



# UNIVERSITY OF PATRAS

School of Natural Sciences

Department of Physics

Applied Science Sector

Laboratory of Atmospheric Physics

Forecasting wind power and turbine energy  
efficiency

---

## **PhD Dissertation**

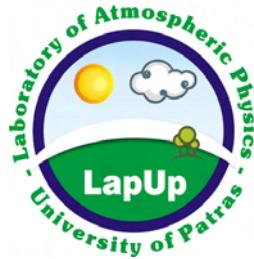
---

Michos K. Dimitrios

### **Supervisor**

Andreas Kazantzidis, Professor

Patras, September 2024



# ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ

Σχολή Θετικών Επιστημών

Τμήμα Φυσικής

Τομέας Εφαρμοσμένων Επιστημών

Εργαστήριο Φυσικής της Ατμόσφαιρας

Πρόγνωση αιολικού δυναμικού και ενεργειακής  
απόδοσης ανεμογεννήτριας

---

Διδακτορική Διατριβή

---

Μίχος Κ. Δημήτριος

Επιβλέπων

Ανδρέας Καζαντζίδης, Καθηγητής

Πάτρα, Σεπτέμβριος 2024

Πανεπιστήμιο Πατρών, Τμήμα Φυσικής  
Μίχος Δημήτριος  
© 2024 - Με την επιφύλαξη παντός δικαιώματος

# **Supervisory committee**

---

Three-Member Advisory Committee

**Andreas Kazantzidis (Supervisor),**  
Professor - Department of Physics - University of Patras

**Dimitrios Soudris,**  
Professor - School of Electrical and Computer Engineering, National  
Technical University of Athens

**Francky Catthoor,**  
Professor - Department of Electrical Engineering, Faculty of Engineering  
Studies, KU Leuven & Senior fellow - Interuniversity Microelectronics  
Centre (imec)

---

Examination Committee

**Athanasis Argyriou,**  
Professor - Department of Physics - University of Patras

**Ioannis Kioutsioukis,**  
Associate Professor - Department of Physics - University of Patras

**Loukopoulos Vassilios,**  
Professor - Department of Physics - University of Patras

**Sirrokostas George,**  
Assistant professor - Department of Physics - University of Patras

Everything's scheduled by chance.  
We can't save a world  
that we leave in.  
Until we meet again...

## Abstract

This dissertation presents a comprehensive methodology for modeling, simulating and forecasting the behaviour of turbulent airflow over complex terrain in the vicinity of wind farms, with a particular focus on addressing challenges associated with limited wind data availability. The goal of this approach is to enable accurate short-term energy production forecasting for wind turbines operating in such challenging environments.

A case study of the Lavrio wind farm in Greece is utilized to illustrate and validate the methodology of this PhD research. The approach leverages vertical wind LIDAR measurements, coupled with a novel spatial extrapolation technique based on a CFD model (Wi.Sp.Ex), to reconstruct the airflow above the complex terrain of the wind farm. The  $k-\epsilon$  turbulence model is employed to efficiently capture the near-wall flow behavior. Through a series of steady-state simulations with varying inlet conditions and initial values, the temporal evolution of the turbulent flow field is investigated, providing valuable insights into the complex interactions between wind and terrain.

The Wi.Sp.Ex. model serves as the core physics-based component of the Wind Energy Extraction Latency (W.E.E.L.) model, which forecasts the 15-minute power production of a wind turbine by identifying its wind energy extraction latency and utilizing the Wi.Sp.Ex simulation results as wind speed forecasts. This integrated approach enables a more accurate and reliable prediction of short-term energy production, facilitating improved wind farm management and grid integration strategies.

This methodology offers a promising solution for enhancing the accuracy and efficiency of wind resource assessment and wind farm design, particularly in complex terrain settings where traditional measurement and modeling approaches might fall short. By addressing the challenges associated with limited wind data availability and incorporating the temporal dynamics of the wind field, this approach contributes to the advancement of wind energy technology and its integration into the broader energy landscape.

## Περίληψη

Αυτή η διατριβή παρουσιάζει μια ολοκληρωμένη μεθοδολογία για τη μοντελοποίηση, την προσομοίωση και την πρόβλεψη της συμπεριφοράς της τυρβώδους ροής του αέρα σε σύνθετο έδαφος στην περιοχή αιολικών πάρκων, με ιδιαίτερη έμφαση στην αντιμετώπιση των προκλήσεων που σχετίζονται με τη περιορισμένη διαθεσιμότητα δεδομένων ανέμου. Ο στόχος αυτής της προσέγγισης είναι να καταστεί δυνατή η ακριβής βραχυπρόθεσμη πρόβλεψη της παραγωγής ενέργειας για τις ανεμογεννήτριες που λειτουργούν σε τέτοια απαιτητικά περιβάλλοντα.

Μια μελέτη περίπτωσης του αιολικού πάρκου στο Λαύριο της Ελλάδας χρησιμοποιείται για να δείξει και να επικυρώσει τη μεθοδολογία αυτής της διδακτορικής έρευνας. Η προσέγγιση αξιοποιεί μετρήσεις από κατακόρυφο ανεμολογικό LIDAR, σε συνδυασμό με μια νέα τεχνική χωρικής παρεκβολής βασισμένη σε ένα μοντέλο CFD (Wi.Sp.Ex), για την ανακατασκευή της ροής του αέρα πάνω από το σύνθετο έδαφος του αιολικού πάρκου. Το μοντέλο τύρβης k-ε χρησιμοποιείται για την αποτελεσματική καταγραφή της συμπεριφοράς της ροής χοντά στον τοίχο. Μέσω μιας σειράς προσομοιώσεων σταθερής κατάστασης με ποικίλες συνθήκες εισόδου και αρχικές τιμές, διερευνάται η χρονική εξέλιξη του πεδίου τυρβώδους ροής, παρέχοντας πολύτιμες πληροφορίες για τις πολύπλοκες αλληλεπιδράσεις μεταξύ του ανέμου και του εδάφους.

Το μοντέλο Wi.Sp.Ex. αποτελεί το βασικό, θεμελιωμένο στη φυσική, συστατικό του μοντέλου Wind Energy Extraction Latency (W.E.E.L.), το οποίο προβλέπει την παραγωγή ισχύος μιας ανεμογεννήτριας σε διάστημα 15 λεπτών, αναγνωρίζοντας τον χρόνο καθυστέρησης στην εξαγωγή αιολικής ενέργειας και χρησιμοποιώντας τα αποτελέσματα της προσομοίωσης Wi.Sp.Ex ως προβλέψεις για την ταχύτητα του ανέμου. Αυτή η ολοκληρωμένη προσέγγιση επιτρέπει μια πιο ακριβή και αξιόπιστη πρόβλεψη της βραχυπρόθεσμης παραγωγής ενέργειας, διευκολύνοντας τη βελτιωμένη διαχείριση των αιολικών πάρκων και τις στρατηγικές ενσωμάτωσης στο δίκτυο.

Αυτή η μεθοδολογία προσφέρει μια πολλά υποσχόμενη λύση για την ενίσχυση της ακρίβειας και της αποτελεσματικότητας της αξιολόγησης των αιολικών πόρων και του σχεδιασμού αιολικών πάρκων, ιδιαίτερα σε περιβάλλοντα σύνθετου εδάφους όπου οι παραδοσιακές προσεγγίσεις μέτρησης και μοντελοποίησης μπορεί να αποτύχουν. Αντιμετωπίζοντας τις προκλήσεις που σχετίζονται με την περιορισμένη διαθεσιμότητα δεδομένων ανέμου και ενσωματώνοντας τη χρονική δυναμική του πεδίου ανέμου, αυτή η προσέγγιση συμβάλλει στην πρόοδο της τεχνολογίας αιολικής ενέργειας και την ενσωμάτωσή της στο ευρύτερο ενεργειακό τοπίο.

## Acknowledgements

My deepest gratitude goes to my professors, and in particular to my supervisor, for their unwavering guidance, support, and encouragement throughout this research journey. Their expertise and insightful feedback have been instrumental in shaping and refining this dissertation.

I am also profoundly grateful to the Centre for Renewable Energy Sources (CRES) for generously providing the wind and terrain data that served as the foundation of this study. Their dedication to advancing wind energy research is truly inspiring.

This research has been funded by the Research Committee - University of Patras, under the project “Short-term wind turbine Energy-yield forecasting” (project code: 81023) and co-financed by IMEC. I extend my heartfelt thanks to the university and IMEC for equipping me with the knowledge and resources necessary to undertake this endeavor. Access to state-of-the-art computational facilities and software tools proved invaluable in conducting the simulations and analyses presented herein.

On a personal note, I overwhelmingly express my profound appreciation to my family and close friends for their unwavering support and encouragement, especially during the challenging moments that inevitably arose throughout these years of research. Their love, patience, and understanding have been my bedrock, reminding me of the importance of self-care even when I was consumed by the demands of my work.

Finally, I wish to acknowledge the profound role of art in complementing the pursuit of science throughout this journey. The creative expression and boundless inspiration found in art have provided a much-needed counterbalance to the rigors of scientific inquiry, reminding me of the beauty and interconnectedness of the world we strive to comprehend.

# **Publications and Conference Participations**

- 1. Participation** in the EMS2022 (European Meteorological Society Conference 2022) with the title: Michos, D., Kazantzidis, A., Catthoor, F., & Foussekis, D. (2022). Preliminary results of a physical model for extrapolating the wind field over complex terrain (No. EMS2022-296). Copernicus Meetings. [10.5194/ems2022-296](https://doi.org/10.5194/ems2022-296)
- 2. Publication** of the article titled: Michos, D., Catthoor, F., Foussekis, D., & Kazantzidis, A. (2024). A CFD model for spatial extrapolation of wind field over complex terrain—Wi.Sp.Ex. *Energies*, 17(16), 4139. doi: [10.3390/en17164139](https://doi.org/10.3390/en17164139).
- 3. Publication** of the article titled: Michos, D., Catthoor, F., Foussekis, D., & Kazantzidis, A. (2024). Ultra-Short-Term Wind Power Forecasting in Complex Terrain: A Physics-Based Approach. *Energies*, 17(21), 5493. doi: [10.3390/en17215493](https://doi.org/10.3390/en17215493) (registering DOI, <https://www.mdpi.com/1996-1073/17/21/5493>).

# List of Figures

1.1	Cartesian grid . . . . .	36
1.2	Curvilinear grid . . . . .	37
1.3	Example of a 2D slice of a grid for fluid flow simulation around the tower of a wind turbine with mixed characteristics. . . . .	39
1.4	Example of a Two-dimensional (2D) unstructured triangular grid. . . . .	42
1.5	Example of a 2D unstructured quadrilateral grid that trans- forms smoothly to a structured Cartesian grid. . . . .	43
1.6	Example of a hybrid grid. . . . .	44
1.7	A composite 2D grid. . . . .	49
1.8	Example of a 2D block-structured grid with non-matching interfaces. . . . .	51
3.1	Setting up a Geometry for real simulations in COMSOL . . .	89
3.2	Meshering a Geometry in COMSOL . . . . .	91
3.3	Defining the fluid properties in COMSOL . . . . .	92
3.4	k-epsilon stationary study equations in COMSOL . . . . .	99
3.5	k-epsilon default turbulence parameters in COMSOL . . . .	100
3.6	Temperature import via interpolation function in COMSOL .	107
3.7	Wall boundary and k-epsilon equations in COMSOL . . . .	109
3.8	Inlet boundary, k-epsilon equations and variables in COMSOL	110
3.9	Open boundary, k-epsilon equations and variables in COMSOL	111
3.10	Ramp function in COMSOL . . . . .	115
3.11	Extrapolated y-component wind speed in COMSOL (negative values in y direction indicate a wind direction toward the south – dominant direction) . . . . .	116
3.12	Segregated solver configuration in COMSOL . . . . .	120
3.13	Segregated Solver configurations in comsol . . . . .	122
4.1	Wind Turbine (WT) information from CRES, 2024. . . . .	131

4.2 CRES satelite view Google Maps, 2024 with Light Detection and Ranging (LIDAR) and WT locations ( <b>left</b> ), meshed geometry (center), and geometry with LIDAR and WT locations ( <b>right</b> ), Michos et al., 2024a . . . . .	132
4.3 Effect of a non-rotating WT on wind speed magnitude (m/s). Iso-surfaces represent the wind speed difference before and after installing a non-rotating WT. <b>(a)</b> Length of iso-surfaces (in WT diameters) showing varying wind speed differences (with transparency), <b>(b)</b> view of these iso-surfaces from below (with transparency), <b>(c)</b> 1 m/s wind speed magnitude difference iso-surface (opaque), Michos et al., 2024a. . . . .	133
4.4 Impact of a small shed on the wind field over flat terrain. The presence of the structure causes alterations in wind speed that can propagate significantly in all directions (Michos et al., 2024a). . . . .	133
4.5 Geometry meshing of the simulation domain. The inner region features a finer mesh to capture detailed flow characteristics near the WTs, while the outer region has a coarser mesh, optimized for computational efficiency (Michos et al., 2024a).	135
4.6 Elevation map of CRES location used for the simulations and distances between LIDAR and WTs(Michos et al., 2024a). . .	137
4.7 Vertical profile extrapolation ramp function $U_{ramp}(z)$ . $z = 0$ represents the ground level at the location of the LIDAR. $U_{ramp}(z) = 1$ for $z \geq 40$ m (First LIDAR measurement height),Michos et al., 2024a. . . . .	138
4.8 ENERCON IAS results(Michos et al., 2024a). . . . .	141
4.9 NEG IAS results(Michos et al., 2024a). . . . .	141
4.10Windroses. . . . .	142
4.11VESTAS IAS results(Michos et al., 2024a). . . . .	142
4.12Enercon EAS results(Michos et al., 2024a). . . . .	144
4.13NEG EAS results(Michos et al., 2024a). . . . .	145
4.14Vestas EAS results(Michos et al., 2024a). . . . .	145
4.15External Area Simulation (EAS) - Corrected LIDAR estimation errors at different heights. . . . .	146
4.16ENERCON E-IAS Results(Michos et al., 2024a). . . . .	147
4.17NEG E-IAS Results(Michos et al., 2024a). . . . .	147
4.18VESTAS E-IAS Results(Michos et al., 2024a). . . . .	148
4.19Wind Energy Extraction Latency forecasting model (WEEL) diagram (Michos et al., 2024b) . . . . .	153

4.20Enercon Power curve scaterplot . . . . .	156
4.21Enercon Power curve scaterplot . . . . .	158
4.22Enercon 15min Power Curves from shifted 10min wind measurements (a,d,f presented in Michos et al., 2024b) . . . . .	159
4.23NEG 15min Power Curves from shifted 10min wind measurements (a,d,f presented in Michos et al., 2024b) . . . . .	160
4.24Time series of 15-minute power forecasts for the Enercon WT (Michos et al., 2024b). . . . .	163
4.25Errors in 15-minute power conversion from measured wind speed for the Enercon WT (Michos et al., 2024b). . . . .	164
4.26Comparison of 15-minute power forecast errors between Wind Spatial Extrapolation model (WiSpEx) and WEEL for the En- ercon WT (Michos et al., 2024b). . . . .	164
4.27Absolute errors (Michos et al., 2024b) . . . . .	165
4.28Absolute Percentage errors (Michos et al., 2024b) . . . . .	166
4.29Time series of 15-minute power forecasts for the NEG WT (Michos et al., 2024b). . . . .	171
4.30Errors in 15-minute power conversion from measured wind speed for the NEG WT (Michos et al., 2024b). . . . .	172
4.31Comparison of 15-minute power forecast errors between WiSpEx and WEEL for the NEG WT (Michos et al., 2024b). . . . .	172
4.32Absolute errors (Michos et al., 2024b) . . . . .	173
4.33Absolute Percentage errors (Michos et al., 2024b) . . . . .	173

# List of Tables

2.1 Comparison of Finite Difference Method Finite Difference Method (FDM), Finite Volume Method (FVM) FVM, and Finite Element Method (FEM). . . . .	83
4.1 IAS – WT Corrected Velocity Magnitude Statistics . . . . .	140
4.2 EAS – LIDAR Corrected Velocity Magnitude Statistics . . . . .	143
4.3 EAS – WT Corrected Velocity Magnitude Statistics . . . . .	143
4.4 E-IAS – WT Corrected Velocity Magnitude Statistics . . . . .	146
4.5 WEEL - Enercon shift stats (part of table presented in Michos et al., 2024b) . . . . .	161
4.6 WEEL - NEG shift stats (part of table presented in Michos et al., 2024b) . . . . .	161
4.7 Enercon 15 minute Power Forecast Errors stats for WEEL at different Time Steps (part of table presented in Michos et al., 2024b) . . . . .	163
4.8 Enercon 15 minute Power Forecast Errors stats for Various Forecast Methods and Time Steps . . . . .	169
4.9 Enercon 15 minute Power Forecast Absolute Errors [kW] for Various Forecast Methods and Time Steps . . . . .	169
4.10 Enercon 15 minute Power Forecast Absolute Percentage Errors [%] for Various Forecast Methods and Time Steps . . . . .	170
4.11 Enercon 15 minute Power Forecast Symmetrical Absolute Percentage Errors [%] for Various Forecast Methods and Time Steps . . . . .	170
4.12 NEG 15 minute Power Forecast Errors stats for WEEL at different Time Steps (part of table presented in Michos et al., 2024b) . . . . .	171
4.13 NEG 15 minute Power Forecast Errors stats for Various Forecast Methods and Time Steps . . . . .	176
4.14 NEG 15 minute Power Forecast Absolute Errors [kW] for Various Forecast Methods and Time Steps . . . . .	176

4.15NEG 15 minute Power Forecast Absolute Percentage Errors [%] for Various Forecast Methods and Time Steps . . . . . 177

4.16NEG 15 minute Power Forecast Symmetrical Absolute Percentage Errors [%] for Various Forecast Methods and Time Steps . . . . . 177

# Abbreviations

**1D** One-dimensional

**2D** Two-dimensional

**3D** Three-dimensional

**AGL** Above Ground Level

**ANN** Artificial Neural Network

**CFD** Computational Fluid Dynamics

**CRES** Center For Renewable Energy Sources

**CV** Control Volume

**DOF** Degrees Of Freedom

**E-yield** Energy Yield

**EAS** External Area Simulation

**E-IAS** Extrapolated-Internal Area Simulation

**FDM** Finite Difference Method

**FEM** Finite Elements Method

**FVM** Finite Volume Method

**IAS** Internal Area Simulation

**LIDAR** Light Detection and Ranging

**MA** Moving Average

**MAE** Mean Absolute Error

**MAPE** Mean Absolute Percentage Error

---

**MOSVaH** Michos Optimal Shift Value Hypothesis

**MSE** Mean Square Error

**NMAE** Normalized Mean Absolute Error

**NRMSE** Normalized Root Mean Square Error

**NWP** Numerical Weather Prediction

**PDE** Partial differential equation

**RANS** Reynolds-averaged Navier–Stokes

**RMSE** Root Mean Square Error

**ShiVa** Shifted Value

**SMAPE** Symmetric Mean Absolute Percentage Error

**SS** Shifted Series

**SSE** Sum of Squared Errors

**WEEL** Wind Energy Extraction Latency forecasting model

**WF** Wind Farm

**WiSpEx** Wind Spatial Extrapolation model

**WT** Wind Turbine

# Contents

<b>Abstract</b>	<b>i</b>
Περίληψη	ii
<b>Acknowledgements</b>	<b>iii</b>
<b>Publications and Conference Participations</b>	<b>iv</b>
<b>List of Figures</b>	<b>v</b>
<b>List of Tables</b>	<b>viii</b>
<b>Abbreviations</b>	<b>x</b>
<b>I Theoretical Framework</b>	<b>1</b>
<b>1 Introduction to CFD</b>	<b>2</b>
1.1 Brief description of CFD modeling . . . . .	2
1.2 Governing Equations of Fluid Flows . . . . .	3
1.2.1 Conservation Laws . . . . .	3
1.2.2 Divergence and Gradient Forms . . . . .	10
1.3 Modeling different Flow Types . . . . .	14
1.3.1 Simplified flow modeling . . . . .	14
1.3.2 Flows and Partial Differential Equations . . . . .	21
1.4 Coordinate and Basis Vector Systems . . . . .	26
1.4.1 Coordinate Systems . . . . .	26
1.4.2 Basis Vector Systems . . . . .	27
1.4.3 Implications for Numerical Methods . . . . .	27
1.4.4 Conservation in Cartesian coordinates . . . . .	28
1.5 Discretization Methods . . . . .	30
1.5.1 Finite Difference Method (FDM) . . . . .	30
1.5.2 Finite Volume Method (FVM) . . . . .	31
1.5.3 Finite Element Method (FEM) . . . . .	32

1.5.4 Conclusion . . . . .	32
1.6 Numerical Grids . . . . .	33
1.6.1 Key Factors Influencing Grid Choice . . . . .	33
1.6.2 Classification of Numerical Grids . . . . .	35
1.6.3 Specialized Grid Types . . . . .	47
1.7 Properties of Numerical Methods . . . . .	52
1.7.1 Conservation . . . . .	53
1.7.2 Boundedness and realizability . . . . .	53
1.7.3 Consistency . . . . .	54
1.7.4 Stability . . . . .	55
1.7.5 Convergence and Accuracy . . . . .	55
1.7.6 Efficiency . . . . .	56
References . . . . .	59
<b>2 Introduction to Discretization Methods</b>	<b>60</b>
2.1 Finite Difference Method . . . . .	60
2.1.1 Fundamental Principles . . . . .	61
2.1.2 Mathematical Formulation . . . . .	62
2.1.3 Discretization of Terms in the Governing Equation . .	64
2.1.4 Conclusion . . . . .	66
2.2 Finite Volume Method . . . . .	67
2.2.1 Fundamental Principles . . . . .	67
2.2.2 Mathematical Formulation . . . . .	67
2.2.3 Discretization of Terms in the Governing Equation . .	68
2.2.4 Conclusion . . . . .	74
2.3 Finite Element Method . . . . .	74
2.3.1 Fundamental Principles . . . . .	74
2.3.2 Mathematical Formulation . . . . .	75
2.3.3 Discretization of Terms in the Governing Equation . .	78
2.4 Comparison of Methods . . . . .	82
References . . . . .	85
<b>II The CRES Experiment</b>	<b>86</b>
<b>3 CFD Modeling in COMSOL Multiphysics®</b>	<b>87</b>
3.1 Geometry and Boolean operators . . . . .	88
3.2 Mesh . . . . .	89
3.3 Material Properties . . . . .	92
3.4 The 'Turbulent k-epsilon' Interface . . . . .	93

---

3.4.1 RANS Turbulence Modeling . . . . .	94
3.4.2 k-epsilon . . . . .	96
3.4.3 Default Parameter Values . . . . .	100
3.4.4 Wall Functions . . . . .	100
3.4.5 P1+P1 Discretization . . . . .	102
3.4.6 Stationary Study implications . . . . .	106
3.4.7 Isothermal Flow implications . . . . .	107
3.5 Boundaries . . . . .	109
3.5.1 Wall Boundary . . . . .	109
3.5.2 Inlet Boundary . . . . .	110
3.5.3 Open Boundaries . . . . .	111
3.5.4 Outlet Boundary . . . . .	112
3.6 Inlet and initial conditions . . . . .	112
3.6.1 Inlet Velocity . . . . .	113
3.6.2 Initial Conditions . . . . .	116
3.6.3 Conclusion . . . . .	117
3.7 Parametric sweeps . . . . .	117
3.8 Study Solver . . . . .	118
3.9 Conclusion . . . . .	123
References . . . . .	126
<b>4 Ultra Short Term Wind Power Forecasting</b>	<b>127</b>
4.1 Experimental setup . . . . .	129
4.2 Impact of Obstacles and Simplifications on Wind Flow . . . . .	132
4.2.1 Obstacles . . . . .	132
4.2.2 CFD modeling . . . . .	134
4.3 Wi.Sp.Ex. . . . .	136
4.3.1 IAS and EAS . . . . .	137
4.3.2 E-IAS . . . . .	139
4.3.3 Results . . . . .	140
4.3.4 Assessing Wi.Sp.Ex.'s forecastability . . . . .	149
4.4 W.E.E.L. . . . .	150
4.4.1 Model Description . . . . .	150
4.4.2 Power curve . . . . .	153
4.4.3 optimal and maximum Shift detection . . . . .	158
4.4.4 Results and Discussion . . . . .	162
4.4.5 conclusion . . . . .	180
References . . . . .	186
<b>5 Conclusion &amp;Future Work suggestions</b>	<b>187</b>

5.1 WiSpEx: Unlocking the Power of Spatial Extrapolation . . . . .	187
5.2 WEEL: Power latency identification "made" simple . . . . .	189
5.3 Implications and Future Directions . . . . .	190
<b>III APPENDIX</b>	<b>193</b>
<b>A Linear Algebra</b>	<b>194</b>
A.1 An Overview of Tensors in Linear Algebra and Physics . . . . .	194
A.2 Tensor operations . . . . .	196
A.2.1 Scalar/Inner Product (Dot Product) . . . . .	196
A.2.2 Outer Product (Tensor Product) . . . . .	196
A.2.3 Vector Product (Cross Product) . . . . .	197
A.2.4 Summary of fundamental Tensor operations . . . . .	197
A.2.5 Tensor Contraction $T : S$ . . . . .	198
<b>B Power Curve</b>	<b>200</b>
B.1 Explanation of WEEL's POWER CURVE . . . . .	200
B.2 Complementary information on approximation . . . . .	202
Bibliography . . . . .	212

# **Part I**

# **Theoretical Framework**

# Chapter 1

## Introduction to CFD

### 1.1 Brief description of CFD modeling

Computational Fluid Dynamics (CFD) employs numerical simulations to analyze fluid flows, complementing experimental and theoretical approaches. It's necessary for studying complex flows lacking analytical solutions due to intricate geometries, external forces, or fluid properties. A studied flow behavior is estimated by numerically solving governing equations, representing the flow field with variables like velocity, pressure, density, and temperature at discrete points.

