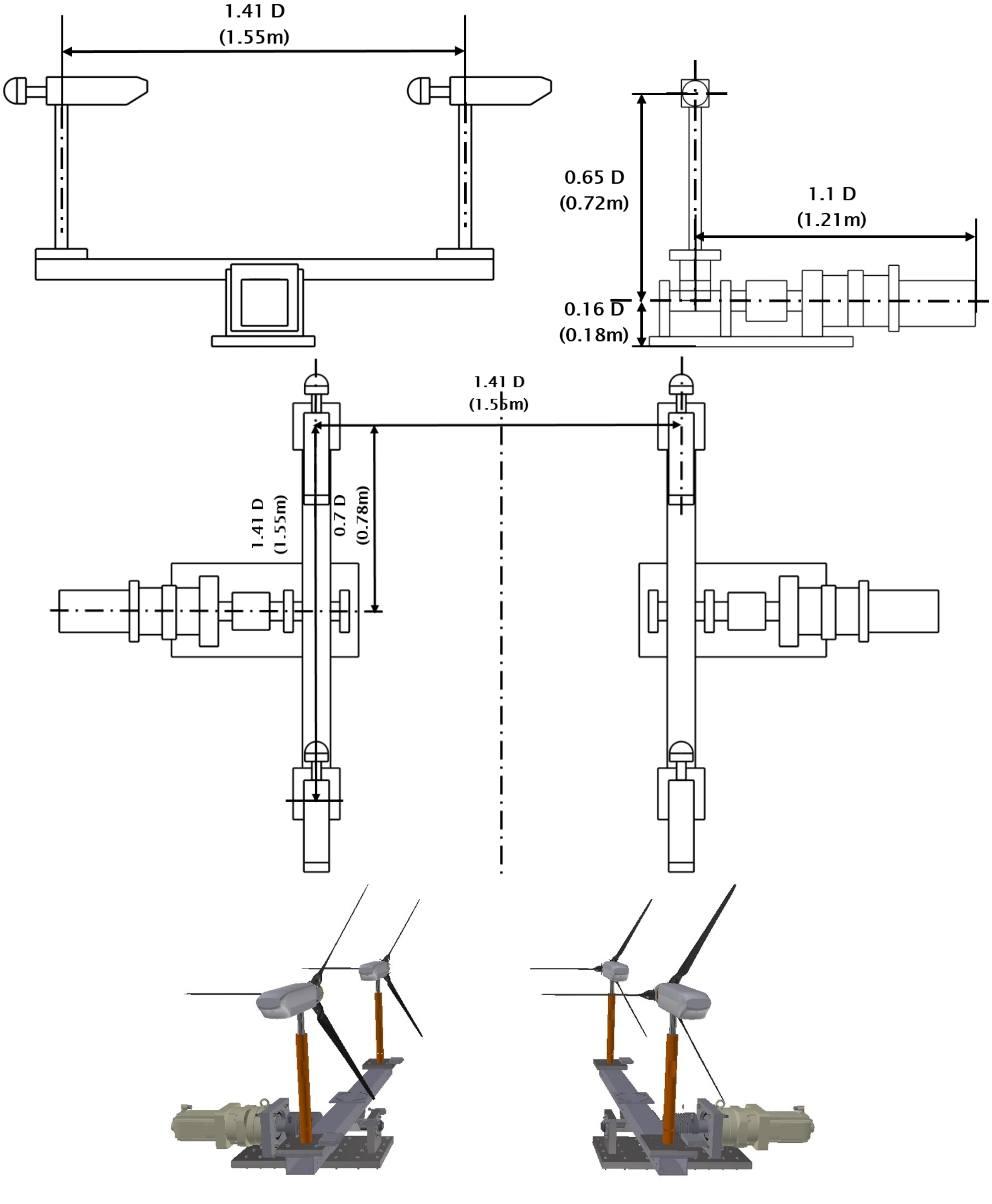

Figure 3.2: Top left: XZ plane of floating platform equiped with two wind turbine models, top right: YZ plane, moddile: XY plane, bottom: 3D platform model.

to the presence of a discrete number of fans and of the structural arrangement of the tunnel just upstream of the test section. The black circle indicates the location of a rotor, which appears to be placed in correspondence with a local peak of the flow velocity.

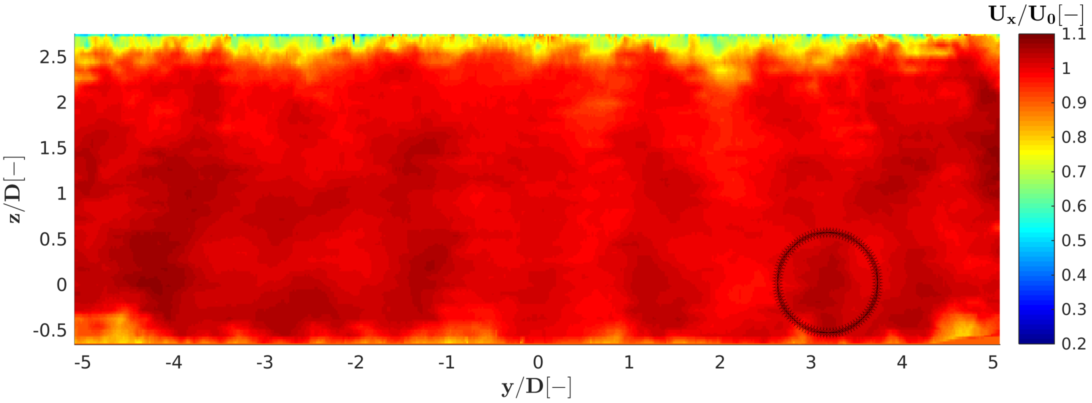

Figure 3.3: Inflow velocity profile (facing upstream) as measured by two scanning LiDARs at a 3D distance upstream of the wind turbine model.

3.1.5 Wake measurement

The flow within the wind tunnel was measured with hot-wire probes or stereo PIV, as shown in Fig. 3.4. The latter technique was used to measure the flow characteristics in the near (0.56D) and far (6D) wake regions. The measurement planes cover a significant fraction of the wind turbine wake. In order to achieve a higher spatial resolution of the velocity field, the measurement area was divided into several smaller windows with small overlaps between them. A rapid scanning of the entire measurement area was achieved by the use of an automated traversing system, moving both the laser and the cameras. The measuring windows were divided into 32×32 pixel interpolation areas, which resulted in an approximatively 15 mm spatial resolution. For each measuring window, 200 pairs of images were acquired (per camera) without phase lock, resulting in time-averaged flow field measurements. Additional details concerning the PIV instrumentation are given in [149]

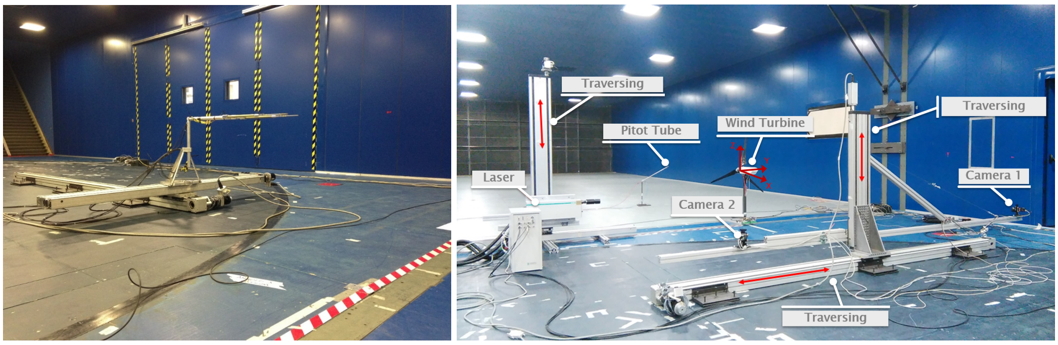

Figure 3.4: Left: hot wire probe and its automated traversing system; right: PIV automated traversing system

3.2 Precursor simulation

The computational setup is reported in the next sections, where a precursor simulation — mimicking the process that takes place in the wind tunnel— is used for the passive generation of the turbulent flow, whose resulting outflow is used as inlet for subsequent wind turbine wake simulations (called successor simulations).

Computational mesh

A first simulation is used to generate the turbulent inflow (precursor) used as an inlet for successive wind turbine/wake (successor) simulations. The layout of the precursor domain is shown in Fig. 3.5, on the left. The precursor domain has the size of 30 m × 6.92 m × 3.84 m. The reduced width of the domain with respect to the actual tunnel size is made to reduce the computational cost. The turbulent inflow map for the successor simulation is sampled at 19.2 m downstream of the domain inlet, as highlighted by the red dash line in the left plot. The simulation mimics the passive turbulence generating system adopted in the wind tunnel where experiments were conducted [52]. A structured body-conforming mesh (purely hexahedral O-grid, see the right plot in Fig. 3.5) is used to discretize the volume around the turbulence-generating spires at the wind tunnel inlet. The average stretching ratio for the O-grid volume mesh is 1.25. The mesh quality is limited by the sharp edges and abrupt surface changes of the spire geometry. The maximum skewness is equal to 2.7, which however does not compromise the simulation convergence. The mesh resolution is designed not to fully resolve the boundary layer, and the average y^+ is equal to 50. Such setup is based on two considerations. First, a wall-modeled simulation significantly reduces computational costs compared with a fully resolved simulation. In fact, the precursor mesh used in this work contains 59 million cells, while one or two orders of magnitude larger grids would be necessary to achieve $y^+ = 1$. Therefore, it is clear that such an approach would lead to extremely high computational costs. Second, as discussed later, results show that the wall-modeled boundary layer approach is capable of providing good agreement with the experimental wind tunnel measurements.

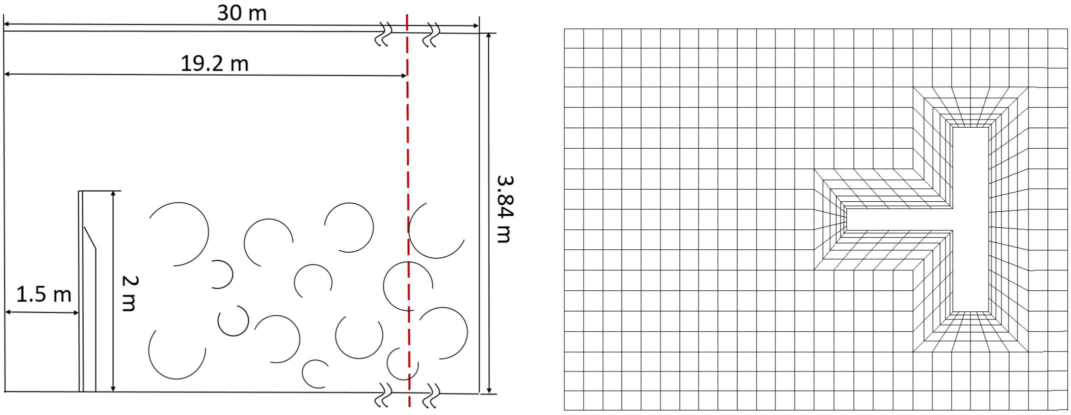

Figure 3.5: Layout of the precursor (left) and successor (right) computational domains.

Boundary conditions

Dirichlet-type non-slip conditions are used for the resolved velocity vector $\tilde{\mathbf{u}}$ on the tunnel side walls and the spire surfaces. Neumann-type conditions are imposed for pressure p on the same boundary surfaces, while Dirichlet-type wall conditions are employed for temperature T, which is assumed to be the same on all surfaces. Regarding the sub-grid scale model, Dirichlet-type surface conditions are used for the eddy viscosity μ_t on the ceiling, using a fixed value equal to 1×10^-5 m^2/s on account of the negligible turbulence at cell centers on a wall; a small positive non-zero value is used, because μ_t is evaluated at cell centroids and not on the wall surface. A wall model is imposed on the other surfaces including spires, left/right walls and floor to adjust wall shear stresses.

The inflow speed at the inlet equals 4.7 m/s, as measured in the wind tunnel, and the maximum Courant number is limited to one. The constant Smagorinsky sug-grid scale model is used, with C_s set to 0.13. In order to reach steady-state conditions, the simulation requires about 15 s of physical time. After achieving a steady mean flow, the outflow of the precursor simulation is collected and stored, to be used as input for subsequent successor simulations.

Comparison of the precursor simulation

Fig. 3.6 shows the normalized time-averaged streamwise velocity $\langle u_x \rangle$ and turbulence intensity $\sigma / \langle u_x \rangle$ profiles measured 20.85 m downstream of the tunnel inlet, which corresponds to 1.5D upstream of the wind turbine rotor. A reference frame is located at the hub, as shown in Fig. 3.8 on the right. The two horizontal and vertical velocity profiles are in good agreement with the experimental data. The average velocity error $\langle \Delta u_x \rangle$ is around 1-2% for both profiles. Turbulence intensity also shows a reasonable agreement, with an average error of 7% and 5% for the horizontal and vertical profiles, respectively. The experimental results for $\sigma / \langle u_x \rangle$ along the horizontal profile show an unexpected discontinuity, not observed in the simulations, which might be due to the effect of the traversing system used for holding and positioning the hot wire probes.

Fig. 3.7 shows the experimental and simulated turbulent kinetic energy spectrum $E(f)$ and autocorrelation $r(\tau)$ at hub height, 1.5D upstream of the rotor. The spectrum plots are reconstructed based on the time history of a sampled point. The LES-computed spectrum appears to be in good agreement with the experimental one. The autocorrelation is computed as:

$$r^j(\tau) = \left\langle (u_x^j(t) - \left\langle u_x^j \right\rangle)(u_x^j(t+\tau) - \left\langle u_x^j \right\rangle) \right\rangle, \tag{3.1}$$

where u_x^j is the streamwise component of the velocity at spatial point j. The integral time scale [150], defined as

$$T_\tau^j = \int_0^\infty \frac{r^j(\tau)}{\left\langle u_x^{j,2} \right\rangle} \mathrm{d}\tau, \tag{3.2}$$

is found to be 0.139 s and 0.143 s for experiment and simulation, respectively. These results indicate a good overall agreement between simulation and experiment even at small scales down to the grid limit, with a consequent correct estimation of flow mixture, wake recovery and other relevant features of the flow.

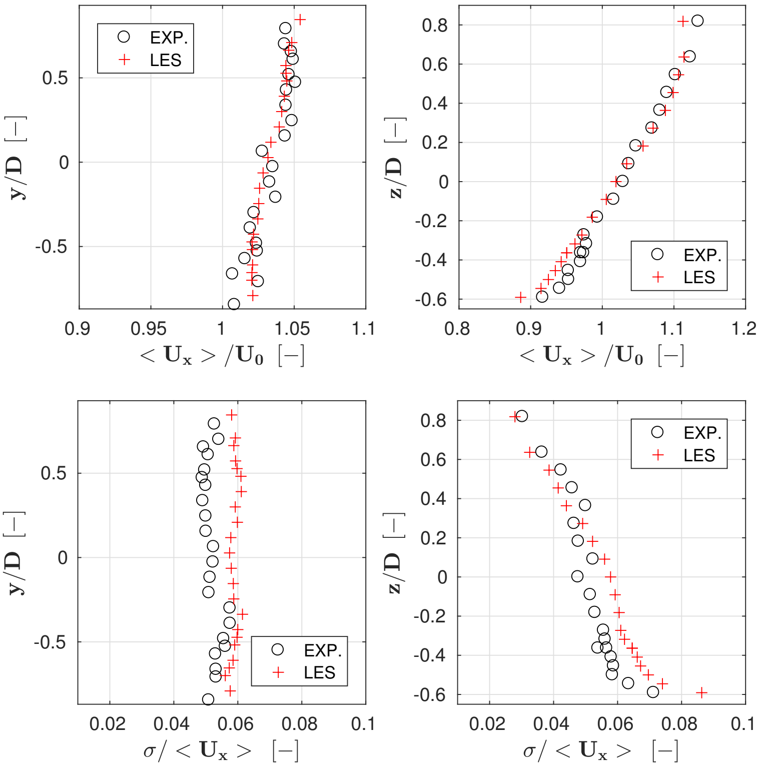

Figure 3.6: Normalized time-averaged streamwise velocity $\langle u_x \rangle$ (top row) and turbulence intensity $\sigma/\langle u_x \rangle$ (bottom row), 1.5D downstream of the rotor. Left column: hub-height horizontal profile; right column: hub-centered vertical profile. Numerical results: red + symbols; experimental measurements: black ∘ symbols.

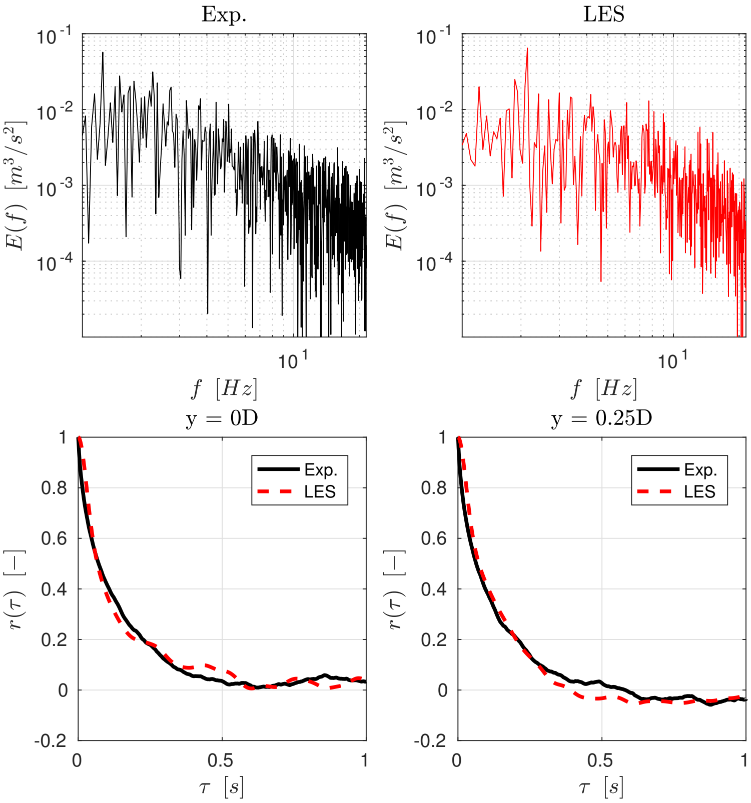

Figure 3.7: Turbulent kinetic energy spectrum $E(f)$ at hub height 1.5D upstream of the rotor, for the experiment (top-left, black line) and simulation (top-right, red line). Autocorrelation $r(\tau)$ at hub height 1.5D upstream of the rotor (bottom-left) and 0.25D to its left, looking downstream (bottom-right).

3.3 Successor simulation

3.3.1 Bottom fixed wind turbine

Numerical scheme

The numerical schemes used for transient term, gradient term, divergence term and Laplacian (diffusion) term writes

```
ddtSchemes
{
    default                 backward;
}
gradSchemes
{
    default                 Gauss linear;
}
divSchemes
{
    default                 Gauss linear;
    div(phi,U)              Gauss localBlended GammaVDC 0.9 GammaVDC 0.1;
    div(phi,k)              Gauss vanLeer01;  //used for SAS and SST model
    div(phi,omega)          Gauss vanLeer01;  //used for SAS and SST model
    div(phi,flm)            Gauss vanLeer01;  //used for LDS
    div(phi,fmm)            Gauss vanLeer01;  //used for LDS

}
laplacianSchemes
{
    default                 Gauss linear uncorrected;
}
```

As explained in section 2.2.3, the deferred-correction Gamma-bounded high interpolation method is used here in conjunction with the IB formulation [87]. The Gamma scheme is parameterized in terms of β_m, a tunable constant that allows one to control the level of upwinding. In general, a larger value of β_m implies a lower dispersion and a higher diffusion (i.e. more upwinding), and vice versa. The value β_m=0.45 is employed in the near wake region to stabilize the simulation, since actuator line body forces and immersed boundary possibly generate numerical dispersion, and β_m=0.05 is used in the far wake to minimize numerical diffusion while retaining a minimum amount of necessary upwinding. A blended algorithm is used to determine the near wake and far wake region. The simulation results indicate that this approach is capable of dealing with the effects caused by the numerical dispersion and the numerical instability.

Regarding sub-grid scale quantities, central differencing is initially used for the convective terms of the k and ω equations for both the SAS and LES model. However, due to the von Kármán length scale in the ω equation of the SAS model, oscillations may be generated that, by affecting the eddy viscosity, can eventually cause the simulation to diverge. Accordingly, a strictly bounded Van Leer differencing scheme is used for the k andω transport equation to minimize numerical dispersion. Although such scheme may cause severe numerical diffusion, there is greater emphasis on the boundedness than accuracy for a scalar field scheme [82].

Accordingly, a relatively diffusive scheme is acceptable for k and ω.

Linear algebraic solver

The linear algebraic solvers and grid quality also play important roles in determining the computational efficiency of the LES framework [82]. Simulations are conducted using high-quality grids, with about 99% of cubic cells for the wind turbine successor cases, and small time-steps (Courant number = 0.3). Moreover, significant flow separation or adverse pressure gradients are typically absent in the present application. Because of this, the use of advanced pre-conditioners and low-quality-mesh-correctors [82] is not essential, with beneficial effects in terms of computational efficiency.

Table 3.3 shows the linear solvers used for the precursor and the wind turbine/wake simulations. The precursor problem has slightly less regular grids, because of the need to mesh the large turbulence generators (termed spires) placed at the tunnel inlet, which requires a slightly different setup of the linear solvers. The conjugate gradient (CG) method is employed to solve the equations with symmetric matrices, associated with pressure p, while a bi-CG is used for the asymmetric matrices associated with $\tilde{\mathbf{u}}$ and T. The geometric-algebraic multi-grid method is used as pre-conditioner for pressure, while the diagonal incomplete-LU factorization is used for asymmetric matrices. Gauss-Seidel is used as a smoother for the pre-conditioning of p for the turbine simulation, while the diagonal incomplete Cholesky factorization Gauss-Seidel is used for the precursor case on account of its lower grid quality, which increases the computational cost by about 5%. The SAS model and SST model implement the same linear solvers used in the LES case for the resolved scales. The k and ω equations for both turbulence models are solved by a conjugate gradient algorithm with diagonal incomplete-LU preconditioning.

Type	Precursor simulation	Wind turbine simulation
p solver	CG	CG
p preconditioner	GAMG	GAMG
p smoother	DIC-GS	GS
No. of p corrector steps	3	3
$\tilde{\mathbf{u}}$ solver	bi-CG	bi-CG
$\tilde{\mathbf{u}}$ preconditioner	DILU	DILU
No. of NOC steps	1	0

Table 3.3: Linear algebraic solvers used for the precursor and the wind turbine/wake simulations (CG = Conjugate gradient; GAMG = geometric-algebraic multi-grid; DIC = diagonal incomplete Cholesky; GS = Gauss-Seidel; DILU = diagonal incomplete LU factorization; NOC = non-orthogonal corrector).

The PISO time marching algorithm allows for recursively solving (or correcting) the pressure flux equation to account for non-orthogonal grid elements [82]. The number of iterations is fixed a priori and set equal to 1 and 0 for the precursor and successor simulations, respectively. Indeed, given the good quality of the grid in the latter case, non-orthogonal corrections are not indispensable, and their elimination lowers the computational cost by about 10%.

Computational mesh

The computational setup for the wind turbine/wake simulation follows [151]. The domain layout is shown in Fig. 3.8. The domain width is reduced to 3.6 D, which is 3.5 times shorter than the test section width to reduce the computational cost while avoiding blockage effects. A coordinate reference frame is centered at the hub of the front turbine, as shown in the same figure. The mesh uses three zones of increasing density. zone 1 is the base mesh, with cubic cells of 0.08 m in size, while the cells in zone 2 have a size of 0.04 m. Zone 3 is the finest grid, used in close proximity of the wind turbines and their wakes. The cell size in this region of the domain differs depending on the turbulence model: for LES the cells have a size of 0.01 m, while for SST and SAS the size is 0.02 m. As a result, the SST and SAS models are run on a grid that has about 7.8 times fewer cells than in the LES case, which has a total cell number of about 39 millions. In all cases), less than 1% of the total mesh is composed of polyhedral cells, while all others are cubic.

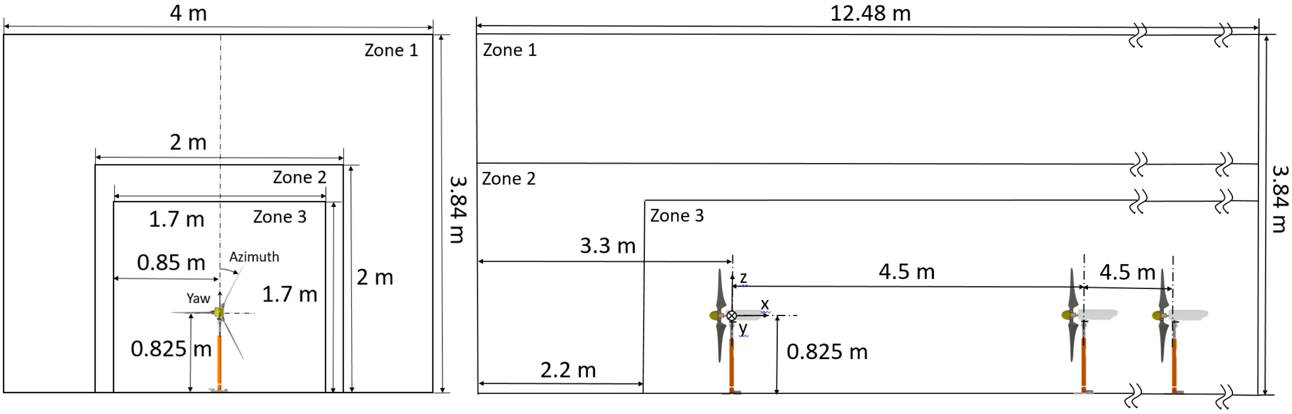

Figure 3.8: Experimental layout and computational domain.

As shown in chapter 5, results indicate that the velocity and turbulence intensity fields are very similar between LES and SAS in moderate and high turbulence conditions. In this sense, SAS is capable of achieving a performance similar to LES, but with much coarser grids. As a validation of mesh convergence, the SAS model was also run on the finer LES grid, obtaining essentially identical results to the ones found on its coarser grid.

On the other hand, finer grids would be necessary for LES for a more accurate solution of the low turbulence case. As discussed later in this thesis, low turbulent conditions have much more stringent requirements in the resolution of the near wake vortical structures and their breakdown. Such a high accuracy is not needed at higher turbulence, as this becomes the dominating factor that dictates vortex breakdown. Finer grids were however not used here, because of the dramatic increase in computational cost caused by the current structured grid approach.

Boundary condition

The same boundary conditions for the flow velocity $\tilde{\mathbf{u}}$, pressure p and temperature T are used for the SST, SAS and LES methods. Three different inflow conditions are considered in this thesis. The low turbulence (< 2%) inflow map is obtained from a LiDAR-scanned [60]. Measurements also accounts for a slight non-uniformity of the flow within the wind tunnel [151]. In the second and third case, as previously explained, the output of the passively-generated turbulent precursor simulation was instead used as inlet for the successor simulation.

The treatment of the domain walls is as follows. Dirichlet-type non-slip wall conditions for $\tilde{\mathbf{u}}$ are used for the tunnel ceiling and floor. Neumann-type conditions for p and T and mixed type conditions for $\tilde{\mathbf{u}}$ are used for the side walls, enforcing a null component of the velocity normal to the side surfaces to ensure mass conservation. As for the precursor case, even here the domain size is reduced to limit the computational cost.

Apart from the resolved flow quantities, boundary conditions are also set for sub-grid scale quantities. The constant Smagorinsky LES model uses Neumann and Dirichlet-type surface conditions for eddy viscosity μ_t at left/right and ceiling/floor faces, respectively. For ceiling and floor, μ_t is set with Dirichlet conditions to the fixed value 1×10^-5 m^2/s in the low turbulence case, while a wall model is used for the moderate and high turbulence condition. The eddy viscosity μ_t, on the other hand, is the ratio of the two additional variables k and ω for the SST and SAS models. Dirichlet-type wall conditions are imposed on both the ceiling and floor surfaces for k and ω, using the values 1×10^{-4} and 2×10^{-2}, respectively, based on [152] and simulation stability tests. Such setup is used for all three turbulence conditions, instead of only for low turbulence condition, because applicability of a wall model to the SAS is still under investigation. Results are also largely insensitive to the boundary values for k and ω, which therefore do not require a precise calibration.

For low turbulence inflow conditions, Dirichlet-type non-slip wall conditions are used for the IB-modeled nacelle and tower. In fact, in these cases a laminar boundary layer (or, at least, a not fully-developed turbulent boundary layer) is expected to extend over the entire IB surface due to the steadiness of the incoming flow. Despite the maximum y^+ being equal to 50 on the IB surfaces, a wall function can no be used here, as it could properly model only a fully developed turbulent boundary layer. Due to the coarse grid, an overestimation of the boundary layer thickness on the IB-modeled bodies is expected, which in turn will lead to an overestimation of the blockage induced by the turbine nacelle and tower.

Slip wall IB surface conditions are used for the moderate turbulence case, in order to mitigate numerical stability issues. Although this neglects the boundary layer induced blockage and turbulence, results indicate a negligible impact on the downstream wake profile. This is probably explained by the background turbulence that, by enhancing mixing, diffuses the signature of tower and nacelle on the downstream flow.

Sub-grid scale filtering

The LES filter size is defined equal to the cell size $\Omega_{CV}^{1/3}$ in OpenFOAM, which, in other words, means that the sub-filter scale (SFS) equals to sub-grid scale (SGS). In OpenFOAM, the grid itself is regarded as the spatial filter if a constant Smagorinsky (CS) model is used, while a top-hat filter is used for a dynamic Smagorinsky model such as the Lagrangian dynamic Smagorinsky (LDS) model.

The grid filter, which uses each grid cell as a spatial filter, is used for both LES and SAS. Moreover, similar to LES, SAS is not associated with a temporal averaging operation. In this regard, instead of a URANS formulation, SAS adopted in this paper can be considered as a LES dynamic sub-grid scale model, which exhibits a LES-like behavior at the resolved scales. The same concept is described in [69]. Moreover, there is no prediction-correction iteration between the momentum and the two transport equations, as explained in 2.2.8. Therefore, the sub-grid scale SAS model not only introduces LES-like characteristics to the formulation but also keeps the PISO algorithm unchanged.

3.3.2 Floating offshore wind turbine

The top plots in Fig. 3.9 show cross-sections of a computational domain. The simulation strictly adheres to the experiment, including the domain size and position of the floating platform and the platform pitching motion. The platform is located 29m downstream of the wind tunnel inlet. It is also fixed in the middle of the wind tunnel, so only half of the domain is included in the simulation model to account for the bilateral symmetry of the setup.

The geometry of the floating platform is significantly simplified. Indeed, minor geometrical features that do not substantially influence the flow evolution are removed.

Three levels of refinement are used. Zone 1 defines a cube-shaped background mesh ($\Delta x_f = \Delta y_f = \Delta z_f$) with a cell size of 0.08 m; zone 2 is the first level of refinement with a cell size of 0.04 m, followed by zone 3, that uses a size of 0.01 m [153]. All the refinement regions extend stream-wise to the end of the domain. The dashed boxes in top plot and boxes in bottom plot in Fig. 3.9 show the customized Scotch decomposition domains. Due to current limitations in the parallelization algorithm, decomposed domains are adjusted to ensure that immersed boundary surfaces do not span multiple processors.