CFD is often faster, safer, and more economical than experiments while it enables the analysis of Complex Phenomena that are challenging for experimental measurements or theoretical analysis. It especially allows researchers to explore and develop novel modeling approaches. While real-world implementation might be constrained by current hardware limitations, this freedom fosters innovation and paves the way for future advancements". CFD finds applications in diverse fields, such as aerospace, automotive, HVAC, marine, turbo—machinery, chemical & process engineering, meteorology, oceanography, hydrology, cardiovascular medicine, respiratory medicine, drug delivery, and sports.

While CFD has seen significant advancements, the development of fast, accurate, widely applicable, and robust simulation methods remains an ongoing pursuit. The fast pace of technological evolution, has given us the opportunity to bring down the computational time cost. This is important, as CFD usage will gradually increase its dominance over statistical methods for real-time applications, that demand precise modeling of physical interactions. That is the case for ultra-short-term forecasting where the integration of thermal conditions, airflow, and optical parameters is

essential for capturing the true complexity of the system.

The general CFD Numerical simulations of fluid flow typically involve the following steps (Kajishima and Taira, 2016):

- **Selection of Governing Equations:** The first step is to choose the appropriate equations that describe the physics of the flow. This could involve selecting the full Navier-Stokes equations, the inviscid approximation, or other simplified models. If necessary, a turbulence model and a non-Newtonian constitutive equation may also be chosen.
- **Discretization:** The governing partial differential equations are then discretized using methods such as finite difference, finite volume, or finite element methods. A suitable grid is also chosen for spatial discretization. This process leads to a system of algebraic equations.
- **Numerical Solution and Visualization:** The algebraic equations are solved using numerical algorithms, and the resulting solutions are visualized using graphs, animations, or other techniques. The general process can be summarized as follows:

In addition to fluid mechanics, knowledge of numerical analysis and computer science is essential for conducting fluid flow simulations. Verification and validation are also critical steps to ensure the reliability of the simulation results.

## 1.2 Governing Equations of Fluid Flows

In this section we discuss the conservation laws, closure of the governing equations, and the divergence and gradient forms of the equations to present the mathematical foundations of CFD.

### 1.2.1 Conservation Laws

Our understanding of fluid dynamics is based on the fundamental principles of conservation - the undeniable belief that mass, momentum, and energy can neither be created nor destroyed, only transformed or transferred. These conservation laws, expressed mathematically, form the governing equations that dictate the behavior of fluids, providing the framework for both theoretical analysis and numerical simulations.

In fluid dynamics, the distinction between intensive and extensive properties is crucial for understanding conservation laws. An intensive

property  $\theta$  is independent of the amount of matter, while an extensive property  $\Theta$  scales with the system's size. The equation

$$\Theta = \iiint_{V_{CM}} \rho \theta \, dV$$

demonstrates how to calculate an extensive property  $\Theta$  from a corresponding conserved intensive property  $\theta$  by integrating over the control mass volume  $V_{CM}$ , considering the density  $\rho$  and highlights the fundamental connection between intensive and extensive properties in expressing conservation principles in fluid mechanics. For generalization purposes, when a variable  $\theta$  may be a scalar or a vector, the notation  $\vec{x}\theta$  will be used for the outer product when  $\theta$  is a vector (like velocity  $\vec{u}$ ) and simple multiplication if  $\theta$  is a scalar (see Appendix A).

In Kajishima and Taira, 2016 (Sections 1.3.1 to 1.3.5) and Ferziger and Perić, 2020 (Sections 2.1 to 2.3), these equations are derived by applying the fundamental conservation principles to a fluid continuum. They start by considering an arbitrary Control Volume (CV) within the fluid and then express the conservation laws in integral form by "dividing" that CV into the inner volume  $V_C$  and Surface  $S_C$ . Subsequently, they employ mathematical tools like Gauss's divergence theorem and the Reynolds transport theorem to transform the integral equations into their equivalent differential forms, which express the conservation laws at a point within the fluid.

We can imagine a fluid with density  $\rho$ , flowing through a CV with volume  $V_C$ , a defined region in space, enclosed by  $S_C$ . Within this volume, a property of the fluid can change over time due to two primary factors:

- **Temporal Change:** A fluid property  $\theta$  within the volume can increase or decrease as density changes over time. The rate of change of  $\theta$  within the CV is:

$$\iiint_{V_C} \frac{\partial \rho \theta}{\partial t} \, dV$$

- **Flux:** A fluid property  $\theta$  can enter or leave the volume through its boundaries, carried by the fluid's velocity. The flux of  $\theta$  through the surface  $\Phi$  is :

$$\iint_{S_C} \rho \vec{\theta}_{flux} \cdot \vec{n} \, dS$$

where  $\vec{\theta}_{flux} = \theta \vec{u}$  is the flux vector of the quantity  $\theta$ , with  $\vec{u}$  representing the velocity (with respect to the CV's velocity) and  $\vec{n}$  the outward-pointing unit normal vector on the surface  $S_C$ .

Various physical processes such as chemical reactions, heat sources, or

external inputs can affect the quantity  $\theta$ . By using a net source term (or sink term)  $\Phi$  within the CV, we account for any generation or absorption of  $\theta$  within the volume :

$$\iiint_{V_C} \Phi dV$$

### Integral and differential Form of the Conservation Equation

The principle of conservation states that the rate of change of  $\theta$  within the CV equals the net flux of  $\theta$  across the control surface plus the net source term acting within the CV. The conservation principle can be applied to any quantity  $\theta$ , which may be a scalar or vector field.

Applying the conservation principle, we obtain:

$$\iiint_{V_C} \frac{\partial \rho \theta}{\partial t} dV = - \iint_{S_C} \rho \vec{\theta}_{flux} \cdot \vec{n} dS + \iiint_{V_C} \Phi dV \quad (1.2.1.1)$$

considering that influx is negative. According to Gauss's divergence theorem, we can transform the surface integral into a volume integral, leading to:

$$\iiint_{V_C} \left( \frac{\partial \rho \theta}{\partial t} + \nabla \cdot (\rho \vec{\theta}_{flux}) + \Phi \right) dV = 0 \quad (1.2.1.2)$$

The term  $\Phi$  accounts for the body forces (gravity, centrifugal and Coriolis forces, electromagnetic forces, etc.), and the surface forces (pressure, normal and shear stresses, surface tension etc.).

Since this equation must hold for any arbitrary CV, the integrand itself must be zero, yielding the differential form conservation for  $\theta$  :

$$\frac{\partial \rho \theta}{\partial t} + \nabla \cdot (\rho \vec{\theta}_{flux}) + \Phi = 0 \quad (1.2.1.3)$$

In many cases and especially for a scalar quantity  $\theta$  (like in Energy conservation),  $\Phi$  includes diffusion transport, which is described by a gradient approximation (e.g., Fourier's law for heat diffusion or Fick's law for mass diffusion). The diffusion flux  $f_\theta$  can be represented as:

$$f_\theta = \iint_{S_C} \Gamma_\theta \nabla \theta \cdot \vec{n} dS$$

where  $\Gamma_\theta$  is the diffusivity for the quantity  $\theta$ .

We are now ready to derive the conservation laws for mass, momentum and energy, leading to a system of equations that governs the dynamics of

fluid flow, based on Kajishima and Taira, 2016, instead of just unifying all possible source terms in a single source function. These laws are not mere mathematical abstractions but have deep physical roots. To simplify the rationale behind the CFD modeling framework, instead of using a single source function, we will assume Newtonian fluid behavior, incompressibility, isotropic stress, Fourier's law, and the ideal gas law. This way we simplify the conservation equations, making them solvable in a wide range of practical applications. However, it is worth mentioning that for more complex phenomena, such as non-Newtonian fluids, compressible flows, or high-speed aerodynamics, these assumptions may break down, requiring more sophisticated models and numerical methods. If  $p$  is the pressure,  $\tau$  is the viscous stress tensor.  $\vec{F}_b$  is the body forces acting on the fluid,  $E$  is the total energy per unit mass and  $q$  is the heat flux vector:

### 1. Conservation of Mass (Continuity Equation)

This law stems from the fundamental principle that matter cannot be created or destroyed. In the context of fluid flow, it implies that the mass within a CV can only change due to the net flow of mass across its boundaries. Mathematically, this is represented by the continuity equation, which can be expressed in both integral and differential forms by replacing  $\theta$  with the mass  $m = 1$ , where  $\vec{\theta}_{\text{flux}} = m\vec{u} = \vec{u}$ , and  $\Phi$  is typically zero, assuming no sources or sinks of mass:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \quad (1.2.1.4)$$

This equation states that the rate of change of density at a point plus the divergence of the mass flux  $\rho \vec{u}$  at that point must be zero.

### 2. Conservation of Momentum (Newton's Second Law)

This law is rooted in Newton's second law of motion, which states that the rate of change of momentum of a body is equal to the net force acting on it. In fluid flows, momentum can change due to both external forces (like gravity or pressure gradients) and internal forces (arising from viscous stresses within the fluid). Mathematically, this is expressed by the momentum equation, when replacing  $\theta$  with the momentum  $m\vec{u}$ , where  $\vec{\theta}_{\text{flux}} = m\vec{u}\vec{u}$  (momentum flux), and  $\Phi$  can include body forces  $\vec{F}_b$  (e.g., gravity) and surface forces (e.g., pressure and viscous stresses). For  $m = 1$ :

$$\frac{\partial(\rho \vec{u})}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) = \nabla \cdot \tau + \rho \vec{F}_b \quad (1.2.1.5)$$

The left side of the equation represents the rate of change of momentum of the fluid parcel, while the right side represents the sum viscous forces, where  $\tau$  represents the momentum exchange due to the stress at the surface of the CV, and body forces acting on it.

### 3. Conservation of Energy (First Law of Thermodynamics)

This law derives from the first law of thermodynamics, which states that energy is conserved in an isolated system. In fluid flows, energy can be transported by the fluid's motion (kinetic energy), stored within the fluid (internal energy), and transferred as heat or work. Mathematically, this is expressed by the energy equation produced by replacing  $\theta$  with the total energy  $E$ , where  $\vec{\theta}_{\text{flux}} = E\vec{u}$  (energy flux), and  $\Phi$  can include heat sources, work done by external forces, and any other sources or sinks of energy:

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot (\rho E \vec{u}) = \nabla \cdot (\tau \cdot \vec{u}) + \nabla \cdot \vec{q} + \rho \vec{u} \cdot \vec{F}_b \quad (1.2.1.6)$$

The left side of the equation represents the rate of change of energy of the fluid parcel, while the right side represents the net rate of heat addition, the work done by viscous forces, and the work done by body forces.

It is worth mentioning that the Second Law of Thermodynamics and entropy play a role in understanding irreversibility and energy loss in fluid systems, but for most CFD applications, the focus remains on solving the conservation of mass, momentum, and energy. The Second Law becomes crucial in compressible flows and turbulent heat transfer when understanding efficiency and irreversibility is key. In this Thesis we focus on incompressible and isothermal turbulent flow modeling, where the primary focus is on the conservation of mass, momentum, and energy, forming the core of the Navier-Stokes equations. The momentum and energy equations already account for irreversibilities like viscous dissipation through terms like viscous stress and dissipation functions, while Entropy increases naturally in turbulent flows due to dissipation of turbulent kinetic energy.

The Navier-Stokes equations, coupled with appropriate constitutive relations and boundary conditions, provide the mathematical framework for understanding and predicting the behavior of fluids in various physical phenomena. By connecting these conservation laws to their underlying physical principles, we can foster a deeper appreciation for their significance and their role in shaping the behavior of fluids. We can also highlight

the challenges and complexities involved in accurately representing and solving these equations numerically, setting the stage for the subsequent chapters on discretization methods and numerical algorithms.

To close the system of equations, we need to express stress and heat flux in terms of the primary variables. For a Newtonian fluid the conservation equations written together as a system in their differential form are:

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho \vec{u} \\ \rho E \end{bmatrix} + \nabla \cdot \begin{bmatrix} \rho \vec{u} \\ \rho \vec{u} \otimes \vec{u} - \tau \\ \rho E \vec{u} - \tau \cdot \vec{u} + \vec{q} \end{bmatrix} + \begin{bmatrix} 0 \\ \rho \vec{F}_b \\ \rho \vec{u} \cdot \vec{F}_b \end{bmatrix} = 0 \quad (1.2.1.7)$$

Here,  $\rho \vec{u} \otimes \vec{u} - \tau$  is the momentum flux tensor, where  $\rho \vec{u} \otimes \vec{u}$  is the flux momentum at velocity  $\vec{u}$  and the stress tensor  $\tau$  describes the internal forces acting within a fluid at any given point. It represents the distribution of stress (force per unit area) across different directions within the fluid, connecting the fluid's motion to the forces acting within it. Its mathematical expression is:

$$\tau = -p\mathbf{I} + 2\mu(D - 1/3I\nabla \cdot u) \quad (1.2.1.8)$$

The term  $-p\mathbf{I}$ , where  $\mathbf{I}$  is the Identity Tensor, represents the isotropic stress due to pressure  $p$ , while the negative sign indicates that pressure  $p$  acts to compress the fluid.

The term  $2\mu(D - 1/3I\nabla \cdot u)$  is the viscous stress, representing the resistance of the fluid to shear deformation, where :

- $D = \frac{1}{2}[(\nabla \vec{u}) + (\nabla \vec{u})^T]$  is the rate of deformation tensor or strain rate tensor, describing how the fluid is deforming or changing its shape over time. It quantifies the rate at which different parts of the fluid are moving relative to each other.
- The term  $1/3I\nabla \cdot u$  is used to subtract out the isotropic part of the deformation rate, ensuring that only the deviatoric part (shear deformation) contributes to the viscous stresses. This is important for incompressible flows where the volume of a fluid element remains constant.
- The dynamic viscosity  $\mu$  often simply referred to as viscosity, determines the magnitude of the fluid's resistance to flow or deformation under shear stress. In simpler terms, it measures how "thick" or "sticky" a fluid is.

The heat flux  $q$  represents the rate of heat energy transfer per unit area. It quantifies how much heat is flowing through a particular surface within the fluid and is expressed via Fourier's law for heat conduction:

$$\vec{q} = -k\nabla T \quad (1.2.1.9)$$

where,  $k$  is the thermal conductivity, and  $T$  is the absolute temperature

The total energy  $E$  is:

$$E = E_U + E_K \quad (1.2.1.10)$$

where, kinetic energy  $E_K$  is the energy of an object due to its motion. For an object with mass  $m$  and velocity  $\vec{v}$ :

$$E_k = \frac{1}{2}m\|\vec{v}\|^2 = \frac{1}{2}m(\vec{v} \cdot \vec{v}) \quad (1.2.1.11)$$

and internal energy  $E_U$  is the total energy contained within a system due to the random motions of its particles, including translational, rotational, vibrational motion, and potential energy interactions between particles. It is a scalar quantity. In specific cases we can derive an expression with the use of equation of state.

The equation of state for fluids describes the relationship between their thermodynamic properties, pressure  $p$ , volume  $V$ , temperature  $T$ . For different types of fluids, these relationships can vary significantly, particularly between incompressible and compressible flows. For an ideal gas, this relationship (The Ideal Gas Law) is given by:

$$pV = nRT \quad (1.2.1.12)$$

where  $n$  is the number of moles and  $R$  is the universal gas constant ( $R \approx 8.314 \text{ J} \cdot \text{mol}^{-1} \cdot \text{K}^{-1}$ ). We want to use density  $\rho = n/V$  to obtain a direct relation between  $T$ ,  $p$  and  $\rho$ :

$$p = \rho RT \quad (1.2.1.13)$$

In thermodynamics, for an ideal gas, the specific heat ratio  $\gamma$  is defined as:

$$\gamma = \frac{c_p}{c_v} \quad (1.2.1.14)$$

where  $c_v$  is the heat capacity at constant volume and  $c_p$  is the heat capacity at constant pressure. The relationship between  $c_p$  and  $c_v$  is given

by:

$$c_p = c_v + R \quad (1.2.1.15)$$

Defining the internal energy  $E_U$  of an ideal gas as

$$E_U = c_v T \quad (1.2.1.16)$$

where  $dE_U = c_v dT$  is its differential form, and by using relationships, a new expression can be derived for the equation of state:

$$p = (\gamma - 1)\rho E_U \quad (1.2.1.17)$$

We have now expressed the system of conservation equations in terms of the variables  $\rho$ ,  $u$ , and  $E$ . Consequently, the number of equations is equal to the number of unknowns, thus closing the system of equations.

## 1.2.2 Divergence and Gradient Forms

In fluid dynamics, we mostly analyze how physical quantities like mass, momentum, and energy change and move within a fluid. This is achieved by using two main mathematical forms, called the divergence form and the gradient form. Both forms are crucial for understanding the behavior of fluids and are used in different numerical methods for simulations.

### Derivation of Divergence and Gradient Forms

To analyze how a physical quantity  $\theta$  changes and moves with the fluid (without the presence of sources or sinks), we decompose the time rate of change and transport as follows:

$$\frac{\partial(\rho\theta)}{\partial t} + \nabla \cdot (\rho\vec{u}\theta) = \rho \left( \frac{\partial\theta}{\partial t} + \vec{u} \cdot \nabla \theta \right) + \theta \left( \frac{\partial\rho}{\partial t} + \nabla \cdot (\rho\vec{u}) \right) \quad (1.2.2.1)$$

Let's break down the terms in this equation:

- **Left Side:**

- $\frac{\partial(\rho\theta)}{\partial t}$  represents the local rate of change of  $\theta$  within the CV over time.
- $\nabla \cdot (\rho\vec{u}\theta)$  captures the flux of  $\theta$  through the boundaries of the CV, influenced by the fluid's velocity  $\vec{u}$ .

- **Right Side:**

- $\rho \left( \frac{\partial \theta}{\partial t} + \vec{u} \cdot \nabla \theta \right)$  is the total rate of change of  $\theta$  as seen from within a fluid particle, often referred to as the material derivative.
- $\theta \left( \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) \right)$  accounts for changes due to variations in density and the divergence of the velocity field.

Using the continuity equation  $\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0$ , the term  $\theta \left( \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) \right)$  simplifies to zero. We can then define the material derivative  $\frac{D\theta}{Dt}$ , which describes the rate of change of a property  $\theta$  experienced by a moving fluid particle, as:

$$\frac{D\theta}{Dt} \equiv \frac{\partial \theta}{\partial t} + \vec{u} \cdot \nabla \theta \quad (1.2.2.2)$$

This leads to the simplified equation in its divergence (or conservative) form:

$$\frac{\partial(\rho\theta)}{\partial t} + \nabla \cdot (\rho \vec{u} \theta) = \rho \frac{D\theta}{Dt} \quad (1.2.2.3)$$

This form is called the divergence form (or conservative form) because it emphasizes the conservation of  $\theta$  by considering its flux through a CV.

The gradient (non-conservative or advective) form of the equation is:

$$\rho \frac{D\theta}{Dt} = \rho \left( \frac{\partial \theta}{\partial t} + \vec{u} \cdot \nabla \theta \right) \quad (1.2.2.4)$$

In this form, the focus is on the changes in  $\theta$  experienced by a fluid particle as it moves with the flow. The term  $\vec{u} \cdot \nabla \theta$  represents the advection of  $\theta$ , or how  $\theta$  changes spatially as the fluid flows.

The divergence and gradient forms are mathematically equivalent, connected through the product rule of differentiation. For instance, expanding the divergence term in the conservation of mass equation yields:

$$\nabla \cdot (\rho \vec{u}) = \rho \nabla \cdot \vec{u} + \vec{u} \cdot \nabla \rho \quad (1.2.2.5)$$

Substituting this back into the original equation and rearranging terms leads to an alternative form of the continuity equation:

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \vec{u} = 0 \quad (1.2.2.6)$$

This equation, expressed in terms of the material derivative of density and the divergence of velocity, can be interpreted as the rate of change of

density following a fluid particle plus the density times the rate of volume expansion of the fluid particle equaling zero.

### Divergence and Gradient Forms of the Conservation equations

Both the divergence and gradient forms are mathematically equivalent ways to describe the same physical phenomenon. In numerical simulations, choosing the appropriate form depends on the method and may affect accuracy and stability. The divergence form is often used in FVM because it aligns with conservation principles, ensuring the conservation of mass, momentum, and energy at the discrete level. This makes it crucial for accurate and stable simulations, as fluxes across the faces of a CV are naturally discretized. On the other hand, the gradient form is typically used in FDM, where derivatives are directly computed at grid points. This form can be advantageous for certain types of boundary conditions and can lead to simpler discretization schemes.

In addition to FDM and FVM, another powerful numerical technique used in computational fluid dynamics is Finite Elements Method (FEM), that can use both forms depending on the focus and is particularly advantageous for complex geometries. When simulating wind flows over complex regions, the divergence form might be used to ensure that the overall mass flow into and out of the region remains balanced, while the gradient form could help in capturing the sharp changes in wind speed and direction as the airflow interacts with the terrain. This dual approach not only enhances the accuracy of the simulation but also provides insights into how terrain features influence flow dynamics. FEM a robust choice for analyzing fluid flows in challenging environments like the one presented in this chapter. All three methods will be described in detail in the next chapter.

Both the divergence and gradient forms of the conservation equations provide valuable insights into the behavior of fluid flows. Understanding both forms is crucial for accurate and efficient fluid dynamics simulations that produce meaningful simulation results. Ferziger and Perić, 2020 (Chapter 4, Sections 4.1 and 4.2) provide further insights into the divergence and gradient forms of the governing equations. They emphasize the importance of choosing the appropriate form depending on the numerical method used for discretization.

We can now present the already discussed conservation laws in both forms:

- **Mass Conservation**

- (a) **Divergence Form:**

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) = 0 \quad (1.2.2.7)$$

This form emphasizes the divergence of the mass flux  $\rho \vec{u}$ , representing the net flow of mass into or out of a CV.

- (b) **Gradient Form:**

$$\frac{D\rho}{Dt} = -\rho \nabla \cdot \vec{u} \quad (1.2.2.8)$$

This form describes the rate of change of density following a fluid particle, considering the expansion or contraction of the fluid.

- **Momentum Conservation**

- (a) **Divergence Form:**

$$\frac{\partial(\rho \vec{u})}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u} - \tau) = \rho \vec{F}_b \quad (1.2.2.9)$$

In this form,  $\rho \vec{u} \vec{u}$  represents the flux of momentum, and  $\nabla \cdot (\rho \vec{u} \vec{u} - \tau)$  accounts for the net force acting on a CV.

- (b) **Gradient Form:**

$$\rho \frac{D\vec{u}}{Dt} = \nabla \cdot \tau + \rho \vec{F}_b \quad (1.2.2.10)$$

This form highlights the forces acting on a fluid particle and their effect on its acceleration.

- **Energy Conservation**

- (a) **Divergence Form:**

$$\frac{\partial(\rho E)}{\partial t} + \nabla \cdot (\rho E \vec{u} - \tau \cdot \vec{u} + \vec{q}) = \rho \vec{u} \cdot \vec{F}_b \quad (1.2.2.11)$$

This form emphasizes the divergence of the energy flux and accounts for the heat and work terms.

- (b) **Gradient Form:**

$$\rho \frac{DE}{Dt} = \nabla \cdot (\tau \cdot \vec{u}) - \nabla \cdot \vec{q} + \rho \vec{u} \cdot \vec{F}_b \quad (1.2.2.12)$$

This form highlights the rate of change of energy experienced by a fluid particle and includes terms for the work done by the stress tensor, heat conduction, and work done by body forces.

Total energy is the sum of Kinetic and Internal Energy. To obtain the conservation of kinetic energy in divergence form we have to take inner product of the momentum equation (divergence form) with  $\vec{u}$ :

$$\rho \vec{u} \cdot \frac{D\vec{u}}{Dt} = \vec{u} \cdot (\nabla \cdot \tau) + \rho \vec{u} \cdot \vec{F}_b$$

The term  $\rho \vec{u} \cdot \frac{D\vec{u}}{Dt}$  is the rate of change of kinetic energy per unit volume, leading to the kinetic energy equation in divergence form:

$$\rho \frac{DE_K}{Dt} = \vec{u} \cdot (\nabla \cdot \tau) + \rho \vec{u} \cdot \vec{F}_b$$

### Divergence forms of Internal and Kinetic Energy

The internal energy  $E_U$  is expressed by :

$$E_U = E - E_K$$

Subtracting the kinetic energy equation from the total energy equation gives the conservation of internal energy:

$$\rho \frac{DE}{Dt} - \rho \frac{Dk}{Dt} = \nabla \cdot (\tau \cdot \vec{u}) - \nabla \cdot \vec{q} - \vec{u} \cdot (\nabla \cdot \tau)$$

With use of the  $E_U = E - E_K$ , we can obtain the internal energy equation in divergence form:

$$\rho \frac{DE_U}{Dt} = \tau : (\nabla \vec{u}) - \nabla \cdot \vec{q}$$

where,  $\tau : (\nabla \vec{u})$  is the contraction of the stress tensor  $\tau$  with the velocity gradient tensor  $\nabla \vec{u}$  (see Chapter A). This term represents the work done by the viscous forces in the fluid, which is converted into internal energy (i.e., heat) due to viscous dissipation.  $\nabla \cdot \vec{q}$  is the divergence of the heat flux, representing heat conduction.

## 1.3 Modeling different Flow Types

### 1.3.1 Simplified flow modeling

The mass and momentum conservation equations, which are described by the Navier-Stokes equations, are complex and nonlinear. Analytical solutions are feasible only for simple geometries and fully developed flows.

For practical purposes, numerical methods are often employed to solve these equations (Kajishima and Taira, 2016 and Ferziger and Perić, 2020). Simplifications can reduce computational effort and are used for various flow types. Complex phenomena such as turbulence, combustion, and multiphase flows often require semi-empirical models (COMSOL Multiphysics® (CFD), 2020, COMSOL Multiphysics® (Reference Manual), 2020). These models, while useful, introduce errors that must be carefully managed.

For air, which can be approximated as an ideal gas under many conditions, the equation of state helps in understanding how changes in temperature and volume affect pressure. The behavior of fluids can be classified based on compressibility. Understanding whether a fluid is compressible or incompressible helps in selecting the appropriate model and equation.

In compressible flows, the fluid density changes significantly with pressure and temperature, requiring more complex equations to accurately describe their behavior. This is typical for gases at high speeds or under large pressure variations. The Ideal Gas Law remains a key tool for describing such flows. For more accurate modeling, particularly at high pressures and low temperatures, equations based on experimental data like the Van der Waals equation (Zemansky and Dittman, 1997) is used:

$$\left(p + \frac{a}{V^2}\right)(V - b) = nRT$$

where,  $a$  = Van der Waals constant for molecular attraction,  $b$  = Van der Waals constant for molecular volume.

For incompressible flows, the density of the fluid remains nearly constant despite changes in pressure, and the fluid behavior is often simplified for easier analysis. This approximation is commonly applied to liquids and for gases at low speeds where density variations are minimal. For incompressible fluids, the pressure variation in a gravitational field is described by:

$$p_h = p_0 + \rho gh$$

where,  $p_h$  is Pressure at height  $h$ ,  $p_0$  is Reference pressure at height  $h_0 = 0$ ,  $\rho$  is Constant density,  $g$  is Gravitational acceleration and  $h$  is Height or depth. In this dissertation we will focus on numerical solutions of incompressible flows but we will give a brief presentation of other flow simplifications.

### Incompressible Flow

Incompressible flow assumes that the fluid density  $\rho$  remains constant along a fluid pathline(Kajishima and Taira, 2016), which is valid for many liquids and gases with Mach numbers below 0.3. For isothermal flows (constant temperature), the viscosity  $\mu$  is also constant, leading to simplifications of the Navier-Stokes equations. For incompressible flow, the material derivative of density is zero:

$$\frac{D\rho}{Dt} = \frac{\partial\rho}{\partial t} + \vec{u} \cdot \nabla\rho = 0. \quad (1.3.1.1)$$

Note that incompressibility does not necessarily mean that the density  $\rho$  is constant, but rather that its rate of change is zero.

The incompressibility condition, also known as the divergence-free constraint, simplifies the continuity equation to:

$$\nabla \cdot \vec{u} = 0. \quad (1.3.1.2)$$

ensuring that the volumetric flux is conserved at each point in time. The stress tensor  $\tau$  in the momentum equation for a Newtonian and incompressible fluid is given by:

$$\tau = -p\mathbf{I} + 2\mu\mathbf{D}, \quad (1.3.1.3)$$

The momentum equation becomes:

$$\rho \frac{\partial \vec{u}}{\partial t} + \nabla \cdot (\vec{u} \otimes \vec{u}) = -\nabla p + \nabla \cdot (2\mu\mathbf{D}) + \rho \vec{F}_b, \quad (1.3.1.4)$$

where  $\vec{u} \otimes \vec{u}$  denotes the outer product of the velocity vector with itself, and  $\vec{F}_b$  represents body forces per unit mass (e.g., gravity).

If the viscosity  $\mu$  is constant, the rate-of-strain tensor  $\mathbf{D}$  simplifies to:

$$\mathbf{D} = \frac{1}{2}(\nabla \vec{u} + (\nabla \vec{u})^T). \quad (1.3.1.5)$$

and the term  $\nabla \cdot (2\mu\mathbf{D})$  simplifies to:

$$\nabla \cdot (2\mu\mathbf{D}) = \mu \nabla^2 \vec{u}, \quad (1.3.1.6)$$

where  $\nabla^2$  denotes the Laplacian operator, applied to each component of

the velocity vector. Thus, the momentum equation simplifies to:

$$\frac{\partial \vec{u}}{\partial t} + \nabla \cdot (\vec{u} \otimes \vec{u}) = -\frac{\nabla p}{\rho} + \nu \nabla^2 \vec{u} + \vec{F}_b, \quad (1.3.1.7)$$

where  $\nu = \frac{\mu}{\rho}$  is the kinematic viscosity, and  $\nabla^2$  denotes the Laplacian operator.

If we take the divergence of Eq. (1.3.1.7) and apply the divergence operator, we get:

$$\nabla \cdot \left( \frac{\partial \vec{u}}{\partial t} \right) + \nabla \cdot (\nabla \cdot (\vec{u} \otimes \vec{u})) = -\nabla \cdot \left( \frac{\nabla p}{\rho} \right) + \nabla \cdot (\nu \nabla^2 \vec{u}) + \nabla \cdot \mathbf{F}_b. \quad (1.3.1.8)$$

Since  $\frac{\partial(\nabla \cdot \vec{u})}{\partial t} = 0$  due to incompressibility (from Eq. (1.3.1.2)), this simplifies to:

$$\nabla \cdot (\nabla \cdot (\vec{u} \otimes \vec{u})) = -\nabla \cdot \left( \frac{\nabla p}{\rho} \right) + \nabla \cdot (\nu \nabla^2 \vec{u}) + \nabla \cdot \mathbf{F}_b. \quad (1.3.1.9)$$

Now, the term  $\nabla \cdot \left( \frac{\nabla p}{\rho} \right)$  simplifies to:

$$\nabla \cdot \left( \frac{\nabla p}{\rho} \right) = \frac{\nabla^2 p}{\rho}. \quad (1.3.1.10)$$

Substituting this into the previous equation, and noting that  $\nabla \cdot (\nu \nabla^2 \vec{u}) = \nu \nabla^2 (\nabla \cdot \vec{u}) = 0$  because  $\nabla \cdot \vec{u} = 0$ , yields:

$$\nabla \cdot (\nabla \cdot (\vec{u} \otimes \vec{u})) = -\frac{\nabla^2 p}{\rho} + \nabla \cdot \mathbf{F}_b. \quad (1.3.1.11)$$

Rearranging this equation gives us the pressure Poisson equation:

$$\frac{\nabla^2 p}{\rho} = -\nabla \cdot (\nabla \cdot (\vec{u} \otimes \vec{u})) + \nabla \cdot \mathbf{F}_b. \quad (1.3.1.12)$$

This equation relates the pressure field to the velocity field and body forces, and is an elliptic partial differential equation that must be solved as a boundary value problem in numerical simulations. Unlike compressible flow, where pressure is determined by the equation of state, pressure in incompressible flow is not derived from thermodynamics but rather from solving Eq. (1.3.1.12).

The conservation of internal energy for incompressible flow is expressed as:

$$\rho \frac{D E_U}{D t} = \mu \mathbf{D} : \mathbf{D} - \nabla \cdot \mathbf{q}, \quad (1.3.1.13)$$

where  $\mathbf{D} : \mathbf{D}$  denotes the double dot product of the rate-of-strain tensor

with itself.

For temperature field, using the thermal conductivity  $k$  and specific heat capacity at constant volume  $c_v$ , we have:

$$\rho c_v \frac{DT}{Dt} = \mu \mathbf{D} : \mathbf{D} + k \nabla^2 T. \quad (1.3.1.14)$$

If the effect of friction on temperature is negligible, the temperature equation simplifies to:

$$\rho c_v \frac{DT}{Dt} = k \nabla^2 T. \quad (1.3.1.15)$$

while, kinetic and internal energies can be considered separately, simplifying the analysis of incompressible flows. In this context, the temperature equation becomes the primary governing equation for thermal analysis.

### Inviscid (Euler) Flow

For flows where viscous effects are negligible (e.g., far from solid surfaces), the Navier-Stokes equations simplify to the Euler equations. The continuity equation remains the same, while the momentum equation becomes:

$$\frac{\partial(\rho \vec{u})}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) = -\nabla p + \vec{F}_b. \quad (1.3.1.16)$$

Inviscid fluids do not stick to walls, allowing slip at boundaries. Euler equations are useful for high-speed compressible flows where viscous effects are minimal near walls. Though challenging, they facilitate simulations over large domains with coarser grids, though resolving boundary layers requires more resources.

### Potential Flow

Potential flow models assume an inviscid and irrotational flow, leading to:

$$\nabla \times \mathbf{v} = 0. \quad (1.3.1.17)$$

The velocity vector  $\mathbf{v}$  can be expressed as the gradient of a scalar potential  $\Phi$ :

$$\mathbf{v} = -\nabla \Phi, \quad (1.3.1.18)$$

resulting in the Laplace equation for  $\Phi$ :

$$\nabla^2 \Phi = 0. \quad (1.3.1.19)$$

The Bernoulli equation can be derived from integrating the momentum equation. Potential flow solutions are described by the Laplace equation and are limited to simple geometries. Although potential flow cannot accurately model drag or lift (D'Alembert's paradox), it provides valuable insights into flow patterns and can be used for more complex flows by combining solutions.

### Creeping (Stokes) Flow

In creeping flow (low Reynolds number), inertial forces are negligible compared to viscous, pressure, and body forces. For constant fluid properties, the momentum equations reduce to the Stokes equations:

$$\nabla \cdot (\mu \nabla u_i) - \frac{1}{\rho} \nabla p_i + b_i = 0 \quad (1.3.1.20)$$

Creeping flow is common in porous media, coating processes, and micro-devices. The equations are linear and simpler to solve compared to general Navier-Stokes equations.

### Creeping (Stokes) Flow

For low Reynolds number flows where inertial forces are negligible compared to viscous, pressure, and body forces and for constant fluid properties, the momentum equations reduce to:

$$\nabla \cdot (\mu \nabla \vec{u}) - \frac{1}{\rho} \nabla p + \vec{F}_b = 0. \quad (1.3.1.21)$$

This simplified form is useful in porous media, coating processes, and micro-devices. The equations are linear and simpler to solve compared to general Navier-Stokes equations.

### Boussinesq Approximation

In flows with significant heat transfer where density varies with temperature, the Boussinesq approximation treats density as constant except in the gravitational term:

$$(\rho - \rho_0)g = -\rho_0 g \beta (T - T_0). \quad (1.3.1.22)$$

where  $\beta$  is the thermal expansion coefficient and the zero indices indicate the reference values of the given property. This approximation introduces

minor errors for small temperature differences but can be significant for larger differences.

### Boundary Layer Approximation

The boundary layer approximation is a significant simplification of the Navier-Stokes equations, applied to flows where one direction is dominant and the geometry changes gradually. This scenario is common in flows through channels, over flat plates, or around mildly curved surfaces. The approximation is based on the following key assumptions:

- **Negligible Diffusive Transport in the Main Flow Direction:** The transport of momentum due to diffusion (viscous effects) in the primary flow direction is much smaller compared to the transport due to convection. This means that viscous forces in the direction of the main flow can be ignored relative to inertial forces.
- **Dominant Velocity Component in the Main Flow Direction:** The velocity component along the primary flow direction is significantly larger than the velocity components in the directions perpendicular to it. This leads to the conclusion that changes in velocity in the primary flow direction dominate the flow behavior.
- **Small Pressure Gradient Normal to the Flow Direction:** The pressure variation perpendicular to the main flow direction is much smaller than the pressure variation along the flow direction. Therefore, the pressure gradient across the flow (normal direction) can be considered negligible.

With these assumptions, the boundary layer equations, which are simplified forms of the Navier-Stokes equations, are derived. These equations primarily focus on the dominant flow direction and include the effects of viscosity in the direction normal to the flow. The resulting equation is:

$$\frac{\partial(\rho\vec{u})}{\partial t} + \nabla \cdot (\rho\vec{u} \otimes \vec{u}) = \mu \frac{\partial^2 \vec{u}}{\partial n^2} - \nabla p, \quad (1.3.1.23)$$

In this equation:

-  $\vec{u}$  represents the velocity vector of the fluid. -  $\rho$  is the fluid density, which is assumed to be constant in many practical applications. -  $\vec{u} \otimes \vec{u}$  denotes the outer product of the velocity vector with itself, which accounts for the convective transport of momentum. -  $\frac{\partial^2 \vec{u}}{\partial n^2}$  is the term representing

the viscous diffusion of momentum in the direction normal to the surface (denoted by  $n$ ). This captures the effect of viscosity within the boundary layer.  $-\nabla p$  is the pressure gradient, with the assumption that the gradient in the direction normal to the surface is negligible, focusing instead on the pressure changes along the main flow direction.

The boundary layer approximation is widely used in aerodynamics and engineering to model flows over surfaces where separation does not occur, Ferziger and Perić, 2020. This approach simplifies the computational analysis of such flows by reducing the complexity of the full Navier-Stokes equations (Wilcox, 2006) while still capturing the essential physics of the boundary layer, such as shear stress and velocity gradients. In cases similar to our PhD study, While the complex terrain violates the strict assumptions of the boundary layer approximation (gradual geometric changes), it can still be employed in certain regions and under specific conditions to gain valuable insights and simplify the simulation (use in less complex regions or in sub-regions of a complex terrain).

### 1.3.2 Flows and Partial Differential Equations

In the study of how fluids like air move, especially in the context of wind energy, understanding the behavior of airflows is crucial (Emeis, 2018). The movement of air around wind turbines, over complex terrains, and through Wind Farm (WF)s can be described using mathematical equations known as PDEs (Ferziger and Perić, 2002). These equations capture how air properties such as velocity, pressure, and temperature evolve in space and time. By classifying these Partial differential equation (PDE)s, we can better understand how air behaves in different situations, enabling more accurate predictions and efficient designs in wind energy and other applications.

PDEs are classified into three main types based on how information, such as changes in pressure or flow patterns, propagates through the fluid: **elliptic**, **parabolic**, and **hyperbolic** (Kajishima and Taira, 2016, Wendt, 2008). Each type corresponds to different physical phenomena and requires different approaches for solving them. Understanding these types is essential not only for optimizing wind turbine performance but also for other engineering and environmental applications.

## Elliptic Equations

Elliptic equations are used to describe steady-state flows, where the properties of the air do not change over time (Wendt, 2008). This is particularly relevant for wind turbines and WFs, where the wind flow is can be assumed to be steady to simplify analysis and design.

For example, consider the airflow around a wind turbine blade. The air moves smoothly and steadily around the blade without sudden changes. This flow can be modeled using elliptic equations like the Poisson equation (previous section) or the "Laplace equation":

$$\nabla^2 \phi = 0$$

Here,  $\phi$  represents the velocity potential. This equation helps predict how the wind will flow over the blade surfaces, ensuring that the turbine operates efficiently by maintaining a stable and predictable flow pattern.

In a WF, multiple wind turbines interact with the incoming wind. The overall airflow through the park can be analyzed using elliptic PDEs to understand how the presence of each turbine affects the others. This analysis is crucial for optimizing the placement of turbines to minimize wake effects (turbulence generated by upstream turbines that can reduce the efficiency of downstream turbines) and maximize energy capture.

Elliptic equations are also vital when dealing with flows over complex terrain, such as hills or ridges. In these cases, the wind flow is steady, but the terrain causes the air to follow the contours of the land. Understanding this steady-state behavior is crucial for siting wind turbines in locations where the terrain significantly influences wind patterns.

Beyond wind energy, elliptic PDEs are used in many other fields. For instance, in electrostatics, the Laplace equation describes how electric potential spreads out in space, and in heat conduction, similar elliptic equations describe steady-state temperature distributions in solids.

## Parabolic Equations

Parabolic equations come into play when a process changes over time but is still influenced by previous conditions. These equations are particularly useful in situations where the flow has a dominant direction, and the changes in the flow are gradual. For flows over complex terrain, parabolic equations help model how the air flows over the land. This is important for predicting how wind behaves in areas with varying topography, which

directly impacts the siting and performance of wind turbines. For such flows, the heat equation is a straightforward parabolic PDE:

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T$$

where  $T$  is the temperature and  $\alpha$  is the thermal diffusivity. This equation can model how temperature changes over time.

In a WF, parabolic equations are also used to model how the boundary layer interacts with the ground and other turbines. As wind flows through the park, the boundary layer grows, and its effects on downstream turbines become more pronounced Emeis, 2018. Understanding this interaction is key to designing WF s that minimize losses due to turbulence and maximize overall efficiency.

Parabolic equations can be used to model the boundary layer near the surface of a wind turbine blade. The boundary layer is a thin region of air close to the blade where the effects of viscosity (friction between the air molecules and the blade) slow down the airflow. As the wind moves over the blade surface, the boundary layer grows thicker, affecting how efficiently the turbine can convert wind energy into electrical energy. The boundary layer approximation, which simplifies the full Navier-Stokes equations into a form that is easier to solve, is an example of a parabolic PDE. It focuses on predicting how the air will behave within this thin layer, which is crucial for optimizing the design of wind turbine blades to reduce drag and increase energy capture.

Parabolic PDEs are not limited to wind energy. They are also used in other areas such as heat conduction, where they describe how heat spreads over time in a material, and in groundwater flow, where they model how water moves through porous media.

## Hyperbolic Equations

Hyperbolic equations are essential for describing wave propagation and rapid changes in the flow. They are used to model situations where sudden disturbances occur. In wind energy, hyperbolic PDEs are used to model shock waves, gust fronts, and other sudden disturbances in the airflow(Emeis, 2018,Ferziger and Perić, 2020).

In complex terrain, hyperbolic equations like the wave equation can model how wind interacts with abrupt landscape features, such as cliffs or ridges, leading to turbulent flows and complex patterns. For instance,

when a sudden gust of wind hits a wind turbine, the air around the turbine experiences rapid changes in velocity and pressure. Hyperbolic equations capture these changes and help predict how the turbine will respond to such events, ensuring that it can withstand the forces generated by sudden wind fluctuations. To model rapid changes such as wind gusts, the wave equation is a suitable hyperbolic PDE:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u$$

where  $u$  represents the wave field and  $c$  is the wave speed. This equation helps predict how sudden wind fluctuations or disturbances affect the airflow around turbines. Accurate modeling of these effects is important for the safe and efficient operation of wind turbines in challenging environments.

Outside of wind energy, hyperbolic PDEs are used to model various wave phenomena, such as sound waves, seismic waves, and electromagnetic waves(Wendt, 2008). They describe how these waves propagate through different media, which is crucial in fields ranging from acoustics to earthquake engineering.

### Mixed Flow Types: A Real-World Perspective

In many practical situations, the flow of air cannot be neatly categorized into just one type of PDE. Instead, we often encounter mixed flow types, where different regions of the flow are governed by different types of equations. For example, consider a WF located in a region with complex terrain. Near the ground or the surface of turbine blades, the flow may be dominated by viscous effects, leading to parabolic behavior. However, as we move away from these surfaces, the flow might become more uniform and steady, exhibiting elliptic behavior. If the park experiences sudden changes in wind speed due to weather conditions, hyperbolic behavior might be observed, particularly in the response of the turbines to gusts.

The challenge in designing and operating such a WF is to account for these different flow behaviors simultaneously. Engineers must use a combination of elliptic, parabolic, and hyperbolic PDEs to accurately model the wind flow and optimize the placement and design of turbines. For example, in areas where the terrain creates complex wind patterns, understanding both the steady-state (elliptic) and time-dependent (parabolic and hyperbolic) aspects of the flow is crucial for predicting the performance

and longevity of the turbines.

An other case of mixed flow conditions is at mountainous regions, where the wind flow can exhibit all three types of PDE behavior depending on the location and flow conditions. In sheltered valleys, the wind might be steady and slow, leading to elliptic behavior. On the windward side of a mountain, the flow could accelerate and create a thin boundary layer, which is parabolic in nature. At the crest of the mountain or in narrow passes, rapid changes in wind speed and direction could generate turbulent flows and even shock waves, which are hyperbolic phenomena.

Understanding these mixed flow types is essential for various applications beyond wind energy. For instance, in environmental engineering, predicting how pollutants disperse in the atmosphere often requires solving mixed PDEs, as the behavior of the flow can change depending on the terrain and weather conditions.

The classification of partial differential equations into elliptic, parabolic, and hyperbolic types provides a powerful framework for understanding and predicting airflows in various contexts. In wind energy, this understanding is crucial for designing efficient wind turbines, optimizing WF layouts, and ensuring the safe operation of turbines in complex terrains. However, the principles outlined here are not limited to wind energy; they apply to a wide range of engineering and environmental problems where fluid flow plays a critical role.

By applying the appropriate mathematical tools and understanding the underlying physics of the flow, engineers and scientists can create more accurate models, leading to better designs and more successful outcomes in both wind energy and other fields. In our PhD work, we use the steady-state (elliptic) form of the conservation equations to approximate transient solutions, which are typically parabolic in nature, significantly reducing the computational time of simulations. Recognizing abrupt changes in wind speed and direction helps interpret approximation errors and informs adjustments to the modeling approach. This could involve combining transient and steady-state solutions under specific conditions or modifying the temporal and spatial resolution of the steady-state approximations to enhance accuracy.