Since the digital model tries to replicate the experiment, the simulation inflow speed, turbulence intensity and shear should be as close as possible to the experimental ones. In order to generate a proper inflow, a separate LES simulation is conducted. The domain models the tunnel inlet, the spires and part of the wind tunnel chamber, as depicted in section 3.2. A developed sheared and turbulent flow outflows from this sub-domain, which is then used as inflow for a successor simulation, where the rest of the tunnel chamber and the floating platform are modeled.

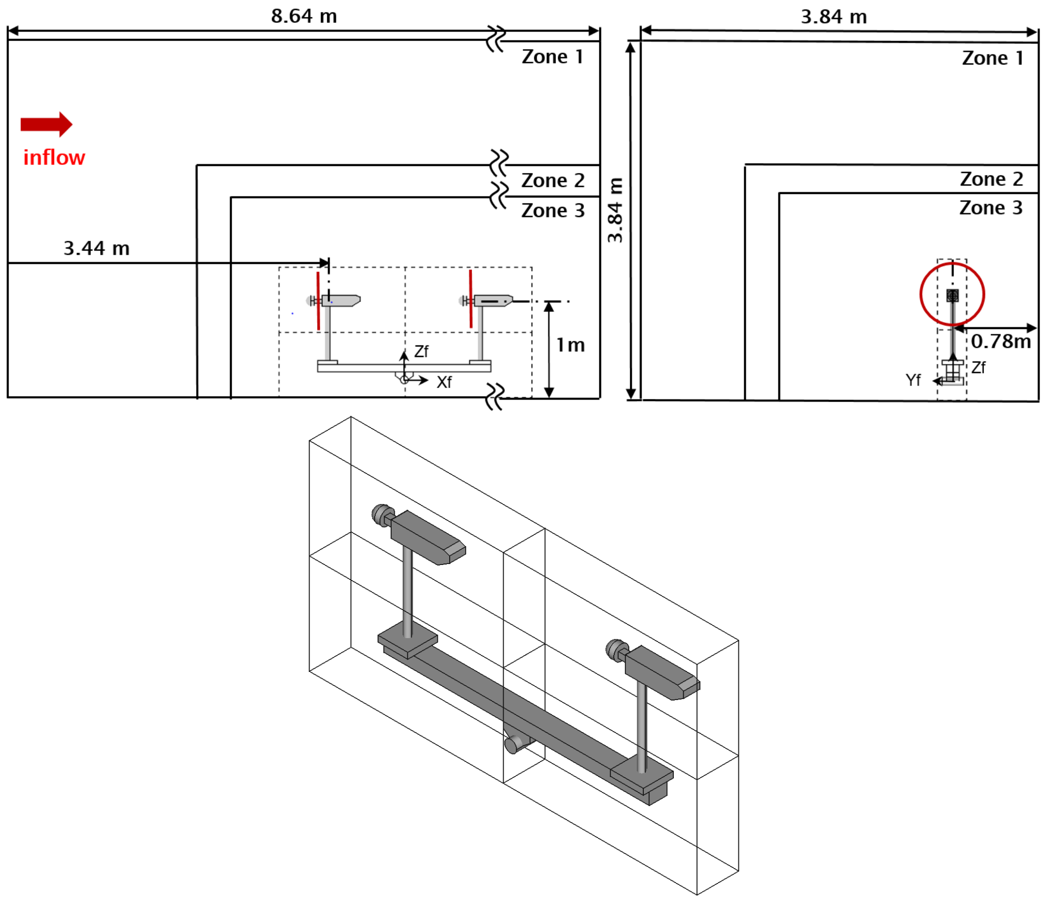

Figure 3.9: CFD computational domain. Left: lateral view; right: streamwise view.

CHAPTER 4

Large-eddy Simulation Framework

In this chapter, the author compares the large-eddy simulation results with experimental measurements performed in a boundary layer wind tunnel with bottom fixed scaled wind turbine models. The study contains normal operating conditions, as well as wake manipulation by derating, yaw misalignment and cyclic pitching of the blades. The framework is compared with the experiments at two turbulence conditions, low turbulence and spire generated moderate turbulence.

Apart from the comparison of the framework with the experimental results, this chapter applies this framework to the research of wake redirection methods. Two control strategies are investigated: yaw misalignment and cyclic pitch control. First, analytical equations are derived for both methods, with the goal of providing a simple physical interpretation of the behavior of the two methods. Next, more realistic results are obtained by simulations and by experiments. Comparing the analytical, numerical and experimental models allows for a cross-validation of the results and a better understanding of the two wake redirection methods.

4.1 Baseline simulation and parameter tuning

The baseline simulation represents an isolated flow-aligned wind turbine. The machine is operated in a low turbulence flow, with a rotor-averaged inflow velocity equal to 5.9 m/s, which is slightly lower than the G1 rated speed (6.0 m/s).

This first case is used to determine the optimal values of the Smagorinsky constant C_s and of the Gamma scheme parameter β_m. The same tuned parameters are used for all other simulations in the rest of this work. This first test case is also used to analyze the effects of the Gaussian width ϵ, which is used to project aerodynamic forces from the lifting lines onto the computational grid. In fact, it was observed that this projection may have a significant effect on the results, including rotor power and thrust. In principle, ϵ should be set equal to 2-3 times the cell size, i.e. $2h \leq \epsilon \leq 3h$ [26]. It was found that the dependency of the rotor aerodynamic power on ϵ is significantly reduced if the integral velocity sampling approach is used [67]. For instance, if ϵ increases by 30%, power will increase by 13% if the traditional point-wise velocity sampling approach is used, but only by 5% when using the integral velocity sampling method. In fact, in the point-wise approach a variation of ϵ reshapes the Gaussian curve, in turn changing the peak value and eventually influencing the calculated aerodynamic power, while the integral approach uses a weighted average that mitigates the reshaping effect [67].

Using a simple trial and error approach, the three parameters ϵ, C_s and β_m (in the near wake) were set to 0.025, 0.13 and 0.45, respectively. Given the low turbulence of the present case, the experimentally measured rotor speed was very nearly constant, and its average value was used in the simulation.

The rotor integral quantities of power and thrust are compared first, by time-averaging over 10 s. The wind turbine power was found to be equal to 45.8 W in the experiment, and equal to 45.5 W for LES, showing a good agreement between these two values. A slightly larger discrepancy was obtained for the thrust, which was found to be 15.3 N and 16.1 N for experiment and simulation, respectively. This may be explained by the fact that thrust is directly measured at the shaft in the numerical simulation, while it is reconstructed from the tower based fore-aft bending moment in the experiment. This requires estimating the contribution of nacelle and tower, which is done by a dedicated experiment performed on the wind turbine without the blades. As a result, this indirect calculation of the experimental thrust is affected by approximations, and it cannot be regarded as accurate as the measurement of rotor torque (and hence of power).

Next, the characteristics of the wake are compared between PIV measurements and CS LES simulation. Fig. 4.1 shows streamwise velocity contours on a plane 0.56D downstream of the rotor. Measurements are missing from two areas left and right of the rotor disk where, due to the close proximity of the measuring plane with the wind turbine, part of the nacelle (which is of a white color) was in the background, leading to a wrong correlation between the PIV images. Apart from the two missing spots, the LES contours are similar to the PIV ones, both in terms of wake width and deficit. The wake deficit for LES is on average 1.3% higher than the experiment.

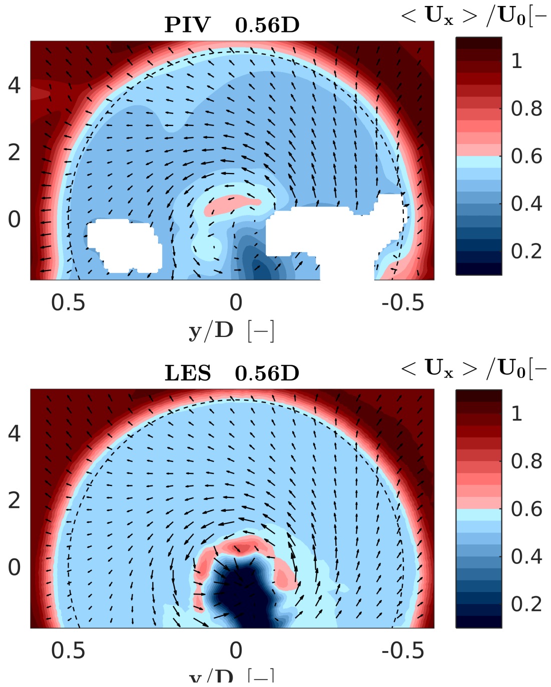

Figure 4.1: Streamwise velocity contours for the constant Smagorinsky model and PIV experimental measurements, on a plane 0.56D downstream of the rotor. Black arrows indicate the cross-wind velocity component at a number of sampling points.

The figure also shows that the simulation overestimates the local wake deficit behind the nacelle and tower, as a results of the enhanced blockage effect mentioned in section 3.3.1. Indeed, the current mesh resolution (high y^+) implies a thicker boundary layer, which in turn produces a higher blockage with a consequent larger flow separation, tower shedding and induced turbulence. This problem could be mitigated by a suitable refinement of the mesh near the IB, which however would come at the price of a significant increase in the computational cost.

Next, hot-wire probe measurements are used to compare wake profiles at 3D, 4D, 7D and 8D downstream positions. Fig. 4.2 shows horizontal (top row) and vertical (central row) profiles of the normalized time-averaged velocity, as well as horizontal profiles of turbulence intensity (bottom row). The plots report results for the CS model, the LDS model, and experimental measurements. The CS case includes two sets of results, one obtained including the effects of nacelle and tower in the model by the IB approach, and one obtained neglecting these two components. Comparing these two curves with the experimental results clearly indicates that the near wake profile is more accurately represented when nacelle and tower are included in the model, as already noted by other authors [76]. This may be particularly true for the present scaled wind turbine, for which these two components are relatively bigger than in full scale machines. Indeed, the sum of the frontal area of the nacelle and of the portion of the tower located within the rotor swept area A is $0.037A$, while it is $0.023A$ for the NREL 5 MW wind turbine [148]. Although this parameter is larger for the G1, it is expected that the effects of nacelle and tower on wake evolution might not be negligible even for typical multi-MW wind turbines [154]. All other simulations reported in this work were performed including nacelle and tower in the model.

Both CS and LDS show a good agreement with the experimental curves. Indeed, the temporally and spatially averaged streamwise velocity difference $\langle \Delta u_x \rangle = (\langle u_{x,\mathrm{LDS}} \rangle - \langle u_{x,\mathrm{CS}} \rangle)/\langle u_{x,\mathrm{CS}} \rangle$ between the CS and LDS models is consistently less than 1% at all downstream distances. Results indicate that the LDS model does not provide significantly more accurate results than the CS model, while at the same time it requires a 20% larger computational effort caused by the solution of its two extra transport equations. Moreover, turbulence intensity plots seem to indicate a slightly better match of CS to the experiments than LDS. Based on these results, all other simulations in the present thesis were based on the CS model.

The rotor-averaged streamwise velocity difference between simulation (with nacelle and tower) and experiment $\langle \Delta u_x \rangle = (\langle u_{x,\mathrm{sim}} \rangle - \langle u_{x,\mathrm{exp}} \rangle)/\langle u_{x,\mathrm{exp}} \rangle$ is equal to -2.7%, -1.6% and -1.3% at 3D, 4D, and 8D, respectively. The root mean square (RMS) error can be used to quantify the spatial fit between simulations and experiments, and it writes

$$\mathrm{RMS}(\cdot) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\left\langle \cdot^{j}_{\mathrm{sim}} \right\rangle - \left\langle \cdot^{j}_{\mathrm{exp}} \right\rangle\right)^2}, \tag{4.1}$$

where $\langle \cdot^j \rangle$ is a generic time-averaged quantity at a given spatial point j. At the various downstream distances, $\mathrm{RMS}(u_x)$ equals 0.34 m/s, 0.33 m/s and 0.15 m/s, respectively. As expected, the matching of simulation with experimental measurements improves moving downstream. Indeed, if rotor thrust is well predicted and numerical diffusion is suitably controlled by the scheme, flow mixture is properly resolved, then the simulation results in a fully developed wake that correlates well with the experiment. The far wake profile can be approximated by the single Gaussian distribution used in some engineering wake models [18, 155].

LES underestimates the rotor-averaged turbulence intensity $\sigma/\langle u_x \rangle$ by 23%, 12% and 12% at 3D, 4D, and 8D, respectively, while the rotor-averaged root mean square error $\mathrm{RMS}\,(\sigma/\langle u_x \rangle)$ is 0.04, 0.02 and 0.02 at these same positions. The turbulence intensity profiles of Fig. 4.2 clearly show that matching is not as good as in the case of the streamwise velocity, especially in the near wake region where tip vortices are not resolved enough and tower shedding is overpredicted. Here again, the problem could mitigated with a finer grid, which however would lead to increased computational costs.

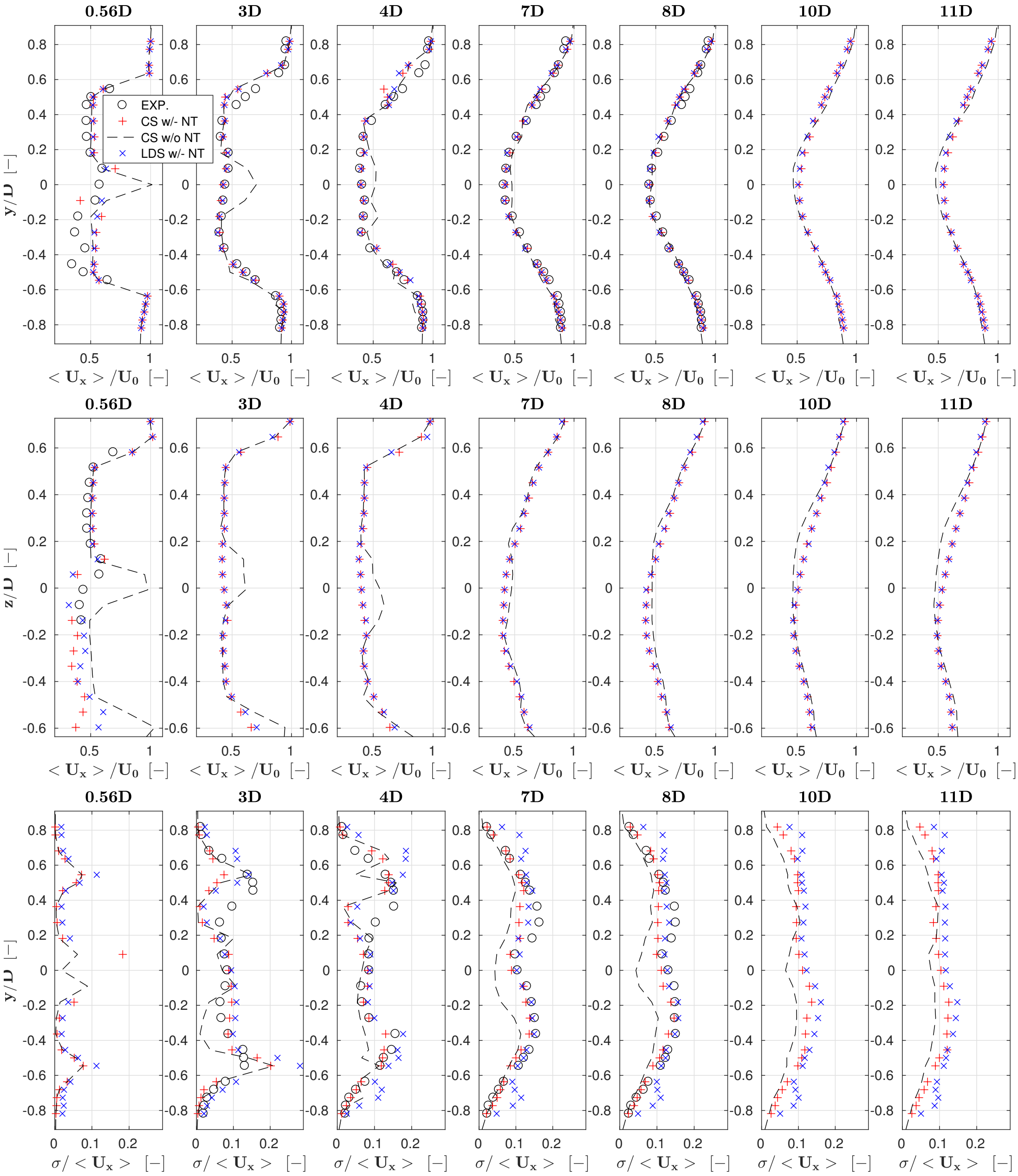

Figure 4.2: Profiles of normalized time-averaged streamwise velocity $\langle u_x \rangle / U_0$ at hub height XY plane (top row), at XZ plane through the tower (middle row), and turbulence intensity $\sigma / \langle u_x \rangle$ (bottom row) at hub height XY plane, at the downstream positions 0.56D, 3D, 4D, 7D, 8D, 10D and 11D (red + symbols: constant Smagorinsky model with nacelle and tower included (marked as CS w/- NT); black −−: constant Smagorinsky model without nacelle and tower included (marked as CS w/o NT); blue × symbols: Lagrangian dynamic Smagorinsky model with nacelle and tower included (marked as LDS w/- NT); black ∘ symbols: experimental results).

Comparing the turbulence intensity results with and without nacelle and tower shows that there is an increased turbulence in the wake of the former case, which causes an earlier vortex breakdown and produces higher turbulence intensity at the far wake. In turn, this generates a higher wake recovery, as shown in the speed deficit plots. Here again, this confirms the need for including nacelle and tower in the simulation.

4.2 Low turbulence inflow simulation

In this section, the characteristics of the LES framework are assessed with reference to three wake control strategies, namely power derating (or axial induction control), wake steering by yaw misalignment and wake enhanced recovery by cyclic pitch control. The flow conditions and setup of the simulations is the same described earlier in the baseline case.

4.2.1 Power derating

Power de-rating was accomplished by providing the turbine power controller with modified values of the rotor speed and torque. Specifically, for a power partialization factor p_f, the reference rotor speed is modified as $\sqrt[3]{p_f}\Omega$, while the torque as $\sqrt[3]{p_f^2}Q$. This corresponds to having set the rated wind speed to the value $\sqrt[3]{p_f}\,U_\infty$; since this is lower than the current wind speed U_∞, the machine is now effectively operating in the full power region. Therefore, the collective blade pitch controller automatically adjusts the pitch setting to track the new reference rotor speed.

The resulting pitch and rotor speed changes modify the angle of attack and Reynolds number at the blade sections. Therefore tests that include power derating are useful for evaluating the quality of the identified multi-airfoil tables. Indeed, to accurately estimate rotor power and thrust, the lifting line airfoil polars need to match the aerodynamic characteristics of the corresponding blade sections, in order to generate and project the proper body forces onto the fluid domain.

Four power settings are considered, namely 100%, 97.5%, 95% and 92.5% of rated power. Fig. 4.3 shows wake velocity profiles measured at hub-height at a 4D downstream position. For all cases, rotor-averaged speed error $\langle\Delta u_x\rangle$ and RMS(u_x) are about 1% and 0.25 m/s, respectively. A quite satisfactory agreement between the simulation and experimental wake profiles can be noticed. Turbulence intensity profiles are not presented here, since the quality of the comparison is very similar to the one of the baseline case. These results seem to indicate that the simulation model has the necessary generality to represent with good accuracy operating conditions that differ from the baseline case, used for its initial tuning.

4.2.2 Wake steering by yaw misalignment

Next, the LES model is compared with the experiments in yaw misalignment conditions, which are relevant to wake deflection control. Hub-height wake profiles measured in low turbulence conditions are used, for yaw misalignment angles of ±20 deg, ±10 deg and ±5 deg.

A comparison of simulated and measured longitudinal speed profiles are presented at a downstream distance of 4D in Fig. 4.4. Similar results were obtained at other distances, but are not reported for space limitations. The maximum rotor-averaged difference $\langle\Delta u_x\rangle$ between simulation and experiment is 4.1% and corresponds to the 20 deg case, while the maximum RMS(u_x) is 0.35 m/s at -10 deg. The average $\langle\Delta u_x\rangle$ and RMS(u_x) over the six yaw cases are

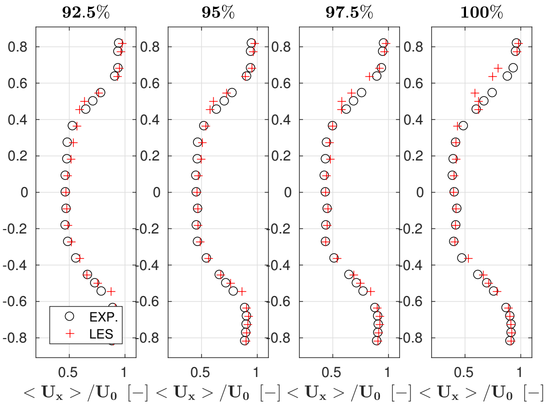

Figure 4.3: Normalized time-averaged streamwise velocity $\langle u_x \rangle / U_0$ profiles at 100%, 97.5%, 95% and 92.5% power settings, measured at hub height and 4D downstream of the rotor (red + symbols: LES; black ∘ symbols: experimental results).

equal to 1.6% and 0.29 m/s, respectively. The results indicate a good agreement between simulation and measurement, both in terms of wake deficit and pattern. Notice, however, that the 1.6% average speed error would correspond to a 4.8% power error for a second wind turbine operating in full wake shading at this downstream difference, a value that is small but not completely negligible.

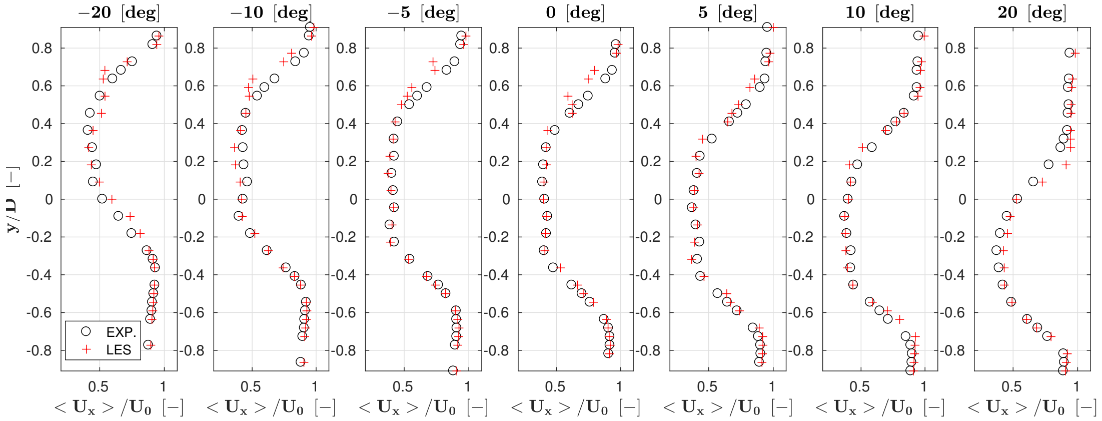

Figure 4.4: Normalized time-averaged streamwise velocity profiles at hub height for different yaw misalignments, 4D downstream of the rotor (red + symbols: LES; black ∘ symbols: experimental results).

4.2.3 Wake-enhanced recovery by cyclic pitch control

A third wake control strategy in the same low turbulence conditions is considered, where the rotor blades are cyclically pitched. The effect of cyclic pitching is that of changing the angle of attack of the blade sections cyclically over one rotor revolution. In turn, this results in an azimuthal change of the out of plane forces generated by the section, which has then the effect of correspondingly modifying the local induced velocity. A simple analytical model of the effects of cyclic pitching was developed in [156]. The analysis showed that, as already noticed by other authors [3], CyPC has some effect on the speed of recovery of the wake, but results only in a very modest deflection of its path. In fact, wake deflection by yawing is driven by the tilting of the rotor thrust, which results in a significant lateral force being applied onto the flow. On the other hand, CyPC modifies the induced velocity, but only generates negligible lateral forces. In addition, it was observed that CyPC also results in large moments being generated in the rotor fixed frame, which further questions the practical applicability of this wake manipulation strategy. Nonetheless, CyPC is considered here to further verify the characteristics of the LES framework in operating conditions that differ significantly from the ones of the previous test cases.

Each blade is pitched according to $\theta_i = \theta_0 + \theta_c \cos(\psi_i + \gamma)$, where θ_0 is the collective pitch constant, θ_c the 1P pitch amplitude, ψ_i the blade azimuth angle (clockwise looking downstream, and null when the blade is pointing vertically up along the tower), and γ is the phase angle (with the same origin and postive sense as ψ). The CyPC parameters were set as θ_0=0 deg, θ_c=5.3 deg, and γ=270 deg.

Given the effects of CyPC on the induced velocity and on the near wake behavior, a more complete analysis can be performed by using the PIV measurements than considering the simple hub-height line scans obtained by hot wire probes.

Fig. 4.5 reports, at left, the streamwise velocity just behind the rotor (x/D=0.56), which is a distance where few results have been previously reported. The images show that the use of CyPC has a strong effect on the wake structure, leading to a marked unsymmetrical shape. Indeed, the phase angle γ=270 deg implies that the blades have maximum pitch, and hence minimum thrust, in the left part of the rotor —as shown in the figure—, which in turn exhibits the lowest induction and highest resulting longitudinal flow speed. A comparison between experimental and numerical results shows that there is, in general, a good qualitative agreement and that the main distortion effects caused by CyPC are reasonably captured. The rotor-average error $\langle \Delta u_x \rangle$ between simulation and measurement is 2.69%, while RMS(u_x) is 0.79 m/s.

The discrepancy between simulation and experiment is two times larger than in the baseline case. One possible reason for this is that unsteady aerodynamic effects of the airfoils (including dynamic stall) are neglected. This could be improved by using unsteady aerodynamics models in the lifting line, including for example a Theodorsen correction and a dynamic stall model. Although the Beddoes-Leishman approach [157] is implemented in FAST and therefore could be readily used in the present LES framework, the model requires the definition of several airfoil-dependent parameters, which would need to be specifically calibrated for the low-Reynolds airfoils used on the G1 scaled wind turbine.

The comparison of LES and experiment in the far wake (6D) is slightly better, as it can be observed in the right part of the same figure. The wake recovery is reasonably good in terms of flow speed, although the slight tilting towards the right shown by the PIV measurements is not apparent in the LES results. Lastly, it should be remarked that CyPC leads to a faster recovery of the wake than in the baseline case, as already noticed by [156]. In principle, this could be of interest for wind farm control, although, as previously mentioned, the large resulting loads exerted on the rotor probably limit the practical applicability of this control concept.

4.3 Moderate turbulence inflow simulation

Next, a turbulent case is considered, where a flow characterized by a 6% hub-height turbulence intensity is generated by the precursor simulation described in 3.2. The wind turbine model is aligned with the streamwise flow direction and the hub-height wind speed is equal to 4.76 m/s,

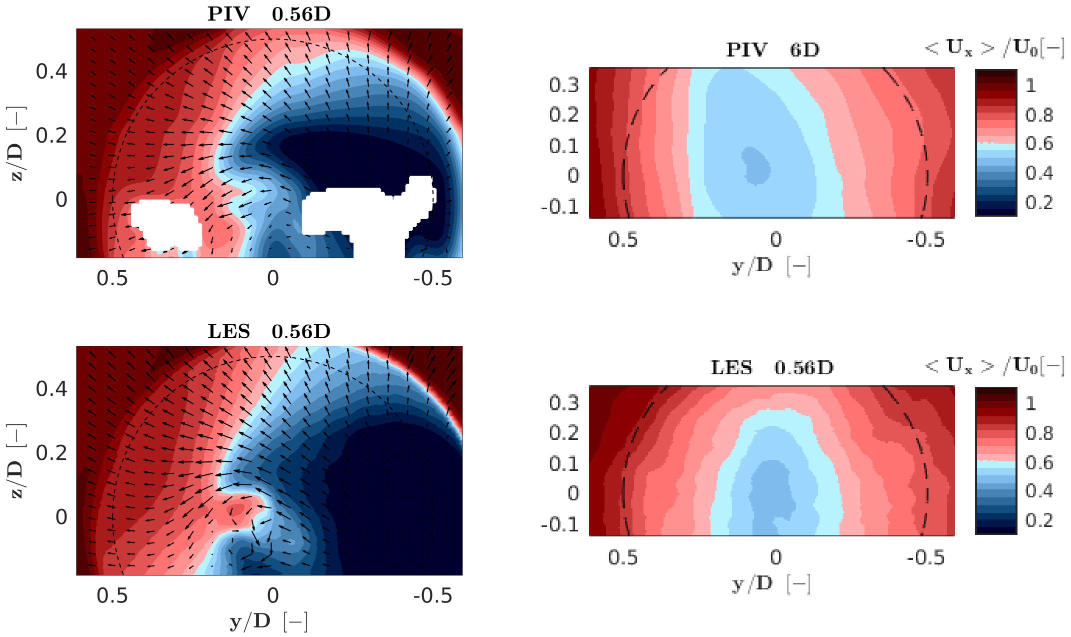

Figure 4.5: Streamwise velocity contour plots for the PIV measurements (top row) and LES model (bottom row), measured 0.56D (left) and 6D (right) downstream of the rotor. Black arrows indicate the cross-wind velocity component at a number of sampling points.

just below the G1 model rated speed. The simulated wind turbine operates in two different modes, namely with a fixed rotating speed of 720 RPM and blade pitch angle of 1.4 deg (which are the values measured on the scaled model in the experiment) or with a controller in the loop [57]. Since the machine operates in the partial load region, the blade pitch setting is constant and torque control is based on a pre-computed look-up table.

The aerodynamic power output, averaged over a 60 s time window, is equal to 31.9 W for the experiment, and to 30.5 W and 31.2 W for the prescribed speed and closed-loop torque simulations, respectively. In this latter case, the average rotor speed was only 2.2% higher than the one measured on the wind turbine, which clearly indicates a good overall match of the numerical model with the experiment. On the other hand, the power standard deviation was 0.2 W, 0.6 W and 0.3 W, respectively for the experiment, prescribed speed and closed-loop simulations. Clearly, prescribing a constant speed to the rotor in the numerical simulation induces significant torque oscillations, because the rotor cannot adjust to the turbulent flow fluctuations. When loads are of interest, it is therefore essential to use a closed-loop controller also in the simulation. However, in this case the simulation might drift away from the operating condition realized in the experiment, if the numerical model has a significant mismatch with respect to reality. Apparently, this is not the case here, and the numerical model seems to be well in line with the experimental one.

Fig. 4.6 shows the normalized time-averaged velocity and turbulence intensity profiles for the LES model and experiment, at distances of -1.5D, 1.4D, 1.7D, 2D, 3D, 4D, 6D and 9D from the rotor. The position at -1.5D is outside of the induction zone, and the flow can be regarded as being the undisturbed free stream. The LES curves show, in general, a good agreement with the experimental ones. Only the case of the closed-loop regulation is reported here, as results are nearly identical to the prescribed-speed case. The rotor-averaged simulation error $\langle \Delta u_x \rangle$ is less than 1% on average across all distances. From the near (1.4D) to the far (9D) wake regions, the root mean square error RMS(u_x) gradually reduces from 0.2 m/s to 0.1 m/s. The comparisons all indicate that the LES results are in good agreement with the experimental ones.