## 1.4 Coordinate and Basis Vector Systems

In fluid dynamics, the representation of conservation equations is highly dependent on the choice of coordinate systems and basis vectors. The mathematical framework allows these equations to be expressed in multiple forms, tailored to the specific needs of the problem being studied. This flexibility is crucial as different flows and physical scenarios may necessitate the use of distinct coordinate systems and bases to achieve more accurate and efficient computational solutions.

### 1.4.1 Coordinate Systems

The selection of an appropriate coordinate system is one of the fundamental decisions in formulating the conservation equations (Ferziger and Perić, 2020; Kajishima and Taira, 2016). Commonly used systems include:

- **Cartesian Coordinates ( $x, y, z$ ):** This is the most straightforward system, ideal for problems with simple, rectangular geometries. The conservation equations in Cartesian coordinates are typically the easiest to solve numerically because of their simplicity and the uniformity of the grid that can be used.
- **Cylindrical Coordinates ( $r, \theta, z$ ):** This system is well-suited for problems with rotational symmetry, such as flow in pipes or around cylindrical objects. The equations take on a different form, with additional terms that account for the curvature and rotational aspects of the system.
- **Spherical Coordinates ( $r, \theta, \phi$ ):** Ideal for problems involving spherical symmetry, such as flow around a sphere. This system introduces more complexity in the equations due to the angular components, but it simplifies the representation of problems with radial symmetry.
- **Curvilinear and Orthogonal/Non-Orthogonal Coordinates:** These are used for more complex geometries where neither Cartesian, cylindrical, nor spherical coordinates are suitable. Curvilinear coordinates can adapt to the shape of the domain, providing a more natural representation of the physical problem. Orthogonal systems have perpendicular coordinate axes, simplifying the mathematical formulation, while non-orthogonal systems, though more complex, offer greater flexibility in handling irregular geometries.

### 1.4.2 Basis Vector Systems

Once the coordinate system is chosen, the next step is to decide on the basis in which vectors and tensors will be represented (Ferziger and Perić, 2020; Wendt, 2008). The basis vectors can be:

- **Fixed vs. Variable Basis:** In a fixed basis, the basis vectors remain constant throughout the domain, which is often the case in Cartesian coordinates. A variable basis, however, changes from point to point, which is common in curvilinear coordinate systems. The choice between a fixed or variable basis affects how derivatives and differential operators are applied to vector and tensor fields.
- **Covariant vs. Contravariant Components:** These terms refer to the way vector components transform under a change of coordinates. Covariant components change with the basis vectors, while contravariant components change inversely to the basis. Understanding this distinction is critical when working with complex geometries and when ensuring the correct transformation properties of physical quantities.
- **Physical vs. Non-Physical Components:** Physical components correspond directly to the measurable quantities in a given direction (e.g., velocity along an axis), whereas non-physical components might be more abstract, relating to the coordinate system rather than the physical space.

### 1.4.3 Implications for Numerical Methods

The choice of coordinate system and basis has significant implications for the discretization method and grid type used in numerical simulations. For example, Cartesian grids are typically easier to implement and more efficient computationally, but may struggle with complex geometries. In contrast, curvilinear grids, while more computationally intensive, offer better accuracy for irregular domains. The decision of which system and basis to use is guided by the problem's specific requirements, including the geometry, boundary conditions, and the desired accuracy of the solution.

In summary, the selection of coordinate and basis vector systems is a critical step in the formulation and solution of fluid dynamics problems. It influences the mathematical form of the governing equations, the complexity of the numerical methods, and ultimately the accuracy and efficiency of the simulations.

#### 1.4.4 Conservation in Cartesian coordinates

In order for a computer to numerically solve the complex equations governing fluid dynamics, a systematic and computationally suitable representation of the problem is essential. This is achieved through component-wise representation, often expressed using indicial notation, which provides a structured framework for translating the continuous partial differential equations into a discrete system of algebraic equations suitable for numerical computation.

For demonstration simplicity, a Three-dimensional (3D) fluid flow in a Cartesian coordinate system, we define:

- $x_1 = x$ ,  $x_2 = y$ , and  $x_3 = z$  as the Cartesian coordinates.
- $u_1 = u$ ,  $u_2 = v$ , and  $u_3 = w$  as the corresponding velocity components in the  $x$ ,  $y$ , and  $z$  directions, respectively.

With these definitions, the fundamental conservation equations that we have presented can be expressed in indicial notation as follows:

- **Conservation of Mass**

- **Divergence Form:**

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} = 0,$$

- **Gradient Form:**

$$\frac{\partial \rho}{\partial t} + u_j \frac{\partial \rho}{\partial x_j} + \rho \frac{\partial u_j}{\partial x_j} = 0,$$

where  $\rho$  is the fluid density,  $t$  is time,  $u_j$  represents the velocity components, and  $x_j$  are the spatial coordinates. This equation essentially states that the rate of change of mass within a fluid element is balanced by the mass flux across its boundaries.

- **Conservation of Momentum**

- **Divergence Form:**

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial}{\partial x_j}(\rho u_i u_j - \tau_{ij}) = \rho F_i,$$

- **Gradient Form:**

$$\rho \left( \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} \right) = \frac{\partial \tau_{ij}}{\partial x_j} + \rho F_i.$$

where  $\tau_{ij}$  is the stress tensor, and  $F_i$  represents the body forces (like gravity) acting on the fluid. This form emphasizes the flux of momentum through a CV's surface.

- **Conservation of Total Energy**

- **Divergence Form:**

$$\frac{\partial(\rho E)}{\partial t} + \frac{\partial}{\partial x_j} (\rho Eu_j - \tau_{ij}u_i + q_j) = \rho u_i F_i,$$

- **Gradient Form:**

$$\rho \left( \frac{\partial E}{\partial t} + u_j \frac{\partial E}{\partial x_j} \right) = \frac{\partial}{\partial x_j} (\tau_{ij}u_i - q_j) + \rho u_i F_i.$$

where  $E$  is the total energy per unit mass, and  $q_j$  represents the heat flux vector.

The equations that relate stress and heat flux to the velocity and temperature fields, are expressed as:

- **Stress Tensor:**

$$\tau_{ij} = -\delta_{ij}p + 2\mu \left( D_{ij} - \frac{1}{3}\delta_{ij}\frac{\partial u_k}{\partial x_k} \right),$$

where  $p$  is the pressure,  $\mu$  is the dynamic viscosity, and  $D_{ij}$  is the rate-of-strain tensor defined as:

$$D_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}.$$

- **Fourier's Law of Heat Conduction:**

$$q_j = -k \frac{\partial T}{\partial x_j},$$

where  $k$  is the thermal conductivity, and  $T$  is the temperature.

In indicial notation, when an index appears twice in a term, summation over that index is implied. For example, in two dimensions,  $a_j b_j$  indicates summation over  $j$  from 1 to 2, and in three dimensions, it sums over  $j$  from 1 to 3. The Kronecker delta,  $\delta_{ij}$ , is a special symbol used in indicial

notation and is defined as:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j, \\ 0 & \text{if } i \neq j. \end{cases}$$

This symbol represents the identity tensor in Cartesian coordinates. In three dimensions, the trace (or sum of the diagonal elements) of the Kronecker delta is  $\delta_{kk} = 3$ .

For an incompressible flow with constant density and viscosity, the continuity equation is:

$$\frac{\partial u_i}{\partial x_i} = 0$$

, the momentum equation is:

$$\frac{\partial u_i}{\partial t} + \frac{\partial(u_i u_j)}{\partial x_j} = \frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + F_i$$

and the pressure Poisson equation derived from the above two equations is:

$$\frac{1}{\rho} \frac{\partial^2 p}{\partial x_i \partial x_i} = -\frac{\partial^2(u_i u_j)}{\partial x_i \partial x_j} + \frac{\partial F_i}{\partial x_i} = -\frac{\partial u_i}{\partial x_j} \frac{\partial u_j}{\partial x_i} + \frac{\partial F_i}{\partial x_i}$$

## 1.5 Discretization Methods

In this section we will give a brief description of the discretization methods presented in the next chapter, created to assist reader in quickly understanding existing CFD applications like the ones presented in COMSOL Multiphysics® (CFD), 2020 which is the software used in this PhD work. The readers that are interested in the discretization methods and want to create custom CFD codes are referred to the work of Dick, 2009; Ferziger and Perić, 2002; Kajishima and Taira, 2016; Wendt, 2008, Wilcox, 2006 and COMSOL Multiphysics® (CYCLOPEDIA), 2017, for an even more detailed description and skip the next chapter.

### 1.5.1 Finite Difference Method (FDM)

The FDM stands as the most conceptually straightforward and historically the oldest approach to numerically solving PDEs in CFD. Its core principle involves directly approximating derivatives in the governing

equations using Taylor series expansions. In practice, the computational domain is discretized into a structured grid, and the derivatives at each grid point are approximated using the values at neighboring grid points. FDM's simplicity makes it particularly appealing for beginners and for problems with simple or regular geometries amenable to structured grids. The method's efficiency in such scenarios stems from the structured grid's facilitation of boundary condition implementation and streamlined solution procedures. Furthermore, FDM boasts a rich history and extensive literature, offering a plethora of resources and examples for learning and troubleshooting.

However, FDM's reliance on structured grids poses a significant limitation when dealing with complex geometries exhibiting curved boundaries or intricate features. Moreover, its accuracy is intrinsically linked to grid resolution and the order of Taylor series approximations. While higher-order schemes can enhance accuracy, they also escalate computational demands and complexity. Additionally, FDM grapples with maintaining accuracy near boundaries or in regions characterized by substantial gradients. Another notable challenge lies in FDM's non-conservative nature, necessitating meticulous formulation and discretization to uphold fundamental conservation laws.

### 1.5.2 Finite Volume Method (FVM)

The Finite Volume Method (FVM) distinguishes itself by its foundation on the integral form of conservation laws. It partitions the computational domain into a collection of non-overlapping control volumes and applies the conservation equations in an integral sense to each volume. This inherent conservation attribute renders FVM a favored choice in CFD, particularly for problems involving shocks, discontinuities, or intricate flow physics where adherence to conservation laws is paramount.

FVM's adaptability to complex geometries through the use of unstructured grids, comprising diverse element shapes, further amplifies its appeal. This geometric flexibility empowers FVM to tackle a wide array of real-world engineering challenges. Furthermore, FVM's robustness and capacity to accommodate a broad spectrum of flow regimes, spanning from low-speed incompressible to high-speed compressible flows with shocks, solidify its position as a cornerstone of CFD.

Nevertheless, FVM is not without its challenges. Its implementation, particularly on unstructured grids, can be more intricate compared to

FDM. Attaining high-order accuracy in FVM can also prove demanding, especially near boundaries or within regions exhibiting steep flow gradients. Moreover, the proper implementation of boundary conditions in FVM can be intricate, particularly for complex geometries or when addressing non-reflecting boundary conditions in wave propagation problems.

### 1.5.3 Finite Element Method (FEM)

The Finite Element Method (FEM) is rooted in the weak or variational formulation of governing PDEs. It subdivides the computational domain into a mesh of finite elements and approximates the solution within each element using piecewise polynomials or other suitable basis functions. The weak form's relaxation of continuity requirements broadens the spectrum of admissible functions.

FEM's prowess in handling complex geometries shines through its capacity to employ unstructured meshes with diverse element types and sizes. It further enables mesh refinement and adaptation, effectively capturing localized flow features. Moreover, FEM inherently facilitates higher-order approximations by utilizing higher-degree polynomials as basis functions, enabling high accuracy with relatively coarser meshes. Its adaptability extends to multiphysics problems involving coupled physical phenomena, owing to its flexibility in accommodating various PDEs and boundary conditions.

However, FEM's computational demands can be substantial, particularly for large 3D problems or when employing higher-order elements. The assembly and solution of the global system of equations can strain computational resources. Additionally, FEM's implementation complexity surpasses that of FDM and FVM, necessitating a firm grasp of the weak formulation, basis functions, and numerical integration techniques. While FEM can be formulated to be conservative, ensuring conservation properties demands careful consideration, especially in problems featuring shocks or discontinuities.

### 1.5.4 Conclusion

The optimal choice among FDM, FVM, and FEM hinges on the specific problem at hand. Factors such as geometry complexity, accuracy requirements, computational resources, the nature of the flow physics, and the availability of existing codes or software all play pivotal roles in this

decision. In practice, hybrid approaches that combine different methods or employ adaptive mesh refinement techniques are often leveraged to harness the strengths of each method and achieve superior solutions for intricate CFD challenges. These methods will be described in the next chapter.

## 1.6 Numerical Grids

Numerical grids are essential in Computational Fluid Dynamics (CFD) simulations, providing a discrete representation of the continuous physical domain (Kajishima and Taira, 2016). They divide the domain into smaller elements, enabling the computation of flow variables. The selection and design of the grid significantly impact the accuracy, efficiency, and stability of the simulation.

The grid's resolution (element size and distribution) directly influences the accuracy of flow variable calculations. Finer grids generally lead to higher accuracy but increased computational cost. The grid's structure (structured vs. unstructured) affects computational efficiency. Structured grids often result in faster computations, while unstructured grids offer greater flexibility for complex geometries (Ferziger and Perić, 2020). The quality of the grid (element shape and distortion) is crucial for numerical stability. Poorly shaped elements can cause convergence issues or divergence. Numerical grids are the foundation of CFD simulations. Careful grid design and selection are vital for achieving accurate, efficient, and stable solutions.

### 1.6.1 Key Factors Influencing Grid Choice

The selection of an appropriate numerical grid is a pivotal decision in CFD simulations, as it profoundly influences the accuracy, efficiency, and overall feasibility of the analysis. Several key factors must be carefully considered when choosing a grid as suggested in CFD literature, like in Dick, 2009; Ferziger and Perić, 2002; Kajishima and Taira, 2016; Wendt, 2008 and Wilcox, 2006. A brief description of these factors follows:

- **Geometric Complexity:** The complexity of the flow domain geometry plays a crucial role in grid selection. For simple geometries, such as rectangular channels or circular pipes, structured grids, characterized by their regular and predictable arrangement, are often the most

suitable choice. Their inherent simplicity translates to computational efficiency and ease of implementation. However, when dealing with intricate geometries, such as those encountered in complex terrain modeling or the intricate blade designs of wind turbines, unstructured grids, with their ability to adapt to irregular shapes and boundaries, become indispensable. Unstructured grids offer the flexibility to accurately represent the geometric details of the flow domain, ensuring that critical features are adequately captured. In some cases, a hybrid approach, combining structured and unstructured grids, may be employed to leverage the advantages of both grid types.

- **Flow Features:** The nature of the flow phenomena being simulated also significantly influences grid selection. Flows characterized by large gradients, localized phenomena (e.g., vortices, shocks), or moving boundaries demand special attention. In such cases, adaptive or dynamic grids, capable of adjusting their resolution and configuration during the simulation, are often necessary to accurately capture these critical flow features. For instance, in wind turbine simulations, adaptive mesh refinement can be employed to dynamically increase grid resolution near the blade tips, where strong vortices and complex flow interactions occur. Similarly, in simulations of atmospheric flows over complex terrain, adaptive grids can refine the mesh near mountains or valleys to resolve the intricate flow patterns and turbulence generated in these regions.
- **Computational Resources:** The available computational resources, including processing power, memory, and storage, impose practical constraints on grid selection. While finer grids generally lead to improved accuracy, they also incur a higher computational cost. Unstructured grids and adaptive mesh refinement techniques, although powerful, often demand greater computational resources compared to structured grids. Therefore, it's crucial to strike a balance between accuracy requirements and computational feasibility. In large-scale simulations, such as those involving WF s or entire atmospheric domains, efficient grid design and parallelization strategies become paramount to manage the computational burden.
- **Numerical Method:** The choice of numerical method used to solve the governing equations also plays a role in grid selection. Certain numerical methods are inherently better suited to specific grid types. For example, finite difference methods are commonly employed on

structured grids due to their reliance on regular grid spacing and connectivity. Finite volume methods, on the other hand, can be applied to both structured and unstructured grids, offering greater flexibility in handling complex geometries. Finite element methods are often preferred for unstructured grids, as they naturally accommodate the irregular element shapes and connectivity patterns. Therefore, the compatibility between the chosen numerical method and the grid type should be carefully considered to ensure optimal performance and accuracy.

### 1.6.2 Classification of Numerical Grids

To generate a grid that properly describes the real scenario modeled in a CFD simulation (Ferziger and Perić, 2020; Wendt, 2008), proper understanding of the grid types and their classification in way that helps us easily relate them to our needs is crucial in fast and efficient mesh generation. During this PhD work and the CFD modeling conducted, our understanding of numerical grid types based on the work of Dick, 2009; Ferziger and Perić, 2002; Kajishima and Taira, 2016; Wendt, 2008, Wilcox, 2006 and COMSOL Multiphysics® (CFD), 2020 led to the modeling approach presented in Michos et al., 2024 and in Part II of this thesis. Numerical grids can be classified based on several key characteristics:

#### A. Topology

The topology of a numerical grid refers to the arrangement and connectivity of its grid lines or elements in relation to the coordinate system and the boundaries of the physical domain. It significantly influences the grid's ability to accurately represent complex geometries and capture flow features near curved surfaces. Let's have a further look into the three primary topological classifications of numerical grids:

**Cartesian Grids** Cartesian grids are characterized by their alignment with the Cartesian coordinate axes ( $x$ ,  $y$ ,  $z$ ) as depicted in figure 1.1. This orthogonal arrangement results in a grid composed of rectangular or cuboidal elements, making them exceptionally straightforward to implement and computationally efficient. However, this simplicity comes at a cost when dealing with complex geometries or curved boundaries.

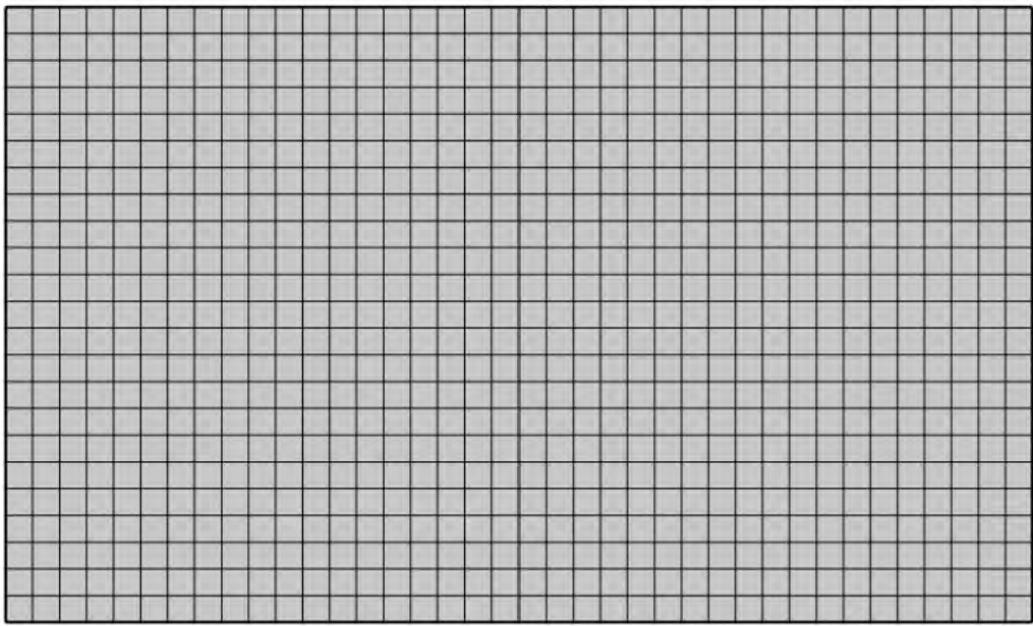


Figure 1.1: Cartesian grid

- **Advantages:**

- Straightforward implementation and computational efficiency
- Well-suited for simple geometries with rectangular or box-shaped domains
- Easy to visualize and interpret results

- **Limitations:**

- Difficulty in accurately representing complex or curved boundaries
- Stair-stepping effect near curved surfaces, leading to potential inaccuracies
- Inefficient for highly irregular geometries, requiring excessive refinement in some regions

For example, in wind turbine simulations, a Cartesian grid might be suitable for modeling the far-field flow domain, where the geometry is relatively simple and uniform. However, near the turbine blades, where the geometry is complex and curved, a Cartesian grid would struggle to accurately capture the flow details.

**Curvilinear Grids** Curvilinear grids like the one presented in 1.2, address the limitations of Cartesian grids by allowing grid lines to curve and conform to the shape of the domain or its boundaries. This enables a more accurate representation of complex geometries and curved surfaces.

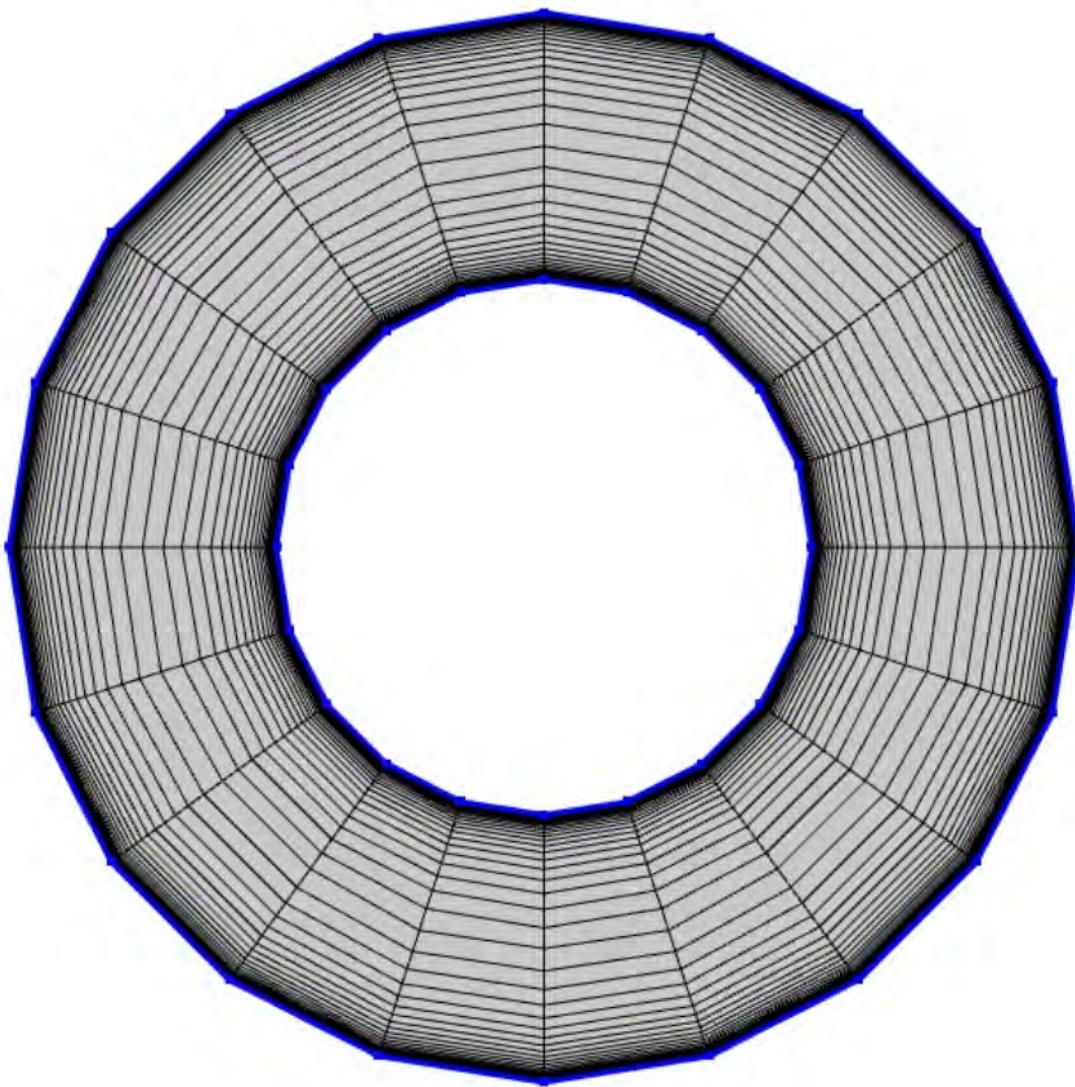


Figure 1.2: Curvilinear grid

- **Advantages:**

- Better representation of complex geometries and curved boundaries
- Reduced grid skewness and distortion near curved surfaces, leading to improved accuracy
- Can be more computationally efficient than unstructured grids for certain geometries

- **Limitations:**

- More challenging to generate than Cartesian grids
- May still exhibit some skewness or distortion in regions of high curvature
- Not as flexible as unstructured grids for handling highly irregular geometries

In simulations of atmospheric flows over complex terrain, curvilinear grids can be used to accurately represent the topography of mountains, valleys, and other terrain features. The curved grid lines allow for better resolution of flow features near the terrain surface, improving the accuracy of the simulation.