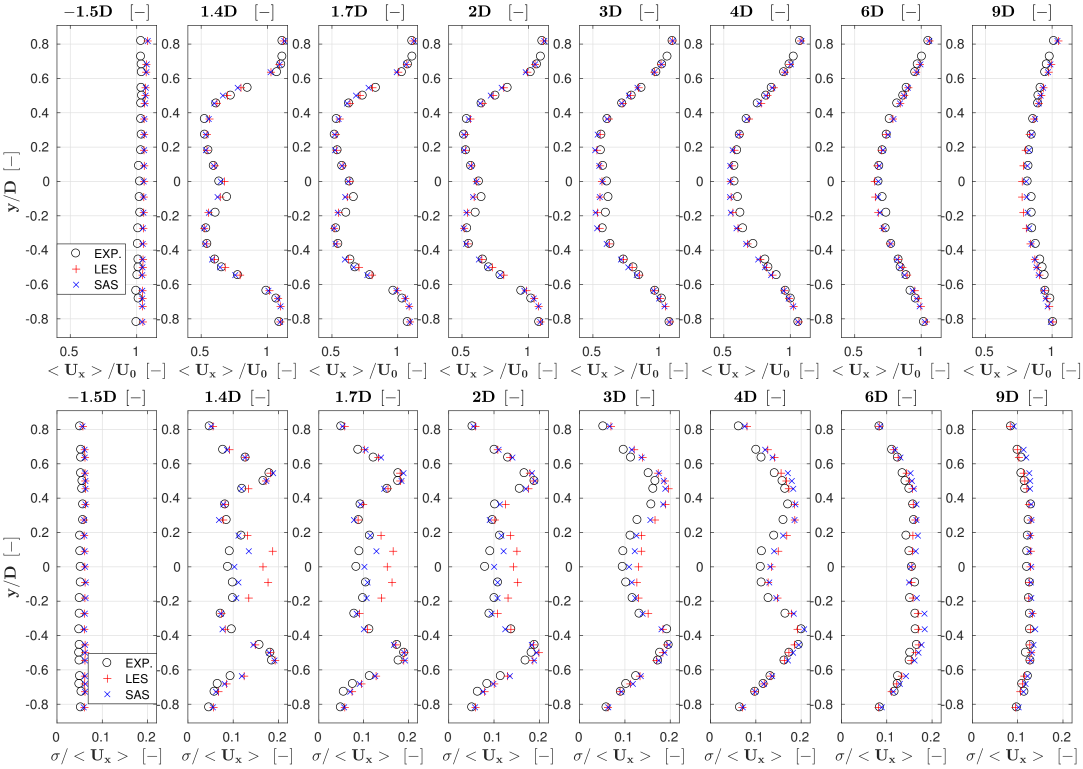

Figure 4.6: Normalized time-averaged streamwise velocity $\langle u_x \rangle / U_0$ (top row) and turbulence intensity $\sigma / \langle u_x \rangle$ (bottom row) profiles at hub height (red + symbols: LES; black ∘ symbols: experimental results).

Contrary to the baseline low-turbulence simulation, the two turbulence intensity peaks induced by the blade tip vortices are well predicted in this case. To explain this phenomenon, the author reports in Fig. 4.7 for the moderate and low turbulence cases the instantaneous streamwise speed component u_x / U_0, the vorticity $\langle \nabla \times \mathbf{u} \rangle$ and the turbulence intensity $\sigma / \langle u_x \rangle$, on a vertical plane through the hub center. As previously observed, the turbulent structures induced by nacelle and tower are different for the two cases, on account of the different slip wall boundary conditions. The turbulence intensity contours are also different close to the blade tip region. On the other hand, the vorticity of the tip vortices in the near wake is quite similar for the two cases. In fact, the actuator line modeled blade tip produces a similar amount of velocity shear for both the low and moderate turbulence conditions. However, the higher level background turbulence of the latter case induces local flow instabilities that rapidly break down the vortical structures into turbulence. The contour plots for the turbulent simulation clearly shows that, starting from 0.1D downstream, the tip vortices generate significant turbulence intensity, while vorticity quickly diminishes from 2D downstream. As expected, the low turbulence case shows a persistent low turbulence and high vorticity up to about 4D downstream of the rotor. Capturing this phenomenon probably requires a denser grid than the one used here so as to introduces proper local flow instabilities and results in accurate vortex breakdown at low turbulence environment, while the vortex breakdown seems capable of representing well in the turbulent case. In fact, [158] has already confirmed that the blade conformed mesh (significantly dense mesh) simulation is capable of triggering the local flow instabilities at low turbulence condition, and results in faster tip vortices breakdown.

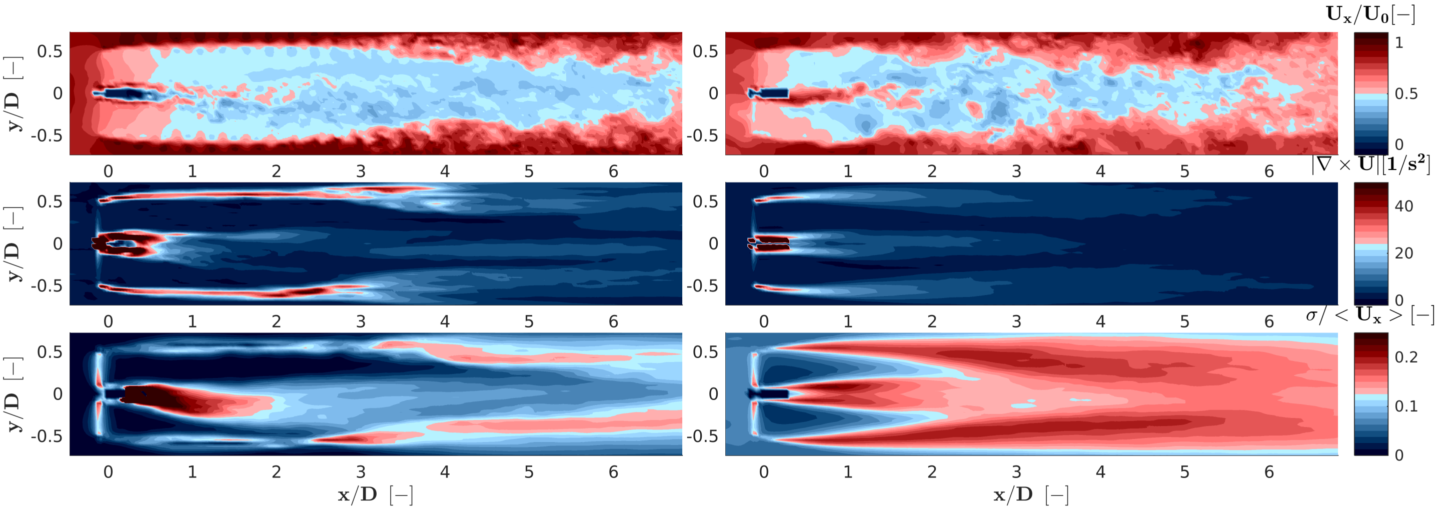

Figure 4.7: Instantaneous streamwise speed component u_x/U_0 (top row), vorticity $\langle \nabla \times \mathbf{u} \rangle$ (central row), and turbulence intensity $\sigma/\langle u_x \rangle$ (bottom row). At left, moderate turbulence case; at right: low turbulence case.

In summary, results indicate a good overall matching of simulations with experiments. Low turbulence test cases appear to be more challenging than moderate turbulence ones, due to the need of having a denser grid to trigger tip vortices breakdown.

4.4 Wake redirection methods

Following the comparison, the framework is used for the research of wake redirection methods. Two control strategies are investigated: yaw misalignment and cyclic pitch control. First, analytical formulas are derived for both methods, with the goal of providing a simple physical interpretation of the behavior of the two methods. Next, more realistic results are obtained by numerical simulations and by experiments conducted with scaled wind turbine models operating in a boundary layer wind tunnel. Comparing the analytical, numerical and experimental models allows for a cross-validation of the results and a better understanding of the two wake redirection methods.

4.4.1 Analytical Model

Fig. 4.8 shows the problem geometry at a hub height horizontal plane. φ and α_l are the yaw misalignment angle and wake skew angle respectively. U_0 is the free stream velocity, parallel to X at s ground-fixed frame of reference XYZ [1]. Force resultants on the rotor are the thrust F_T and the in-plane lateral force F_H, aligned with a nacelle-attached frame xyz. The resultant total aerodynamic force of YM has a non-zero component, $F_T \sin\varphi - F_H \cos\varphi$, along with a

[1] Follows the convention of the right-hand rule in this thesis. The arrows of axes and angles point to the positive directions.

wind-orthogonal ground-fixed unit axis Y, while cyclic pitch control produces only $F_H \cos\varphi$ along with the same axis. By the principle of action and reaction, an equal and opposite force will be exerted by the rotor onto the air flow, resulting in a lateral displacement noted l at a distance X downstream of the wind turbine.

A simple analytical model has been widely used for estimating the lateral wake displacement for YM [17]. It employs the simplified momentum transport equations to connect $F_T \sin\varphi$ and the wake skew angle α_l, but with no F_H considered in the formulation. In this research, the $F_H \cos\varphi$ term is introduced into the Jiménez's model to improve the estimation of wake steering and extend this model to the CyPC application.

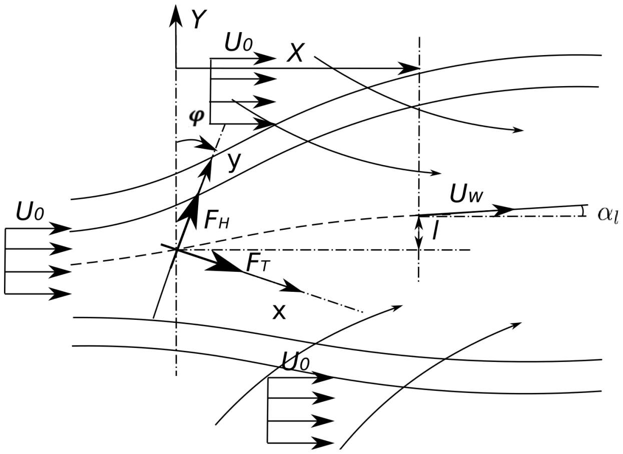

Figure 4.8: Rotor disk velocities & Coordinate system & Wake deflection and downstream position

A schematic view of a wind turbine is shown in Fig. 4.9. A nacelle-attached frame is noted as xyz, rotated about the vertical axis z by the yaw misalignment angle φ. The rotor is spinning at an angular velocity Ω, while θ_i and ψ_i are, respectively, the pitch and azimuthal angles of the ith of B blades. The tangential force acting locally at a generic blade section is noted as F_t, while F_r is the radial component at that same location. The in-plane vertical forces are denoted as F_G in Fig. 4.9.

The velocity triangle at a generic blade section is shown on the Fig. 4.10, where U_T is the tangential velocity component, U_P the perpendicular one, χ is the inflow angle and α the angle of attack (α). To derive a linearized aerodynamic model for the two wake redirection methods, several assumptions are determined first. The rotor speed is significantly higher than the inflow speed, $U_T \gg U_P$. Small-angle is used as appropriate for α. The blade chord and twist are assumed to be constant along the blade span [121]. As well, the flapwise and edgewise motion of each blades are not allowed, so no aeroelastic motion is introduced in the formulation. Based on aforementioned assumptions, the velocity components are readily found to be

$$U_P = U_0(1-a), \tag{4.2a}$$

$$U_T = \Omega R(1+a') - U_0\varphi\cos(\psi_i), \tag{4.2b}$$

where a and a' are the axial and swirl inductions respectively (assumed to be spanwisely constant), and R the rotor radius. Both induction factors are assumed to be cyclically varying

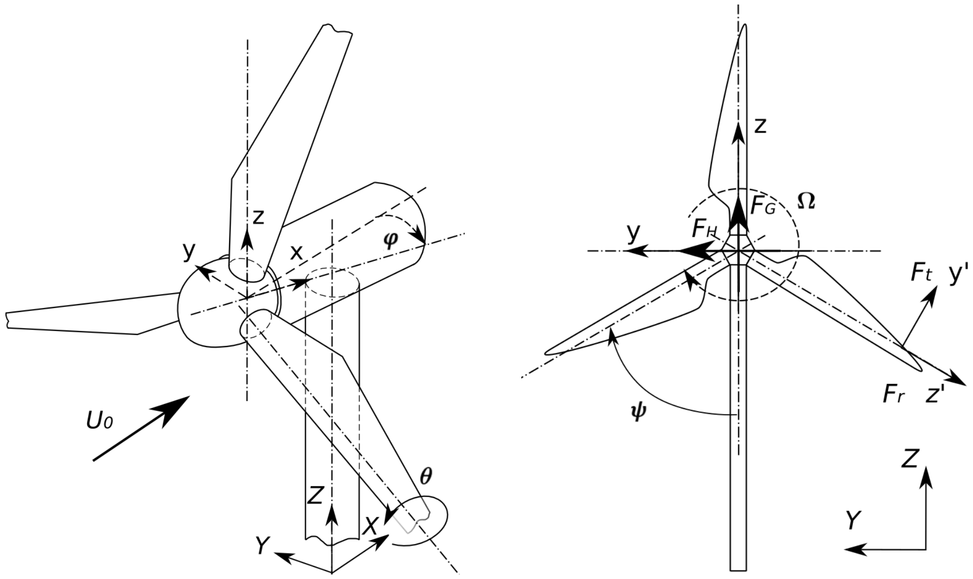

Figure 4.9: Left: yaw and C_yPC configurations & Coordinate systems, right: wind turbine configuration & rotor loading & coordinate systems & Sign conventions

as

$$a = a_0 + a_s \sin(\psi_i + \gamma) + a_c \cos(\psi_i + \gamma), \tag{4.3a}$$

$$a' = a'_0 + a'_s \sin(\psi_i + \gamma) + a'_c \cos(\psi_i + \gamma). \tag{4.3b}$$

The cosine terms a_c and a'_c are induced by the yaw misalignment or by cyclically pitching the blades with a phase angle of γ. The sine terms a_s and a'_s account for the phase lag due to unsteady aerodynamics [159] caused by the cosine disturbances.

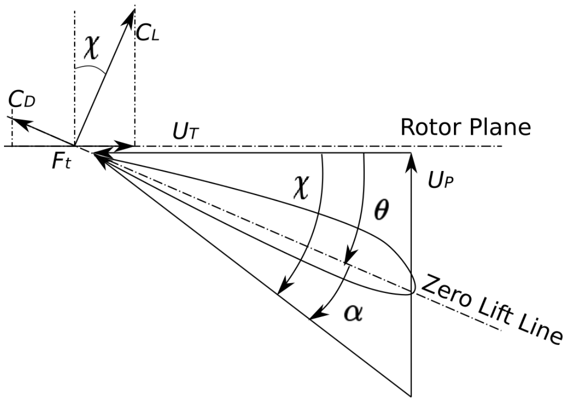

Figure 4.10: Airfoil velocity triangles & Local blade forces

In-plane F_H force

Looking at Fig. 4.10, the sectional tangential force is computed for a small inflow angle as $F_t = \mathtt{L}\chi - \mathtt{D}$, where $\mathtt{L}$ is the airfoil lift and $\mathtt{D}$ the drag. On the other hand, the radial component F_r

is mainly driven by centripetal force, a time-independent value. The lateral force F_H averaged over a rotor revolution is obtained as

$$F_H = \frac{1}{2\pi}\sum_{i=1}^{B}\int_0^{2\pi}\int_0^{R}(\mathtt{L}\phi - \mathtt{D})\cos(\psi_i)\mathrm{d}\psi_i\mathrm{d}r, \tag{4.4}$$

where the local lift $\mathtt{L} = \frac{1}{2}\rho c C_{\mathrm{L}\alpha}(U_P U_T - \theta U_T^2)$ and drag $\mathtt{D} = \frac{1}{2}\rho c C_D U_T^2$ (per unit span) are expressed as functions of tangential and perpendicular velocity components, while c is the chord length, and the coefficients of lift and drag are $C_L = C_{L\alpha}\alpha$ and C_D, respectively.

Consider now the YM case, wherein the partial load region, $\theta \approx 0$ deg. The cyclic pitch phase is not relevant and therefore $\gamma = 0$ deg. After simplifications and assuming radially constant inductions, the expression for the in-plane force becomes

$$F_{\mathrm{H,YM}} = \frac{1}{4}\rho\hat{C}_D A(U_0\cos\varphi)^2\lambda\varphi - \frac{1}{2}\rho\hat{C}_L A(U_0\cos\varphi)^2 a_c, \tag{4.5}$$

where A is the rotor area, $\lambda = \Omega R/U_0$ the tip speed ratio, $\hat{C}_D = C_D\sigma(1 + a_0')$ with σ indicating the rotor solidity, and $\hat{C}_L = C_{L\alpha}\sigma(1 - a_0)$. The first term in Eq. (4.5) is proportional to the yaw misalignment angle, while the second term is induced by the cosine oscillation of the axial induction (which are also assumed to be primarily caused by the yaw misalignment, i.e. $a_c \approx a_{c\varphi}\varphi$).

In the CyPC case, the blade is pitched according to

$$\theta_i = \theta_0 + \theta_c\cos(\psi_i + \gamma), \tag{4.6}$$

where θ_0 is the collective pitch constant, θ_c the 1P pitch amplitude and γ is the phase angle of the blade. γ is used to control the blade azimuth angle where maximum and minimum α is reached.

All the essential parameters and variables are employed to replace the terms in Eq. 4.4. Assuming $\theta_0 \approx 0$ deg in the partial load region, the rotor in-plane force becomes

$$F_{\mathrm{H,CyPC}} = -\frac{1}{2}\rho\hat{C}_L A U_0^2\lambda\left(\frac{2a_c}{\lambda} + \frac{1}{2}\theta_c\right)\cos\gamma. \tag{4.7}$$

This expression shows that the rotor in-plane force depends on the cyclic pitch amplitude and on the cosine fluctuations of the axial induction caused by pitching (i.e. $a_c \approx a_{c\theta_c}\theta_c$).

The inflow model in Eq. 4.5 and 4.7 is not an accurate assumption for the sheared velocity profile. However, by replacing the constant inflow model in Eq. 4.2b with the power law (vertical shear flow) yields sophisticated formulations and is, therefore, not derived in this thesis.

In-plane F_G force

According to high-fidelity simulations, F_G is non-negligible for both YM and CyPC cases. The author models F_G force using the same way as for the derivation of F_H force. The in-plane vertical force is formulated and given as

$$F_G = \frac{1}{2}\sum_{i=1}^{B}\int_0^{2\pi}\int_0^{R}(L\phi - D)\sin(\psi_i)\mathrm{d}\psi_i\mathrm{d}r, \tag{4.8}$$

where cos term is replaced by sin term compared to Eq. 4.4. The yaw angle term, $U_0\varphi\cos\psi_i$, is dropped after the integration. With the assumption of $\gamma = 0$ deg, $F_{\mathrm{G,YM}}$ equals to $F_{\mathrm{G,CyPC}}$, and writes

$$F_{\mathrm{G,YM}} \approx F_{\mathrm{G,CyPC}} = -U_0^2 \rho A [a_s \hat{C}_L + \frac{1}{3}\lambda^2 a_s' \hat{C}_D]. \tag{4.9}$$

The equation reveals that the physical interpretations of F_G for YM and CyPC are similar. They are mainly induced by the unsteady aerodynamics of the blade motion. The equation implies that F_G highly depends on the amplitude of a_s and a_s', which account for the unsteady aerodynamics effects. The equation also indicates that the in-plane vertical force equals to zero if the induction factors are assumed to be circumferentially constant ($a_s' = 0$).

Lateral wake displacement

Jiménez's wake deflection model can be reformulated by introducing the F_H force to the equation. The derived formula writes

$$\alpha_l = \frac{\cos^2\varphi}{1+\zeta\dfrac{X}{D}}\left(\frac{C_T}{2}\sin\varphi - \frac{\hat{C}_D\lambda\sin\varphi\cos\varphi}{4} + \frac{\hat{C}_L a_c \cos\varphi}{2}\right), \tag{4.10}$$

where α_l is the wake skew angle, C_T the thrust coefficient of a rotor, ζ the tuning parameter for the wake expansion, and X the streamwise distance from a rotor plane. The third term in the equation, $\frac{1}{2}\hat{C}_L a_c \cos\varphi$, is one order of magnitude smaller than the second term, $\frac{1}{4}\hat{C}_D\lambda\sin\varphi\cos\varphi$, and therefore can be neglected. By rearranging the formula, the lateral wake displacement $\frac{l}{X}$ writes

$$\left(\frac{l}{X}\right)_{YM} = \tan\left\{\frac{\cos^2\varphi\sin\varphi}{1+\zeta\dfrac{X}{D}}\left[\frac{C_T}{2} - \frac{\hat{C}_D\lambda\cos\varphi}{4}\right]\right\}, \tag{4.11}$$

where the sign of the F_H force term, $\hat{C}_D\lambda\cos\varphi$, implies a negative impact on the wake steering motion. Therefore, instead of improving the wake lateral displacement, the F_H force will, in fact, reduce the steering motion. It requires further investigations of how significant the F_H force is, contributing to the lateral motion of a wake. The lateral displacement of CyPC is derived based on Eq. 4.7 and eq. 4.11 ($\frac{2a_c}{\lambda}$ term is neglected), and is formulated as

$$\left(\frac{l}{X}\right)_{CyPC} = -\tan\left\{\frac{\hat{C}_L\lambda\theta_c}{4(1+\beta\dfrac{X}{D})}\cos\gamma\right\}, \tag{4.12}$$

where the lateral component of rotor thrust, $\frac{C_T}{2}\sin\varphi$ equals to 0 due to the fully aligned inflow and is dropped from the equation. Since the C_T term is significantly larger than the F_H term, this expression indicates CyPC being unable to generate sizeable wake lateral displacement.

4.4.2 Model verification by LES framework

The 5MW NREL wind turbine is used as the wind turbine model for this verification. A spire generated inflow model is replaced by a stochastic atmospheric boundary layer generator. The computational domain is adjusted accordingly.

ABL inflow map

A boundary layer simulator is used to provide a realistic turbulent wind inflow for the wake redirection research. Initial efforts have been focused on generating an artificial boundary layer [160] and on investigating an ABL precursor inflow model [21]. However, a self-developed inflow generator cannot offer a feasible turbulent wind map for high fidelity CFD simulations, while a wall model generated inflow requires a significantly large computational domain and long time length (20,000 s for a stable stratified ABL [63]). To reduce the computational costs and maintain the accuracy, a stochastic, full-field, turbulent-wind simulator, TurbSim, is used in this Ph.D. study [161].

The parameter tuning follows Lavely et al. [162] and Rai et al. [163]. The author used the flat terrain turbulence model, SMOOTH, which offers the applicable wind estimation till 100 to 200 m. This range of height allows one to run a typical 5 MW NREL simulation. The data is superimposed with the coherent turbulence structure obtained from the high fidelity simulations of a Kelvin-Helmholtz billow [161]. Surface roughness uses 0.01 m, the exponent of a power-law velocity profile equals to 0.085, and the friction velocity is set to 0.23 $m \cdot s^{-1}$. A stable ABL boundary condition is employed in this research. According to Abkar et al. [29], a stable ABL allows for a more clear observation of the wake deficit from 4D to 7D downstream, a region where downstream wind turbine models are often located. Moreover, the SMOOTH model offers appropriate estimations of wind flows, which is valid only at near surface layer. Therefore, the momentum transport to the upper layer needs to be minimized to prevent the near-surface layer from being contaminated. A stable boundary condition can provide a relatively low turbulence condition, which limits the momentum transport between the upper and near surface layers. The Richardson number, a measure of thermal convection [164], reads

$$R_B = \frac{(g/T_v)\Delta\Theta_v \Delta Z}{(\Delta U)^2 + (\Delta V)^2}, \tag{4.13}$$

where T_v is the absolute virtual temperature, $\Delta\Theta_v$ the potential temperature difference across a layer of thickness ΔZ. ΔU and ΔV are the changes in the horizontal wind components across that same layer. The potential temperature $\Theta = T\left(\frac{1000}{p}\right)^{0.286}$ is calculated using the mean absolute air temperature, T, and atmospheric pressure. $R_B = 0.261$ in this thesis

Verification

Table 4.1 shows the F_H and F_G in-plane force calculated by the analytical model and by the simulations. All the sampled simulation forces were subtracted from the baseline forces (simulation without CyPC and YM). The comparison of $F_{\mathrm{H,YM}}$ force shows a good agreement between the two models, with errors of 4.5 %, while the comparison shows a large discrepancy of $F_{\mathrm{H,CyPC}}$ force between the simulation and the analytical model. It is likely that assumptions used in the formulation of induction factors may not be an appropriate approximation. The CyPC seems to be more severally influenced by the unsteady aerodynamic effects, which dramatically increases the complexity of the problem. a_{CyPC} and a'_{CyPC} are not the simply a function of trigonometric terms. Therefore, the F_T force cannot be easily modeled.

The analytical solutions for $F_{\mathrm{G,CyPC}}$ and $F_{\mathrm{G,YM}}$ equal to 33 % and 77 % of the simulation forces, not as accurate as the estimation of the $F_{\mathrm{H,YM}}$ force. It is likely that there is not sufficient accuracy regarding the calibration of the sinusoidal coefficients for the axial and tangential induction factors. For instance, a'_s is generally below the order of 10^{-2} and is difficult to be measured, but a'_s cannot be ignored in Eq. 4.9, since the second term is not negligible.

Table 4.1: Analytical and CFD rotor in-plane force F_H and F_G (in kN), CFD rotor nodding moment M_y and yawing moment M_z (in kNm).

	Type	F_H	F_G	M_y	M_z
CyPC	Analytical	-0.99	≈-0.89	-	-
θ_c=7 deg	Simu. uni.	-3.94	-2.69	4414.96	74.59
	Simu. ABL	-5.31	1.94	3918.85	40.55
YM	Analytical	2.10	≈-0.52	-	-
φ=20 deg	Simu. uni.	2.20	-0.67	329.22	248.60
	Simu. ABL	1.94	0.05	333.42	101.68

The sign of F_G forces is negative for both the analytical model and the uniform flow simulation, which is determined by the sign of a_s'. The yaw misalignment angle φ and cyclic pitch angle θ, on the other hand, controls the sign of a_s'. Moreover, the $F_{G,\text{YM}}$ and $F_{G,\text{CyPC}}$ obtained from ABL inflow simulations are positive, opposite to the uniform inflow case. This phenomenon can be explained by the vertical pressure and velocity gradient in a boundary layer. An ascending flow coming from the lower height produces a lift force to the rotor.

Moreover, according to the LES results, although the CyPC force is larger than the YM force, the rotor of CyPC is fully aligned with the incoming wind; therefore, this is all the lateral force that the rotor is capable of exerting on the flow. On the other hand, for the YM case, the thrust contributes to a sizeable lateral force (much larger than F_T), which in turn is responsible for causing a significant lateral wake displacement. The same table also reports the nodding moments of a rotor (located at the tip of a low-speed shaft). It appears that CyPC is not only unable to generate large lateral forces, but it also generates large nodding moments, as already noticed in [3]. Clearly, this is undesirable, because when the machine is subjected to the gusts or enters into the shutdown from an overloaded condition, it will typically generate higher peak loads, which in turn might become the design drivers [165].

4.4.3 Wake characteristics

The velocity slices sampled from the simulations are examined in this section to understand the roles, which YM and CyPC are involved in wake steering and wake recovery. Fig. 4.11 shows the streamwise velocity maps U_X on a hub-height horizontal plane for YM case with $\varphi = 20$ deg and for CyPC with $\theta_c = 7$ deg ($\gamma = 180$ deg). Time-averaged U_X contours are on the left side, while the plots on the right are instantaneous velocity contours. The subplots of YM show respectively the marked wake deflection for uniform inflow without NT (case 1), uniform inflow with NT (case 2) and turbulent inflow with NT (case 3). The wake in case 1 has a typical expected symmetrical shape and breaks down at far wake, while the wake collapses earlier for case 2, where the nacelle and tower are included due to the enhanced flow mixture.

On the other hand, the velocity plot for CyPC shows a drastically different behavior. First, there is little if none lateral movement for all three subplots (uniform inflow without NT (case 4), uniform inflow with NT (case 5) and turbulent inflow with NT (case 6)). The wake profiles are far from symmetrical, revealing a region of significantly lower speed on one side of the rotor disk. This is caused by cyclic pitching that, by changing the azimuthal loading on the disk, creates a significant tilting of the axial induction plane, which then generates a non-symmetrical warped downstream wake shape. The figure also shows that this lack of symmetry generates a stronger mixing with the external flow than the mixing in the YM cases, which clearly facilitates the wake recovery. After 4D, the angular momentum (swirling motion) of the wake decays rapidly, also on account of the coherent vortical structures breaking down and of the mixing of the wake with the external flow. Apart from the unique characteristics of CyPC, similar to YM, the presence of nacelle and tower in CyPC simulations also enhance the wake recovery, as shown in case 4 and 5.

The large-scale turbulent motion driven by a turbulent boundary layer influences both the YM and CyPC cases in a number of aspects, in particular, the yaw misalignment. The wake breakdown takes place approximately at 2.5D downstream of the rotor in case 3, which is at least 3.5D earlier than the case 2. A similar phenomenon can be observed in case 6, where the wake of CyPC is collapsed earlier. Moreover, both YM and CyPC perform faster wake recovery under the turbulence condition, which may enhance the global power capture of multiple interacting wind turbines. On the other hand, the lateral wake motion of YM is reduced by the fluctuating turbulent structure and the enhanced flow mixture, which will certainly reduce the effectiveness of wake steering.

The wake structure is further illustrated in Fig. 4.12, which reports the time-averaged streamwise velocity component U_X on vertical planes at 1D, 3D and 5D downstream of the rotor for uniform inflow without NT(case 1 and 4), uniform inflow with NT (case 2 and 5) and turbulent inflow with NT (case 3 and 6). For YM, the wake shows nearly no lateral displacement and axisymmetrical behavior for case 1 at 1D. The noticeable lateral wake deflection by yaw misalignment is observed at further downstream, and its own lateral motion also deforms the wake shape.

For CyPC case, the plots for all three cases (case 4, 5 and 6) show the effects of a tilting of the axial induction at 1D, which is largely responsible for the generation of large nodding moments on the rotor. CyPC redistributes the wake profile, at the cost of a surge in loadings, which is however not in charge of continuously generating a large & stable wake deflection. A unique non-symmetrical velocity profile appears quickly after the rotor and is coupled with the wake swirling effect. The average swirl rate from 0D to 4D is about 0.47 rad/D for case 4. The influence of tower in case 5 enhances the wake mixture and slightly reduce the swirling motion.

Another observation from Fig. 4.12 and Fig. 4.11 is that YM does not induce as much wake recovery as CyPC. Therefore it usually requires a large misalignment angle to generate a large wake deflection so as to have a partial or even no wake condition for the downstream interacting wind turbines.

4.4.4 Wake steering

Section 4.4.3 shows that only YM can effectively deflect the wake, while CyPC is mainly in charge of increasing the downstream wake recovery. This section tries to quantify the wake deflection by YM. The effectiveness of wake steering for YM will be further verified by comparing simulations with experiments. Additionally, a modified Jiménez's analytical model is computed to quantify the wake steering motion, cross-verify with the simulation and experimental results, and investigate the influence of F_H force to the wake deflection.

Equation (4.11) points to the fact that the wake displacement is, to the first-order approximation, linearly dependent on the misalignment angle φ. As well, the same displacement is only marginally correlated with rotor sizes and operating conditions. To help corroborate these conclusions, different rotors and operating conditions were used for the experiments (G2 model) and simulations (5 MW NREL wind turbine).

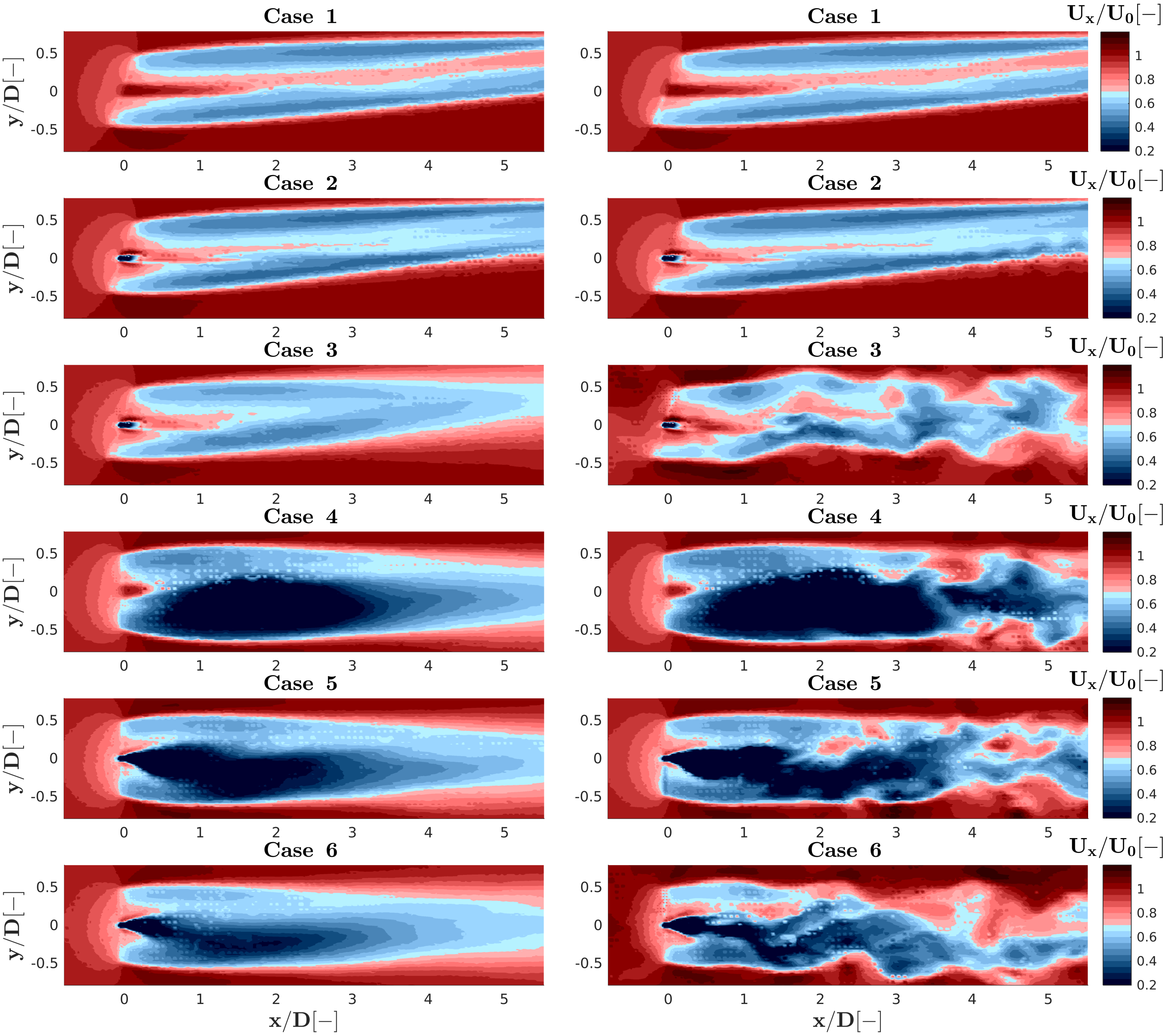

Figure 4.11: 7 deg CyPC & 20 deg YM streamwise velocity at hub height horizontal plane. From case 1 to 6 are: YM without NT with uniform inflow; YM with NT with uniform inflow; YM with NT with turbulent inflow; CyPC without NT with uniform inflow, CyPC with NT with uniform inflow; CyPC with NT with turbulent inflow. Column left and right are respectively the time averaged and instantaneous U_X.

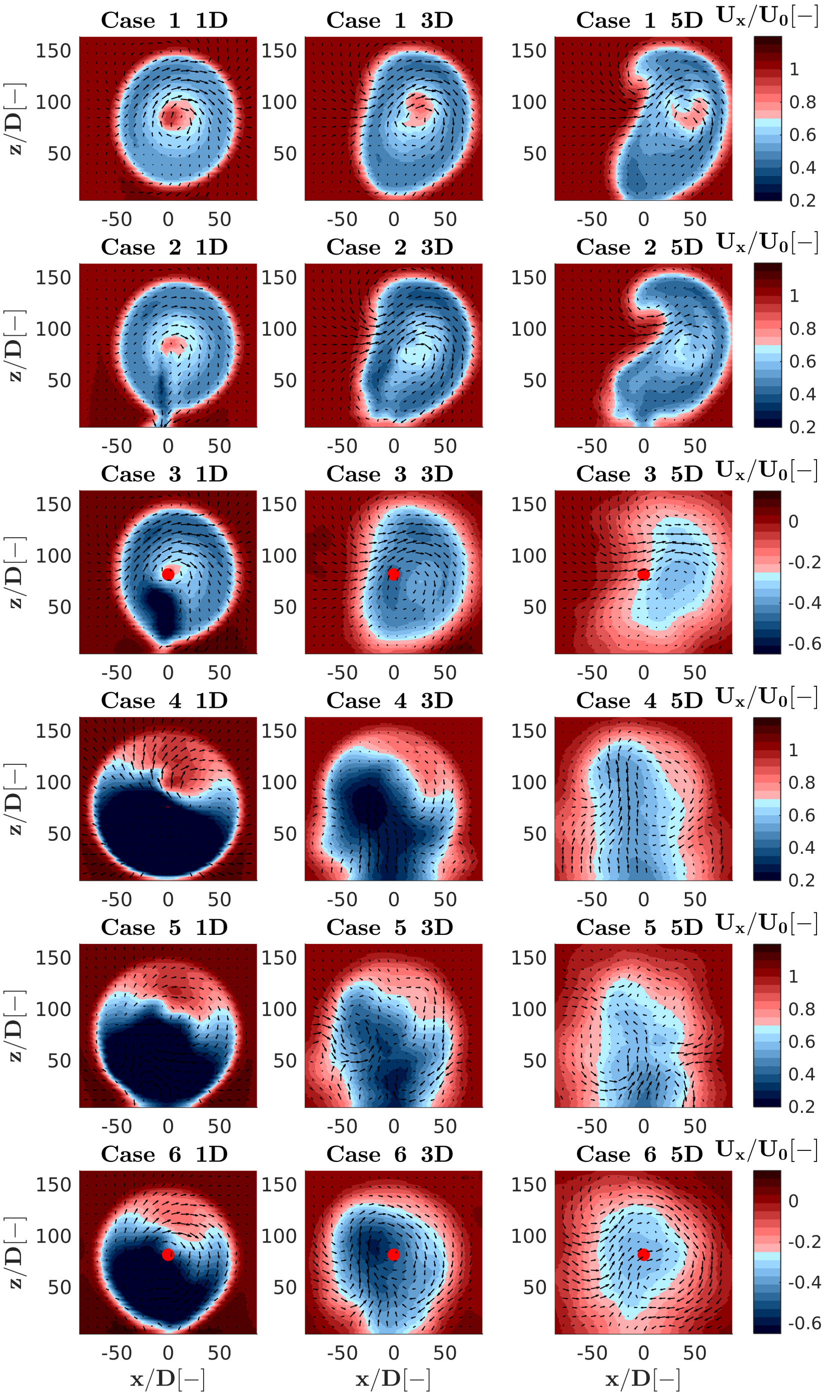

Figure 4.12: 7 deg CyPC & 20 deg YM U_X at 1D, 3D and 5D (from column 1 to 3, left to right) downstream position. From case 1 to 6 are: YM without NT with uniform inflow; YM with NT with uniform inflow; YM with NT with turbulent inflow; CyPC without NT with uniform inflow, CyPC with NT with uniform inflow; CyPC with NT with turbulent inflow.

To identify the wake center, the author follows the concept of the available mean specific power in the wind (AP) [166]. YM and CyPC induce unsymmetrical wake profiles, particularly if a large shear is included. The deflected wake centers are often estimated based on a Gaussian curve, fitted to a wake profile associated with the least square approach [167]. Such method does not allow the wake profile to be highly unsymmetrical, which is in contrast with the wake profile of YM and CyPC. The AP estimation method, on the other hand, enables to identify the wake center for an unsymmetrical profile. The equation reads

$$\begin{aligned} f_{AP}(Y_c) =& \frac{1}{2}\int\int U_X^3(X,Y,Z)dZdY \\ &(Y-Y_c)^2 + (Z-Z_c)^2 \leqslant (D/2)^2 \end{aligned} \tag{4.14}$$

Where $U_X(X,Y,Z)$ is the streamwise velocity at a point in space, Y_c and Z_c the center of a sampled circle (size of a rotor). Y_c moves horizontally to find the location of the minimum $f_{AP}(Y_c)$, which, in other words, is the wake center. The sampled velocity slice of each φ are subtracted by a slice sampled at the same location from an empty wind turbine simulation to remove the background vertical shear.

The results are shown in Fig. 4.13, which reports the wake displacement at 4D distance downstream from a rotor. The displacement is plotted as a function of misalignment angle φ. The simulation curve is plotted using the triangles connected by a blue line, the experimental curve with the cross symbols connected by a red line, and finally the modified Jiménez's analytical wake steering model (Eq. (4.11) is shown with a yellow line.

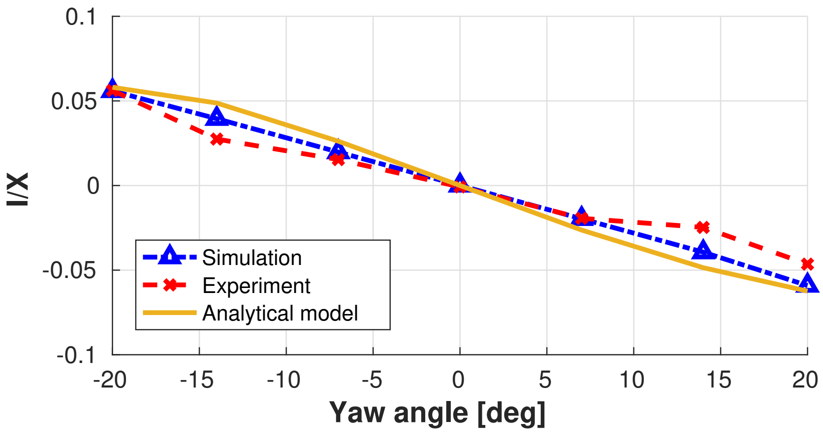

Figure 4.13: Wake displacement for YM vs. yaw misalignment φ

The curves represent a good agreement between the Jiménez's analytical model and the other two curves. The analytical curve is governed by two parameters, C_T and $\hat{C}_D$ in Eq. 4.11, which were measured directly from the simulations. C_T and $\hat{C}_D$ can be considered as constants, as they both oscillate within 5% of their mean values. $\cos\varphi$ is assumed to be one if φ remains small. The wake expansion factor ζ is considered as a tuning parameter in the equation. $\zeta = 0.1$ is used in this thesis. Such assumptions significantly simplify the formulation, and the only remaining variable is $\sin\varphi$.

All three curves in the figure show a similar linear pattern between the wake deflection and the yaw misalignment. The slopes for the experiment and the simulation are -0.14 1/rad and -0.16 1/rad, respectively. The slopes imply close aerodynamic characteristics of the G2 model and 5MW wind turbine. In fact, although the two wind turbine models have distinguished rotor sizes and rotor speeds, the rotor design parameters in Eq. 4.11 are similar. A well designed scaled wind turbine model has comparable design parameters to a normal scale wind turbine. The rotor solidity σ is 0.052 for a 5 MW wind turbine, and 0.047 for a G2 wind turbine model. The tip speed ratio λ is 8.1 for a 5MW wind turbine model and 7.5 for a scaled wind turbine. These values highlight the fact that a non-dimensional wake displacement is, roughly, the same for different rotors operated in different conditions in the partial load regime.

The ratio of $\hat{C}_D$ term and C_T term in Eq. 4.11 is small for both the 5MW NREL and G2 model ($\approx$1 %). The low ratio indicates a negligible contribution of F_H term to the wake steering, which has been already observed in section 4.4.2. Nonetheless, the ratio increases for a scaled wind turbine model with a larger solidity σ. The F_H term $\hat{C}_D\lambda/4$ cannot be neglected if the wind turbine blade with large averaged chord length is used for wind tunnel tests. This non-negligible F_H force will eventually reduce the wake steering, as it is negatively contributed to the wake deflection.

4.4.5 Global power capture

The objective of wind plant control is to achieve an increased net power gain of interacting wind turbines. The previous analysis has indicated that YM is capable of attaining significant lateral wake displacement that, in principle, can be used for reducing the interaction of its wake with a downstream wind turbine. On the other hand, although CyPC is unable to laterally move the wake, it seems to produce faster wake recovery that may result in an increased power capture of downstream wind turbines. Clearly, in all situations, these strategies will result in a net power increase only if what is gained on the downstream machine is more than what is lost on the front one.

A comprehensive analysis of this problem is beyond the scope of the present work, and it would require simulating at least two interacting wind turbines in a wide range of relative positions. Here the author considers several most representative conditions to highlight the key performance in terms of the global power capture for YM and CyPC.

Downstream power estimation

The analysis is performed in a simplified manner by simulating one single wind turbine. The power on a wind-aligned downstream machine is estimated by averaging in space and time U_X on a rotor swap area at several downstream distances. The change of the global power capture ΔP_{tot} for the two interacting wind turbines is computed as [15]

$$\Delta P_{\text{tot}}(X) = \Delta P_u + \frac{1}{2}\rho A \left(U_{X1}^3(X) - U_{X0}^3(X)\right) C_P, \tag{4.15}$$

where ΔP_u is the power loss/gain for the upstream machine, $C_P = 0.482$ the power coefficient of a 5 MW NREL wind turbine [148], U_{X1} the rotor-averaged mean streamwise velocity for CyPC or YM as a function of downstream position X, and U_{X0} the same quantity for a baseline case without CyPC nor YM.

The verification of Eq. 4.15 is briefly introduced next. The computational domain contains the two wind-aligned and identical wind turbines. The setpoint for the upstream machine equals to 7 deg of C_yPC with 180 deg of γ, while the second wind turbine is located at 4D downstream, the location where the estimated power is also measured. Instead of estimating the theoretical power output of the downstream wind turbine, this second wind turbine model is simulated at the same position to generate the actual power. The same ABL inflow is used for the verification case, associated with other essential boundary conditions. $\Delta P(4D)$ from the two-aligned wind turbines simulation is less than 5 % of uncertainty concerning the value calculated directly by Eq. 4.15.

Power capture

The rotor-averaged streamwise velocity U_X is plotted vs. downstream position in Fig. 4.14. Fig. 4.15 plots the total power change formulated as Eq. 4.15. The CyPC cases with 7 deg pitch amplitude are reported with triangle (without NT, uniform inflow, marked as case 1), circle (with NT, uniform inflow, marked as case 2) and cross (with NT, turbulent inflow, marked as case 3) symbols connected by the blue curves, while the YM cases with 20 deg misalignment angle are plotted by the corresponding symbols and red curves on the same figures (marked as case 4, 5, 6).

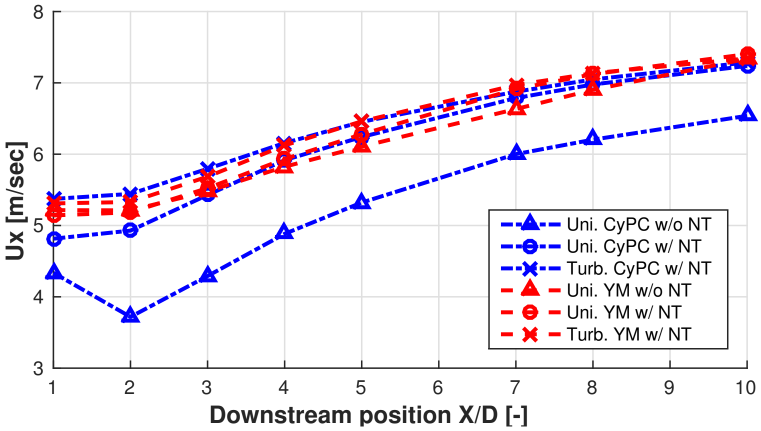

Figure 4.14: Downstream longitudinal flow velocity U_X for the CyPC 7 deg and YM 20 deg cases.

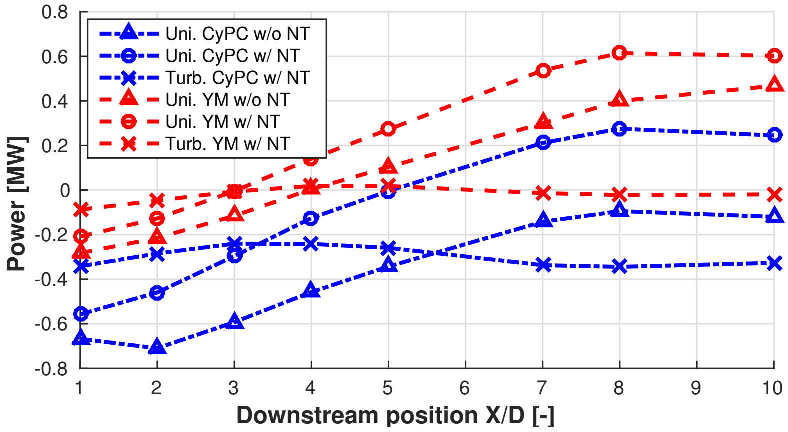

Figure 4.15: total power change vs. downstream position X/D for the CyPC 7 deg and YM 20 deg cases.

Fig. 4.14 shows that, even at this small interaction distance of 4D and in a fully aligned configuration, the wake displacement induced by YM is able to obtain a significant increase in the average wind speed at the downstream wind turbine rotor for all three curves. The uniform w/o NT YM (case 1) results in a net power gain starting from a downstream distance of 3D in this particular configuration, while the signature of NT appears to have positive influence on the global power capture, as its ΔP_{tot} is consistently higher than case 1, and reaches $\Delta P_{\text{tot max.}}$ at 8D. The discrepancies highlight the importance of modeling the nacelle and tower, especially at low turbulence condition.

ΔP_{tot} under turbulent inflow for YM (case 3) doesn't perform similarly as the uniform inflow ΔP_{tot}. From 1D to 3D, ΔP_{tot} increases due to the wake steering motion. After 4D, as shown in Fig. 4.11, large-scale turbulence motions jeopardize the lateral wake displacement, so the maximal ΔP_{tot} remains around 0 MW. It is essential to test even larger yaw angle, possibly coupled with the reasonable Coriolis force at full wake condition, trying to observe higher net power gain under a turbulent inflow condition.

The same plots show a downstream velocity increment for all three CyPC cases as a result of the enhanced mixing of the wake. The power capture recovery rate from 2D to 7D is around $\Delta P_{\text{tot}}/X \approx 0.13$ MW/D for the two uniform inflow cases (case 4 and 5), which is close to the recovery rate of YM. CyPC with nacelle and tower (case 5) provides a consistently higher global power capture than the case 4. The increment of ΔP_{tot} from 4D to 7D are 0.37 MW and 0.38 MW for case 4 and case 5 respectively. The CyPC turbulent inflow curve is similar to the curve for YM. The net power gain is not proportional to the increase of downstream position. ΔP_{tot} of CyPC reaches the maximum at -0.24 MW and stabilizes at -0.29 MW, which clearly indicates a net power loss. Moreover, It is addressed, even if CyPC obtains a modest power increase in a particular configuration and operating condition, this positive effect would have to be traded against a significant increase in the loading of the machine, as previously shown.

In summary, the global power capture by YM is superior to $\Delta P_{\text{tot}}(X)$ by CyPC in both uniform and turbulence flow conditions. The effectiveness of the wake recovery decreases with an increase of the background turbulence. It is also important to model the nacelle and tower in a low turbulence environment.

CHAPTER 5

Scale-adaptive Simulation Framework

This chapter considers the application of a scale-adaptive simulation CFD formulation for the modeling of single and waked wind turbines in flows of different turbulence intensities. The SAS approach is compared to a large-eddy simulation formulation, as well as to experimental measurements performed in a boundary layer wind tunnel with scaled wind turbine models.

5.1 Flow quantities

Three flow quantities are used to evaluate the simulation accuracy with respect to experimental measurements in this chapter. The average velocity $\langle u \rangle$ is computed based on time-averaged velocity components sampled along a hub-height horizontal line in the flowfield, which are then spatially averaged along the rotor diameter. The percentage average velocity error is defined as $\langle \Delta(u) \rangle = (\langle u_{\text{sim}} \rangle - \langle u_{\text{exp}} \rangle)/\langle u_{\text{exp}} \rangle$. The root mean square (RMS) error is used to quantify a spatial fit between simulations and experiments [168], and it is defined as

$$\text{RMS}(u) = \sqrt{\frac{1}{N} \sum_{j=1}^{N} \left(\left\langle u^j_{\text{sim}} \right\rangle - \left\langle u^j_{\text{exp}} \right\rangle \right)^2}, \tag{5.1}$$

where $\langle u^j \rangle$ is a time-averaged velocity component at a given spatial point j. The calculation of turbulence intensity $\sigma/\langle u \rangle$ needs to account for the turbulence both at the resolved and modeled scales. To this end, modeled fluctuations are summed to the resolved ones, yielding

$$\frac{\sigma^j}{\langle u^j \rangle} = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(u^{i,j} - \langle u^j \rangle \right)^2 + \langle 2/3\, k^j \rangle}}{\langle u^j \rangle}, \tag{5.2}$$

where $u^{i,j}$ represents a velocity component at a given spatial point j and at time step i. The term $\sqrt{\langle 2/3\, k^j \rangle}$ is the velocity fluctuation corresponding to the modeled turbulence kinetic energy k^j. In addition to the point-wise turbulence intensity $\sigma^j/\langle u^j \rangle$, rotor-average turbulence intensity $\sigma/\langle u \rangle$ and turbulence intensity $\text{RMS}(\sigma/\langle u \rangle)$ are defined similarly to the velocity case.

5.2 Parameter tuning

The time step length for LES, SAS and SST turbulence models is limited by imposing that the lifting line representing the blade does not cross more than one cell in one step [26], which is equivalent to 0.3 of the maximum Courant number. Since the size of the smallest cells for the LES grid is half that of the SAS and SST ones, the time step is accordingly smaller. The difference in grid size (and hence the number of cells) and in time step length are the main factors driving the different computational costs of these methods. In fact, although SAS and SST require the solution of two additional transport equations, the resulting additional cost is negligible when compared to the effects of grid size and time step. On average, SST and SAS are roughly 13 times faster than LES for the simulation cases considered here.

The Gaussian width ϵ of the ALM is set to 2.5 times the cell size at the rotor disk, which once again results in different values on account of the different grid densities of the turbulence models. The constant Smagorinsky model is employed for the LES case, using C_s equal to 0.13. The scaling parameter F_{SAS} is set to 2 for the SAS model. Tests have shown that the performance of the SAS model is dependent on this parameter, so that its careful calibration becomes essential for the accuracy of the results.

5.3 Single-turbine baseline case

A first baseline case is used to tune the parameters for the three turbulence models, parameters that are then used unchanged in the other cases considered herein. The baseline case represents an isolated flow-aligned wind turbine operating in a low turbulence environment in the partial load region. The CPU time ratio of LES and SAS for this case is $\mathrm{CPU_{LES}}/\mathrm{CPU_{SAS}}$=16.9.

The wind turbine power measured in the experiment is 45.8 W, while for SST, SAS and LES it is 44.8 W, 45.1 W and 45.5 W, respectively. Hence, the power output predicted by SAS appears to be in good agreement with both LES and measurements. As previously stated, the same ALM Gaussian width in terms of cell size is used for all methods.

Fig. 5.1 shows vorticity contours for the LES, SAS and SST simulations, respectively from left to right. It appears that LES is capable of a significantly higher resolution of the tip and root vortices than SAS, thanks to its denser grid. Additionally, a much higher vorticity is produced by SAS compared to SST, due to its enhanced ability of resolving small scale features.

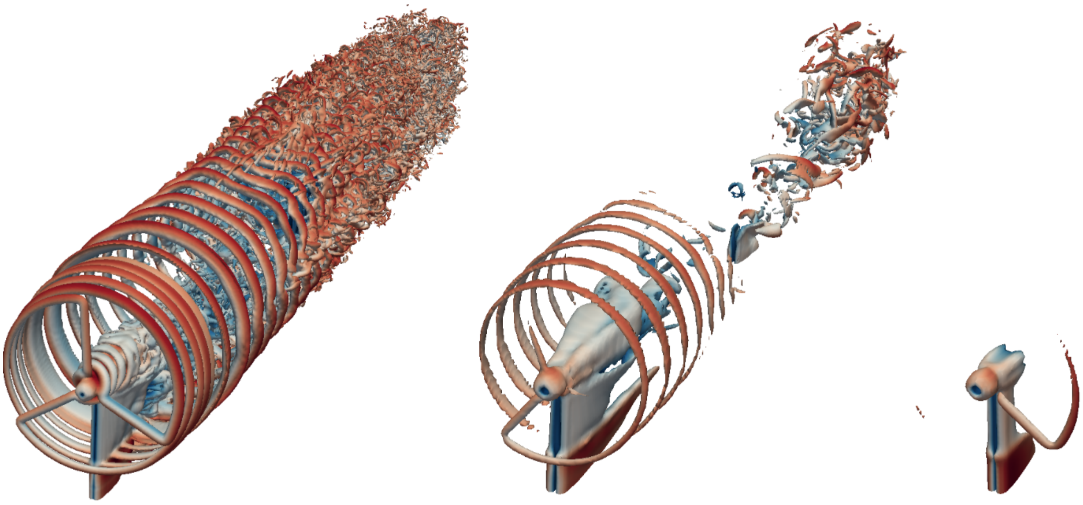

Figure 5.1: Vorticity iso-surface plots ($|\nabla \times \mathbf{u}| = 40$ 1/s) for LES, SAS and SST (from left to right).

For a more precise understanding of these different representations of vorticity and of the overall modeling of wake structures, experimental measurements at hub height and at different distances downstream of the rotor are considered. Fig. 5.2 shows the normalized time-averaged longitudinal velocity (top) and turbulence intensity (bottom) profiles at several downstream distances for LES, SST, SAS and experiment.

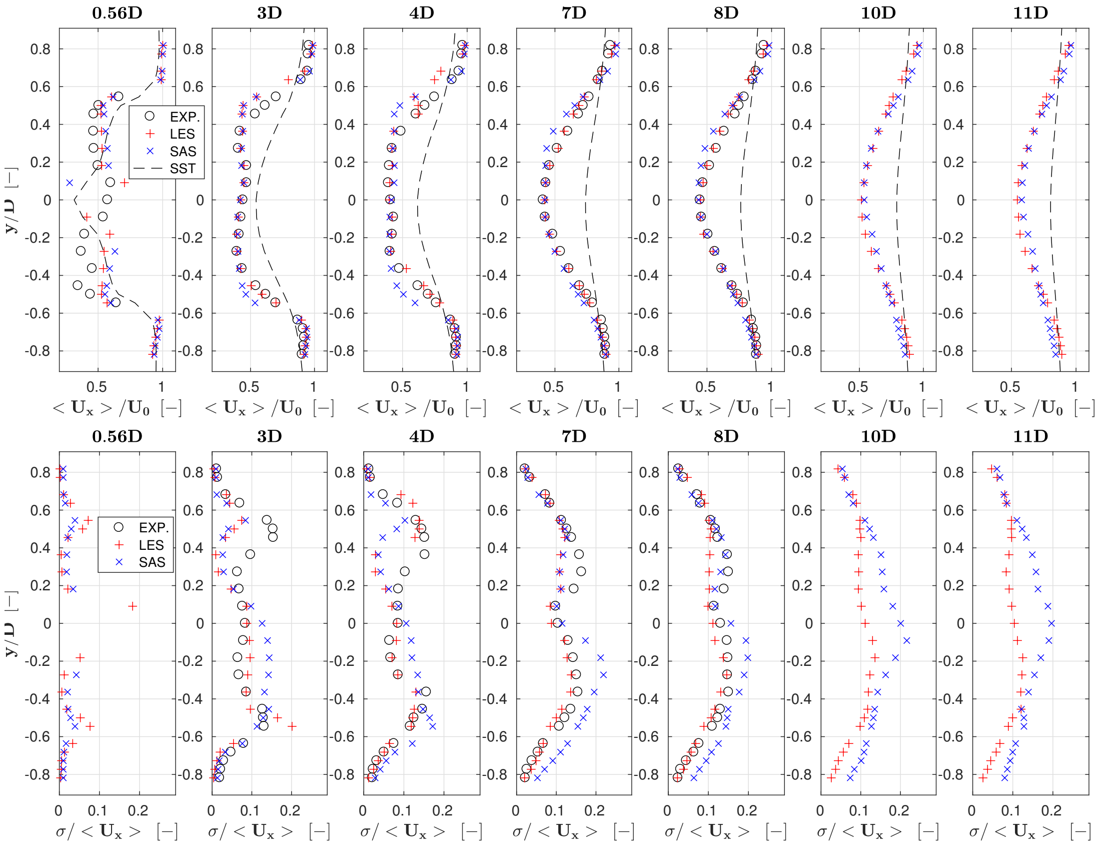

Figure 5.2: Normalized time-averaged stream-wise velocity (top) and turbulence intensity (bottom) profiles at hub height and at several downstream positions. Experiment: black ∘ symbols; LES: red + symbols; SAS: blue × symbols; SST: black dashed line.

The average percent error $\langle\Delta(u_x)\rangle$ between simulation and experiment for LES is equal to -2.7%, -1.6% and -1.1% at 3D, 4D and 8D, respectively. For SAS, the error is -4.1%, -5.4% and -3.3% at the same downstream positions. RMS(u_x) for LES equals 0.34 m/s, 0.33 m/s and 0.12 m/s at 3D, 4D and 8D, respectively, and it is 0.44 m/s, 0.50 m/s and 0.25 m/s for SAS at those same locations. The SAS velocity profiles show a reasonable agreement with both LES and the experimental curves.

On the other hand, the error at the same distances for SST is 11%, 13% and 13%, which is significantly larger than for SAS. The SST RMS(u_x) is also on average twice as large than for SAS. These results suggest that, by including local flow inhomogeneities through $\left|\nabla^2\tilde{\mathbf{u}}\right|$, the modeling of the wake is significantly improved, by a locally adjusted eddy viscosity and limited numerical diffusion.