**Body-Fitted Grids** Body-fitted grids take the concept of curvilinear grids a step further by aligning the grid lines directly with the boundaries of the object being simulated. This close conformity to the geometry ensures excellent accuracy in capturing flow features near the object's surface.

- **Advantages:**

- Superior accuracy for complex geometries, especially near boundaries
- Minimal grid skewness and distortion, leading to improved numerical stability
- Ideal for simulations where accurate boundary layer resolution is critical

- **Limitations:**

- Can be challenging and time-consuming to generate, particularly for highly irregular shapes
- May require specialized grid generation techniques and software
- Not as flexible as unstructured grids for handling moving boundaries or evolving geometries

A good example is the simulations of flow around the tower of a wind turbine or its blades. A body-fitted grid can be used to accurately capture the boundary layer development and flow separation near the blade surface.

This is crucial for predicting the aerodynamic performance and structural loads on the turbine.

In summary, the choice of grid topology depends on the specific requirements of the CFD simulation. Cartesian grids are suitable for simple geometries, while curvilinear and body-fitted grids are necessary for accurately representing complex geometries and curved boundaries, as seen in figure 1.3. Unstructured grids offer the greatest flexibility for handling highly irregular shapes and moving boundaries and are widely used for body-fitted grids, but they may come with increased computational complexity. The optimal grid topology balances accuracy, efficiency, and the specific challenges posed by the flow domain and the phenomena being simulated.

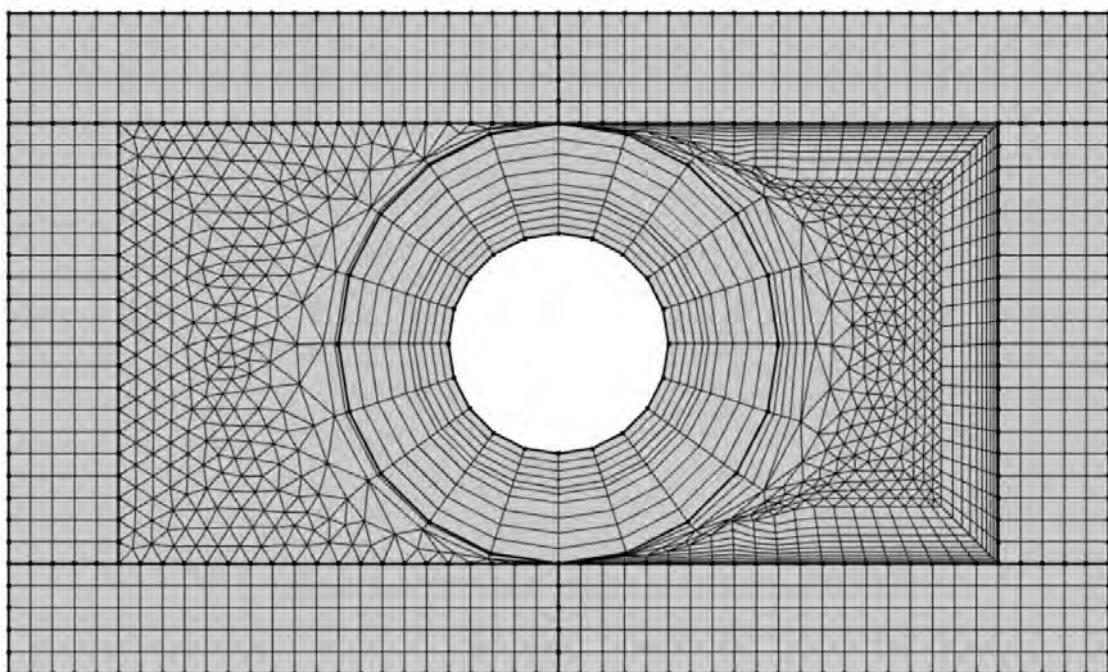


Figure 1.3: Example of a 2D slice of a grid for fluid flow simulation around the tower of a wind turbine with mixed characteristics.

Figure 1.3 shows the significance of the chosen grid's characteristics. The left side shows how an unstructured grid directly interconnects the boundaries of the Cartesian grid with the curvilinear grid, while the right side uses a boundary structured body fitted approach which struggles at sharp angles, along with use of unstructured grid.

## B. Structure

The structural arrangement of grid points and their connectivity defines the fundamental distinction between structured and unstructured grids. Each type presents unique advantages and challenges, making them suitable for different CFD applications.

**Structured Grids** Structured grids are characterized by a regular and predictable arrangement of grid points. This organization often manifests as a logically ordered pattern, reminiscent of rows and columns in a table. Such a structured layout brings forth several key benefits:

- **Efficient Data Storage and Access:** The inherent regularity of structured grids allows for the use of simple indexing schemes (e.g.,  $i, j, k$ ) to identify and access grid points and their associated data. This translates to efficient memory usage and streamlined computational operations.
- **Simplified Numerical Implementation:** Many numerical methods, particularly finite difference schemes, are naturally well-suited for structured grids due to their reliance on regular grid spacing and connectivity. This often leads to simpler code implementation and faster execution times.
- **Well-Suited for Simple Geometries:** Structured grids excel in representing simple geometries with regular shapes, such as rectangular channels, circular pipes, or annular domains. The grid lines can be easily aligned with the boundaries and coordinate axes, facilitating accurate representation and boundary condition enforcement.
- **Challenges with Complex Geometries:** The rigid structure of structured grids can pose limitations when dealing with complex or irregular geometries. Generating a high-quality structured grid for such domains can be challenging and may result in skewed or distorted elements, which can adversely affect accuracy and stability.

For example, Cartesian grids are the simplest type of structured grid and are often used for flows in rectangular or box-shaped domains. In case of cylindrical or axisymmetric geometries the use of Cylindrical grids is suitable option (1.2). The grid lines follow the radial, azimuthal, and axial directions of the cylindrical coordinate system. In cases of more complex structures, Body-Fitted Structured grids can be a solution.

**Unstructured Grids** Unstructured grids, in contrast to their structured counterparts, embrace irregularity in the arrangement of grid points. This flexibility empowers them to handle complex geometries and adapt to evolving flow features with remarkable ease.

- **Geometric Flexibility:** The ability to position grid points arbitrarily allows unstructured grids to conform to intricate shapes and boundaries, making them ideal for simulations involving complex geometries, such as those encountered in aerospace, automotive, and environmental applications.
- **Adaptability to Flow Features:** Unstructured grids can be readily refined or coarsened in specific regions to capture localized flow phenomena, such as boundary layers, vortices, or shocks. This adaptability ensures efficient resolution of critical flow features while minimizing computational overhead in regions with smoother flow behavior.
- **Handling Moving Boundaries:** The inherent flexibility of unstructured grids makes them well-suited for simulations with moving boundaries or interfaces, such as those encountered in fluid-structure interaction problems or free surface flows. The grid can be dynamically adjusted to accommodate the changing domain configuration.
- **Complex Data Structures:** The irregular connectivity of unstructured grids necessitates the use of specialized data structures to efficiently store and access grid information. This can lead to increased memory requirements and slightly more complex code implementation compared to structured grids.
- **Computational Overhead:** The irregular nature of unstructured grids can also introduce some computational overhead, particularly in terms of matrix assembly and solution. However, advancements in numerical algorithms and computing power have significantly mitigated this challenge.

#### Examples of Unstructured Grids:

- **Triangular Grids (2D):** These grids consist of triangular elements, offering flexibility in representing complex 2D domains as shown in figure 1.4. They are widely used in simulations of aerodynamics, hydrodynamics, and heat transfer.

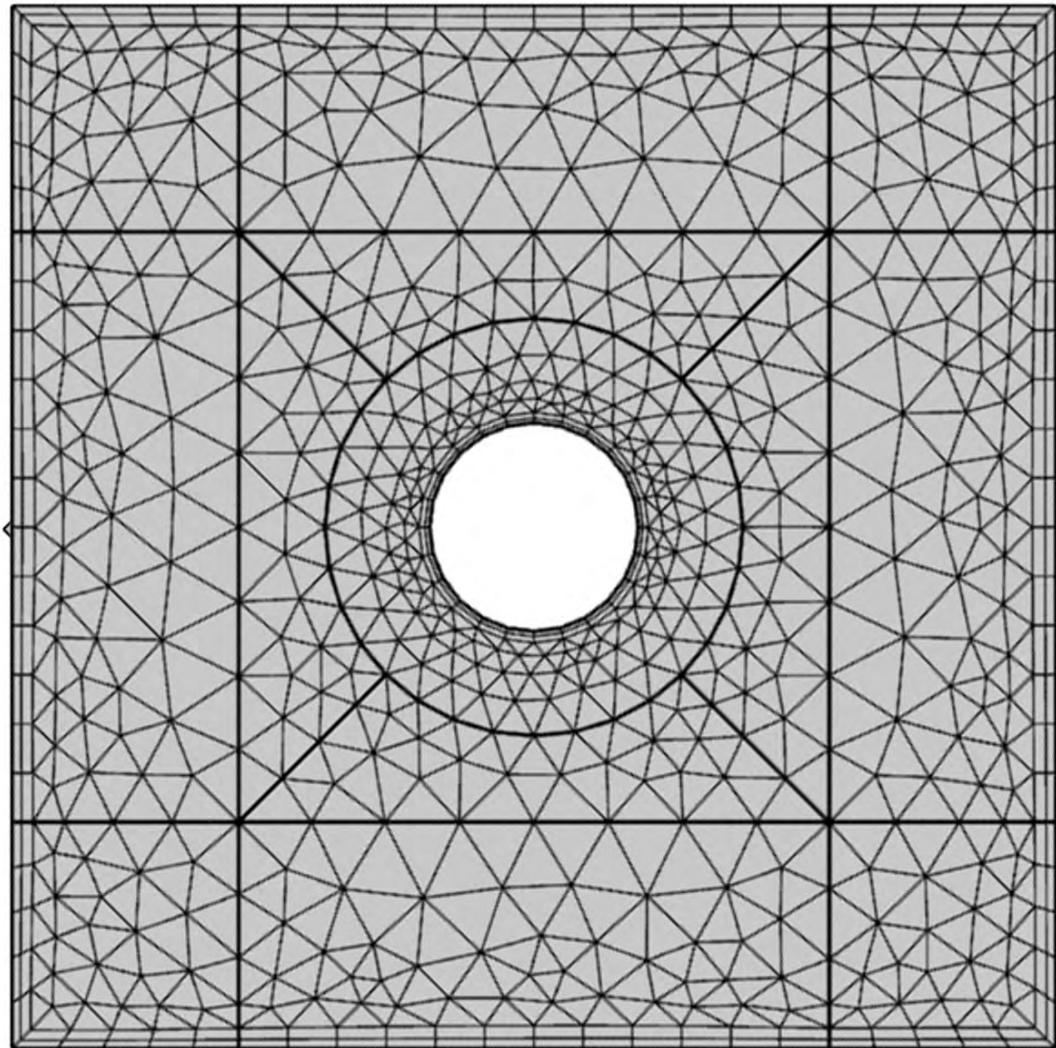


Figure 1.4: Example of a 2D unstructured triangular grid.

- **Tetrahedral Grids (3D):** The 3D counterpart of triangular grids, tetrahedral grids are composed of **tetrahedral elements** and are commonly **employed in simulations involving complex 3D geometries**, such as those encountered in biomedical engineering, environmental modeling, and industrial processes.
- **Quadrilateral Grids (2D):** Similar to triangular grids, while composed of elements with **4 sides** and are commonly **employed in simulations involving parts of complex geometries, offering a smooth transition to a Cartesian grid** (figure 1.5).

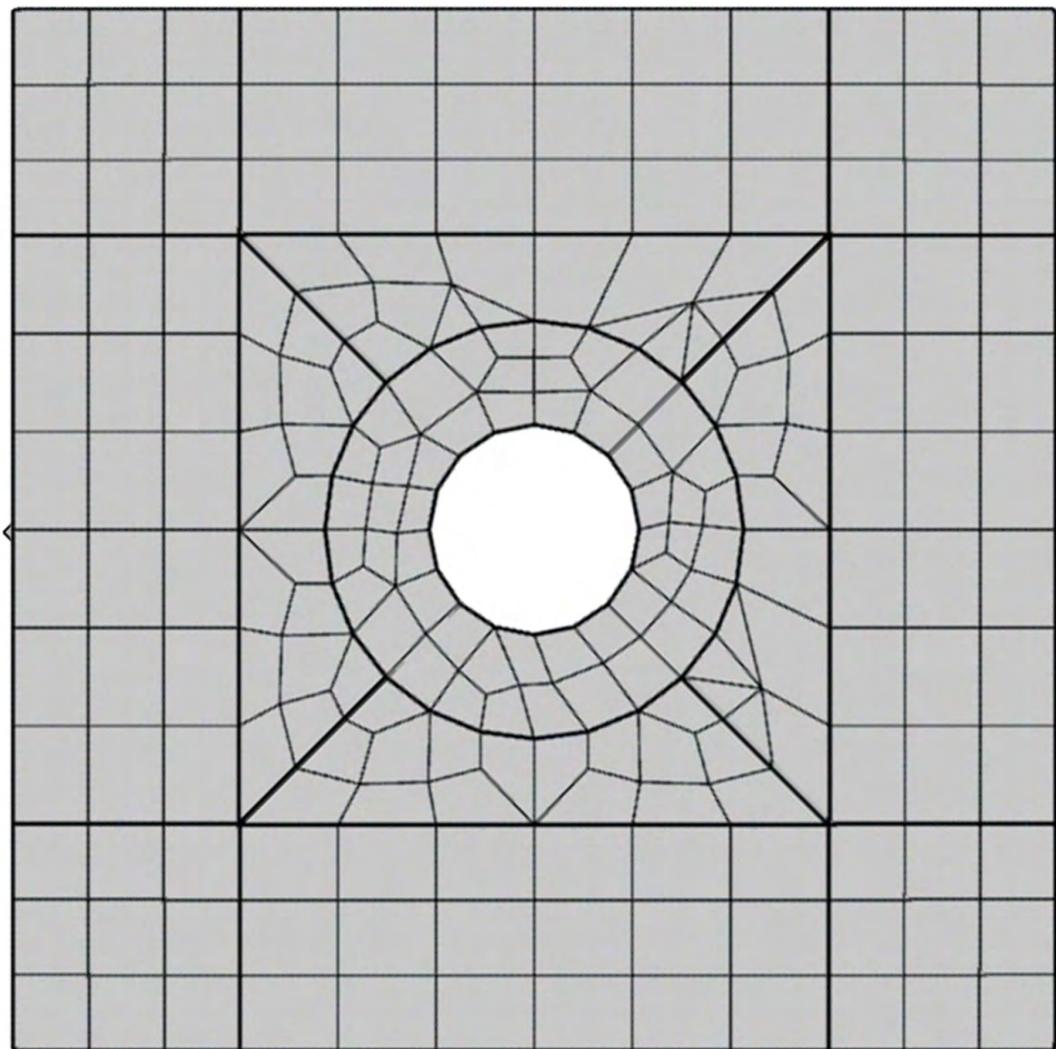


Figure 1.5: Example of a 2D unstructured quadrilateral grid that transforms smoothly to a structured Cartesian grid.

- **Hybrid Grids:** These grids combine different element types, such as triangles and quadrilaterals in 2D or tetrahedra and hexahedra in 3D. This approach offers a compromise between the flexibility of unstructured grids and the computational efficiency of structured grids, allowing for tailored grid design in different regions of the domain (figure 1.6).

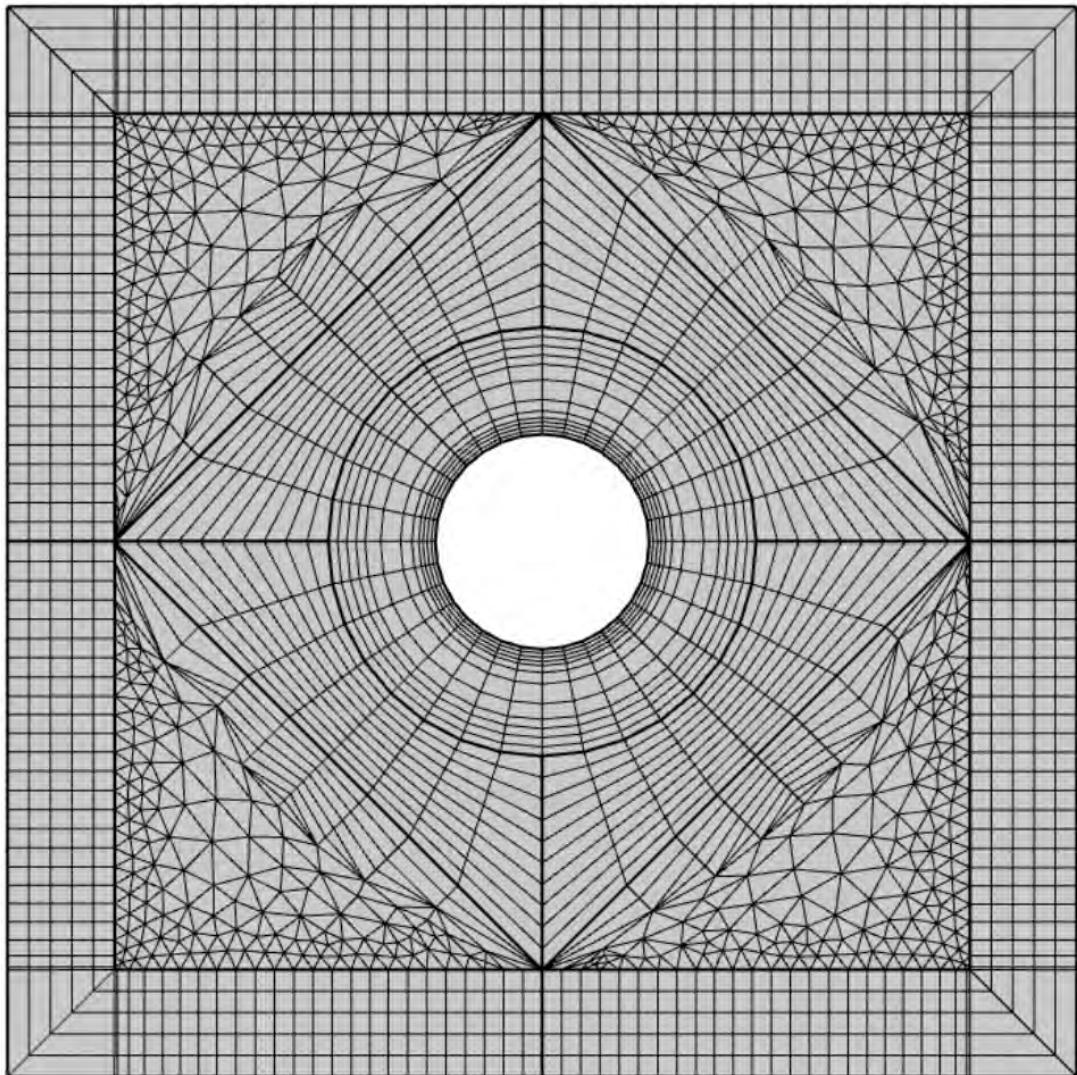


Figure 1.6: Example of a hybrid grid.

In summary, the choice between structured and unstructured grids hinges on a careful balance between geometric complexity, flow features, computational resources, and numerical method compatibility. Structured grids are favored for their simplicity and efficiency in handling simple geometries, while unstructured grids excel in representing complex shapes and adapting to evolving flow phenomena. The optimal grid choice ultimately depends on the specific requirements and constraints of the CFD simulation at hand.

### C. Adaptability

The adaptability of a numerical grid refers to its ability to modify its configuration during the simulation to better resolve critical flow features

or accommodate changes in the domain geometry. This dynamic nature of adaptive and dynamic grids offers significant advantages in terms of accuracy and computational efficiency, but it also comes with increased complexity in implementation and numerical solution procedures.

**Static Grids** Static grids maintain a fixed configuration throughout the simulation. The grid points and element connectivity remain unchanged, regardless of the evolving flow field or any changes in the domain geometry.

- **Advantages:**

- Simplicity in implementation and data management
- Predictable computational performance
- Suitable for problems with stationary boundaries and relatively uniform flow features

- **Limitations:**

- Inefficient for flows with localized phenomena or large gradients
- May require excessive refinement in regions where flow features are not well-resolved
- Unable to handle moving boundaries or deforming geometries

In simulations of steady-state flow around a wind turbine in a uniform wind field, a static grid might suffice if the focus is on capturing the overall flow pattern and wake characteristics. However, if the goal is to accurately resolve the complex flow structures near the blade tips or the turbulent wake development, an adaptive or dynamic grid would be more appropriate.

**Adaptive Grids** Adaptive grids possess the ability to dynamically modify their configuration during the simulation, either by refining the mesh in regions of high gradients or flow complexity or by coarsening it in areas where the flow is relatively smooth. This adaptability leads to several benefits:

- **Advantages:**

- By concentrating grid points in regions of interest, adaptive grids can significantly enhance the resolution of critical flow features, leading to improved accuracy and capturing of fine-scale details.

- Adaptive grids optimize computational resources by allocating more elements to regions where they are most needed, while using coarser elements in less critical areas. This can result in significant savings in computational time and memory.
- Adaptive grids excel in capturing localized flow phenomena, such as shocks, boundary layers, vortices, and turbulent structures. They can dynamically refine the mesh near these features, ensuring their accurate representation and tracking their evolution over time.

- **Limitations:**

- Increased complexity in implementation and numerical solution procedures
- Requires robust error estimation and adaptation criteria
- May introduce additional computational overhead due to dynamic mesh modifications

In simulations of atmospheric flows over complex terrain, adaptive grids can be used to refine the mesh near mountain peaks, valleys, and other topographic features where flow gradients are steep and turbulence is intense. This allows for better resolution of the complex flow patterns and the interaction of the flow with the terrain.

**Dynamic Grids** Dynamic grids take adaptability a step further by allowing grid points to move and deform to track moving boundaries or interfaces within the flow domain. This capability is essential for simulations involving:

- **Fluid-Structure Interaction (FSI):** In FSI simulations, the fluid flow interacts with a deformable structure, such as a wind turbine blade or an aircraft wing. Dynamic grids can adapt to the changing shape of the structure, ensuring accurate representation of the fluid-structure interface and the resulting flow field.
- **Free Surface Flows:** These flows involve the interface between two immiscible fluids, such as air and water. Dynamic grids can track the movement of the free surface, enabling accurate simulation of waves, splashes, and other complex interfacial phenomena.

- **Moving Objects:** Dynamic grids are also useful for simulations involving moving objects within the flow domain, such as rotating machinery or projectiles. The grid can be deformed or remeshed to accommodate the motion of the object, ensuring accurate representation of the flow field around it.

Dynamic grids have the following Limitations:

- Significant increase in complexity compared to static or adaptive grids
- Requires specialized algorithms for mesh movement and deformation
- Can introduce additional challenges in terms of numerical stability and conservation

In simulations of wind turbines operating in turbulent atmospheric conditions, a dynamic grid can be used to capture the complex interaction between the rotating blades and the turbulent wind field. The grid can deform to follow the motion of the blades, allowing for accurate resolution of the unsteady flow features and the resulting aerodynamic loads on the turbine.