It is interesting to observe that, close to the rotor (0.56D), SST and SAS predict nearly the same speed profile. In fact, in the near wake region the flow is not yet strongly affected by mixing and numerical diffusion, so that differences in the modeling of unresolved scales play a lesser role. In this region of the wake, the behavior is mostly governed by the rotor thrust, which indeed is quite similar for all three models. The 10 s time-averaged thrust is in fact 16.1 N, 15.9 N and 15.7 N for LES, SAS and SST, respectively. On the other hand, moving downstream away from the rotor, the overestimated eddy viscosity of the SST model begins to show its effects on the wake deficit, as apparent in the plots starting at the 3D location all the way to the end of the domain.

It should be noticed that at 3D and 4D the velocity profiles of LES match very well those of the experiments, while SAS predicts a slightly larger wake width. This phenomenon is due to a lack of resolution of the blade tip vortices, which diffuses the local flow fluctuation. Further downstream, the tip vortices collapse, break down and convert from local velocity shear (tip vortices) to the turbulence, and therefore this effect is reduced. In particular, SAS curves show a very good match with LES at 10D and 11D. This indicates that numerical diffusion is well controlled by the SAS model throughout the propagation of the wake, and flow mixture is properly resolved.

It should also be remarked that the resolution of tip vortices plays a lesser role than in the present case for moderate/high turbulence inflows. In fact, in those conditions vortex breakdown will take place earlier, similar to the breakdown in the experiment, due to the higher background ambient turbulence. Therefore, the accuracy of SAS is expected to improve for more turbulent cases, as in fact confirmed by results shown later on in this work. From this point of view, this initial baseline scenario represents a particularly difficult problem.

LES underestimates turbulence intensity by 23% and 12% at 3D and 4D, respectively, while for SAS the error is 11% and 10%. The consistent underprediction of turbulence intensity for low turbulence inflow conditions in the near wake region has already been observed by [158]. However, results are similar for the two models considered here, which indicates the ability of SAS in resolving second order quantities in the near wake region.

There is a significant lack of symmetry in the profiles left (looking downstream, i.e. for positive y values) and right of the rotor axis for both methods, as in fact the left peak is significantly underpredicted. This lack of symmetry however does not appear in the experimental results. This is probably due to a combination of lack of resolution of the tip vortices and their interaction with the wake shed by nacelle and tower. This problem is analyzed in detail further on, in reference to a yaw misaligned case. The effects of this lack of symmetry on turbulence intensity is consistent with a small lack of symmetry in wake recovery. In fact, especially at 7D and 8D, the numerical velocity profiles exhibit a reduced wake recovery on the left of the wake compared with the experimental measurements. This fact is attributable to the lower upstream turbulence intensity on this same side of the wake, shown in the bottom row of plots of Fig. 5.2.

Regarding the far wake at 10D and 11D, SAS overestimates turbulence intensity by more than 50%, which may lead to a faster wake recovery further downstream. However, such a problem is only limited to low turbulence conditions, and the situation improves for higher turbulence.

Since SST is clearly unable to provide for sufficiently accurate estimates of the wake behavior, it is not considered further in the present work.

5.4 Single-turbine yaw-misaligned case

A correct estimation of wake behavior in yaw misaligned conditions is crucial, especially when significant intentional misalignment angles are generated for wake deflection wind plant control. LES and SAS are compared to experiments, by considering hub-height velocity profiles 4D downstream of the rotor for six different yaw misalignment angles, namely ±5 deg, ±10 deg and ±20 deg. The flow conditions are the same low turbulence ones of the baseline case. Therefore, as previously observed, SAS results are somewhat affected by a lack of resolution of the tip vortices in the near wake region. The CPU time consumption ratio for this case is $\mathrm{CPU}_{\mathrm{LES}}/\mathrm{CPU}_{\mathrm{SAS}}$=14.4.

Fig. 5.3 shows a comparison of the velocity (top) and turbulence intensity (bottom) profiles for the considered yaw misalignment angles. The velocity profile of the SAS model shows a relatively good agreement with both the experiment and LES. The maximum average-velocity error for SAS is 5.2% at -10 deg, while it is 4.1% for LES at 20 deg. The overall error over the six yaw configurations is 4% and 1% for SAS and LES, respectively. Likewise, the maximum RMS(u_x) for SAS is 0.54 m/s and it is 0.35 m/s for LES. The overall RMS(u_x) over the 6 yaw configurations is 0.45 m/s and 0.29 m/s for SAS and LES, respectively. As noted in the baseline case, SAS overpredicts the wake width, due to a lack of resolution of the tip vortices. However, other than this, it is in a reasonably good agreement with LES.

The overall average turbulence intensity error over the 6 yaw configurations is 11% and 20% for SAS and LES, respectively. In both cases, turbulence intensity is not everywhere matching well with the experiments. Here again, one of the two external peaks of the profiles is typically severely underpredicted by both methods, similarly to what was observed for the baseline case.

In fact, the turbulence intensity peaks correspond to the blade tip region, where the mesh is not fine enough for an accurate modeling of the tip vortices. For LES, the blade tip chord length is 1.8 times the cell size, which is clearly not enough to precisely resolve the tip vortices. The situation would be clearly much different for a body conforming blade meshing approach, however at the cost of an increased computational effort. The left peak in Fig. 5.3 is particularly much lower than in the experimental case. Since the SAS cell size at the blade tip is twice as large as in the LES case, this peak for SAS is even lower than the one for LES. The same phenomenon can also be observed in Fig. 5.4, where the blade tip vorticity and turbulence intensity contours all indicate that the SAS model is less capable of resolving tip vortices than LES.

On the other hand, the turbulence intensity peak on the right is well predicted by both SAS and LES. This phenomenon can be explained by the interaction of the wake shed by the tower with the tip vortices. Since the rotor spins in a clockwise direction (looking downstream), the wake has a counterclockwise swirling motion, as shown by the two bottom plots of Fig. 5.4. In turn, this causes the tower wake to move slightly to the right and upwards, increasing the turbulence intensity at hub height in this region of the wake. This effect is well illustrated

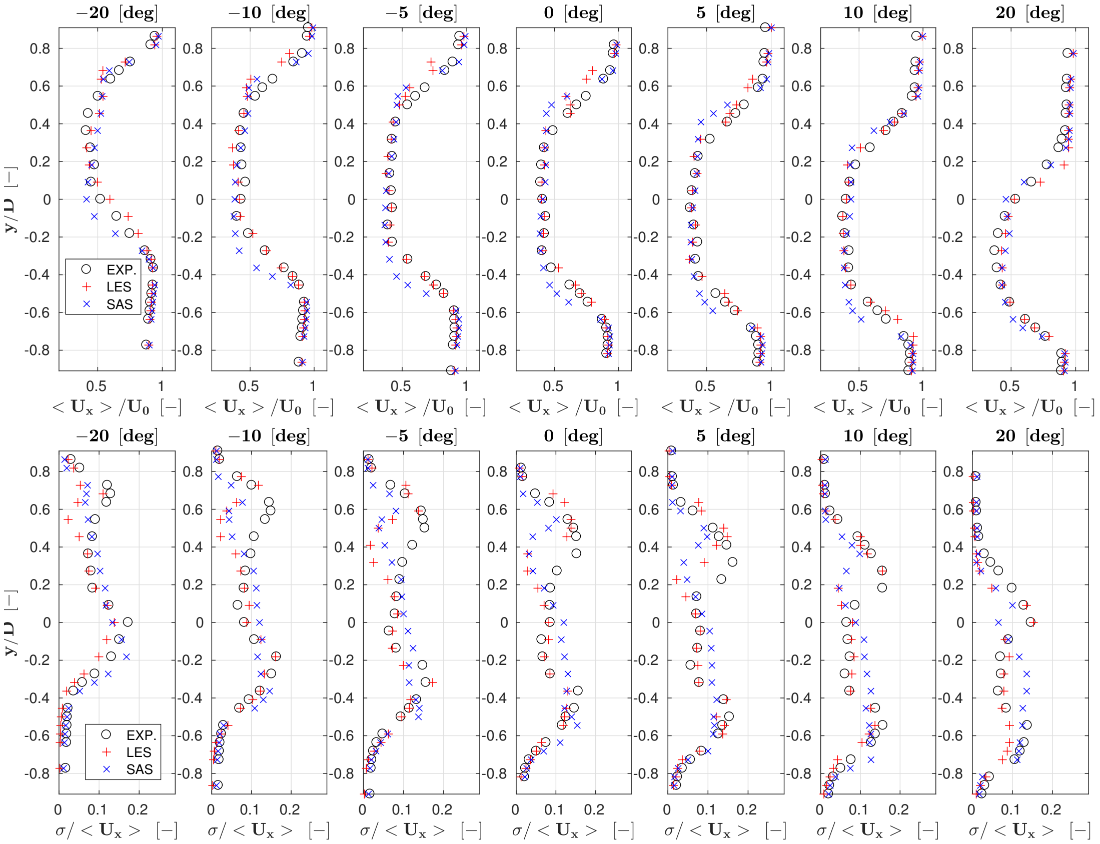

Figure 5.3: Normalized time-averaged stream-wise velocity (top) and turbulence intensity (bottom) profiles at hub height and 4D downstream, for various yaw misalignment angles. Experiment: black ∘ symbols; LES: red + symbols; SAS: blue × symbols.

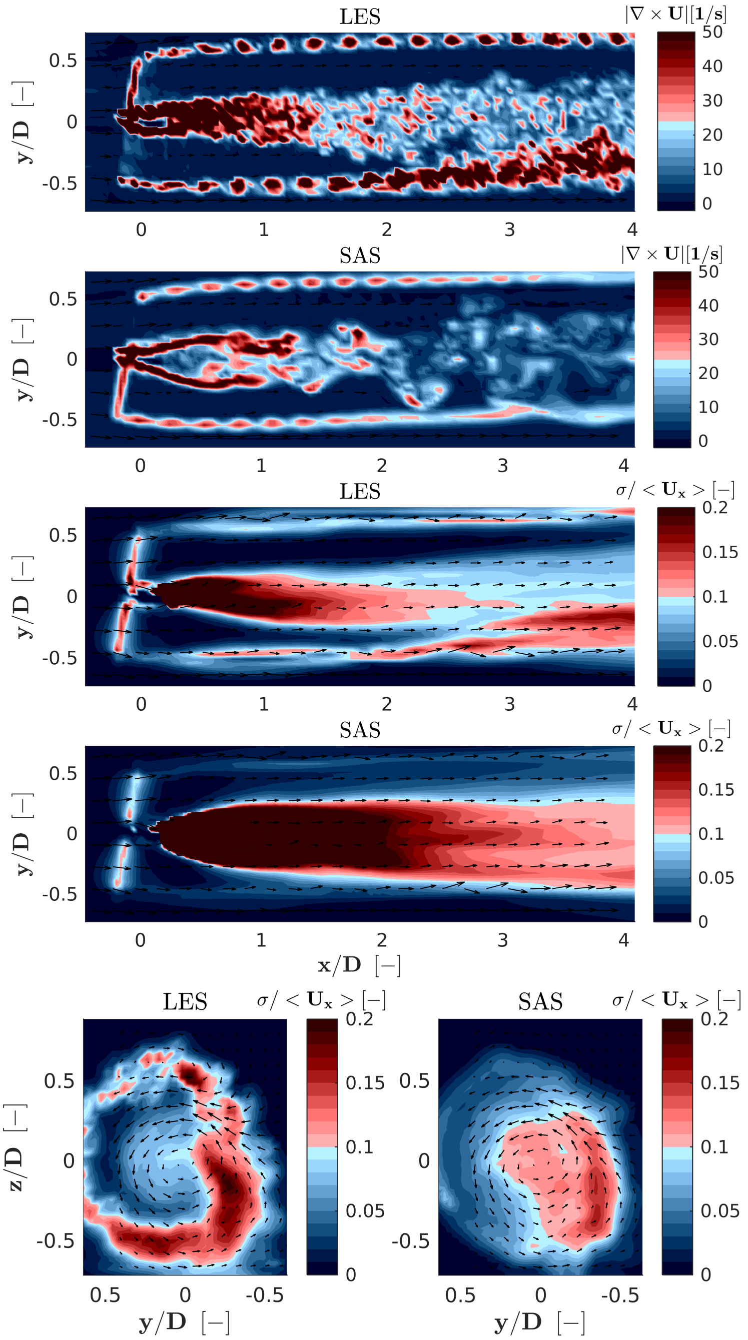

Figure 5.4: From top to bottom, vorticity magnitude $|\nabla \times \mathbf{u}|$ and turbulence intensity $\sigma/\langle u_x \rangle$ on a hub-height horizontal plane, and turbulence intensity on a vertical plane (looking downstream) 4D behind of the rotor. Black arrows indicate the cross-flow velocity vector at a number of sampling points.

by the LES turbulence intensity color plot of Fig. 5.4. The higher turbulence intensity on the right of the wake promotes a faster decay of the tip vortices than on the left, as shown by the vorticity color plot of the same figure. Because of this enhanced mixing, it is not necessary to have an extremely fine resolution of the grid, and even the relatively coarse mesh used here is enough to capture reasonably well the turbulence intensity peak on the right of the wake. The situation is different on the left: here, there is a very low background turbulence, as the incoming flow is almost uniform and there is little or no effect from the tower wake. Hence, to estimate the correct turbulence intensity one would have to resolve very accurately the tip vortices, something that is however not possible with the current grid density.

This interpretation of the results was confirmed by a simulation conducted without nacelle and tower. In that case, which is not reported here for brevity, very similar turbulence intensity peaks were observed to both the right and left of the wake.

The good matching of the right peak deserves a further comment. In fact, here the turbulence intensity matches well the experiment, while the results on the left peak demonstrate a general lack of resolution of the tip vortices. Hence, to compensate for this, the turbulence generated by the immersed boundaries of tower (and nacelle) must probably be overestimated. Indeed, this is very probable, based on the large difference between SAS and LES (which has a twice as dense mesh) in turbulence intensity in the hub-height core of the wake shown in Fig. 5.4.

The analysis can be applied to the baseline case described in section 5.3, which shows a very similar behavior of the turbulence intensity peaks.

5.5 Two-aligned-turbine baseline case

In this two-aligned-turbine case, the operating condition of the upstream wind turbine is identical to one of the isolated case, but a second machine operates in its wake at a downstream distance of 4.1D. The power capture of the second wind turbine measured in the experiment is 4.5 W, while for the SAS and LES simulations it is 3.9 W and 4.4 W, respectively. The relatively large error of the SAS case is probably due to a few reasons. First, the power output is quite small and close to the cut-in limit of the wind turbine. Second, power capture depends on the cube of the wind speed, and therefore it is highly sensitive to errors in the predicted wake speed. The 4D velocity profile is located just in front of the second wind turbine, as shown in Fig. 5.5. As already noticed, at this downstream distance the wake width of the SAS solution is larger than for the LES case, because of the poor resolution of the blade tip-vortices. The average speed $\langle u_x \rangle$ is 3.9% smaller than in the LES case, which roughly equals the 14% of power error between the experiment and the SAS model. Third, the calculation of the angle of attack at the airstations along the lifting lines are affected by the different densities of the LES and SAS grids cut by the blades as they sweep across the rotor disk, which in turn may affect the resulting aerodynamic forces and hence power. Here again, this problem might be alleviated by a denser SAS mesh.

Fig. 5.5 shows the average streamwise speed $\langle u_x \rangle$ and turbulence intensity curves for SAS, LES and experiment. $\langle \Delta u_x \rangle$ for LES is equal to -4.0%, -2.3% and -1.2% at 8D, 10D and 11D, respectively. The same quantity is 0.1%, 3.4% and 4.1% for the SAS model at the same positions. The velocity profiles at the far wake show that wake recovery for the SAS model is somewhat higher than for the LES and experimental curves. The wind turbine nacelle and tower may contribute to this effect. In fact, the immersed boundary interpolation algorithm operates on a larger volume due to the coarser SAS mesh. A large volume implies an extended cross-sectional area of nacelle and tower, which introduce extra mixing and, in turn, may generate a somewhat faster wake recovery. The LES approach underestimates turbulence intensity by 10%, 8% and 7% for 8D, 10D, and 11D, respectively, while at the same position, SAS overestimates the turbulence intensity by 0.8%, 9% and 11%.

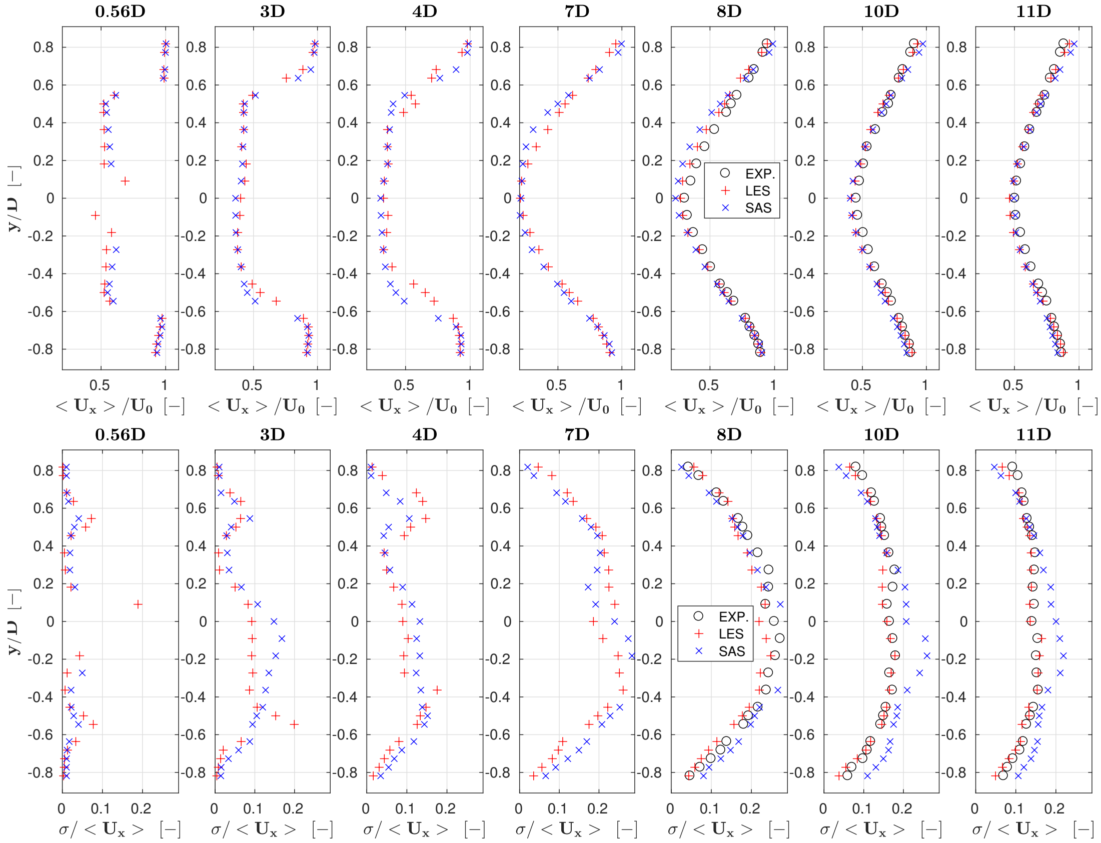

Figure 5.5: Normalized time-averaged stream-wise velocity (top) and turbulence intensity (bottom) profiles at hub height and at several downstream positions. Experiment: black ∘ symbols; LES: red + symbols; SAS: blue × symbols; SST: black dashed line.

5.6 Turbulent cases

5.6.1 Single-turbine moderate-turbulence case

Next, an isolated wind turbine is considered in a moderate turbulence environment. The turbulent inflow is generated by the precursor simulation, as described in section 3.2. The machine is aligned with the flow and operates at a fixed rotating speed of 720 RPM with a collective pitch of 1.4 deg. The numerical models use the same exact parameters employed for the low turbulence cases. The CPU time ratio between the two turbulence models is $\text{CPU}_{\text{LES}}/\text{CPU}_{\text{SAS}}$=9.37 in this case. The 60 s average rotor power is equal to 31.0 W for the experiment, 30.5 W for LES and 30.1 W for SAS.

Fig. 5.6 shows the normalized velocity and turbulence intensity profiles for the experiment, LES and SAS at -1.5D, 1.4D, 1.7D, 2D, 3D, 4D, 6D and 9D. The first measurement station at -1.5D is upstream of the wind turbine, where the flow can be regarded as the undisturbed free stream. The velocity profiles are all, in general, in a good agreement with one another. The overall simulation error $\langle\Delta(u_x)\rangle$, averaged over all distances, is equal to 0.9% for LES and it is 1.1% for SAS. At 4D downstream, where a second wind turbine is located in other experiments, $\langle\Delta(u_x)\rangle$ is 2.1% for LES and 3.2% for SAS. Throughout the wake propagation from near (1.4D) to far wake (9D), the RMS(u_x) for LES gradually reduces from 0.18 m/s to 0.13 m/s, while for SAS it decreases from 0.21 m/s to 0.08 m/s. Comparing to the low turbulence case of section 5.3, the RMS values are drastically reduced, which indicates a significant increase of the simulation accuracy for the present moderate turbulence condition.

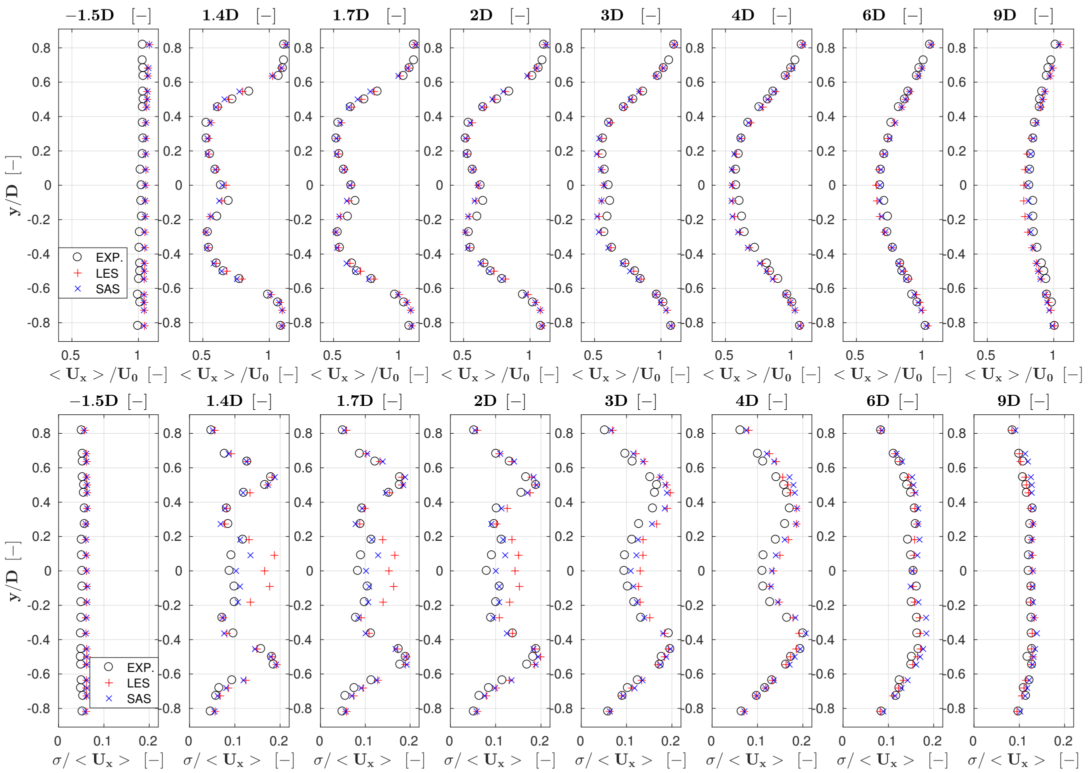

Figure 5.6: Normalized time-averaged stream-wise velocity (top) and turbulence intensity (bottom) profiles at hub height and at several positions. Experiment: black ∘ symbols; LES: red + symbols; SAS: blue × symbols.

From 1.4D to 9D, the average turbulence intensity error for SAS is 2.6%, 5.1%, 6.3% 10.7%, 12.1%, 8.2% and 7.6%, respectively. Contrary to the low turbulence (baseline) case, turbulence intensity for SAS significantly improves in the far wake (10D and 11D) in the moderate turbulence environment. For instance, the turbulence intensity RMS at 8D was 0.03 in the low turbulence case, while it is 0.01 in the present case at 9D. A good estimation of turbulence intensity is necessary for the correct estimation of wake deficit. The good match observed here at 9D is therefore encouraging for the use of the present simulation models both for closely spaced wind farms, where the wake might be interacting with multiple machines, and for largest spacings, where one needs to account for impingement of wakes shed by machines far upstream.

In the near wake, a proper estimation of the effects of tip vortices can be observed, differently from the low turbulence case discussed in section 5.3. The two turbulence intensity peaks can be clearly observed from 1.4D to 4D, covering the whole near wake range. It is possible that even a coarser grid could be used in this case, although a precise characterization of the degradation of the results with decreasing mesh density was not performed.

5.6.2 Single-turbine high-turbulence case

Next, a high turbulence (20%) condition is considered. In this case, experimental measurements are not available, and SAS is compared directly to LES, which serves as a benchmark. The turbulent inflow is generated as previously explained, using the corresponding high-turbulence spires and wind tunnel configuration. The CPU time consumption ratio is $\mathrm{CPU_{LES}}/\mathrm{CPU_{SAS}}$=9.37.

Results indicate a good agreement between SAS and LES, both in terms of velocity and of turbulence intensity, as shown in Fig. 5.7. Here again, one can notice that the difference between the two models tends to decrease in higher turbulence conditions. This is particularly true in the near hub region, which is probably due to the higher mixture created by the background turbulence.

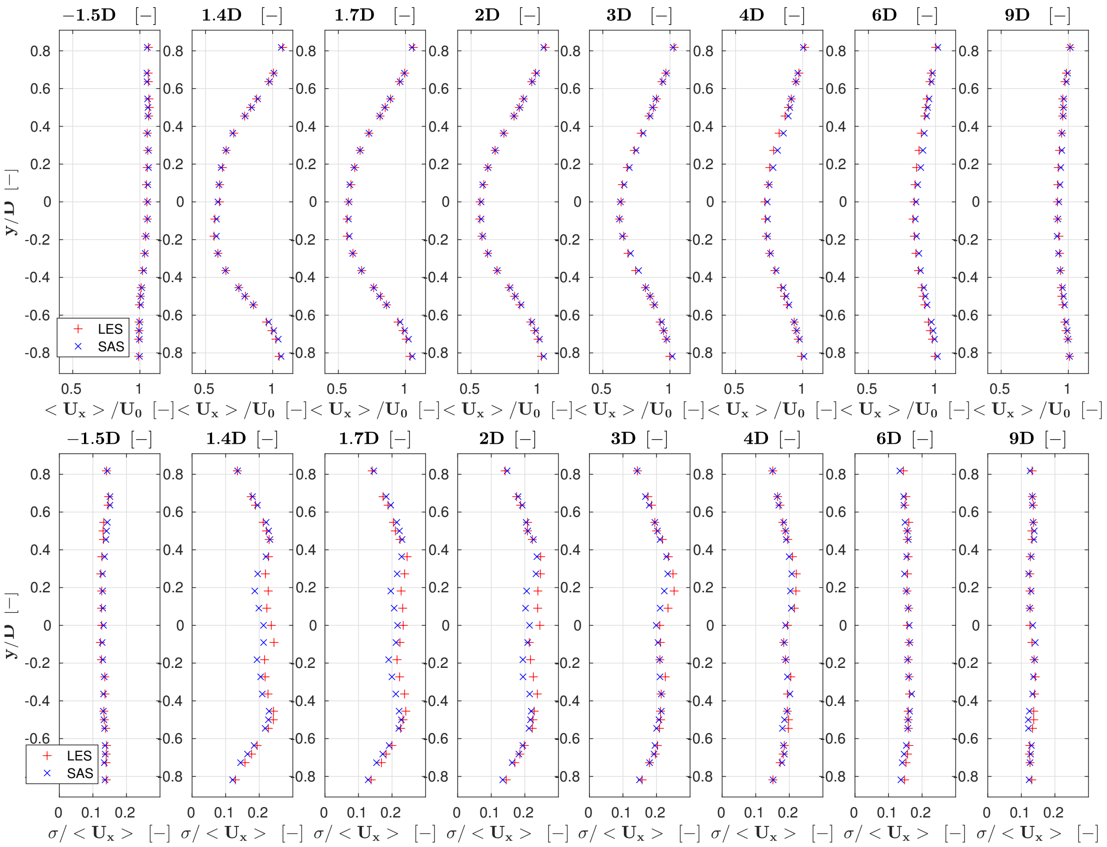

Figure 5.7: Normalized time-averaged streamwise velocity and turbulence intensity profiles at hub height at downstream positions: -1.5D, 1.4D, 1.7D, 2D, 8D, 3D, 4D, 6D and 9D. blue x symbols: SAS; red crosses: LES; black circles: experimental results.

5.7 Three aligned turbines

Results shown up to here indicate that SAS achieves in general a good agreement with LES both in terms of wake deficit and turbulence intensity. The match is of a better quality for increasing turbulence, while it is less satisfactory for low turbulence conditions. However, the moderate and high turbulence flows considered here represent more realistic atmospheric boundary layers, while very low turbulence conditions are less likely to be encountered in actual conditions in the field.

To characterize the performance of SAS in conditions characterized by wake interactions, three fully-waked wind turbines are considered, The machines are aligned with the flow, with their rotors pointing into the incoming wind and spaced among themselves of 4.1D. Experimental measurements are not available in this case.

The same moderate turbulence flow of section 5.6.1 is used even in this case. The first upstream machine operates in the same exact conditions of the isolated wind turbine case, while the two downstream machines are operated in closed by a pitch-torque controller, in order to adjust their operating points to the local incoming wind.

Fig. 5.8 shows velocity and turbulence intensity profiles at various distances from the rotor. Notice that the wind turbines are located at 0D, 4.1D and 8.2D. The two simulation models show nearly identical velocity profiles at all downstream positions. The average error $\langle\Delta(u_x)\rangle$ over all stations is less than 1%. The turbulence intensity profiles are also in a good agreement, with only small discrepancies at 11D.

Fig. 5.9 shows color plots of the normalized instantaneous streamwise velocity u_x/U_0, time-averaged streamwise velocity $\langle u_x\rangle/U_0$, vorticity magnitude $|\nabla\times\mathbf{u}|$ and turbulence intensity $\sigma/\langle u_x\rangle$ for LES and SAS on a vertical plane across the hub. Coherently with Fig. 5.8, the two turbulence models exhibit a nearly identical behavior for average velocity and turbulence intensity. The wake deficit behind the first wind turbine is accurately simulated by the SAS model, as already shown for the single turbine case. A correct modeling of the wake deficit behind the second and third wind turbine is more challenging, as wakes overlap and interact. In particular, the turbulence shed by the rotor enhances mixing between the wake and the free-stream flow. This effect has to be precisely resolved by the model for the correct wake deficit to be captured. For multiple interacting turbines in a row, as in the present case, errors or lack of accuracy in the modeling of these processes will propagate and amplify at downstream, eventually increasing the simulation uncertainty at the back of the cluster. However, velocity contours show a consistent behavior between LES and SAS even for the second and third wake.