In summary, the choice of grid adaptability depends on the specific requirements of the CFD simulation. Static grids are suitable for simple, steady-state problems, while adaptive grids are beneficial for capturing localized phenomena and optimizing computational resources. Dynamic grids are essential for handling moving boundaries and complex fluid-structure interactions, but they come with increased complexity and computational challenges. The selection of the appropriate grid adaptability level is crucial for achieving accurate, efficient, and meaningful CFD simulations.

### 1.6.3 Specialized Grid Types

In addition to the fundamental classifications based on structure, topology, and adaptability, several specialized grid types have emerged to address specific challenges and enhance the flexibility and accuracy of CFD simulations. These specialized grids often combine elements from different categories or introduce novel approaches to grid generation and adaptation.

**Overset Grids (or Chimera Grids)** Overset grids, also known as Chimera grids, employ a multi-grid approach where multiple structured or unstructured grids overlap to cover the entire computational domain. As shown in Ferziger and Perić, 2020 this technique offers several advantages:

- **Handling Complex Geometries:** Overset grids excel in representing complex configurations involving multiple components or objects with relative motion. Each component can be discretized with its own grid, tailored to its specific geometry, and these grids can overlap and interact seamlessly.
- **Efficient Grid Generation:** Generating a single, high-quality grid for complex configurations can be challenging. Overset grids simplify this process by allowing the use of simpler, component-specific grids that are easier to generate and manage.
- **Simulating Moving Objects:** Overset grids are particularly useful for simulating scenarios where objects move relative to each other, such as in rotorcraft simulations or store separation studies. The grids associated with the moving objects can be translated or rotated independently, maintaining their integrity and accuracy.

The challenges that are introduced by this types of structures are:

- **Interpolation at Overlapping Regions:** Accurate and conservative interpolation of flow variables between overlapping grids is essential to ensure solution continuity and avoid numerical errors.
- **Computational Overhead:** Managing multiple grids and performing interpolation can introduce additional computational overhead.
- **Complex Data Structures:** Overset grids require sophisticated data structures to handle the overlapping regions and grid connectivity, adding complexity to the implementation.

In simulations of wind turbines in complex terrain, an overset grid approach can be employed. A body-fitted grid can be used to discretize the wind turbine blades or the Tower, while a separate, coarser grid can be used for the surrounding terrain. The two grids overlap near the turbine, allowing for accurate resolution of the flow field in the vicinity of the WT while maintaining computational efficiency in the far field, as illustrated in figure 1.7.

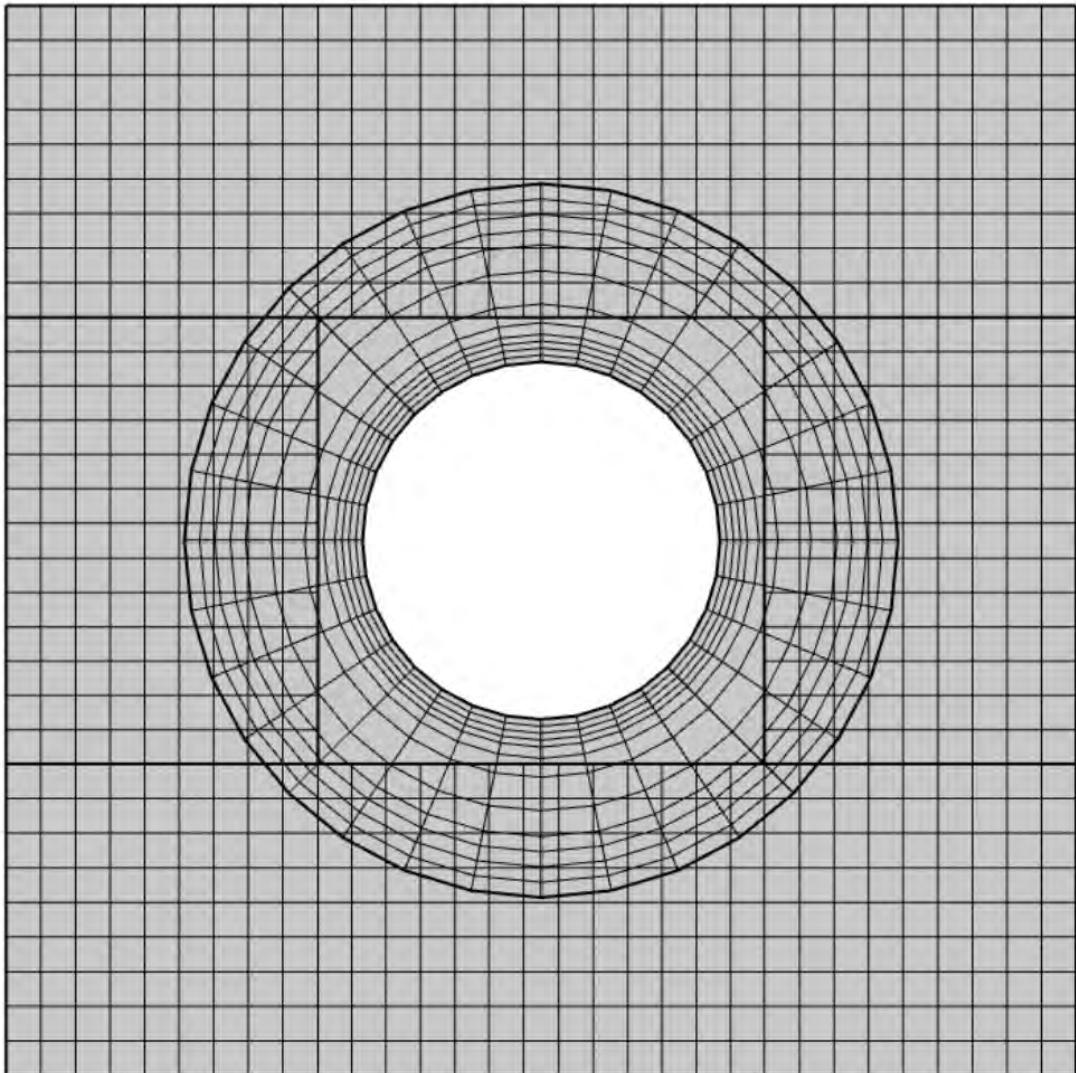


Figure 1.7: A composite 2D grid.

**Hierarchical Grids** Hierarchical grids utilize multiple levels of refinement to capture multi-scale flow phenomena. The idea is to employ coarser grids in regions where the flow is relatively smooth or less critical, while finer grids are used in areas with high gradients, complex flow structures, or where greater accuracy is required.

- **Advantages:**

- Efficient resolution of multi-scale phenomena
- Significant reduction in computational cost compared to using a uniformly fine grid throughout the domain
- Adaptability to evolving flow features

- **Challenges:**

- Requires robust refinement criteria and load balancing strategies
- Can introduce complexity in data management and inter-grid communication
- May require specialized numerical algorithms for handling grid interfaces

In simulations of atmospheric boundary layer flows, a hierarchical grid can be employed. A coarse grid can be used to represent the upper atmosphere, where the flow is relatively uniform. Progressively finer grids can then be introduced closer to the ground to capture the turbulent boundary layer and the complex flow interactions near the surface.

**Block-Structured Grids** Block-structured grids partition the computational domain into blocks, each of which is discretized with its own structured grid. This approach offers a balance between the flexibility of unstructured grids and the computational efficiency of structured grids.

- **Advantages:**

- Flexibility in grid refinement: Different blocks can have varying levels of refinement, allowing for efficient allocation of grid points in regions of interest.
- Facilitates handling of complex geometries: The domain decomposition into blocks enables the use of simpler, structured grids within each block, even for complex overall geometries.
- Can leverage existing structured grid generation tools and numerical methods

- **Challenges:**

- Requires careful treatment of block interfaces to ensure conservation and solution continuity
- Can introduce additional complexity in data management and parallelization

**Types of Block Interfaces:**

- **Matching Interfaces:** When blocks share common boundaries with perfectly aligned grid points, the interfaces are considered matching. This simplifies the implementation and ensures seamless communication between blocks.
- **Non-Matching Interfaces:** In cases where grid points do not align at block boundaries, the interfaces are non-matching like the grid shown in figure 1.8. This requires specialized interpolation or conservative treatment to maintain accuracy and avoid numerical errors.

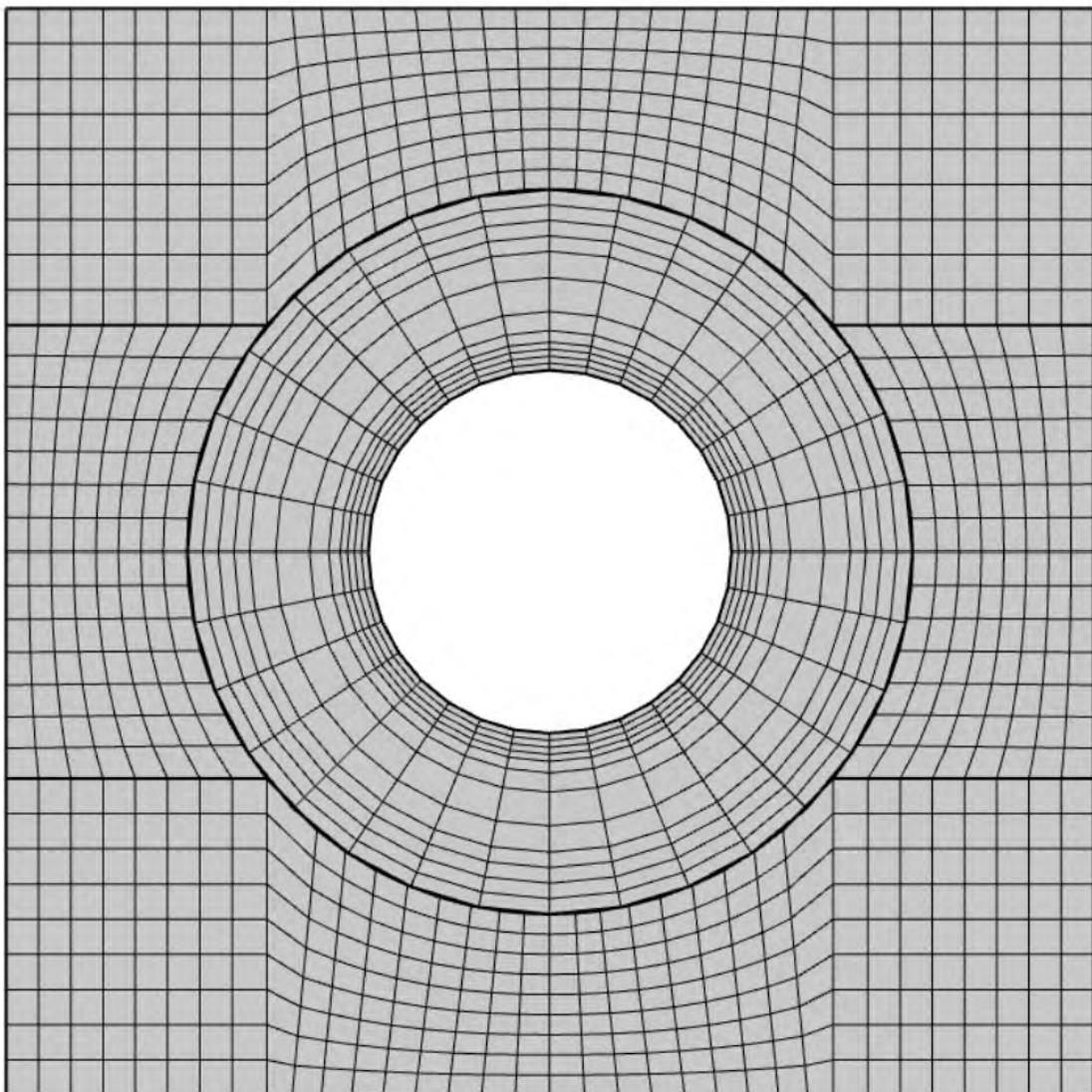


Figure 1.8: Example of a 2D block-structured grid with non-matching interfaces.

**Composite or Chimera Grids** A specialized type of block-structured grid, where blocks can overlap, enabling the combination of multiple structured grids for simulating complex flow fields, particularly those involving moving bodies. While Chimera grids offer flexibility, they introduce challenges in enforcing conservation at block boundaries, requiring careful treatment and specialized algorithms.

Both "Chimera grids" in the context of "Overset Grids" and "Composite Grids" refer to a technique where overlapping grids are used to manage complex simulations, but they might be applied slightly differently based on the context. The key difference lies in the emphasis: "Overset grids" focus on moving grids relative to each other, while "Composite Chimera grids" might focus more on handling complex static geometries through domain decomposition. However, in essence, they are variations of the same overarching methodology.

In conclusion, specialized grid types offer powerful tools for handling complex geometries, multi-scale phenomena, and moving boundaries in CFD simulations. The choice of the appropriate grid type depends on the specific problem at hand, balancing the trade-offs between accuracy, efficiency, and implementation complexity.

Selecting the appropriate grid type is crucial for effective numerical simulations. Structured grids are advantageous for simpler geometries due to their computational efficiency. For more complex domains, unstructured grids or hybrid approaches offer greater flexibility but may require additional computational resources. Boundary-fitted grids and adaptive techniques further enhance simulation accuracy and capabilities. Understanding these classifications helps balance accuracy, efficiency, and computational feasibility in CFD simulations. For those interested in a more in-depth exploration of grid generation methods, consider consulting Ferziger and Perić, 2020 , Kajishima and Taira, 2016 Thompson et al., 1985 and Wendt, 2008.

## 1.7 Properties of Numerical Methods

The success and reliability of any numerical method in CFD derives from a set of fundamental properties that govern its behavior and performance. These properties, encompassing conservation, boundedness, realizability , consistency, stability, convergence, accuracy, and efficiency, provide crucial insights into the method's ability to accurately and effi-

ciently solve the complex fluid flow problems encountered in CFD. In this section we will briefly describe each of these properties, based on Dick, 2009, Kajishima and Taira, 2016 and Ferziger and Perić, 2020.

### 1.7.1 Conservation

The governing equations in CFD express the fundamental physical laws of conservation of mass, momentum, and energy. A conservative numerical method ensures that these conservation laws are correctly transferred to the discrete level. Conservation properties ensure that the numerical solution respects the underlying physics of fluid flow, leading to more realistic and meaningful results. The choice of discretization schemes significantly impacts conservation. FVM are inherently conservative due to their integral formulation, while FDM and FEM require careful formulation to ensure conservation. Conservation can be enforced both globally (over the entire domain) and locally (at each control volume or element). Local conservation is often preferred as it provides a more stringent constraint on the numerical solution.

### 1.7.2 Boundedness and realizability

Boundedness ensures that the numerical solution remains within physically meaningful limits. This property prevents the occurrence of unrealistic or unbounded values that could compromise the integrity of the simulation and lead to numerical instabilities. Boundedness is closely tied to the physical constraints of the problem. For example, in incompressible flows, the density should remain constant, and the velocity components should not exceed physically realistic values. The choice of numerical schemes and the implementation of boundary conditions play a crucial role in maintaining boundedness. Techniques such as flux or slope limiters can be employed to prevent the solution from exceeding reasonable bounds.

While Boundedness ensures that the numerical solution remains within physically meaningful limits or ranges, we need to encompass a broader spectrum of physical constraints and principles that ensure the numerical solution is physically plausible and consistent with the underlying physics. This includes adherence to thermodynamic laws, turbulence characteristics, and other relevant physical constraints and is achieved by Realizability requirements.

Realizability mandates that computed solutions remain consistent with

the fundamental principles of physics, thereby ensuring their physical plausibility. This property transcends mere numerical correctness, preventing the occurrence of non-physical artifacts that could undermine the simulation's integrity and predictive capability. Realizability constraints are derived from various physical laws and principles, such as the positivity of density, pressure, and temperature, the second law of thermodynamics, and the positive definiteness of turbulent kinetic energy and dissipation rates. These constraints ensure that the numerical solution remains within physically meaningful bounds and does not violate fundamental physical principles.

Ensuring realizability often necessitates careful consideration in the formulation and implementation of numerical schemes. Techniques such as flux limiters, slope limiters, and artificial viscosity can be employed to prevent non-physical oscillations, negative values, or other violations of realizability constraints. Additionally, the choice of turbulence models, especially in Reynolds-averaged Navier-Stokes (RANS) simulations, should prioritize those known to satisfy realizability conditions, such as the Realizable k- $\varepsilon$  model. This model incorporates modifications to the turbulent viscosity formulation and the dissipation rate equation to ensure that the turbulent normal stresses remain positive and the Schwarz inequality for shear stresses is satisfied, promoting physically consistent and reliable predictions of turbulent flows.

### 1.7.3 Consistency

Consistency is fundamental for any numerical method. It ensures that as the discretization parameters, such as grid spacing or time step, approach zero, the discretized equations converge to the original PDEs governing the fluid flow. In essence, consistency guarantees that the numerical solution asymptotically approaches the exact solution as the resolution of the discretization is infinitely refined.

Consistency is connected to the mathematical formulation of the numerical method. It requires a careful discretization of the PDEs, ensuring that the truncation errors associated with the approximations of derivatives and integrals vanish as the discretization parameters tend to zero. A consistent method provides confidence that the numerical solution is a proper representation of the underlying physical phenomena, at least in the limit of infinite resolution. However, consistency alone does not guarantee a satisfying solution. Stability and convergence are equally

important.

#### 1.7.4 Stability

Stability is a critical property that prevents errors introduced at any stage of the computation from amplifying uncontrollably as the solution progresses. A stable numerical method ensures that small perturbations or numerical errors do not grow exponentially, leading to a divergent or non-physical solution.

Stability is closely linked to the control of numerical errors. It ensures that the errors remain bounded and do not overwhelm the true solution, preserving the integrity of the simulation. The choice of numerical schemes for approximating derivatives and integrals plays a crucial role in stability. Implicit schemes, which involve solving a system of equations at each time step, are generally more stable than explicit schemes, which directly update the solution based on previous values. In explicit time-marching methods, stability often imposes restrictions on the maximum allowable time step. The Courant-Friedrichs-Lowy (CFL) condition is a prime example of such a stability criterion.

#### 1.7.5 Convergence and Accuracy

Convergence is the most important part of any numerical method. It signifies that the numerical solution approaches the exact solution as the discretization is refined, either by reducing the grid spacing or the time step. A convergent method provides assurance that the computed solution is a reliable approximation of the true solution.

Convergence is intimately related to accuracy. As the discretization is refined, the numerical solution captures finer details of the flow field, leading to improved accuracy. In iterative solution procedures, convergence refers to the process of successive approximations approaching a fixed point or solution. Convergence criteria are employed to determine when the iterations have sufficiently converged to the solution. While convergence is desirable, it is not always guaranteed. Factors such as non-linearity, complex geometries, and boundary conditions can pose challenges to achieving convergence.

Accuracy quantifies how closely the numerical solution approximates the exact solution. Higher-order methods, which employ more sophisticated approximations of derivatives and integrals, generally offer improved

accuracy compared to lower-order methods.

The order of accuracy characterizes the rate at which the error decreases as the discretization is refined. A method with a higher order of accuracy converges faster to the exact solution as the grid is refined or the time step is reduced. While higher-order methods offer improved accuracy, they often come at the cost of increased computational complexity and memory requirements. The choice of accuracy level depends on the specific problem and the desired level of fidelity in the solution.

### 1.7.6 Efficiency

Efficiency pertains to the computational resources, both in terms of time and memory, required to obtain the numerical solution. An efficient method strikes a balance between accuracy and computational cost, delivering a solution of acceptable accuracy within a reasonable time-frame and with manageable memory usage.

The efficiency of a numerical method is influenced by the underlying algorithms and data structures used in its implementation. Optimized algorithms and efficient data handling can significantly improve computational performance. For large-scale CFD simulations, parallel computing techniques can be employed to distribute the computational workload across multiple processors or cores, thereby accelerating the solution process. The choice of hardware, such as CPUs, GPUs, or specialized accelerators, can also impact the efficiency of the numerical method.

These trade-offs create a scenario where it's impossible to optimize all objectives (accuracy, computational time, and memory usage) simultaneously. This is where Pareto trade-offs and the concept of Pareto fronts can become essential (Catthoor et al., 2020).

In CFD simulations, a Pareto front represents the set of solutions where no one criterion can be improved without worsening another. For example, one solution on the Pareto front may offer higher accuracy but require more computational time, while another may provide faster performance at the cost of some accuracy. Each point on the Pareto front represents a non-dominated solution, meaning that no solution is strictly better in all aspects. Engineers and scientists must then select a solution that best meets the needs of their specific project.

Thus, Pareto optimization is a vital for managing the trade-offs between accuracy, computational cost, and resource consumption in a finalizing a product focusing on ultra-short-term-applications. Understanding and

navigating these trade-offs is crucial for creating efficient and accurate simulations. In practice, the choice of where to position a solution on the Pareto front depends on the specific goals of the simulation—whether high precision is required, or if faster computational results are more critical.

By applying the appropriate mathematical tools and understanding the underlying physics of the flow, engineers and scientists can create models that balance these factors, leading to better designs and more successful outcomes in fields like wind energy and beyond. In our PhD, the steady-state form of the conservation equations is used to approximate transient solutions, which are typically parabolic, reducing the computational time of the simulations. Understanding that abrupt wind speed and direction changes can affect the approximation errors provides insights into potential adjustments, such as combining transient and steady-state solutions in specific situations or refining temporal or spatial resolution in the steady-state approximations. The choice of these adjustments reflects a position on the Pareto front, balancing efficiency and accuracy based on the simulation's requirements.

---

## References

- Catthoor, F., Basten, T., Zompakis, N., Geilen, M., & Kjeldsberg, P. G. (2020). *System-scenario-based design principles and applications* (Vol. 1). Springer. <https://doi.org/10.1007/978-3-030-20343-6>
- COMSOL Multiphysics® (CFD). (2020). *COMSOL Multiphysics® CFD module user's guide, version 5.6*. COMSOL. <https://doc.comsol.com/5.6/doc/com.comsol.help.cfd/CFDModuleUsersGuide.pdf>
- COMSOL Multiphysics® (CYCLOPEDIA). (2017). COMSOL Multiphysics® CYCLOPEDIA [[Online]. Last accessed: 24 August 2024]. <https://www.comsol.com/multiphysics/finite-element-method>
- COMSOL Multiphysics® (Reference Manual). (2020). *COMSOL Multiphysics® Reference Manual, version 5.6*. COMSOL. [https://doc.comsol.com/5.5/doc/com.comsol.help.comsol/COMSOL\\_ReferenceManual.pdf](https://doc.comsol.com/5.5/doc/com.comsol.help.comsol/COMSOL_ReferenceManual.pdf)
- Dick, E. (2009). Introduction to finite element methods in computational fluid dynamics. In J. F. Wendt (Ed.), *Computational fluid dynamics* (pp. 235–274). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-85056-4\\_10](https://doi.org/10.1007/978-3-540-85056-4_10)
- Emeis, S. (2018). *Wind energy meteorology: Atmospheric physics for wind power generation*. Springer. <https://doi.org/10.1007/978-3-642-30523-8>
- Ferziger, J. H., & Perić, M. (2002). *Computational methods for fluid dynamics*. Springer. <https://doi.org/10.1007/978-3-642-56026-2>
- Ferziger, J. H., & Perić, M. (2020). *Computational methods for fluid dynamics* (4th). Springer. <https://doi.org/10.1007/978-3-319-99693-6>
- Kajishima, T., & Taira, K. (2016). *Computational fluid dynamics: Incompressible turbulent flows*. Springer. <https://doi.org/10.1007/978-3-319-45304-0>
- Michos, D., Catthoor, F., Foussekis, D., & Kazantzidis, A. (2024). A cfd model for spatial extrapolation of wind field over complex terrain—wi. sp. ex. *Energies*, 17(16), 4139. <https://doi.org/10.3390/en17164139>
- Thompson, J. F., Warsi, Z. U., & Mastin, C. W. (1985). *Numerical grid generation: Foundations and applications*. Elsevier North-Holland, Inc.
- Wendt, J. (2008). *Computational fluid dynamics: An introduction*. Springer Berlin Heidelberg. <https://books.google.gr/books?id=IIUkqI-HNbQC>

- 
- Wilcox, D. C. (2006). *Turbulence modeling for cfd* (3rd) [Includes bibliography, index and Compact Disk.]. DCW Industries, Inc. <http://www.dcwindustries.com>
- Zemansky, M., & Dittman, R. (1997). *Heat and thermodynamics: An intermediate textbook*. McGraw-Hill. <https://books.google.gr/books?id=4vVAAQAAIAAJ>

# **Chapter 2**

## **Introduction to Discretization Methods**

In CFD, the discretization process serves as a translator between the continuous real-world mathematical representation of fluid flow, described in the previous chapter, embodied in PDEs, and the discrete world of numerical computation. By transforming these continuous equations into a system of algebraic equations, discretization methods enable us to harness the power of computers to solve complex flow problems. There are three primary discretization methods widely used in CFD: the FDM, the FVM, and the FEM. Understanding their modeling framework is crucial for their selection in a real case application. Each method has its own advantages and disadvantages in terms of speed, mathematical complexity and geometrical representation.

In this chapter, we will introduce each of these methods and provide the necessary details for someone new to the subject, based on the works of Dick, 2009; Ferziger and Perić, 2002; Kajishima and Taira, 2016; Wendt, 2008; Wilcox, 2006; Zemansky and Dittman, 1997 and COMSOL Multiphysics® (CFD), 2020; COMSOL Multiphysics® (CYCLOPEDIA), 2017; COMSOL Multiphysics® (Reference Manual), 2020, to whom the reader is referred for the deep understanding needed for the creation of custom CFD code and use of available CFD software like COMSOL, whose CFD module is used for our FEM simulations.

### **2.1 Finite Difference Method**

The Finite Difference Method (FDM) is one of the most conceptually straightforward and foundational discretization techniques in Computa-

tional Fluid Dynamics (CFD). It provides a direct and intuitive approach to transforming the continuous governing equations of fluid flow into a system of algebraic equations amenable to numerical solution. FDM directly discretizes the differential form of these equations, approximating derivatives using difference quotients based on the values of the flow variables at neighboring grid points.

### 2.1.1 Fundamental Principles

At its core, the FDM is predicated upon the approximation of derivatives using finite differences. This entails replacing the continuous derivatives in the governing PDEs with discrete approximations based on the values of the flow variables at neighboring grid points. The accuracy of these approximations hinges on the choice of finite difference stencils and the grid spacing.

The mathematical underpinning of FDM lies in the Taylor series expansion, which expresses the value of a function at a neighboring point in terms of its value and derivatives at a reference point. By truncating the Taylor series after a certain number of terms, we obtain finite difference approximations of varying orders of accuracy.

FDM is typically employed in conjunction with structured grids, characterized by their regular and predictable arrangement of grid points. This facilitates the application of finite difference stencils and simplifies the implementation of boundary conditions. The grids in FDM are usually staggered. 3D staggered grids store the scalar quantities  $\theta$  (like temperature and pressure) at the center of the control volume. On the other hand, the velocity components ( $\vec{u} = (u, v, w)$ ) are stored at the faces of the control volume:  $u$  is stored at the faces normal to the x-direction,  $v$  to the y-direction and  $w$  to the z-direction.

This discretization technique provides several advantages in 3D CFD simulations (Ferziger and Perić, 2020). It prevents pressure-velocity decoupling, which is common in collocated grids, thereby avoiding unphysical oscillations in the pressure field. By placing velocity components on the faces of control volumes, staggered grids allow for more accurate flux calculations, which are essential for conserving mass, momentum, and energy. Staggered grids often lead to more stable numerical schemes, especially in simulations with high Reynolds numbers or sharp gradients.

While staggered grids offer numerous benefits, they are not devoid of challenges. The storage of different variables at distinct locations intro-

duces additional complexity in data management and access. Moreover, although pressure gradients can be computed directly, interpolation is often necessary to obtain velocity values at cell centers or other locations where they are not directly stored. This interpolation can introduce additional computational overhead and potential sources of error.

In FDM, the continuous derivatives in the governing equations are replaced with their corresponding finite difference approximations. This transforms the PDEs into a system of algebraic equations that relate the values of the flow variables at the grid points. FDM is particularly well-suited for problems involving simple geometries and smooth solutions, where the flow field can be accurately captured without the need for complex grid generation techniques.

However, while FDM excels in scenarios with simple geometries, it can encounter challenges when applied to more complex flow problems. Accurately representing intricate geometries or capturing sharp gradients in the flow field, such as those found in turbulent flows or flows with boundary layers, can be difficult with FDM, and staggered grids may not be readily adaptable to unstructured grids, which are often favored for complex geometries. These challenges necessitate careful consideration of grid resolution and discretization schemes to ensure that the numerical solution remains stable and accurate.

In this section, we will describe the FDM based on insights and conventions presented in Kajishima and Taira, 2016. We will explore the fundamental principles underlying FDM, its mathematical formulation, and its application to the discretization of a general scalar like various terms encountered in the governing equations of fluid dynamics.

### 2.1.2 Mathematical Formulation

Let's consider a generic scalar transport equation, often encountered in CFD, to illustrate the application of FDM:

$$\frac{\partial \theta}{\partial t} + \nabla \cdot (\vec{u}\theta) = \nabla \cdot (\Gamma \nabla \theta) + \Phi \quad (2.1.2.1)$$

where,  $\theta$  is Scalar variable (e.g., temperature, concentration),  $t$  is Time,  $\vec{u}$  is Velocity vector,  $\Gamma$  is Diffusion coefficient and  $\Phi$  is Source term

For simplicity, we will discretize this equation on a uniform structured Cartesian grid, where the grid points are identified by indices  $(i, j, k)$  in the  $x$ ,  $y$ , and  $z$  directions, respectively. The grid spacing in each direction is

denoted by  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$ . To illustrate this discretization notation, we can imagine that the notation  $i, i+1, \dots, i+n$  indicates that  $n+1$  equally spaced and ordered grid points are present in the  $x$  direction. If we now think of  $x_i$  representing  $x$  value in  $i$ -th position of the grid, then  $x_{i+n} = x_i + n(\Delta x)$ ,  $n \in \mathbb{N}$

A fundamental way to obtain Finite difference schemes is by using Taylor series expansion. The Taylor series expansion of a function  $f(x)$  around a point  $x$  can be expressed as:

$$f(x + \Delta x) = \sum_{n=0}^{\infty} \frac{(\Delta x)^n}{n!} \frac{\partial^n f}{\partial x^n}$$

Expanding this for the first few terms gives:

$$f(x + \Delta x) = f(x) + \Delta x \frac{\partial f}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 f}{\partial x^2} + \frac{(\Delta x)^3}{3!} \frac{\partial^3 f}{\partial x^3} + \dots + O((\Delta x)^n)$$

Where  $O((\Delta x)^n)$  represents the truncation error, which includes all the terms from  $n$ -th order onwards. This notation indicates that the error introduced by truncating the series is of the same order as the next term in the series that is not included.

We'll derive the forward and backward difference approximations using Taylor series expansions but focus on the central difference scheme for the first derivative, as it is commonly used in FDM and highlighted in Kajishima and Taira, 2016 and Ferziger and Perić, 2020.

### **Forward Difference:**

Using the Taylor series expansion of  $f(x + \Delta x)$  around the point  $x$ :

$$f(x + \Delta x) = f(x) + \Delta x \frac{\partial f}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 f}{\partial x^2} + O((\Delta x)^3)$$

Rearranging to solve for the first derivative:

$$\frac{\partial f}{\partial x} = \frac{f(x + \Delta x) - f(x)}{\Delta x} + O(\Delta x)$$

This is the forward difference approximation for the first derivative, which has a first-order accuracy.

**Backward Difference:**

Using the Taylor series expansion of  $f(x - \Delta x)$  around the point  $x$ :

$$f(x - \Delta x) = f(x) - \Delta x \frac{\partial f}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 f}{\partial x^2} + O((\Delta x)^3)$$

Rearranging to solve for the first derivative:

$$\frac{\partial f}{\partial x} = \frac{f(x) - f(x - \Delta x)}{\Delta x} + O(\Delta x)$$

This is the backward difference approximation for the first derivative, which also has a first-order accuracy.

**Central Difference:**

Consider a function  $f(x)$ . The Taylor series expansion of  $f(x + \Delta x)$  and  $f(x - \Delta x)$  around the point  $x$  are:

$$\begin{aligned} f(x + \Delta x) &= f(x) + \Delta x \frac{\partial f}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 f}{\partial x^2} + \frac{(\Delta x)^3}{3!} \frac{\partial^3 f}{\partial x^3} + O((\Delta x)^4) \\ f(x - \Delta x) &= f(x) - \Delta x \frac{\partial f}{\partial x} + \frac{(\Delta x)^2}{2!} \frac{\partial^2 f}{\partial x^2} - \frac{(\Delta x)^3}{3!} \frac{\partial^3 f}{\partial x^3} + O((\Delta x)^4) \end{aligned}$$

Subtracting these two equations, we get:

$$f(x + \Delta x) - f(x - \Delta x) = 2\Delta x \frac{\partial f}{\partial x} + O((\Delta x)^3) \quad (2.1.2.2)$$

Rearranging to solve for the first derivative:

$$\frac{\partial f}{\partial x} = \frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x} + O((\Delta x)^2) \quad (2.1.2.3)$$

This is the central difference approximation for the first derivative, which has a second-order accuracy.

### 2.1.3 Discretization of Terms in the Governing Equation

Now let's apply finite difference approximations to discretize the terms in the governing equation. The time derivative  $\frac{\partial \theta}{\partial t}$  can be approximated using a finite difference scheme. For simplicity, let's consider the first-order

forward difference approximation:

$$\frac{\partial \theta}{\partial t} \approx \frac{\partial \theta_{i,j,k}}{\partial t} = \frac{\theta_{i,j,k}^{n+1} - \theta_{i,j,k}^n}{\Delta t} \quad (2.1.3.1)$$

where,  $\theta_{i,j,k}^{n+1}$  is the value of  $\theta$  at grid point  $(i, j, k)$  at the next time step  $(n + 1)$ ,  $\theta_{i,j,k}^n$  is the value of  $\theta$  at grid point  $(i, j, k)$  at the current time step  $(n)$  and  $\Delta t$  is the time step size.

The term  $\nabla \cdot (\vec{u}\theta)$  represents the transport of  $\theta$  due to the fluid flow. Let's consider the central difference approximation for the  $x$ -component of the term:

$$\frac{\partial(u\theta)}{\partial x} \approx \frac{\partial(u\theta)_{i,j,k}}{\partial x} = \frac{u_{i+1/2,j,k}\theta_{i+1/2,j,k} - u_{i-1/2,j,k}\theta_{i-1/2,j,k}}{\Delta x} \quad (2.1.3.2)$$

where,  $u_{i+1/2,j,k}$  is the  $x$ -component of velocity at the cell face between grid points  $(i, j, k)$  and  $(i + 1, j, k)$ , and  $\theta_{i+1/2,j,k}$  is the value of  $\theta$  at the cell face between grid points  $(i, j, k)$  and  $(i + 1, j, k)$ , which can be approximated using interpolation. Similar approximations can be applied to the  $y$  and  $z$  components, leading to the discretized (on the staggered grid) convective flux  $\nabla \cdot (\vec{u}\theta)$  as:

$$\begin{aligned} \nabla \cdot (\vec{u}\theta)_{i,j,k} &\approx \frac{u_{i+\frac{1}{2},j,k}\theta_{i+\frac{1}{2},j,k} - u_{i-\frac{1}{2},j,k}\theta_{i-\frac{1}{2},j,k}}{\Delta x} \\ &+ \frac{v_{i,j+\frac{1}{2},k}\theta_{i,j+\frac{1}{2},k} - v_{i,j-\frac{1}{2},k}\theta_{i,j-\frac{1}{2},k}}{\Delta y} \\ &+ \frac{w_{i,j,k+\frac{1}{2}}\theta_{i,j,k+\frac{1}{2}} - w_{i,j,k-\frac{1}{2}}\theta_{i,j,k-\frac{1}{2}}}{\Delta z}, \end{aligned} \quad (2.1.3.3)$$

where the  $\theta$  values at the face centers are interpolated from the neighboring cell centers. For example:

$$\theta_{i+\frac{1}{2},j,k} = \frac{\theta_{i,j,k} + \theta_{i+1,j,k}}{2}. \quad (2.1.3.4)$$

The diffusion term  $\nabla \cdot (\Gamma \nabla \theta)$  represents the spreading of  $\theta$  due to molecular motion. It involves second-order derivatives, which can be approximated using central difference schemes. For the  $x$ -component of the diffusion term, we have:

$$\frac{\partial}{\partial x} \left( \Gamma \frac{\partial \theta}{\partial x} \right) \approx \frac{\Gamma_{i+1/2,j,k} \left( \frac{\theta_{i+1,j,k} - \theta_{i,j,k}}{\Delta x} \right) - \Gamma_{i-1/2,j,k} \left( \frac{\theta_{i,j,k} - \theta_{i-1,j,k}}{\Delta x} \right)}{\Delta x} \quad (2.1.3.5)$$

where  $\Gamma_{i+1/2,j,k}$  and  $\Gamma_{i-1/2,j,k}$  are the values of the diffusion coefficient at the cell faces. Again, similar approximations can be applied to the  $y$  and  $z$

components of the diffusion term.

By substituting the finite difference approximations into the original transport equation, we obtain the discretized equation at grid point  $(i, j, k)$ :

$$\begin{aligned} \frac{\theta_{i,j,k}^{n+1} - \theta_{i,j,k}^n}{\Delta t} + \frac{(u\theta)_{i+1/2,j,k} - (u\theta)_{i-1/2,j,k}}{\Delta x} + \frac{(v\theta)_{i,j+1/2,k} - (v\theta)_{i,j-1/2,k}}{\Delta y} \\ + \frac{(w\theta)_{i,j,k+1/2} - (w\theta)_{i,j,k-1/2}}{\Delta z} = \frac{(\Gamma \frac{\partial \theta}{\partial x})_{i+1/2,j,k} - (\Gamma \frac{\partial \theta}{\partial x})_{i-1/2,j,k}}{\Delta x} \\ + \frac{(\Gamma \frac{\partial \theta}{\partial y})_{i,j+1/2,k} - (\Gamma \frac{\partial \theta}{\partial y})_{i,j-1/2,k}}{\Delta y} \\ + \frac{(\Gamma \frac{\partial \theta}{\partial z})_{i,j,k+1/2} - (\Gamma \frac{\partial \theta}{\partial z})_{i,j,k-1/2}}{\Delta z} + \Phi_{i,j,k} \end{aligned} \quad (2.1.3.6)$$

This discretized equation represents a linear algebraic equation relating the value of  $\theta$  at grid point  $(i, j, k)$  at the next time step ( $n + 1$ ) to its value at the current time step ( $n$ ) and the values of  $\theta$  and other variables at neighboring grid points.

## 2.1.4 Conclusion

The FDM offers a conceptually straightforward and computationally efficient approach to discretizing the governing equations of fluid flow. Its reliance on structured grids and Taylor series expansions provides a direct path to transforming PDEs into solvable algebraic equations. While FDM excels in handling simple geometries and smooth solutions, its accuracy can be compromised in the presence of complex geometries or sharp gradients. Nonetheless, FDM remains a valuable tool in the CFD arsenal, particularly for fundamental research and problems where computational efficiency is paramount.

Remember that the specific choice of finite difference stencils and interpolation schemes can significantly influence the accuracy and stability of the numerical solution. Careful consideration of these factors, along with appropriate grid design and boundary condition implementation, is crucial for achieving reliable and meaningful CFD simulations using the Finite Difference Method.

## 2.2 Finite Volume Method

The FVM, leverages the integral form of the governing equations, expressing conservation laws in terms of fluxes across CV boundaries. Flow variables are stored at cell centers, and the method focuses on balancing the influx and efflux of conserved quantities within each CV. This approach lends itself well to both structured and unstructured grids, making it a versatile choice for a wide range of CFD applications. FVM is known for its inherent conservation properties, ensuring that mass, momentum, and energy are preserved within the computational domain. We suggest the readers that are interested in the subject to read Kajishima and Taira, 2016 and Ferziger and Perić, 2020, as we will briefly describe the FVM.

### 2.2.1 Fundamental Principles

FVM discretizes the integral form of the governing equations, ensuring that the conservation laws of mass, momentum, and energy are inherently satisfied at the discrete level. This conservation property makes FVM a robust and reliable choice for a wide range of CFD simulations. The computational domain is divided into a collection of non-overlapping CV, each associated with a computational node where the flow variables are stored. These CVs can be of various shapes and sizes, accommodating both structured and unstructured grids.

The integral form of the governing equations is applied to each CV, expressing the conservation of a quantity (e.g., mass, momentum) as a balance between the net flux of that quantity across the CV boundaries and any sources or sinks within the volume. The fluxes across CV faces are approximated using numerical schemes that ensure consistency and stability. The choice of numerical flux scheme plays a crucial role in determining the accuracy and robustness of the FVM solution.

### 2.2.2 Mathematical Formulation

Let's consider the generic scalar transport equation in integral form to illustrate the application of FVM:

$$\frac{d}{dt} \iiint_{V_C} \rho\theta dV + \iint_{S_C} \rho\theta \vec{u} \cdot \vec{n} dS = \iint_{S_C} \Gamma \nabla \theta \cdot \vec{n} dS + \iiint_{V_C} \Phi dV \quad (2.2.2.1)$$

where,  $\vec{n}$  is the outward unit normal vector on the control surface.

FVM focuses on integral conservation laws applied directly to discrete CVs that tessellate the computational domain. Each CV serves as a fundamental unit where the conservation of relevant physical quantities is meticulously enforced. This approach inherently ensures that any changes in a conserved quantity within a CV are solely attributed to fluxes across its bounding faces and contributions from internal sources or sinks.

The discretization process in FVM commences with the transformation of the governing PDEs into their integral counterparts, applied over each CV. This integral formulation intrinsically upholds the conservation principle, stipulating that the temporal variation of a conserved quantity within a CV is precisely balanced by the net flux (convective and diffusive) traversing its boundaries, augmented by any internal generation or depletion.

### 2.2.3 Discretization of Terms in the Governing Equation

Within this integral framework, the discretization of FVM is often connected to collocated grids. In a collocated grid arrangement, all the primary variables, such as velocity components, pressure, temperature, and other scalar quantities, are stored at the same grid points. This contrasts with staggered grids, where different variables are stored at different locations. Collocated grids are advantageous in terms of simplicity and ease of implementation, especially in complex geometries. However, they require special treatment to avoid issues such as pressure-velocity decoupling.

The arrangement of variables on a collocated grid simplifies the data structure and is more straightforward to implement in complex geometries compared to staggered grids, since all variables are stored at the same location, the data structure is simpler and easier to implement. This is particularly beneficial in complex geometries where staggered grids may become cumbersome. They also offer greater flexibility in handling boundary conditions and are more adaptable to unstructured meshes.

A significant challenge in collocated grids is the issue of pressure-velocity decoupling, which can lead to spurious oscillations in the pressure field. Special techniques like Rhie-Chow interpolation are often used to mitigate this issue. Ensuring numerical stability can be more challenging in collocated grids, particularly in high Reynolds number flows or simulations with sharp gradients.

To understand how discretization in FVM collocated grids works, consider a general scalar quantity  $\theta$  (such as temperature or concentration).

The governing equation for  $\theta$  in 3D. The domain is divided into control volumes, and the governing equations are integrated over each control volume. For a collocated grid, all variables are stored at the centers of these control volumes.

Similarly to FDM, the temporal evolution of the conserved quantity is approximated through finite difference schemes. The forward difference approximation, a first-order accurate scheme, is frequently employed for its simplicity:

$$\frac{\partial}{\partial t} \iiint_{V_C} \rho\theta dV \approx \frac{(\rho\theta V)^{n+1} - (\rho\theta V)^n}{\Delta t}$$

In the discretization part,  $V$  represents the volume of the CV.

### **Convection Term:**

The convective transport of the conserved quantity, governed by the fluid flow, is represented by the cumulative convective fluxes across all faces of the CV. The convective flux across a face 'f' is generally approximated as:

$$F_f = \iint_{A_f} \rho \vec{u} \cdot \vec{n} \theta dA \approx (\rho \vec{u} \cdot \vec{n} \theta)_f A_f$$

Numerical schemes like upwind or central differencing (Ferziger and Perić, 2020 presents more schemes, like TVD) are leveraged to evaluate  $(\rho \vec{u} \cdot \vec{n} \theta)_f$  based on the flow velocity and the scalar quantity at the face of a control volume.

- **Central Differencing Scheme** This scheme is a second-order accurate method that uses the average of values from both sides of the face to approximate the quantity at the face. This method is simple and works well in regions where the solution is smooth. For a scalar quantity  $\theta$ , the approximation at the face  $f$  between two neighboring control volumes  $P$  and  $N$  is given by:

$$\theta_f = \frac{\theta_P + \theta_N}{2}$$

where  $\theta_P$  is the value of  $\theta$  at the center of the upstream control volume  $P$  and  $\theta_N$  is the value of  $\theta$  at the center of the downstream control volume  $N$ . The convective flux at the face  $f$  using central differencing can then be written as:

$$(\rho \vec{u} \cdot \vec{n} \theta)_f \approx \rho_f (\vec{u} \cdot \vec{n})_f \frac{\theta_P + \theta_N}{2}$$

Central differencing provides higher accuracy (second-order) in smooth regions of the flow but it can lead to numerical oscillations, especially in regions with sharp gradients (e.g., near shocks) or in cases of strong convective dominance.

- **Upwind Differencing Scheme** This scheme is a first-order accurate method that evaluates the flux at the face based on the direction of the flow. The basic idea is to use the value of  $\theta$  from the upstream (or upwind) side, assuming that the influence on the face primarily comes from the direction of the flow. For a face  $f$  between two control volumes  $P$  and  $N$ :

- If the flow is from  $P$  to  $N$  (i.e.,  $\vec{u} \cdot \vec{n} > 0$ ), the upwind scheme uses the value from the upstream node  $P$ .
- If the flow is from  $N$  to  $P$  (i.e.,  $\vec{u} \cdot \vec{n} < 0$ ), it uses the value from the upstream node  $N$ .

Mathematically:

$$\theta_f = \begin{cases} \theta_P, & \text{if } \vec{u} \cdot \vec{n} > 0 \\ \theta_N, & \text{if } \vec{u} \cdot \vec{n} < 0 \end{cases}$$

The convective flux at the face  $f$  using upwind differencing can then be written as:

$$(\rho \vec{u} \cdot \vec{n} \theta)_f \approx \begin{cases} \rho_f (\vec{u} \cdot \vec{n})_f \theta_P, & \text{if } \vec{u} \cdot \vec{n} > 0 \\ \rho_f (\vec{u} \cdot \vec{n})_f \theta_N, & \text{if } \vec{u} \cdot \vec{n} < 0 \end{cases}$$

Upwind differencing is more stable in cases of strong convective dominance because it reduces numerical oscillations that may arise with central differencing and particularly effective when the flow directionality is clear and dominant. On the other hand, it is only first-order accurate, meaning it introduces more numerical diffusion and can smear out steep gradients.

For simplicity, in a uniform, structured grid in a Cartesian system, the faces of the CVs are aligned with the coordinate axes. This simplifies the calculation of  $(\vec{u} \cdot \vec{n})_f$ .

For a face normal to the x-axis:

- $\vec{n} = (1, 0, 0)$  or  $\vec{n} = (-1, 0, 0)$  depending on whether it's the right or left face of the CV.
- $(\vec{u} \cdot \vec{n})_f = \pm u_f$ , where  $u_f$  is the x-component of the velocity at the face.

For a face normal to the y-axis:

- $\vec{n} = (0, 1, 0)$  or  $\vec{n} = (0, -1, 0)$ .
- $(\vec{u} \cdot \vec{n})_f = \pm v_f$ , where  $v_f$  is the y-component of the velocity at the face.