A more pronounced difference between the two models is apparent for the vorticity contour plots. In fact, because of its higher mesh density, LES is capable of resolving finer vortical structures. This is also noticeable for the blade root vortices, which are significantly diffused for SAS. Nonetheless, these differences still appear to have only a negligible impact on the overall behavior of the wake.

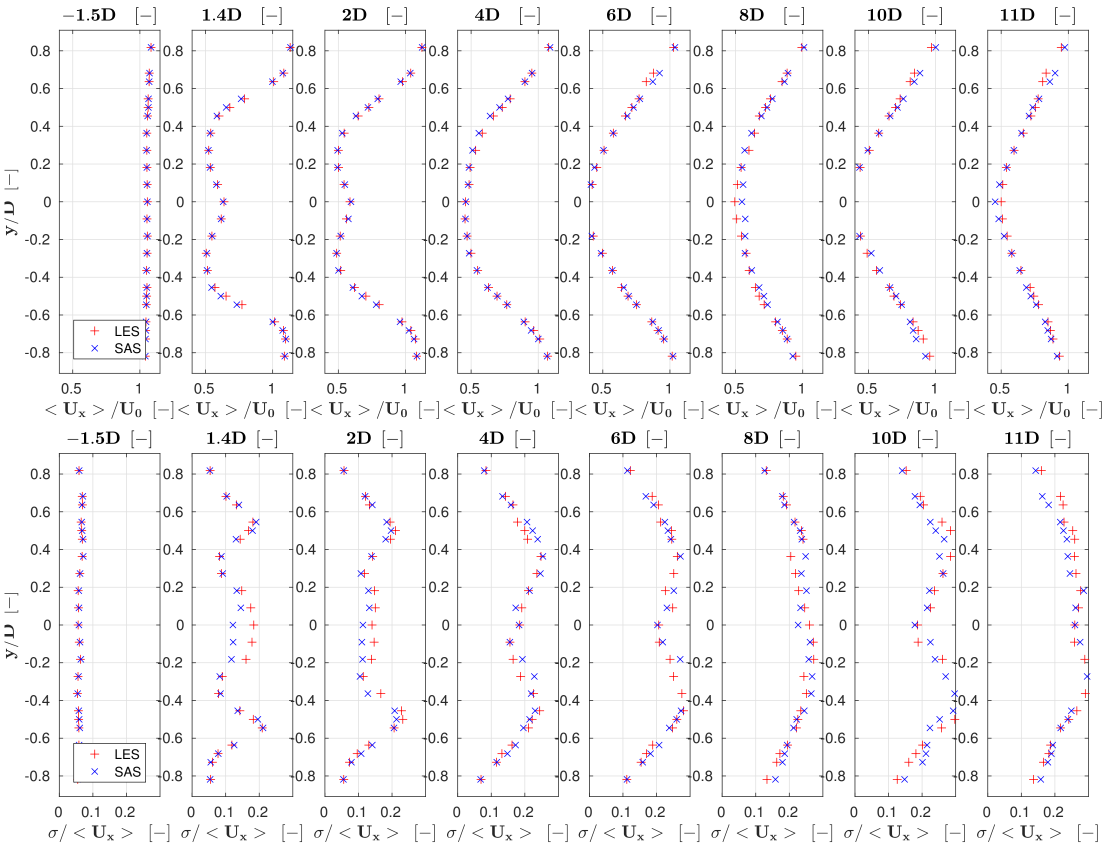

Figure 5.8: Normalized time-averaged stream-wise velocity (top) and turbulence intensity (bottom) profiles at hub height and at several positions. Experiment: black ∘ symbols; LES: red + symbols; SAS: blue × symbols.

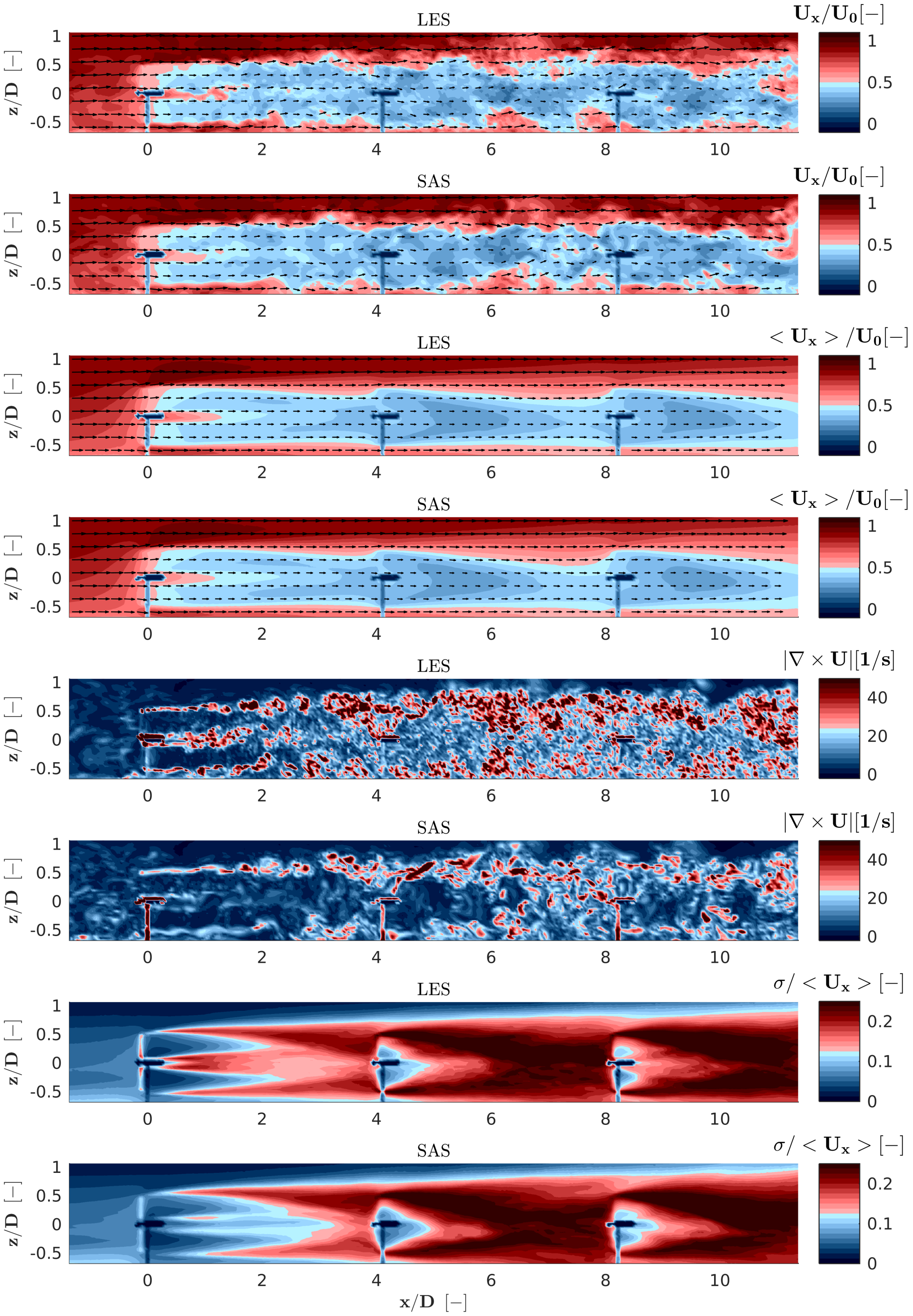

Figure 5.9: From top to bottom, normalized instantaneous streamwise velocity u_x/U_0, 60 s time-averaged streamwise velocity $\langle u_x \rangle/U_0$, vorticity magnitude $\|\nabla \times \mathbf{u}\|$ and turbulence intensity $\sigma/\langle u_x \rangle$, for both LES and SAS along a streamwise vertical plane across the hub.

CHAPTER 6

Floating Offshore Wind Turbine Framework

In this chapter, the author demonstrates the main features of a novel simulation framework for floating offshore wind turbine. The floating algorithm is developed based on an existing LES solver, which was modified to handle a prescribed rigid body motion of wind turbines. Rotor blades are modelled by lifting lines, while nacelle, tower and floating platform by an immersed boundary method. A complete restart capability allows breaking long simulations into shorter consecutive segments. The wind turbines are operated in a closed loop by a pitch and torque controller. The developed FOWT simulation framework is compared with wind tunnel experimental data, including inflow conditions and turbine response parameters.

6.1 FOWT experiment

FOWT experiments were performed with the scaled model of a square-shaped floating platform, equipped with four identical wind turbines located at its corners in a boundary wind tunnel. The description of FOWT wind tunnel tests follows the work of Campagnolo et al. [169].

6.1.1 Floating platform

The geometry of the floating platform is introduced in section 3.1.3. This section focuses on the design principles and characteristics of this floating platform, as well as the platform pitching motion mechanism. The model of the floating platform was conceived to satisfy two design requirements:

1. To allow single DOF oscillation, i.e. prescribed pitch motion (rotation along y_f, see Fig. 3.9 and Fig. 6.1) or alternatively roll, in order to simulate the simplified motion of floating offshore wind turbines.

2. To be sufficiently stiff in order to have natural frequencies higher than the 1st fore-aft and side-side frequencies of the nacelle and tower, which limits the influence of the platform flexibility on the wind turbine dynamics and acts similar to the full-scale model.

The platform model was built with two separated steel beams, each carrying two WTs at its ends, installed on two bearing-supported shafts, in turn, connected by joints to two Siemens synchronous motors containing a 2-stages 50:1 gear-head and Drive-clio encoder. The two motors can either be operated in a fully synchronized mode or independent mode since the two beams are mechanically decoupled. For the current applications, the whole structure behaves as a single floating platform.

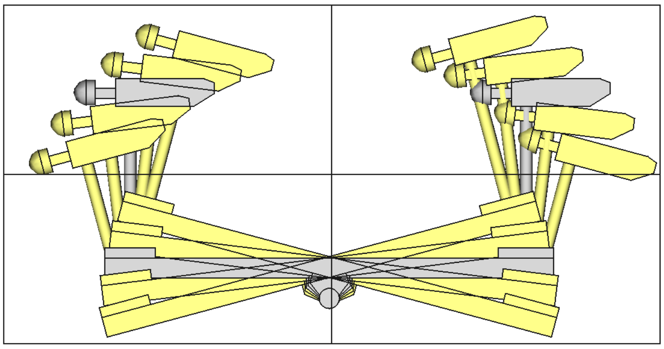

Figure 6.1: 1 DOF motion (oscillation along y_f) of the floating platform.

6.1.2 Prescribed pitching motion

The floating algorithm is introduced in section 2.2.9. The immersed boundary modelled floating platform uses a non-topological approach to achieve the pitching motion. The non-topological change allows a floating platform to move freely inside the domain without re-generation of mesh. However, the pitching motion still requires a proper way to describe the geometry movement.

Preliminary estimates provided by KRISO show that, for the specific configuration considered here, the dynamic effects of the wind turbines on the platform motion are negligible. This is because the ratio between platform and wind turbine inertia is significantly larger than one of the other semi-submersible systems. It is therefore expected that the floating platform dynamics will primarily be affected by forces induced by currents, waves and moorings lines [170]. In such a case, a simple one-way coupling exists between the floating platform and the wind turbines dynamics. In this regard, only the platform pitch rotation was prescribed in the experimental setup. In fact, of the six rigid body degrees of freedom of the platform, the pitching motion is the one that is expected to have a larger impact on the wind turbine aerodynamics. The same input files used for the experiments are prescribed for the floating platform simulations, which are *IB Transformation Input File FAST Transformation Input File* in Fig. 2.5.

A periodic sinusoidal pitch motion was prescribed to simplify the experimental implementation and to account for limitations of the hardware. The prescribed motion function used in this analysis equals to $\beta_P = 6.5\sin(2\pi t/T + c)$ deg. where 6.5 deg is the pitching amplitude, T is the oscillation period, and c is the phase angle. The prescribed motion is not scaled as the rest of the model and results in a higher amplitude and smaller ratio of frequency/rotor-speed than a full-scale system. Despite the prescribed platform motion not being properly scaled, the motion-induced wind speed changes with respect to the free stream at hub height are indeed correctly scaled. This ensures that the effects produced by the platform pitch motion on the turbine aerodynamics are indeed properly captured.

6.1.3 Inflow conditions

The FOWT simulation used a turbulence inflow map generated by precursor simulations. The turbulence inflow is a power-law wind profile simulated in the wind tunnel, similar to the one observed for a neutrally stratified atmosphere in open field. A pitot tube was used to measure time-averaged inflow speed and turbulence intensity 4D in front of an upstream wind turbine on the floating platform. The time-averaged velocity equals to 6 m/s, and turbulence intensity equals to 6%. On the other hand, the rated wind speed, rotor speed and power of the wind turbine models used in the experiment were 4.65 m/s, 650 rpm and 23.2 W. Since the rated wind speed for the wind turbine models in this experiment was 30% lower than the inflow speed (6 m/s), the turbine model (upstream), therefore, operated in region III. The pitching axis of the floating platform is orthogonal to the inflow direction, and the platform is located at 29 m downstream of the wind tunnel inlet.

6.1.4 Data analysis

In this section, the measurement data from the FOWT experiment is post-processed to analyze the rotor integral quantities and flow characteristics of a floating platform. The analysis helps to obtain a deep-understanding of the platform motion and its induced aerodynamic effects.

In Fig. 6.2, the platform pitch $\beta_P = 6.5\sin(2\pi t/T)$ deg induces a maximum horizontal velocity of the rotor center equal to approximately 3.1% of the undisturbed flow speed, as well as an upflow angle χ that varies between ±6.5 deg, where a positive inflow means that the rotor is impinged by a vertical wind pointing upward. It is important to highlight that the main contribution to the upflow is due to the rotor tilt angle, which periodically varies due to the platform motion. Moreover, despite the induced horizontal velocity is the same for both the machine, the upflow is different: when the upstream wind turbine is moving up, the downstream one is moving in the opposite direction, thus leading to opposed platform-induced vertical wind speeds.

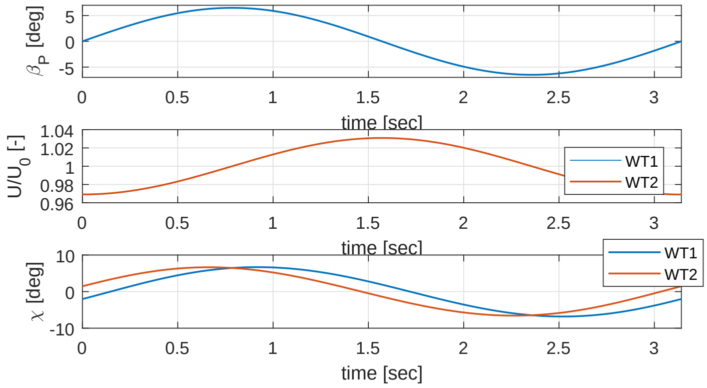

Figure 6.2: Platform pitch, induced horizontal wind speed and upflow angle χ plotted for a single period.

The collective blade pitch, rotor speed and bending moment are reported, for the pitching platform, in Fig. 6.3. Since the wind speed is higher than rated, the upstream machine is operating in region III and its pitch is automatically varied by the pitch controller in order to keep constant the rotational speed and the power output. The average values of the blade pitch (higher than the region II fine pitch plotted in the figure) and the rotor speed of the downwind wind turbine also highlight that the same is operating in region III. Black circles in the plot indicate the time instants when the downwind machine enters into region II. By looking at the blade pitch time evolution, it is possible to observe that the machine is operating occasionally in region II, which seems to take place with quite a regular pattern, despite it is not periodic.

Such a behavior was not observed for an experiment where the inflow speed was the same (6.5 m/s) and with the floating platform kept fixed. Indeed, Fig. 6.4 shows that both the wind turbines are normally operating in region III and only rarely the downstream one experiences, due to turbulence, a wind speed lower than rated. But the lower blade pitch on average, with respect to the upstream turbine, does show the effects of the wake impingement

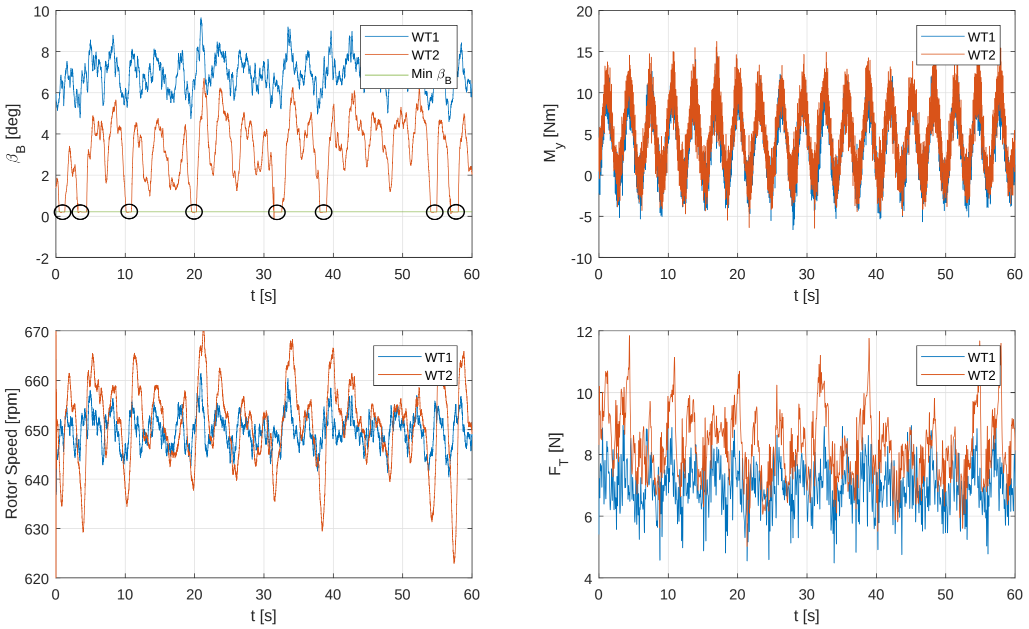

Figure 6.3: Blade collective pitch (top left, black circles indicate the time instants when the machine enters into region II), rotor speed (top right), tower base fore-aft bending moment (bottom left) and rotor thrust measured with the pitching platform (bottom right)

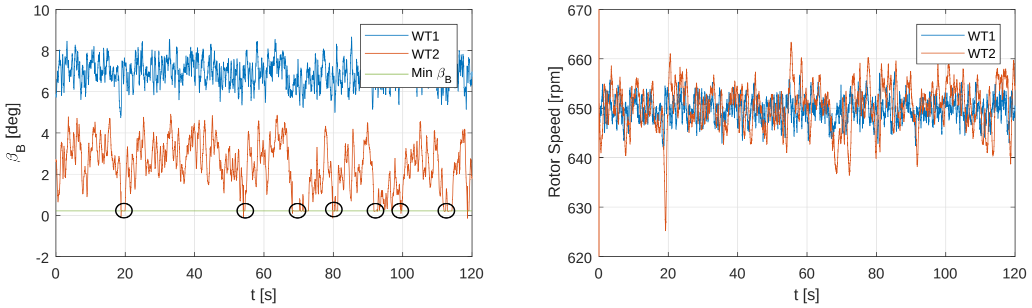

Figure 6.4: Blade collective pitch (left), rotor speed (right) measured with the fixed platform

Next, the dynamic behavior of the upstream wind turbine is analyzed. Since the platform pitch variation ($\Omega_{p,max}$=2 rad/s) is quite slow with respect to the rotational speed of the wind turbine (Ω_R=68 rad/s), it is reasonable to assume that the pitch controller is capable of quickly adjusting the pitch of the blade in order to keep the rotational speed of the wind turbine constant, and equalize the rated pitch. This implies that the wind turbine will experience a variation of its λ only due to the horizontal velocity (± 3.1%) generated by the pitching motion. In details, it is expected that the λ of the wind turbine will vary from a minimum of $\lambda_{min} = \dfrac{\Omega_R R}{1.031U_0}$ to a maximum of $\lambda_{max} = \dfrac{\Omega_R R}{0.969U_0}$, with an average of $\lambda_{avrg} = \dfrac{\Omega_R R}{U_0}$. This λ variation is reported in Fig. 6.5 (red crosses) together with the $C_P(\lambda, \beta_B)$ curves of the rotor (derived from the BEM model that makes use of identified polars) and the regulation trajectory that the wind turbine is expected to follow in region III. Note that the region II operating point of the model has been on purpose selected to be different from the one of maximum C_P, but rather the one that provides the maximum C_P at λ roughly equals to 8.15. The plot depicted in the figure shows that the blade pitch should vary of roughly 0.6 deg around an average of roughly 7.5 deg.

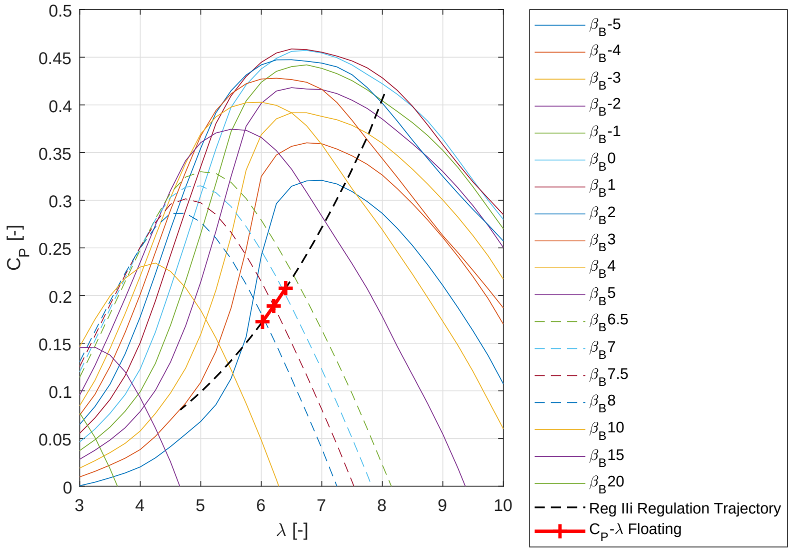

Figure 6.5: $C_P(\lambda, \beta_B)$ curves of the G1 model together with the region III regulation trajectory and the three expected power coefficients (red crosses) for λ_{min}, λ_{avrg} and λ_{max} due to the platform-induced longitudinal speed

These findings show good agreement with the pitch time history (Fig. 6.3) and the frequency spectra of the pitch signals, reported in Fig. 6.6. The data depicted in the figures, indeed, highlight that the measured average pitch is approximately 6.95 deg, while the amplitude of the blade pitch variation at 2 rad/s is equal to approximately 0.78 deg. The difference between the expected mean blade pitch, based on BEM computation, and the mean measured one is probably due to minor mismatches between the plant and the BEM model, or either to uncertainties in the wind speed measurements. The small uncertainties between the estimated and measured amplitude of the pitch variation at 2 rad/s could be due to the fact that, during the experiment, the height of the rotor center for each wind turbine with respect to the ground changed of approximately ±0.1 m. Such an effect could lead to a higher variation of the wind speed experienced by the rotor due to the presence of a vertical wind profile (simulated with spires within the wind tunnel).

Similar to the analysis of rotor power coefficient, the expected variation of the thrust coefficient due to the combined effects of platform motion and the wind turbine controller can be obtained by looking at $C_T(\lambda, \beta_B)$ curves and the regulation trajectory that the wind turbine is expected to follow in region III. The curves depicted in Fig. 6.7 show that the thrust

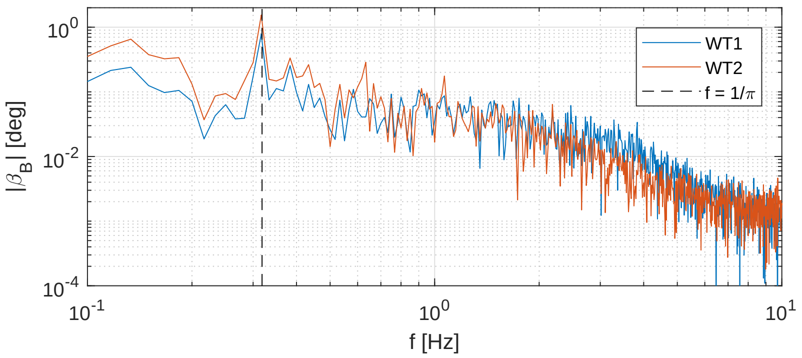

Figure 6.6: Frequency spectra of the collective pitch for the upstream and downstream wind turbine

coefficient should have an average value of about 0.313 and varies between a minimum of 0.285, experienced when the rotor effective speed is at its maximum due to the platform-induced longitudinal speed, and a maximum of 0.345, experienced when the rotor effective speed is at its minimum. A thrust varying between 6.22 and 6.65 N is therefore obtained, with an average value of approximately 6.42 N.

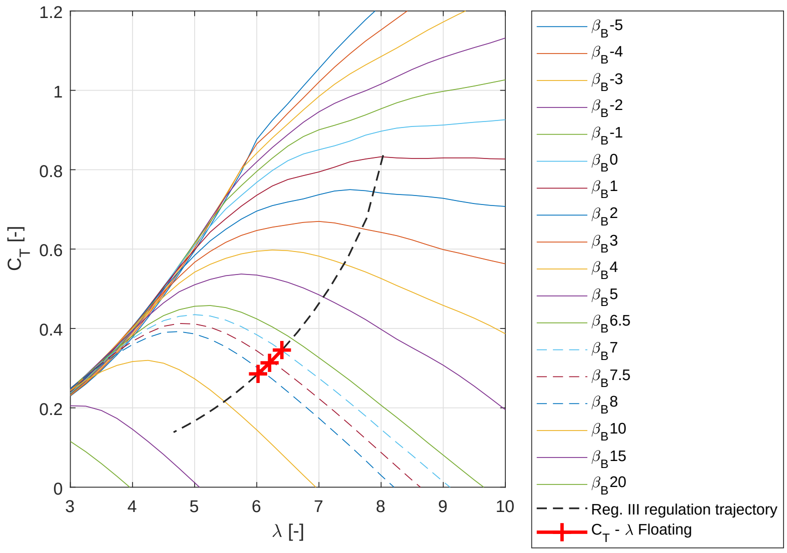

Figure 6.7: $C_T(\lambda, \beta_B)$ curves of the G1 model together with the region III regulation trajectory and the three expected thrust coefficients (red crosses) for λ_{min}, λ_{avrg} and λ_{max} due to the platform-induced longitudinal speed

These findings again show good agreement with the experimental observations, where the time-averaged thrust equals to 6.95 N. Moreover, the frequency spectra of the thrust signals, reported in Fig. 6.8, depicts that an amplitude of the thrust variation at 2 rad/s equals to approx. 0.55 N. The mismatch observed between the measurements and experimental data could be again due to the same reasons given for the mismatch of the blade pitch.

The dynamic behavior of the downstream wind turbine is quite different with respect to

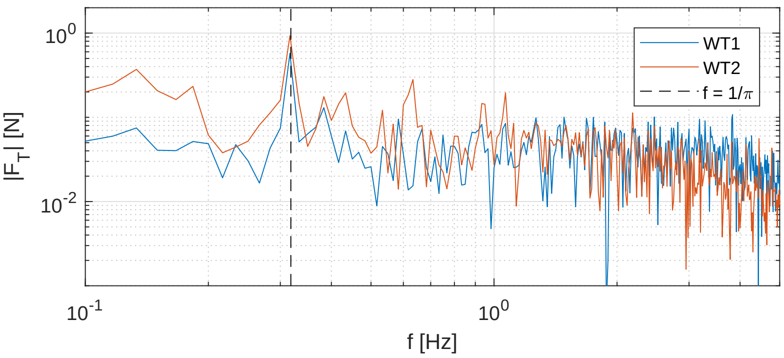

Figure 6.8: Frequency spectra of the rotor thrust for the upstream and downstream wind turbine

the one of the upstream machine. Indeed, the turbine occasionally operates in region II (see Fig. 6.3), despite the average values of the blade pitch (higher than the region II fine pitch) and rotor speed (equal to rated) indicate that the turbine is, on average, operating in region III. Moreover, the blade pitch variation associated with the platform movement (amplitude of the spectra of the blade pitch at 2 rad/s) is higher than the one observed for the upstream machine (1.55 deg instead of 0.75 deg). This can be only explained by a stronger fluctuation of the flow speed that impinges the downstream rotor. This is again due to the platform pitch movement which not only induces variable longitudinal speed on the downstream rotor but also affects the wake shed of the upstream wind turbine, whose deficit also varies due to the unsteadiness of the upstream rotor thrust coefficient (see Fig. 6.7). The way these two terms sum up together is driven by the time delay associated to the wake advection, which in turn depends on the wake speed and the longitudinal distance between the machines. By using the well-known relationship between the thrust coefficient and the axial induction factor of the axial-momentum theory, it is possible to estimate an average value of the axial induction factor $a = \dfrac{1-\sqrt{1-C_T}}{2} = 0.086$, i.e. a speed in the wake of approximately $U_0(1-2a) = 4.97$ m/s, where any wake recover has been reasonably neglected considering the very close distance among the turbines. Similar computations (and accounting for the thrust coefficient variation) lead to a variation of the speed in the wake of roughly ±0.1 m/s. This information enables an approximate estimation of wake advection time, which equals approximately 0.3 s, i.e. almost 10% of the platform pitch oscillating period. This means that the flow speed variation felt by the downstream rotor due to the wake unsteadiness (±0.1 m/s) and the platform-induced longitudinal speed (±0.19 m/s) are almost synchronized. This, in turn, implies that the sum of the two terms leads to a stronger fluctuation, in the order of plus-minus 0.28 m/s (i.e. roughly 5.6% of the mean wake speed), of the flow speed that impinges the downstream rotor, as shown in Fig. 6.9. This finding qualitatively agrees with the experimental observations: a variation of the blade pitch of the downstream machine associated with the platform movement that is almost the double of the one observed for the first one.

The aforementioned results also agree with the time evolution of the collective blade pitch of the second turbine plotted together with the platform pitch (see Fig. 6.10). The curves depicted in the figure, indeed, show that the wind turbine occasionally starts operating in region II when the pitch of the platform gets null and its pitch rate is positive and maximum, a condition that induces the highest negative longitudinal speed.

It should be addressed that the performed data analysis, based on physical considerations

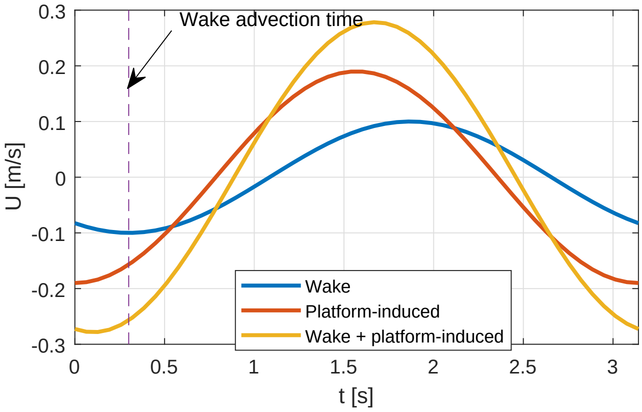

Figure 6.9: Variation (plotted for one period) of the speed that impinges the downstream rotor (orange line). The figure also reports the variation of the speed in the wake shed by the upstream wind turbine (blue line) and the platform-induced longitudinal speed variation (red line)

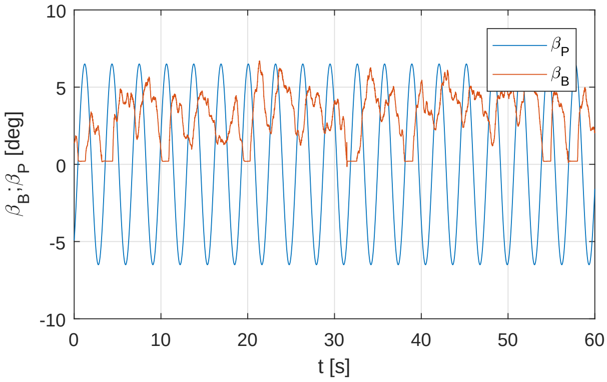

Figure 6.10: Collective blade pitch β_B of the downstream wind turbine plotted together with the platform pitch angle β_P

and accounting for the effects induced by the wind turbine control, allows only to qualitatively discuss the results and to partially explain the physics that governs the problem at hand. Some phenomena, e.g. the effects of the variable upflow angle on the wake shed by the upstream wind turbine or the effect of the downstream wind turbine on the wake itself, were not addressed and their proper understanding should be properly studied by means of dedicated CFD tools.

6.2 Comparison for FOWT framework

The velocity contours of a floating platform at maximum and minimum pitch are shown in this section to present that the framework is capable of simulating complex wake interaction induced by the platform pitching motion. The comparison of downstream rotor thrust and blade pitching is presented next. However, since it is computationally expensive to run a complete 60 s FOWT simulation, only 16 s of simulation results were obtained from a number of consecutive simulation runs. The simulation results were therefore too short to analyze the frequency spectra of quantities.

6.2.1 Flow properties

Fig. 6.11 shows instantaneous streamwise velocity (left) and vorticity (right) contours. The top part of the figure corresponds to a case where the platform is held fixed at the null pitch. The pictures below, on the other hand, correspond to the prescribed pitching case, the one in the centre representing an instant in time when the platform is pitched forward (platform pitch = -6.5 deg, induced normalized horizontal velocity = 0, upflow angle of upstream WT = -6.5 deg, $\lambda \approx 6.2$, $C_P \approx 0.19$, see Fig. 6.2 and Fig. 6.5), and the one on the bottom when the platform is pitched backward (platform pitch = 6.5 deg, induced normalized horizontal velocity = 0, upflow angle of upstream WT = 6.5 deg, $\lambda \approx 6.2$, $C_P \approx 0.19$, see Fig. 6.2 and Fig. 6.5. The pitching oscillation period of the simulation is π s, which equals to the period prescribed in the experiment. Since the inflow speed of the simulation is very close to the value measured in the experiment, the induced horizontal velocity and the rotor upflow angle of the simulation are similar to the experimental values. A complex vertical wake interaction is observed in the pitching case (center and bottom), with a strong influence on the downstream machine and the overall evolution of the combined wakes. Further analysis of the results is required to verify whether or not the pitching motion —and its induced wake interaction— influences wake recovery.

A distinguishing characteristic of this and similar systems is that there is not only a complex interaction between the atmospheric flow and the wind turbines but also among the wind turbines themselves. Indeed, wakes shed by the upwind wind turbines affect power and loads of downstream machines. In this sense, wakes represent a major form of coupling within the power plant. Besides, the platform dynamics will affect the wind turbine response, modifying performance and loads.

6.2.2 Rotor thrust

In the experiments, the bending moment at tower base is measured. Fig. 6.12 depicts the system geometry, with the purpose of reconstructing the thrust F_T from the bending moment M_y, measured close to the tower base by strain gauges located at point o_s. Given the moving frame (x_s, z_s) attached to the tower, the equilibrium of moments around point o_s can be written

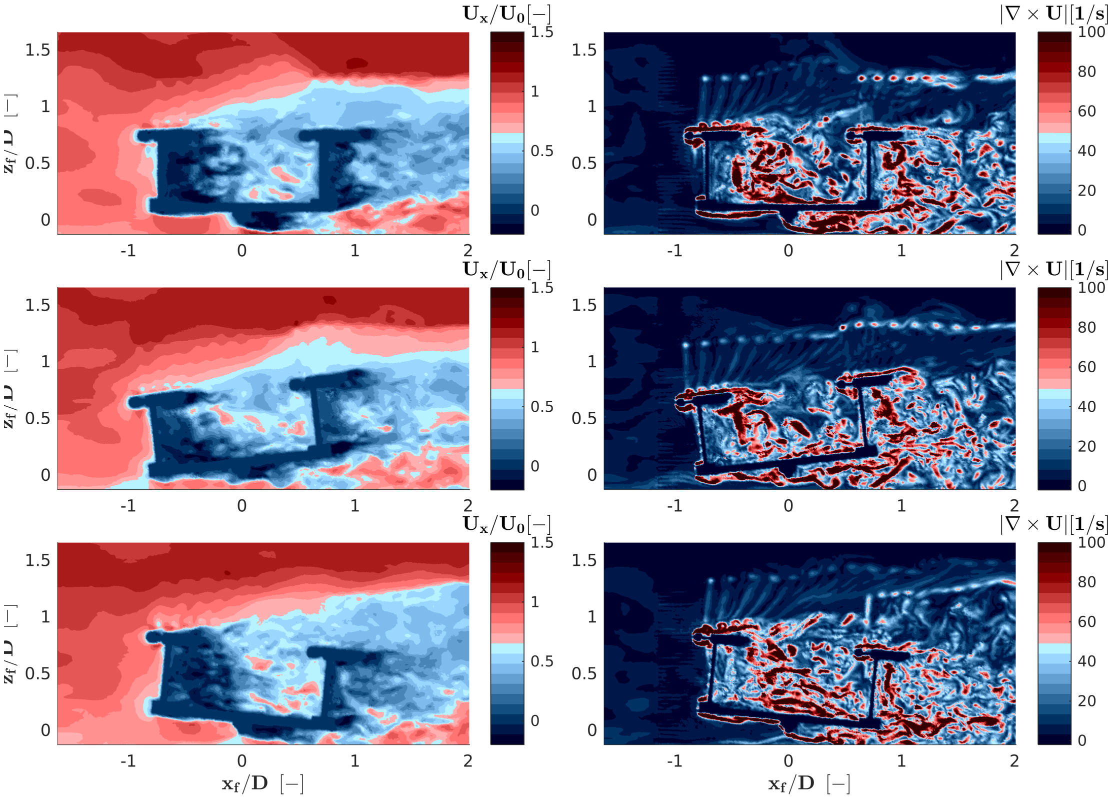

Figure 6.11: Normalized streamwise velocity (left) and vorticity (right) snapshots on the $x_f z_f$ plane through the turbine hubs. top: fixed platform case; center and bottom: pitching case at the maximum and minimum pitch angle.

as:

$$ma_{x,f}L_{cs}cos(\theta_{to}-\beta_P)+m\dot{\Omega}L_{cs}^2+mgL_{cs}(sin(\theta_{to}-\beta_P))+ma_{zf}L_{cs}sin(\theta_{to}-\beta_P) = M_y+F_T H_{hs}, \tag{6.1}$$

where $a_{Os,xf}$ and $a_{Os,zf}$ are, respectively, the horizontal and vertical accelerations of point o_s, while $\dot{\Omega}$ is the angular acceleration of the platform, β_P the platform pitch angle, while the other geometrical quantities are defined in the same figure. Since (x_s, z_s) is a non-inertial frame, one has to consider inertial forces applied at the center of mass o_m of the portion of wind turbine located above the load sensors. $F_c = \Omega^2 L_{cs} m$ is the centrifugal force due to the fact that the non-inertial frame (x_s, z_s) has an angular velocity equal to Ω, which however does not contribute to the moment equation since it passes through o_s (lever arm equals to 0); forces $ma_{Os,xf}$ and $ma_{Os,zf}$ are produced by the translational accelerations of o_s, while $m\dot{\Omega}L_{cs}$ is due to the angular acceleration. Finally, m is the mass of the portion of wind turbine model located above the strain gauge. Coriolis forces are neglected, since only minor oscillations of the model center of gravity are expected due to tower vibrations.

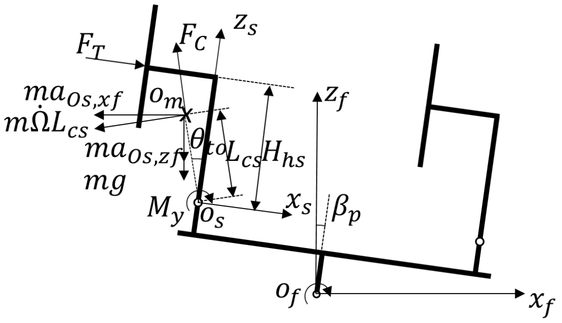

Figure 6.12: Sketch of the model geometry for the calculation of thrust from tower base bending moment.

Fig. 6.13 shows the numerical and experimental time histories of the blade pitch angle $\beta_{B,i}$ and rotor thrust F_T of the downstream machine, as well as the platform pitch angle β_P. While the pitch motion of the platform is fully known in the simulation, the turbulent wind can only be matched in a statistical sense. Therefore, one should not expect to observe an exact correspondence between simulation results and experimental measurements. Moreover, due to the relatively high computational cost of the CFD analysis, the current simulation only lasts for about 16 sec of physical time.

In both the simulation and experiment, the wind turbines are governed by a collective pitch and torque controller. Since the machines operate in a full power regime, the goal of the controller is to keep the rotor speed constant, compensating for the fluctuations due to pitching-induced wind changes and wind turbulence. Indeed, the power of the downstream machine in the simulation and experiment are both around the desired power output, i.e. 23.2 W. The blade pitch angle of the simulation shows satisfactory accordance, in terms of its magnitude and variation. The periodic change of the blade pitch angle can be explained as follows: when the platform pitches nose down, i.e., the downstream wind turbine is moving towards the incoming wind, the blade pitch angle increases to compensate for the increased relative inflow speed, and vice-versa. In addition, the simulation thrust shows good agreement with the experimental thrust. The mean thrust of the downstream wind turbine is equal to 9.6 N for the simulation and to 8.7 N for the experiment, while their standard deviations are

0.88 N and 1.05 N, respectively. The accurate estimation of the downstream rotor thrust shows that the developed framework is capable of simulating platform pitch induced wake motion and the relevant wake interaction. Nonetheless, it is still necessary to improve the simulation accuracy and run longer simulations in order to evaluate the frequency spectra of rotor thrust and other higher order quantities.

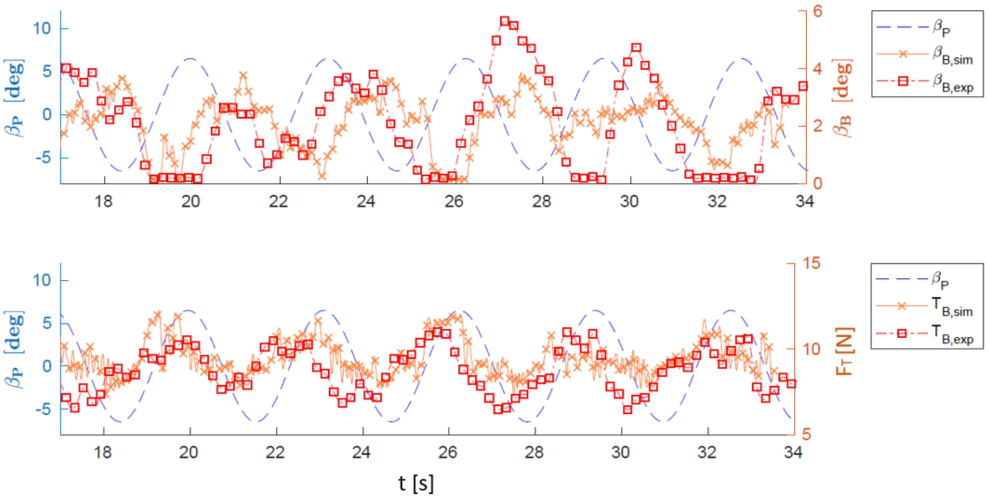

Figure 6.13: Comparison of numerical and experimental results for the downstream machine: blade pitch angle $\beta_{B,1}$ (top) and rotor thrust F_T (bottom).

CHAPTER 7

Conclusions and outlook

7.1 Conclusions

This thesis has described high-fidelity numerical approaches for the simulation of wind turbine wakes. A simulation framework was used to simulate scaled wind tunnel experiments. This framework contains several numerical approaches to improve the simulation accuracy.

a) An immersed boundary method, categorized to the discrete forcing approach, is used to model nacelle and tower.

b) A low-cost scale-adaptive approach, an alternative to large-eddy simulation, is used as a sub-grid scale model to reduce computational costs significantly.

c) A floating algorithm is implemented to enable FOWT simulations.

d) The simulation of one experiment is divided into two steps: a precursor first generates a sheared and turbulent flow, whose outflow becomes the inflow for a successor that simulates the wind turbine and its wake. A body-conformed mesh approach is used for the precursor domain to generate a high-quality turbulent inflow for wake simulations.

e) An integral approach for the velocity sampling is applied to the actuator line method.

f) An identification method is used for the calibration of airfoil polars, accounting for the low and variable Reynolds at which the scaled wind turbines operate.

g) The simulation model operates with the same pitch-torque controller in the loop used on board the experimental model for the increased realism of simulation results.

In addition, the Gamma-bounded differencing scheme is used in conjunction with the IB formulation to limit numerical dispersion and diffusion. A low Courant number and high-quality meshes are used to speed up simulations, which in fact are run without sophisticated preconditioners or non-orthogonal grid correctors. A few decomposition and compiling techniques are also investigated to improve the runtime performance.
comparison for the framework consists of three parts.

a) Bottom fixed wind turbine large-eddy simulation

   A low turbulence normal-operation problem is considered first, showing that simulations are in good agreement with experiments, both in terms of rotor quantities (thrust and power) and wake behavior. Next, the three wake control strategies of power derating, wake steering by yaw misalignment and wake enhanced recovery by cyclic pitch control are studied. Results show a good agreement of simulations with experiments for both derating and yaw misalignment. The wake turbulence intensity shows some discrepancies, which were here attributed to a lack of refinement of the grid that in turn affects the breakdown of near wake vortical structures. Slightly less accurate results are obtained for cyclic pitching, possibly due to un-modeled unsteady airfoil aerodynamics.

The thesis continues by considering a moderately turbulent wind. The characteristics of the simulated turbulent flow are in good agreement with measurements. The average streamwise velocity is within 1% of the experiments, the average turbulence intensity within 5-7%, while the turbulent kinetic energy spectrum and integral time scale also exhibit a good matching. The wake characteristics are in very good agreement with the experiments, since tip vortices break down earlier than in the low turbulence condition, relaxing a bit the need for very dense grids in near wake region. The use of a controller in the loop leads to a more realistic response of the model turbine to the turbulent flow, which is important in case the load response of the machine is of interest. Remarkably, the model in the loop also operates at essentially the same rotor speed as the experiment, which demonstrates the overall fidelity of the digital model to the experimental one.

b) Bottom fixed wind turbine scale-adaptive simulation

An isolated wind turbine has been considered in a low turbulence inflow, to serve as baseline test case for the tuning of the simulation environment. Next, other test cases have been studied, considering inflows of increasing turbulence, cases where one machine is misaligned with respect to the main flow direction, and a case with three aligned and fully waked wind turbines.

The analysis of SAS draws the following conclusions:

- In general, SAS appears to achieve a good agreement with both LES and experimental measurements in terms of time-averaged rotor integral quantities and velocity profiles. Results are not only qualitatively very similar, but also quantitative differences are typically limited to a few percentage points.
- In low turbulence conditions, blade tip-vortices appear not to be properly resolved by SAS. It was found that this difference is not caused by the different modeling of turbulence between the two approaches, but it is in fact driven by the coarser mesh used by SAS. Because of the smearing of tip vortices, there is a defect of turbulence intensity in the near wake region for SAS, which in turn causes some small differences in far wake recovery. Although this problem is quite apparent in the low turbulence inflow conditions, it is less severe in higher ambient turbulence cases.
- LES requires on average roughly 8 times larger meshes and 13 times longer CPU time than SAS.

As a general remark, one should keep in mind that moving from LES to SAS does not only entail a different turbulence model but also implies the choice of denser grids. While this is exactly the reason that might motivate the use of SAS, one should remember that coarser grids will also imply effects that are not directly related to turbulence, as for example the accuracy of lifting line calculations (both in terms of angle of attack computation and projection of the aerodynamic forces back onto the fluid domain), the mesh resolution around nacelle and tower, and the numerical diffusion of tip vortices.

Given that differences in results appear to be quite limited, at least in the conditions examined here, the significant difference in computational burden between the two methods indicates that SAS could be an interesting alternative to LES, at least in turbulent conditions. In this sense, one could use SAS for tuning of wind plant control laws, repetitive runs or other tasks that do not require extremely accurate results, leaving LES for the final higher-accuracy runs.

c) Floating wind turbine large-eddy simulation

Data from a scaled FOWT experiment was used to compare with the simulation model. The computational setup of the simulation strictly adheres to the experiment. A turbulent and sheared flow is passively generated in the experiment with spires placed at the tunnel inlet. The same process of turbulence generation was realized in the digital model, using an ad-hoc precursor simulation.

Results show that FOWT pitching introduces a vertical wake meandering, which affects the downstream machine. Both simulation and experimental results clearly indicate this effect and show the compensation performed by a closed-loop controller to keep a steady power output.

Given the small time length of numerical simulations performed so far, only a preliminary and somewhat qualitative comparison between simulation and experiment has been possible. Nonetheless, the simulation model seems to be capable of reproducing at least some of the characteristics of the response observed in the experiments.

In addition to the validation work, the developed simulation framework is used to investigate the wake redirection methods. The author used a three-pronged approach to study the wake redirection. A simplified analytical model was developed to gain some insight on the main parameters affecting this phenomenon. A CFD simulation model was used for a more refined representation of the complex dynamics of the fluid flow and its interaction with the wind turbine. Finally, scaled models were tested in a boundary layer wind tunnel, obtaining direct measurements of performance and operational parameters.

The main conclusions regarding the wake redirection research are as follows. First, YM is more effective than CyPC in displacing the wake laterally, because it can generate a significant lateral force by tilting the thrust with respect to the incoming wind, which in turn generates a lateral component of the flow velocity. The amount of wake displacement is roughly linear with the misalignment angle (for small angles) and largely independent of design and operational parameters. In this sense, most rotors can be expected to produce similar non-dimensional wake displacements when yawed away from the wind. Third, CyPC, by tilting the axial inflow plane, generates a high nodding moment on the machine and a faster wake recovery due to the enhanced mixing.

7.2 Outlook

In continuation of this Ph.D. study, work is progressing on multiple fronts. On the CFD side, several things need to be investigated.

a) A mesh independence study for LES and SAS under turbulence inflow condition.

   The original mesh independence study is only limited to the low turbulence condition. The same mesh resolution used for the low turbulence case is also applied to moderate and high turbulence simulations. However, recent studies have indicated the potential of adopting coarser grid for both large-eddy simulation and scale-adaptive simulation. Indeed, the immersed boundary surfaces may not be identified by mesh cells if a coarser SAS grid is eventually used. Therefore, mapping of IB surfaces to the fluid domain is not needed. The computation cost is reduced not only due to the mesh size, but also the removing of immersed boundary surfaces.

b) Full parallelization of the immersed boundary method.

   Due to current limitations in the parallelization algorithm, decomposed domains are adjusted to ensure that immersed boundary surfaces do not span multiple processors. However, such user-defined decomposed approach enforces the mesh domain decomposed under sub-optimal conditions. Sometimes, the parallel efficiency drops more than 40%, if the immersed boundary surface, and consequently the decomposed domain containing this IB surface, is too large. In this sense, a full parallelization of the IB method is necessary for obtaining the optimal decomposed domain, so as to avoid consuming additional computational resources.

c) Multi-airfoil polar identification

   The current blade airfoil polar shows reasonable agreements between simulations and experiments in this thesis except for the power derating and cyclic pitch control. The lift and drag coefficients at each Reynolds numbers need to be reidentified using more advanced identification methods. Moreover, a dynamic stall model needs to be implemented to the code to adopt large angles of attack.

d) Adaptive local mesh refinement

   As mentioned several times in the thesis, several mesh regions in a wake simulation need to be refined, i.e., the blade tip and root vortices and the flow field close to nacelle and tower. The adaptive local mesh refinement is a numerical method for self-detecting and refining local mesh based on user-defined criteria, for instance, the large velocity gradient. This way, a simulation model is capable of refining only the crucial areas, improving simulation accuracy and reducing computational costs.

On the simulation side, several things also need to be investigated.

a) Framework validation

   Results shown in this work indicate that the present LES-ALM approach is a viable way of simulating scaled wind tunnel experiments. These encouraging results motivate and justify the application of the same model to the analysis of clusters of wake-interacting wind turbines, for which we have gathered an ample collection of data sets in multiple operating conditions. Hopefully, this will lead to a better understanding of wake behavior, which is of crucial importance for the design and operation of wind turbines and wind power plants. The final validation of the present and similar simulation approaches can undoubtedly benefit from the use of scaled wind tunnel experiments, as attempted in this Ph.D. study.

b) Floating wind turbine

Longer simulations will be performed, which will enable more decent comparison of the results. The response of the machine in terms of mean and standard deviation of power, rotor speed, pitch, and torque activity will be validated. The spectrum analysis of power and rotor thrust will also be performed to clarify the frequency content of the simulation results. In addition, the effect of platform pitching on wake recovery will be investigated. Different prescribed motions should be tested to better understand the resulting complex turbine to turbine interactions, as well as to back-scale the achieved results to a full-scale floating system.

a) Full-scale wind turbines simulation

The scale wind turbine research is considered as an intermediate step towards their application to the full-scale case. The main discrepancies between the wind turbine model at two scales are the airfoil polar, the aero-servo-elasticity of real wind turbine models, the turbulence inflow generation and the Coriolis force. In principle, the identified airfoil polar is more accurate at the full scale than the small scale due to the Reynolds number effects. The full-scale simulation usually requires a wall model to generate turbulence boundary layer instead of utilizing turbulence generating spires. Several research institutes [14, 21, 22] have developed suitable turbulence boundary layer models for full-scale simulations. These models contain the capabilities of modelling thermal stratification, wind veer, terrain effects, and surface roughness, etc. The Coriolis force is also implemented in the boundary layer models to improve the fidelity of modelling for large-size wind farms. In fact, the original SOWFA framework is designed for full-scale simulations, which provides a set of feasible methodologies for generating the high fidelity atmospheric boundary layer. Nonetheless, the aero-servo-elasticity of full-scale wind turbine models remains to be an issue for further investigations. Although flexible turbine blades may have very modest effects on the downstream wake, the fluid-structure interaction of blades and past fluid needs to be carefully examined. In addition, among those modelling capabilities, a faster convergence method of the stable thermal stratification may still be an issue and needs to be investigated. A more advanced wall model for complex terrain simulations is also an important topic for full-scale simulation.

On the other hand, the knowledge obtained from the scaled wind turbines simulation can be used to simulate full-scale machines. For instance, the immersed boundary modelled geometry, the velocity sampling method and, more importantly, the SAS subgrid scale model. The SAS allows for a significant reduction of computational costs, which is particularly attractive to simulate a large size wind plant. Therefore, the value of this Ph.D. work goes beyond the scaled cases that the author considered.

CHAPTER 8

Dissemination

8.1 Journal papers

- Wang, J., Wang, C., Campagnolo, F. and Bottasso, C.L. A Large-Eddy Simulation Approach for Wind Turbine Wakes and its Verification with Wind Tunnel Measurements. In Journal of Wind Energy Science (Under review).

- Wang, J., Wang, C., Campagnolo, F. and Bottasso, C.L. Scale-adaptive simulation of wind turbines, and its verification with respect to wind tunnel measurements. In Journal of Wind Energy Science (Under review).

8.2 Peer-reviewed conference papers

- Wang, J., McLean, D., Campagnolo, F., Yu, T. and Bottasso, C.L., 2017, May. Large-eddy simulation of waked turbines in a scaled wind farm facility. In Journal of Physics: Conference Series (Vol. 854, No. 1, p. 012047). IOP Publishing. DOI: 10.1088/1742-6596/854/1/012047.

- Wang, J., Foley, S., Nanos, E.M., Yu, T., Campagnolo, F., Bottasso, C.L., Zanotti, A. and Croce, A., 2017, May. Numerical and Experimental Study of Wake Redirection Techniques in a Boundary Layer Wind Tunnel. In Journal of Physics: Conference Series (Vol. 854, No. 1, p. 012048). IOP Publishing. DOI: 10.1088/1742-6596/854/1/012048.

- Wang, J., Liu, Y., Campagnolo, F. and Bottasso, C.L., 2018. Verification of a Fast Scale-Adaptive CFD Formulation for Waked Wind Turbines. In 2018 Wind Energy Symposium (p. 0251). AIAA SciTech Forum. DOI: 10.2514/6.2018-0251.

- Wang, J., Wang, C., Castañeda, O D., Campagnolo, F. and Bottasso, C.L., 2018, June. Large-eddy simulation of scaled floating wind turbines in a boundary layer wind tunnel. In Journal of Physics: Conference Series (Under review).

Bibliography

[1] C. B. Hasager, N. G. Nygaard, P. J. Volker, I. Karagali, S. J. Andersen, and J. Badger, “Wind farm wake: The 2016 horns rev photo case,” *Energies*, vol. 10, no. 3, p. 317, 2017.

[2] T. Knudsen, T. Bak, , and M. Svenstrup, “Survey of wind farm control—power and fatigue optimization,” *Wind Energy*, vol. 18, p. 1333–1351, 2015.

[3] P. A. Fleming, P. M. Gebraad, S. Lee, J.-W. van Wingerden, K. Johnson, M. Churchfield, J. Michalakes, P. Spalart, and P. Moriarty, “Evaluating techniques for redirecting turbine wakes using sowfa,” *Renewable Energy*, vol. 70, pp. 211–218, 2014.

[4] P. Fleming, A. Scholbrock, A. Jehu, S. Davoust, E. Osler, A. D. Wright, and A. Clifton, “Field-test results using a nacelle-mounted lidar for improving wind turbine power capture by reducing yaw misalignment,” in *Journal of Physics: Conference Series*, vol. 524, no. 1. IOP Publishing, 2014, p. 012002.

[5] K. S. Hansen, R. J. Barthelmie, L. E. Jensen, and A. Sommer, “The impact of turbulence intensity and atmospheric stability on power deficits due to wind turbine wakes at horns rev wind farm,” *Wind Energy*, vol. 15, no. 1, pp. 183–196, 2012.

[6] R. J. Barthelmie, S. T. Frandsen, M. Nielsen, S. Pryor, P.-E. Rethore, and H. E. Jørgensen, “Modelling and measurements of power losses and turbulence intensity in wind turbine wakes at middelgrunden offshore wind farm,” *Wind Energy*, vol. 10, no. 6, pp. 517–528, 2007.

[7] R. J. Barthelmie, K. Hansen, S. T. Frandsen, O. Rathmann, J. Schepers, W. Schlez, J. Phillips, K. Rados, A. Zervos, E. Politis *et al.*, “Modelling and measuring flow and wind turbine wakes in large wind farms offshore,” *Wind Energy*, vol. 12, no. 5, pp. 431–444, 2009.

[8] J. Bartl, F. Pierella, and L. Sætrana, “Wake measurements behind an array of two model wind turbines,” *Energy Procedia*, vol. 24, pp. 305–312, 2012.

[9] 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.

[10] 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.

[11] R. J. Barthelmie, G. Larsen, S. Frandsen, L. Folkerts, K. Rados, S. Pryor, B. Lange, and G. Schepers, “Comparison of wake model simulations with offshore wind turbine wake profiles measured by sodar,” *Journal of atmospheric and oceanic technology*, vol. 23, no. 7, pp. 888–901, 2006.

[12] M. Gaumond, P.-E. Réthoré, S. Ott, A. Pena, A. Bechmann, and K. S. Hansen, "Evaluation of the wind direction uncertainty and its impact on wake modeling at the horns rev offshore wind farm," *Wind Energy*, vol. 17, no. 8, pp. 1169–1178, 2014.

[13] J. M. Kennedy, B. Fox, T. Littler, and D. Flynn, "Validation of fixed speed induction generator models for inertial response using wind farm measurements," *IEEE Transactions on Power Systems*, vol. 26, no. 3, pp. 1454–1461, 2011.

[14] F. Porté-Agel, Y.-T. Wu, H. Lu, and R. J. Conzemius, "Large-eddy simulation of atmospheric boundary layer flow through wind turbines and wind farms," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 99, no. 4, pp. 154–168, 2011.

[15] P. Gebraad, F. Teeuwisse, J. Wingerden, P. A. Fleming, S. Ruben, J. Marden, and L. Pao, "Wind plant power optimization through yaw control using a parametric model for wake effects—a cfd simulation study," *Wind Energy*, vol. 19, no. 1, pp. 95–114, 2016.

[16] N. O. Jensen, *A note on wind generator interaction.* DTU, 1983.

[17] Á. Jiménez, A. Crespo, and E. Migoya, "Application of a les technique to characterize the wake deflection of a wind turbine in yaw," *Wind energy*, vol. 13, no. 6, pp. 559–572, 2010.

[18] G. C. Larsen, H. Madsen Aagaard, F. Bingöl, J. Mann, S. Ott, J. N. Sørensen, V. Okulov, N. Troldborg, N. M. Nielsen, K. Thomsen *et al.*, "Dynamic wake meandering modeling," Risø National Laboratory, Tech. Rep., 2007.

[19] C. E. Carcangiu, "Cfd-rans study of horizontal axis wind turbines," Ph.D. dissertation, Universita'degli Studi di Cagliari, 2008.

[20] T. Stovall, G. Pawlas, and P. Moriarty, "Wind farm wake simulations in openfoam," *AIAA Paper*, vol. 825, p. 2010, 2010.

[21] M. J. Churchfield, S. Lee, P. J. Moriarty, L. A. Martinez, S. Leonardi, G. Vijayakumar, and J. G. Brasseur, "A large-eddy simulation of wind-plant aerodynamics," *AIAA paper*, vol. 537, p. 2012, 2012.

[22] M. Calaf, C. Meneveau, and J. Meyers, "Large eddy simulation study of fully developed wind-turbine array boundary layers," *Physics of fluids*, vol. 22, no. 1, p. 015110, 2010.

[23] N. Troldborg, "Actuator line modeling of wind turbine wakes," 2009.

[24] J. N. Sorensen and W. Z. Shen, "Numerical modeling of wind turbine wakes," *Journal of fluids engineering*, vol. 124, no. 2, pp. 393–399, 2002.

[25] R. Mikkelsen *et al.*, "Actuator disc methods applied to wind turbines," *Technical University of Denmark*, 2003.

[26] L. A. Martinez, S. Leonardi, M. J. Churchfield, and P. J. Moriarty, "A comparison of actuator disk and actuator line wind turbine models and best practices for their use," *AIAA Paper*, vol. 900, 2012.

[27] 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.

[28] P. J. Oliveira, Raad I. Issa, “An improved piso algorithm for the computation of buoyancy-driven flows,” *Numerical Heat Transfer: Part B: Fundamentals*, vol. 40, no. 6, pp. 473–493, 2001.

[29] M. Abkar, A. Sharifi, and F. Porté-Agel, “Wake flow in a wind farm during a diurnal cycle,” *Journal of Turbulence*, vol. 17, no. 4, pp. 420–441, 2016.

[30] M. J. Churchfield, S. Lee, J. Michalakes, and P. J. Moriarty, “A numerical study of the effects of atmospheric and wake turbulence on wind turbine dynamics,” *Journal of turbulence*, no. 13, p. N14, 2012.

[31] Q. Wang, H. Zhou, and D. Wan, “Numerical simulation of wind turbine blade-tower interaction,” *Journal of Marine Science and Application*, vol. 11, no. 3, pp. 321–327, 2012.

[32] N. Troldborg, J. N. Sørensen, and R. Mikkelsen, “Actuator line simulation of wake of wind turbine operating in turbulent inflow,” in *Journal of physics: conference series*, vol. 75, no. 1. IOP Publishing, 2007, p. 012063.

[33] J. N. Sørensen, R. F. Mikkelsen, D. S. Henningson, S. Ivanell, S. Sarmast, and S. J. Andersen, “Simulation of wind turbine wakes using the actuator line technique,” *Phil. Trans. R. Soc. A*, vol. 373, no. 2035, p. 20140071, 2015.

[34] L. A. Martínez-Tossas, M. J. Churchfield, and S. Leonardi, “Large eddy simulations of the flow past wind turbines: actuator line and disk modeling,” *Wind Energy*, vol. 18, no. 6, pp. 1047–1060, 2015.

[35] D. Yang, C. Meneveau, and L. Shen, “Large-eddy simulation of offshore wind farm,” *Physics of Fluids*, vol. 26, no. 2, p. 025101, 2014.

[36] D. Matha, “Impact of aerodynamics and mooring system on dynamic response of floating wind turbines,” *PhD dissertation*, 2017.

[37] D. Matha, F. Sandner, C. Molins, A. Campos, and P. W. Cheng, “Efficient preliminary floating offshore wind turbine design and testing methodologies and application to a concrete spar design,” *Phil. Trans. R. Soc. A*, vol. 373, no. 2035, p. 20140350, 2015.

[38] F. Sandner, D. Schlipf, D. Matha, and P. W. Cheng, “Integrated optimization of floating wind turbine systems,” in *ASME 2014 33rd International Conference on Ocean, Offshore and Arctic Engineering*. American Society of Mechanical Engineers, 2014, pp. V09BT09A030–V09BT09A030.

[39] D. Matha, G. P. Morán, K. Müller, and F. Lemmer, “Comparative analysis of industrial design methodologies for fixed-bottom and floating wind turbines,” in *ASME 2016 35th International Conference on Ocean, Offshore and Arctic Engineering*. American Society of Mechanical Engineers, 2016, pp. V006T09A055–V006T09A055.

[40] S. Rockel, E. Camp, J. Schmidt, J. Peinke, R. B. Cal, and M. Hölling, “Experimental study on influence of pitch motion on the wake of a floating wind turbine model,” *Energies*, vol. 7, no. 4, pp. 1954–1985, 2014.

[41] J. Azcona, D. Bekiropoulos, H. Bredmose, N. Heilskov, A. Krieger, T. Lutz, A. Manjock, D. Manolas, D. Matha, K. Meister *et al.*, “State-of-the-art and implementation of design tools for floating structures,” *Innwind*, 2013.

[42] Network, “Floating wind turbine model test,” *Marine Renewables Infrastructure*, 2014.

[43] K. Müller, F. Sandner, H. Bredmose, J. Azcona, A. Manjock, and R. Pereira, “Improved tank test procedures for scaled floating offshore wind turbines,” *IWEC*, 2014.

[44] C. Bjerkseter and A. Ågotnes, “Levelised costs of energy for offshore floating wind turbine concepts,” Master’s thesis, Norwegian University of Life Sciences, Ås, 2013.

[45] F. Campagnolo, V. Petrovic, E. M. Nanos, C. W. Tan, C. L. Bottasso, I. Paek, H. Kim, K. Kim *et al.*, “Wind tunnel testing of power maximization control strategies applied to a multi-turbine floating wind power platform,” in *The 26th International Ocean and Polar Engineering Conference*. International Society of Offshore and Polar Engineers, 2016.

[46] K.-H. Kim, K. Lee, J. M. Sohn, S.-W. Park, J.-S. Choi, K. Hong *et al.*, “Conceptual design of 10mw class floating wave-offshore wind hybrid power generation system,” in *The Twenty-fifth International Ocean and Polar Engineering Conference*. International Society of Offshore and Polar Engineers, 2015.

[47] J. Counihan, “An improved method of simulating an atmospheric boundary layer in a wind tunnel,” *Atmospheric Environment (1967)*, vol. 3, no. 2, pp. 197–214, 1969.

[48] N. Cook, “Wind-tunnel simulation of the adiabatic atmospheric boundary layer by roughness, barrier and mixing-device methods,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 3, no. 2-3, pp. 157–176, 1978.

[49] H. Kozmar, “Scale effects in wind tunnel modeling of an urban atmospheric boundary layer,” *Theoretical and Applied Climatology*, vol. 100, no. 1-2, pp. 153–162, 2010.

[50] H. Hu, Z. Yang, and P. Sarkar, “Dynamic wind loads and wake characteristics of a wind turbine model in an atmospheric boundary layer wind,” *Experiments in fluids*, vol. 52, no. 5, pp. 1277–1294, 2012.

[51] L. P. Chamorro and F. Porté-Agel, “Effects of thermal stability and incoming boundary-layer flow characteristics on wind-turbine wakes: a wind-tunnel study,” *Boundary-layer meteorology*, vol. 136, no. 3, pp. 515–533, 2010.

[52] A. Zasso, S. Giappino, S. Muggiasca, and L. Rosa, “Optimization of the boundary layer characteristics simulated at politecnico di milano boundary layer wind tunnel in a wide scale ratio ranger,” in *Proceedings of the 6th Asia-Pacific Conference on Wind Engineering*, 2005.

[53] B. Blocken, T. Stathopoulos, and J. Carmeliet, “Cfd simulation of the atmospheric boundary layer: wall function problems,” *Atmospheric environment*, vol. 41, no. 2, pp. 238–252, 2007.

[54] F. Porté-Agel, Y.-T. Wu, and C.-H. Chen, “A numerical study of the effects of wind direction on turbine wakes and power losses in a large wind farm,” *Energies*, vol. 6, no. 10, pp. 5297–5313, 2013.

[55] N. Troldborg, J. N. Sørensen, R. Mikkelsen, and N. N. Sørensen, “A simple atmospheric boundary layer model applied to large eddy simulations of wind turbine wakes,” *Wind Energy*, vol. 17, no. 4, pp. 657–669, 2014.

[56] H. Sarlak, “Large eddy simulation of turbulent flows in wind energy,” *PhD thesisTechnical University of Denmark2014*, 2014.

[57] C. L. Bottasso, F. Campagnolo, and V. Petrović, “Wind tunnel testing of scaled wind turbine models: Beyond aerodynamics,” *Journal of wind engineering and industrial aerodynamics*, vol. 127, pp. 11–28, 2014.

[58] H. Kim, K. Kim, I. Paek, C. L. Bottasso, and F. Campagnolo, “A study on the active induction control of upstream wind turbines for total power increases,” in *Journal of Physics: Conference Series*, vol. 753, no. 3. IOP Publishing, 2016, p. 032014.

[59] F. Campagnolo, V. Petrović, J. Schreiber, E. M. Nanos, A. Croce, and C. L. Bottasso, “Wind tunnel testing of a closed-loop wake deflection controller for wind farm power maximization,” in *Journal of Physics: Conference Series*, vol. 753, no. 3. IOP Publishing, 2016, p. 032006.

[60] M. F. van Dooren, F. Campagnolo, M. Sjöholm, N. Angelou, T. Mikkelsen, and M. Kühn, “Demonstration and uncertainty analysis of synchronised scanning lidar measurements of 2-d velocity fields in a boundary-layer wind tunnel,” *Wind Energy Science*, vol. 2, no. 1, pp. 329–341, 2017. [Online]. Available: https://www.wind-energ-sci.net/2/329/2017/

[61] C. L. Bottasso, S. Cacciola, and X. Iriarte, “Calibration of wind turbine lifting line models from rotor loads,” *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 124, pp. 29–45, 2014.

[62] C. Bottasso, S. Cacciola, and A. Croce, “Estimation of blade structural properties from experimental data,” *Wind Energy*, vol. 16, no. 4, pp. 501–518, 2013.

[63] M. Churchfield and S. Lee, “Nwtc design codes-sowfa,” 2012. [Online]. Available: http://wind.nrel.gov/designcodes/simulators/SOWFA

[64] J. M. Jonkman and M. L. Buhl Jr, “Fast user’s guide-updated august 2005,” National Renewable Energy Laboratory (NREL), Golden, CO., Tech. Rep., 2005.

[65] H. Jasak, A. Jemcov, Z. Tukovic *et al.*, “Openfoam: A c++ library for complex physics simulations,” in *International workshop on coupled methods in numerical dynamics*, vol. 1000. IUC Dubrovnik, Croatia, 2007, pp. 1–20.

[66] D. P. Rodgers, “Improvements in multiprocessor system design,” in *ACM SIGARCH Computer Architecture News*, vol. 13, no. 3. IEEE Computer Society Press, 1985, pp. 225–231.

[67] M. J. Churchfield, S. Schreck, L. A. Martınez-Tossas, C. Meneveau, and P. R. Spalart, “An advanced actuator line method for wind energy applications and beyond,” in *35th Wind Energy Symposium*, 2017, p. 1998.

[68] P. K. Jha and S. Schmitz, “Actuator curve embedding–an advanced actuator line model,” *Journal of Fluid Mechanics*, vol. 834, 2018.

[69] F. Menter and Y. Egorov, “A scale-adaptive simulation model using two-equation models,” *AIAA paper*, vol. 1095, p. 2005, 2005.

[70] H. Jasak, “Dynamic mesh handling in openfoam,” in *47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2009, p. 341.

[71] C.-H. Moeng, “A large-eddy-simulation model for the study of planetary boundary-layer turbulence,” *Journal of the Atmospheric Sciences*, vol. 41, no. 13, pp. 2052–2062, 1984.

[72] M. J. Churchfield, S. Lee, S. Schmitz, and Z. Wang, “Modeling wind turbine tower and nacelle effects within an actuator line model,” in *33rd Wind Energy Symposium*, 2015, p. 0214.

[73] R. J. Stevens, L. A. Martínez-Tossas, and C. Meneveau, “Comparison of wind farm large eddy simulations using actuator disk and actuator line models with wind tunnel experiments,” *Renewable energy*, vol. 116, pp. 470–478, 2018.

[74] X. Yang and F. Sotiropoulos, “A new class of actuator surface models for wind turbines,” *Wind Energy*, vol. 21, no. 5, pp. 285–302, 2018.

[75] R. Mittal and G. Iaccarino, “Immersed boundary methods,” *Annu. Rev. Fluid Mech.*, vol. 37, pp. 239–261, 2005.

[76] C. Santoni, K. Carrasquillo, I. Arenas-Navarro, and S. Leonardi, “Effect of tower and nacelle on the flow past a wind turbine,” *Wind Energy*, vol. 20, no. 12, pp. 1927–1939, 2017.

[77] H. Bandringa, “Immersed boundary methods,” *Masters thesis, University of Groningen*, vol. 9700, 2010.

[78] H. Jasak and D. Rigler, “Finite volume immersed boundary method for turbulent flow simulations,” in *9th OpenFOAM Workshop*, 2014.

[79] E. N. Jacobs and A. Sherman, “Airfoil section characteristics as affected by variations of the reynolds number,” 1937.

[80] F. Moukalled, L. Mangani, M. Darwish *et al.*, “The finite volume method in computational fluid dynamics,” 2016.

[81] D. Mehta, “Large eddy simulation of wind farm aerodynamics with energy-conserving schemes,” in *MARE-WINT*. Springer, 2016, pp. 347–358.

[82] C. J. Greenshields, “Openfoam user guide,” *OpenFOAM Foundation Ltd, version*, vol. 3, no. 1, 2015.

[83] T. Holzmann, “Mathematics, numerics, derivations and openfoam®,” 2016.

[84] J. H. Ferziger and M. Peric, *Computational methods for fluid dynamics*. Springer Science & Business Media, 2012.

[85] B. P. Leonard, “A stable and accurate convective modelling procedure based on quadratic upstream interpolation,” *Computer methods in applied mechanics and engineering*, vol. 19, no. 1, pp. 59–98, 1979.

[86] B. Leonard, “A survey of finite differences of opinion on numerical muddling of the incomprehensible defective confusion equation,” *Finite element methods for convection dominated flows*, vol. 34, pp. 1–10, 1979.

[87] H. Jasak, H. Weller, and A. Gosman, “High resolution nvd differencing scheme for arbitrarily unstructured meshes,” *International journal for numerical methods in fluids*, vol. 31, no. 2, pp. 431–449, 1999.

[88] H. Jasak, “Error analysis and estimation for finite volume method with applications to fluid flow,” 1996.

[89] P. Gaskell and A. Lau, “Curvature-compensated convective transport: Smart, a new boundedness-preserving transport algorithm,” *International journal for numerical methods in fluids*, vol. 8, no. 6, pp. 617–641, 1988.

[90] M. Darwish, “A new high-resolution scheme based on the normalized variable formulation,” *Numerical Heat Transfer, Part B Fundamentals*, vol. 24, no. 3, pp. 353–371, 1993.

[91] B. P. Leonard, “Simple high-accuracy resolution program for convective modelling of discontinuities,” *International journal for numerical methods in fluids*, vol. 8, no. 10, pp. 1291–1318, 1988.

[92] T. Theodoropoulos, “Prediction of three dimensional engine flow on unstructured meshes,” 1990.

[93] A. Gosman, “Developments in industrial computational fluid dynamics,” *Chemical Engineering Research and Design*, vol. 76, no. 2, pp. 153–161, 1998.

[94] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*. NBS Washington, DC, 1952, vol. 49, no. 1.

[95] T. Behrens, “Openfoam’s basic solvers for linear systems of equations,” *Chalmers, Department of Applied Mechanics*, vol. 18, no. 02, 2009.

[96] A. Van der Sluis and H. A. van der Vorst, “The rate of convergence of conjugate gradients,” *Numerische Mathematik*, vol. 48, no. 5, pp. 543–560, 1986.

[97] R. P. Fedorenko, “A relaxation method for solving elliptic difference equations,” *USSR Computational Mathematics and Mathematical Physics*, vol. 1, no. 4, pp. 1092–1096, 1962.

[98] F. De la Vallee Poussin and W. P. Timlake, “An accelerated relaxation algorithm for iterative solution of elliptic equations,” *SIAM Journal on Numerical Analysis*, vol. 5, no. 2, pp. 340–351, 1968.

[99] A. Settari and K. Aziz, “A generalization of the additive correction methods for the iterative solution of matrix equations,” *SIAM Journal on Numerical Analysis*, vol. 10, no. 3, pp. 506–521, 1973.

[100] W. L. Briggs, S. F. McCormick *et al.*, *A multigrid tutorial*. Siam, 2000, vol. 72.

[101] S. Elias, G. Stubley, and G. Raithby, “An adaptive agglomeration method for additive correction multigrid,” *International Journal for Numerical Methods in Engineering*, vol. 40, no. 5, pp. 887–903, 1997.

[102] D. Mavriplis and V. Venkatakrishnan, “Agglomeration multigrid for viscous turbulent flows,” in *Fluid Dynamics Conference*, 1994, p. 2332.

[103] R. Lonsdale, “An algebraic multigrid scheme for solving the navier-stokes equations on unstructured meshes,” *Numerical methods in laminar and turbulent flow*, pp. 1432–1442, 1991.

[104] M. Imano, “Optimization of parameter settings for gamg solver in simple solver,” 2012. [Online]. Available: https://www.slideshare.net/MasashiImano/20120908ofbatkansai

[105] O. Tatebe, “The multigrid preconditioned conjugate gradient method,” 1993.

[106] Y. Saad, *Iterative methods for sparse linear systems.* siam, 2003, vol. 82.

[107] R. I. Issa, “Solution of the implicitly discretised fluid flow equations by operator-splitting,” *Journal of computational physics*, vol. 62, no. 1, pp. 40–65, 1986.

[108] A. A. AlOnazi, “Design and optimization of openfoam-based cfd applications for modern hybrid and heterogeneous hpc platforms,” Ph.D. dissertation, 2014.

[109] C. Chevalier and F. Pellegrini, “Pt-scotch: A tool for efficient parallel graph ordering,” *Parallel computing*, vol. 34, no. 6-8, pp. 318–331, 2008.

[110] W. D. Gropp and D. E. Keyes, “Domain decomposition on parallel computers,” *IMPACT of Computing in Science and Engineering*, vol. 1, no. 4, pp. 421–439, 1989.

[111] G. M. Amdahl, “Validity of the single processor approach to achieving large scale computing capabilities,” in *Proceedings of the April 18-20, 1967, spring joint computer conference.* ACM, 1967, pp. 483–485.

[112] J. L. Gustafson, “Reevaluating amdahl’s law,” *Communications of the ACM*, vol. 31, no. 5, pp. 532–533, 1988.

[113] X.-H. Sun and Y. Chen, “Reevaluating amdahl’s law in the multicore era,” *Journal of Parallel and Distributed Computing*, vol. 70, no. 2, pp. 183–188, 2010.

[114] S. Keough, “Optimising the parallelisation of openfoam simulations,” DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION FISHERMANS BEND (AUSTRALIA) MARITIME DIV, Tech. Rep., 2014.

[115] M. Culpo, “Current bottlenecks in the scalability of openfoam on massively parallel clusters,” *PRACE white paper to appear on http://www. praceri. eu*, 2011.

[116] D. Kanter, “Intel’s haswell cpu microarchitecture,” *Real World Technologies, November*, 2012.

[117] T. Jain and T. Agrawal, “The haswell microarchitecture-4th generation processor,” *International Journal of Computer Science and Information Technologies*, vol. 4, no. 3, pp. 477–480, 2013.

[118] Wikipedia, “Haswell (microarchitecture),” 2012. [Online]. Available: https://www.wind-energ-sci.net/2/329/2017/

[119] O. Schroder, “Runtime performance optimizations for an openfoam simulation,” in *HPC Advisory Council European Conference*, 2014.

[120] Wyldckat, "Installation/linux/foam-extend-4.0," 2018. [Online]. Available: https://openfoamwiki.net/index.php/Installation/Linux/foam-extend-4.0

[121] J. F. Manwell, J. G. McGowan, and A. L. Rogers, *Wind energy explained: theory, design and application.* John Wiley & Sons, 2010.

[122] J. Smagorinsky, "General circulation experiments with the primitive equations: I. the basic experiment," *Monthly weather review*, vol. 91, no. 3, pp. 99–164, 1963.

[123] S. B. Pope, "Turbulent flows," 2001.

[124] P. Sagaut, *Large eddy simulation for incompressible flows: an introduction.* Springer Science & Business Media, 2006.

[125] P. J. Mason, "Large-eddy simulation: A critical review of the technique," *Quarterly Journal of the Royal Meteorological Society*, vol. 120, no. 515, pp. 1–26, 1994.

[126] J. W. Deardorff, "A numerical study of three-dimensional turbulent channel flow at large reynolds numbers," *Journal of Fluid Mechanics*, vol. 41, no. 2, pp. 453–480, 1970.

[127] N. S. Ghaisas, "Subgrid scale modeling for large eddy simulation of buoyant turbulent flows," Ph.D. dissertation, Purdue University, 2013.

[128] C. Meneveau, T. S. Lund, and W. H. Cabot, "A lagrangian dynamic subgrid-scale model of turbulence," *Journal of Fluid Mechanics*, vol. 319, pp. 353–385, 1996.

[129] M. Germano, U. Piomelli, P. Moin, and W. H. Cabot, "A dynamic subgrid-scale eddy viscosity model," *Physics of Fluids A: Fluid Dynamics*, vol. 3, no. 7, pp. 1760–1765, 1991.

[130] D. K. Lilly, "A proposed modification of the germano subgrid-scale closure method," *Physics of Fluids A: Fluid Dynamics*, vol. 4, no. 3, pp. 633–635, 1992.

[131] S. Ghosal, T. S. Lund, P. Moin, and K. Akselvoll, "A dynamic localization model for large-eddy simulation of turbulent flows," *Journal of Fluid Mechanics*, vol. 286, pp. 229–255, 1995.

[132] C. Meneveau and J. Katz, "Scale-invariance and turbulence models for large-eddy simulation," *Annual Review of Fluid Mechanics*, vol. 32, no. 1, pp. 1–32, 2000.

[133] F. Menter and Y. Egorov, "Sas turbulence modelling of technical flows," *ERCOFTAC SERIES*, vol. 10, p. 687, 2006.

[134] Y. Egorov and F. Menter, "Development and application of sst-sas turbulence model in the desider project," *Advances in Hybrid RANS-LES Modelling*, pp. 261–270, 2008.

[135] D. Lindblad, A. Jareteg, and O. Petit, "Implementation and run-time mesh refinement for the k- ω sst des turbulence model when applied to airfoils." *Chalmers University of Technology, USA*, 2014.

[136] F. Menter and Y. Egorov, "The scale-adaptive simulation method for unsteady turbulent flow predictions. part 1: theory and model description," *Flow, Turbulence and Combustion*, vol. 85, no. 1, pp. 113–138, 2010.

[137] Y. Egorov, F. Menter, R. Lechner, and D. Cokljat, "The scale-adaptive simulation method for unsteady turbulent flow predictions. part 2: Application to complex flows," *Flow, Turbulence and Combustion*, vol. 85, no. 1, pp. 139–165, 2010.

[138] M. Younsi, A. Djerrada, T. Belamri, and F. Menter, "Application of the sas turbulence model to predict the unsteady flow field behaviour in a forward centrifugal fan," *International journal of computational fluid dynamics*, vol. 22, no. 9, pp. 639–648, 2008.

[139] J. C. Familton, "Quaternions: a history of complex noncommutative rotation groups in theoretical physics," *arXiv preprint arXiv:1504.04885*, 2015.

[140] B. A. Robinson, H. T. Yang, and J. T. Batina, "Aeroelastic analysis of wings using the euler equations with a deforming mesh," *Journal of Aircraft*, vol. 28, no. 11, pp. 781–788, 1991.

[141] J. T. Batina, "Unsteady euler airfoil solutions using unstructured dynamic meshes," *AIAA journal*, vol. 28, no. 8, pp. 1381–1388, 1990.

[142] T. J. Mueller, *Fixed and flapping wing aerodynamics for micro air vehicle applications*. AIAA, 2001, vol. 195.

[143] F. Campagnolo, V. Petrović, J. Schreiber, E. M. Nanos, A. Croce, and C. L. Bottasso, "Wind tunnel testing of a closed-loop wake deflection controller for wind farm power maximization," in *Journal of Physics: Conference Series*, vol. 753, no. 3. IOP Publishing, 2016, p. 032006.

[144] F. Campagnolo, V. Petrović, C. L. Bottasso, and A. Croce, "Wind tunnel testing of wake control strategies," in *American Control Conference (ACC), 2016*. IEEE, 2016, pp. 513–518.

[145] F. Campagnolo, V. Petrovic, E. M. Nanos, C. W. Tan, C. L. Bottasso, I. Paek, H. Kim, K. Kim *et al.*, "Wind tunnel testing of power maximization control strategies applied to a multi-turbine floating wind power platform," in *The 26th International Ocean and Polar Engineering Conference*. International Society of Offshore and Polar Engineers, 2016.

[146] E. Bossanyi, "The design of closed loop controllers for wind turbines," *Wind Energy*, vol. 3, no. 3, pp. 149–163, 2000.

[147] P. Fleming, P. M. Gebraad, S. Lee, J.-W. Wingerden, K. Johnson, M. Churchfield, J. Michalakes, P. Spalart, and P. Moriarty, "Simulation comparison of wake mitigation control strategies for a two-turbine case," *Wind Energy*, vol. 18, no. 12, pp. 2135–2143, 2015.

[148] J. Jonkman, S. Butterfield, W. Musial, and G. Scott, "Definition of a 5-mw reference wind turbine for offshore system development," National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2009.

[149] G. Campanardi, D. Grassi, A. Zanotti, E. M. Nanos, F. Campagnolo, A. Croce, and C. L. Bottasso, "Stereo particle image velocimetry set up for measurements in the wake of scaled wind turbines," in *Journal of Physics: Conference Series*, vol. 882, no. 1. IOP Publishing, 2017, p. 012003.

[150] P. O'neill, D. Nicolaides, D. Honnery, J. Soria *et al.*, "Autocorrelation functions and the determination of integral length with reference to experimental and numerical data," in *15th Australasian Fluid Mechanics Conference*, vol. 1. The University of Sydney, 2004, pp. 1–4.

[151] J. Wang, S. Foley, E. Nanos, T. Yu, F. Campagnolo, C. Bottasso, A. Zanotti, and A. Croce, "Numerical and experimental study for wake redirection techniques in a boundary layer wind tunnel," in *Journal of Physics: Conference Series*, 2017.

[152] F. Menter and T. Esch, "Elements of industrial heat transfer predictions," in *16th Brazilian Congress of Mechanical Engineering (COBEM)*, vol. 109, 2001.

[153] J. Wang, y. Liu, C. Filippo, and C. L. Bottasso, "Verification of a low cost scale-adaptive cfd formulation for waked wind turbines," in *AIAA Science and Technology Forum and Exposition (SciTech 2018)*, 2018.

[154] J. Wang, D. Mclean, F. Campagnolo, T. Yu, and C. Bottasso, "Large-eddy simulation of waked turbines in a scaled wind farm facility," in *Journal of Physics: Conference Series, preprint (under review)*, 2017.

[155] D. J. Renkema, "Validation of wind turbine wake models," *Master of Science Thesis, Delft University of Technology*, vol. 19, 2007.

[156] J. Wang, C. L. Bottasso, and F. Campagnolo, "Wake redirection: comparison of analytical, numerical and experimental models," in *TORQUE 2016, The Science of Making Torque from Wind*. IOP Publishing, 2016.

[157] P. J. Moriarty and A. C. Hansen, "Aerodyn theory manual," National Renewable Energy Lab., Golden, CO (US), Tech. Rep., 2005.

[158] N. Troldborg, F. Zahle, P.-E. Réthoré, and N. N. Sørensen, "Comparison of wind turbine wake properties in non-sheared inflow predicted by different computational fluid dynamics rotor models," *Wind Energy*, vol. 18, no. 7, pp. 1239–1250, 2015.

[159] R. L. Bisplinghoff and H. Ashley, *Principles of aeroelasticity*. Courier Corporation, 2013.

[160] N. G. Jørgensen and H. NILSSON, "Implementation of a turbulent inflow boundary condition for les based on a vortex method," *A course at Chalmers University of Technology*, 2012.

[161] B. J. Jonkman, "Turbsim user's guide: Version 2.00," 2016.

[162] A. W. Lavely, G. Vijayakumar, J. G. Brasseur, E. G. Paterson, and M. P. Kinzel, "Comparing unsteady loadings on wind turbines using turbsim and les flow fields," in *Proceedings of the 50th AIAA Aerospace Sciences Meeting, Nashville, TN, USA*, vol. 912, 2012.

[163] R. K. Rai, M. J. Singh, and J. W. Naughton, "Investigation of turbine response to various wind inflow models," in *Proceedings of the 49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition. Orlando, FL, USA*, 2011.

[164] J. Rohr, E. Itsweire, K. Helland, and C. Van Atta, "Growth and decay of turbulence in a stably stratified shear flow," *Journal of Fluid Mechanics*, vol. 195, pp. 77–111, 1988.

[165] C. Bottasso, A. Croce, C. Riboldi, and M. Salvetti, "Cyclic pitch control for the reduction of ultimate loads on wind turbines," in *Journal of Physics: Conference Series*, vol. 524. IOP Publishing, 2014, p. 012063.

[166] L. Vollmer, G. Steinfeld, D. Heinemann, and M. Kühn, "Estimating the wake deflection downstream of a wind turbine in different atmospheric stabilities: an les study," *Wind Energy Science*, vol. 1, no. 2, pp. 129–141, 2016.

[167] J.-J. Trujillo, F. Bingöl, G. C. Larsen, J. Mann, and M. Kühn, "Light detection and ranging measurements of wake dynamics. part ii: two-dimensional scanning," *Wind Energy*, vol. 14, no. 1, pp. 61–75, 2011.

[168] T. Chai and R. R. Draxler, "Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature," *Geoscientific Model Development*, vol. 7, no. 3, pp. 1247–1250, 2014.

[169] F. Campagnolo, V. Petrovic, E. M. Nanos, C. W. Tan, C. L. Bottasso, I. Paek, H. Kim, K. Kim *et al.*, "Wind tunnel testing of power maximization control strategies applied to a multi-turbine floating wind power platform," in *The 26th International Ocean and Polar Engineering Conference.* International Society of Offshore and Polar Engineers, 2016.

[170] H. Kim, M. Kim, K. Kim, K. Hong, Y. Bae *et al.*, "Global performance of a square-type semi-submersible kriso multi-unit floating wind turbine; numerical simulation vs model test," in *The 26th International Ocean and Polar Engineering Conference.* International Society of Offshore and Polar Engineers, 2016.

ACKNOWLEDGMENTS

I would like to extend thanks to the many people, in many countries, who so generously contributed to the work presented in this thesis.

I would like to first express my sincere gratitude to my Prof. Dr. Carlo L. Bottasso for the continuous support of my Ph.D. study and relevant research, for his patience and motivation. His guidance helped me in all the time of research and writing of this thesis. I also thank my fellow labmates, Dr. Filippo Campagnolo, Mr. Chengyu Wang, Mr. Johannes Schreiber, Mr. Alberto Fortes Plaza, Mr. Emmanouil Nanos, Mr. Robin Weber and my former colleague Dr. Stefano Cacciola and Dr. Vlaho Petrović for the stimulating discussions, for the moments we were working together before deadlines, and for all the fun we have had in the last four years. Moreover, I also thank my supervised student, Mr. Gabriel Naccache, Mr. Patrick Larin, Ms. Alessandra Fabbri, Mr. Kutay Yilmazlar, Mr. Spencer Foley, Mr. Dan Mclean, Ms. Tong Yu, Mr. Yang Liu, Mr. Omar David Castañeda Mr. Kai Lin, Ms. Leslie Taylor and Mr. Sina Khedmati for their assistance to my research work.

In addition, I would like to thank Chinese Scholarship Council for the financial support of this Ph.D. thesis. This work has also been supported by the CL-WINDCON project, which receives funding from the European Union Horizon 2020 research and innovation program under grant agreement No. 727477. All tests were performed at the wind tunnel of Politecnico di Milano, with the support of Prof. A. Croce, Mr. G. Campanardi, Dr. Zanotti and Mr. D. Grassi. Indeed, I also would like to thank Dr. Matthew Churchfield and Prof. Dr. Fernando Porté-Agel for their insightful comments and encouragement for my Ph.D. research. Moreover, I express my appreciation to the Leibniz Supercomputing Centre (LRZ) for providing access and computing time on the SuperMUC Petascale System.

To this end, I would like to thank my parents for supporting me spiritually throughout writing this thesis and my life in general. My friend, Bing Li, Yu Wang, Shengyang Kang, Wei Wu, Fangzhou Liu, etc. also helped me in multiple aspects of my Ph.D. life.