For a face normal to the z-axis:

- $\vec{n} = (0, 0, 1)$  or  $\vec{n} = (0, 0, -1)$ .
- $(\vec{u} \cdot \vec{n})_f = \pm w_f$ , where  $w_f$  is the z-component of the velocity at the face.

The  $\pm$  sign depends on the orientation of the face (whether it's the positive or negative face in the corresponding coordinate direction).

### **The 3 Equations in Cartesian System**

For a 3D problem, the convective flux will be calculated for each of the three coordinate directions. Let's consider a control volume (CV) with center at P  $(i, j, k)$ . The convective fluxes through its six faces can be approximated as follows:

#### **x-direction:**

- Left face  $(i - 1/2, j, k)$ :
  - The neighboring CV center is at W  $(i - 1, j, k)$ ,
  - $F_{x,i-1/2,j,k} \approx (\rho u \theta)_{i-1/2,j,k} \Delta y \Delta z$ ,
  - Upwind scheme:
    - \* If  $u_{i-1/2,j,k} > 0$  (from W to P), then

$$(\rho u \theta)_{i-1/2,j,k} \approx \rho_{i-1/2,j,k} u_{i-1/2,j,k} \theta_W$$

- \* If  $u_{i-1/2,j,k} < 0$  (from P to W), then

$$(\rho u \theta)_{i-1/2,j,k} \approx \rho_{i-1/2,j,k} u_{i-1/2,j,k} \theta_P$$

- Right face  $(i + 1/2, j, k)$ :
  - The neighboring CV center is at E  $(i + 1, j, k)$ ,

- $F_{x,i+1/2,j,k} \approx (\rho u \theta)_{i+1/2,j,k} \Delta y \Delta z$ ,
- Upwind scheme:
  - \* If  $u_{i+1/2,j,k} > 0$  (from P to E), then

$$(\rho u \theta)_{i+1/2,j,k} \approx \rho_{i+1/2,j,k} u_{i+1/2,j,k} \theta_P$$

- \* If  $u_{i+1/2,j,k} < 0$  (from E to P), then

$$(\rho u \theta)_{i+1/2,j,k} \approx \rho_{i+1/2,j,k} u_{i+1/2,j,k} \theta_E$$

#### **y-direction:**

- Bottom face ( $i, j - 1/2, k$ ):
  - The neighboring CV center is at S ( $i, j - 1, k$ ),
  - $F_{y,i,j-1/2,k} \approx (\rho v \theta)_{i,j-1/2,k} \Delta x \Delta z$ ,
  - Upwind scheme: Apply similar logic as in the x-direction, using  $v_{i,j-1/2,k}$ ,  $\theta_S$ , and  $\theta_P$ .
- Top face ( $i, j + 1/2, k$ ):
  - The neighboring CV center is at N ( $i, j + 1, k$ ),
  - $F_{y,i,j+1/2,k} \approx (\rho v \theta)_{i,j+1/2,k} \Delta x \Delta z$ ,
  - Upwind scheme: Apply similar logic, using  $v_{i,j+1/2,k}$ ,  $\theta_P$ , and  $\theta_N$ .

#### **z-direction:**

- Front face ( $i, j, k - 1/2$ ):
  - The neighboring CV center is at B ( $i, j, k - 1$ ),
  - $F_{z,i,j,k-1/2} \approx (\rho w \theta)_{i,j,k-1/2} \Delta x \Delta y$ ,
  - Upwind scheme: Apply similar logic, using  $w_{i,j,k-1/2}$ ,  $\theta_B$ , and  $\theta_P$ .
- Back face ( $i, j, k + 1/2$ ):
  - The neighboring CV center is at T ( $i, j, k + 1$ ),
  - $F_{z,i,j,k+1/2} \approx (\rho w \theta)_{i,j,k+1/2} \Delta x \Delta y$ ,
  - Upwind scheme: Apply similar logic, using  $w_{i,j,k+1/2}$ ,  $\theta_P$ , and  $\theta_T$ .

Here,  $\Delta x$ ,  $\Delta y$ , and  $\Delta z$  are the grid spacings in the  $x$ ,  $y$ , and  $z$  directions, respectively. The values at the face centers (e.g.,  $(\rho u \theta)_{i-1/2,j,k}$ ) are typically obtained using interpolation schemes like central or upwind differencing, as in our example. The choice of numerical scheme (e.g., upwind, central differencing) will impact the accuracy and stability of the solution.

**Diffusion Term:**

The diffusive transport, connected to molecular motion, is similarly accounted for by summing the diffusive fluxes across all CV faces. These fluxes are typically proportional to the gradient of the conserved quantity at the face, and are approximated using central differencing or other suitable schemes discussed in Ferziger and Perić, 2002.

$$D_f = \iint_{A_f} \Gamma \nabla \theta \cdot \vec{n} dS \approx (\Gamma \nabla \theta \cdot \vec{n})_f A_f$$

The central differencing scheme, which is a second-order scheme with great accuracy, is commonly employed to evaluate the gradient at the face, although alternative schemes may be used in specific scenarios to enhance stability or for addressing discontinuities.

**Source Term:**

Internal sources or sinks of the conserved quantity within the CV are represented by the source term. This contribution is approximated by evaluating the source term at the CV's centroid and scaling it by the CV's volume.

$$\iiint_{V_C} \Phi dV \approx \Phi_P V$$

where  $\Phi_P$  is the Source term evaluated at the cell center P

**The Discretized Equation**

The culmination of this discretization process yields the discretized equation for a control volume:

$$\frac{(\rho\theta V)^{n+1} - (\rho\theta V)^n}{\Delta t} + \sum_{\text{faces}} (\rho \vec{u} \cdot \vec{n} \theta)_f A_f = \sum_{\text{faces}} (\Gamma \nabla \theta \cdot \vec{n})_f A_f + \Phi_P V$$

This equation accurately describes the connection between the temporal variation of the conserved quantity, convective and diffusive fluxes across the CV boundaries, and internal source contributions. The successful application of the FVM requires the understanding of these discretization principles, coupled with the correct numerical schemes and grid configurations. This method is comprehensively described in Ferziger and Perić,

2020; Kajishima and Taira, 2016; Wendt, 2008.

### 2.2.4 Conclusion

The FVM is a powerful and widely used discretization technique in CFD, due to its conservation properties and geometric flexibility. Its focus on flux balances across CV boundaries provides a physically meaningful and robust approach to simulating fluid flow phenomena. By understanding the fundamental principles and implementation details of FVM, researchers and engineers can effectively leverage this method to tackle a diverse range of CFD challenges and gain valuable insights into the behavior of fluids in various engineering and scientific applications.

## 2.3 Finite Element Method

FEM employs a variational or weak formulation of the governing equations, transforming them into an integral equation that holds over the entire domain. The solution is approximated using a set of basis functions, often associated with the grid nodes. FEM excels in handling complex geometries and accommodating irregular grid structures, making it a popular choice for problems with intricate shapes or adaptive mesh refinement. However, FEM implementations can be more mathematically involved and computationally demanding than FDM or FVM.

FEM is the method we use for the purposes of this PhD research. This powerful and versatile discretization technique is modeled within COMSOL—Multiphysics. It is capable to handle complex geometries and adapt to a wide range of flow phenomena. In this section, we will briefly describe the Finite Element Method, based on the insights and conventions presented in Dick, 2009, COMSOL Multiphysics® (CFD), 2020 and COMSOL Multiphysics® (CYCLOPEDIA), 2017. We will explore the fundamental principles of FEM, its mathematical formulation, and its application to the discretization of various terms encountered in the governing equations of fluid dynamics.

### 2.3.1 Fundamental Principles

The Finite Element Method rests on several key principles that distinguish it from other discretization techniques. Instead of directly discretizing the strong form of the governing PDEs, FEM employs a weak or variational

formulation. This involves multiplying the PDEs by test functions and integrating over the domain, leading to an integral equation that holds in a weighted-average sense. The computational domain is divided into a collection of non-overlapping elements, typically triangles or quadrilaterals in 2D and tetrahedra or hexahedra in 3D. The choice of element type and size significantly influences the accuracy and efficiency of the FEM solution.

A set of basis functions, often associated with the nodes of the elements, is used to approximate the solution within each element. These basis functions can be of varying orders, with higher-order basis functions generally leading to improved accuracy. The most common approach in FEM is the Galerkin method, where the same set of basis functions is used for both the solution approximation and the test functions in the weak formulation. This leads to a system of algebraic equations that relate the nodal values of the flow variables.

### 2.3.2 Mathematical Formulation

The field of computational fluid dynamics (CFD) and heat transfer analysis frequently involves the solution of partial differential equations (PDEs) that describe the transport of scalar quantities such as temperature, concentration, or velocity potential. These PDEs can be complex, often involving terms that represent convection, diffusion, and source effects. Numerical methods such as the FEM and FVM have been developed to solve these equations with a high degree of accuracy.

For demonstration purposes, we will focus on the weak formulation, a fundamental concept in FEM, which serves as an alternative to the direct discretization of the strong form of the governing PDEs. We will also compare this approach to the FVM, a method that is widely used for solving conservation laws in fluid mechanics. By expanding on these concepts, we aim to provide a comprehensive understanding of their application in engineering simulations.

#### The Weak Formulation

The weak formulation, also known as the variational formulation, is a foundational concept of the FEM. It involves rewriting the governing PDEs in an integral form that relaxes the continuity requirements on the solution, making it possible to approximate solutions that may not be

smooth enough to satisfy the strong form of the PDEs.

To illustrate this concept, let us consider the generic scalar transport equation for a scalar, often encountered in CFD:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\vec{u}\phi) = \nabla \cdot (\Gamma \nabla \phi) + \Phi_S, \quad (2.3.2.1)$$

where  $\phi$  (instead of  $\theta$ ) represents the scalar variable of interest (like temperature  $T$ ). The term  $t$  denotes time,  $\vec{u}$  is the velocity vector field,  $\Gamma$  is the diffusion coefficient, and  $\Phi_S$  represents the source term, which accounts for any internal generation or consumption of  $\phi$ .

In the strong form, this equation imposes pointwise constraints on the solution, meaning that the solution must be sufficiently smooth to allow for the computation of all derivatives involved. However, in practical situations, especially in the presence of sharp gradients or discontinuities (e.g., shock waves, boundary layers), the solution may not be smooth.

To overcome this, we multiply the governing equation by a test function  $\psi$  (which belongs to a suitable function space) and integrate over the computational domain  $\Omega$ . The result is the weak form of the PDE:

$$\iiint_{\Omega} \left( \frac{\partial \phi}{\partial t} \psi + \nabla \cdot (\vec{u}\phi) \psi \right) d\Omega = \iiint_{\Omega} (\nabla \cdot (\Gamma \nabla \phi) + \Phi_S) \psi d\Omega. \quad (2.3.2.2)$$

For the following calculations we will unite the terms of the right side in Eq 2.3.2.1 to a general source term :

$$\Phi_G = \nabla \cdot (\Gamma \nabla \phi) + \Phi_S$$

We now have:

$$\iiint_{\Omega} \frac{\partial \phi}{\partial t} \psi dV + \iiint_{\Omega} (\nabla \cdot (\vec{u}\phi)) \psi dV = \iiint_{\Omega} \Phi_G \psi dV \quad (2.3.2.3)$$

By applying the divergence theorem to  $\nabla \cdot ((\vec{u}\phi)\psi)$  we get:

$$\iiint_{\Omega} \nabla \cdot ((\vec{u}\phi)\psi) dV = \iint_{\partial\Omega} (\vec{u}\phi)\psi \cdot \vec{n} dS \quad (2.3.2.4)$$

where,  $\partial\Omega$  denotes the boundary of the domain. Integration by parts on the left side yields:

$$\iiint_{\Omega} \nabla \cdot ((\vec{u}\phi)\psi) dV = \iiint_{\Omega} (\nabla \cdot (\vec{u}\phi))\psi dV + \iiint_{\Omega} (\vec{u}\phi) \cdot \nabla \psi dV \quad (2.3.2.5)$$

substitution to Eq. 2.3.2.4, yields:

$$\iiint_{\Omega} (\nabla \cdot (\vec{u}\phi))\psi dV + \iiint_{\Omega} (\vec{u}\phi) \cdot \nabla\psi dV = \iint_{\partial\Omega} ((\vec{u}\phi)\psi) \cdot \vec{n} dS \quad (2.3.2.6)$$

A simple rearrangement leads us to the following expression:

$$\iiint_{\Omega} (\nabla \cdot (\vec{u}\phi))\psi dV = - \iiint_{\Omega} (\vec{u}\phi) \cdot \nabla\psi dV + \iint_{\partial\Omega} ((\vec{u}\phi)\psi) \cdot \vec{n} dS \quad (2.3.2.7)$$

We have now expressed the second term in Eq. 2.3.2.3 in a way that we can integrate the flux boundary condition while removing the differentiability constraint for the flux. Substitution of 2.3.2.7 to 2.3.2.3 yields:

$$\iiint_{\Omega} \frac{\partial\phi}{\partial t}\psi dV - \iiint_{\Omega} (\vec{u}\phi) \cdot \nabla\psi dV + \iint_{\partial\Omega} (\vec{u}\phi) \cdot \vec{n}\psi dS = \iiint_{\Omega} \Phi_G\psi dV \quad (2.3.2.8)$$

To obtain the FEM weak formulation we replace the general Source term  $\Phi_G$ :

$$\iiint_{\Omega} \frac{\partial\phi}{\partial t}\psi dV - \iiint_{\Omega} (\vec{u}\phi) \cdot \nabla\psi dV + \iint_{\partial\Omega} (\vec{u}\phi) \cdot \vec{n}\psi dS = \iiint_{\Omega} [\nabla \cdot (\Gamma\nabla\phi) + \Phi_S]\psi dV \quad (2.3.2.9)$$

We can further apply the divergence theorem to simplify the diffusion term  $\nabla \cdot (\Gamma\nabla\phi)$ , with the aid of the vector calculus identity:

$$\nabla \cdot (\Gamma\nabla\phi\psi) = \nabla \cdot (\Gamma\nabla\phi)\psi + \Gamma\nabla\phi \cdot \nabla\psi. \quad (2.3.2.10)$$

Rearrange to solve for  $\nabla \cdot (\Gamma\nabla\phi)\psi$ :

$$\nabla \cdot (\Gamma\nabla\phi)\psi = \nabla \cdot (\Gamma\nabla\phi\psi) - \Gamma\nabla\phi \cdot \nabla\psi. \quad (2.3.2.11)$$

Substitute this result into the integral of the diffusion term of EQ 2.3.2.9 after applying integration by parts:

$$\begin{aligned} & \iiint_{\Omega} \frac{\partial\phi}{\partial t}\psi dV - \iiint_{\Omega} (\vec{u}\phi) \cdot \nabla\psi dV + \iint_{\partial\Omega} (\vec{u}\phi) \cdot \vec{n}\psi dS \\ &= \iiint_{\Omega} [\nabla \cdot (\Gamma\nabla\phi\psi) - \Gamma\nabla\phi \cdot \nabla\psi] d\Omega + \iiint_{\Omega} \Phi_S\psi dV \end{aligned} \quad (2.3.2.12)$$

Apply the Divergence Theorem to the term involving the divergence on

the right side of the equation:

$$\iiint_{\Omega} \nabla \cdot (\Gamma \nabla \phi \psi) d\Omega = \iint_{\partial\Omega} (\Gamma \nabla \phi \psi) \cdot \vec{n} dS. \quad (2.3.2.13)$$

with the use of the property  $(\lambda \vec{a}) \cdot \vec{b} = \lambda(\vec{a} \cdot \vec{b})$  of the dot product with a scalar function, we can substitute back into the weak form and separate the boundary and volume integrals, obtaining:

$$\begin{aligned} & \iiint_{\Omega} \left( \frac{\partial \phi}{\partial t} \psi - \vec{u} \phi \cdot \nabla \psi + \Gamma \nabla \phi \cdot \nabla \psi \right) d\Omega \\ &= \iiint_{\Omega} \Phi_S \psi d\Omega + \iint_{\partial\Omega} (\Gamma \nabla \phi \cdot \mathbf{n}) \psi dS - \iint_{\partial\Omega} (\vec{u} \phi \cdot \mathbf{n}) \psi dS. \end{aligned} \quad (2.3.2.14)$$

In this final expression:

- $\iint_{\partial\Omega} (\Gamma \nabla \phi \cdot \mathbf{n}) \psi dS$  represents the boundary flux of the diffusion term.
- $\iint_{\partial\Omega} (\vec{u} \phi \cdot \mathbf{n}) \psi dS$  represents the boundary flux of the convective term.

This weak form is suitable for numerical methods like the Finite Element Method (FEM), as it allows the handling of boundary conditions and integrates the PDE in a weak sense, accommodating solutions with potential discontinuities or irregularities. It must hold for all test functions  $\psi$ , which leads to a system of algebraic equations when the domain is discretized. A common choice for test functions  $\psi$  is to use polynomials, but they can also be other types of functions. The test function  $\psi$  is typically chosen from the same function space as the solution  $\phi$ , resulting in what is known as the Galerkin method.

It is worth mentioning that if  $\psi = 1$ , we obtain the formulation for FVM, showing that this is just a special case of the generic weak formulation of FEM.

### 2.3.3 Discretization of Terms in the Governing Equation

#### Basis Functions

In FEM, the solution to a PDE is not computed directly but is instead approximated by expressing it as a linear combination of predefined functions called **basis functions**. These basis functions, often denoted as  $\psi_j$ , are defined over small subdomains of the problem domain  $\Omega$ , known as finite elements.

Basis functions typically have local support, meaning that each function is nonzero over only a small region (element) of the domain and zero

elsewhere. This property simplifies the computation of integrals over the entire domain, as only a small number of basis functions will contribute to the solution in any given element. These functions are designed to interpolate the solution at specific points within each element, called nodes. The value of the solution at any point within the element can thus be determined by these functions.

The approximate solution  $\phi_h$  within the domain  $\Omega$  is expressed as a linear combination of these basis functions:

$$\phi_h(x) = \sum_{j=1}^N \phi_j \psi_j(x),$$

where  $\phi_j$  are the unknown coefficients that represent the value of the solution at the nodes. These coefficients are what we aim to determine through the FEM.

Additionally, basis functions are often chosen as piecewise polynomials, which can be linear (P1 elements), quadratic (P2 elements), or of even higher order. The choice of basis functions influences the accuracy and computational cost of the FEM. Higher-order basis functions generally lead to more accurate solutions but require more computational resources.

For example, Tetrahedral elements are commonly used in 3D FEM due to their flexibility in meshing complex geometries and Basis functions are typically chosen as linear functions over the element. Imagine a tetrahedral element with vertices  $\mathbf{v}_1 = (x_1, y_1, z_1)$ ,  $\mathbf{v}_2 = (x_2, y_2, z_2)$ ,  $\mathbf{v}_3 = (x_3, y_3, z_3)$ , and  $\mathbf{v}_4 = (x_4, y_4, z_4)$ . A simple basis function  $\psi_j(x, y, z)$  for vertex  $j$  is:

$$\psi_j(x, y, z) = \frac{\det \begin{bmatrix} x - x_1 & y - y_1 & z - z_1 \\ x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \end{bmatrix}}{\det \begin{bmatrix} x_2 - x_1 & y_2 - y_1 & z_2 - z_1 \\ x_3 - x_1 & y_3 - y_1 & z_3 - z_1 \\ x_4 - x_1 & y_4 - y_1 & z_4 - z_1 \end{bmatrix}}$$

Here, the tetrahedral element is defined by four vertices. The basis function  $N_i$  represents how the solution varies linearly within the element relative to vertex  $i$ . The numerator determinant represents the volume formed by vectors from vertex  $\mathbf{v}_1$  to the point  $(x, y, z)$  and two other vertices. The denominator determinant represents the volume of the tetrahedron formed by the four vertices of the element.

$\psi_j(x, y, z)$  is a linear shape function that describes how the field variable changes linearly from vertex  $j$  to the rest of the element, and interpolates the value of a field variable (e.g., temperature, pressure) at any point within the tetrahedron by approximating it as a linear combination of values at the vertices. It is worth noting that  $\psi_j(x, y, z)$  equals 1 at its associated vertex  $v_j$  and 0 at the other vertices, ensuring proper interpolation. The basis function contributes to the field variable based on the volume of the sub-tetrahedron formed by  $v_j$  and the point  $(x, y, z)$ , partitioning the volume among vertices.

### Test Functions

The simplest test functions have the restriction to be nonzero in the vicinity of the "nodes" of the local functions leading to the computations of the integrals over the elements in this vicinity. For continuous base functions, the contribution from the boundary fluxes is used for elements that have a face at the boundary of a 3D flow, since internal elements' boundary fluxes cancel out.

In many cases, the test functions are chosen to be the same as the basis functions,  $\psi_i = \psi_j$ . This choice simplifies the formulation and leads to a symmetric system of equations, which is computationally efficient to solve. The Galerkin method, which is central to the FEM, involves selecting the test functions from the same space as the basis functions. This choice ensures that the resulting system of equations is symmetric and positive definite, properties that are highly desirable for numerical stability and the efficiency of iterative solvers.

The test functions, denoted by  $\psi_i$ , play a crucial role in the Galerkin method. The idea behind the Galerkin method is to minimize the residual of the PDE over the domain  $\Omega$  in a weighted sense. The residual is the difference between the exact PDE and the PDE when the approximate solution  $\phi_h$  is substituted into it. The test functions serve as these weights. This method involves multiplying the PDE by each test function  $\psi_i$  and integrating over the entire domain  $\Omega$ , ensuring that the weighted average of the residual is minimized, leading to a more accurate approximation of the solution.

### Application of the Galerkin Method

Consider the following PDE, which describes a general conservation law:

$$\begin{aligned} & \iiint_{\Omega} \left( \frac{\partial \phi}{\partial t} \psi - \vec{u} \phi \cdot \nabla \psi + \Gamma \nabla \phi \cdot \nabla \psi \right) d\Omega \\ &= \iiint_{\Omega} \Phi_S \psi d\Omega + \iint_{\partial\Omega} (\Gamma \nabla \phi \cdot \mathbf{n}) \psi dS - \iint_{\partial\Omega} (\vec{u} \phi \cdot \mathbf{n}) \psi dS. \end{aligned}$$

Here,  $\phi$  represents the unknown field variable (e.g., temperature, displacement, or pressure),  $\vec{u}$  is a velocity field,  $\Gamma$  is a diffusion coefficient,  $\Phi_S$  is a source term, and  $\mathbf{n}$  is the outward normal vector on the boundary  $\partial\Omega$ .

To apply the Galerkin method, we approximate  $\phi$  using the basis functions:

$$\phi_h(x) = \sum_{j=1}^N \phi_j \psi_j(x).$$

Substituting  $\phi_h$  into the PDE, and then multiplying by the test function  $\psi_i$  (where  $i = 1, 2, \dots, N$ ), we obtain the following integral expression:

$$\begin{aligned} & \sum_{j=1}^N \left[ \iiint_{\Omega} \frac{\partial \phi_j \psi_j}{\partial t} \psi_i d\Omega - \iint_{\partial\Omega} \vec{u} \phi_j \psi_j \cdot \nabla \psi_i d\Omega + \iiint_{\Omega} \Gamma \nabla(\phi_j \psi_j) \cdot \nabla \psi_i d\Omega \right] \\ &= \iiint_{\Omega} \Phi_S \psi_i d\Omega + \sum_{j=1}^N \left[ \iint_{\partial\Omega} (\Gamma \nabla \phi_j \psi_j \cdot \mathbf{n}) \psi_i dS - \iint_{\partial\Omega} (\vec{u} \phi_j \psi_j \cdot \mathbf{n}) \psi_i dS \right]. \end{aligned}$$

This equation represents the Galerkin discretization of the PDE. It transforms the continuous problem into a discrete one, resulting in a system of algebraic equations.

The unknowns in this system are the coefficients  $\phi_j$ , which can be solved using numerical methods. Each term in the equation represents an integral over the domain or its boundary, and each of these integrals is now expressed in terms of the basis functions and their gradients. The goal is to solve for the coefficients  $\phi_j$ . We can organize the terms as follows:

- **Mass Matrix (M):** This matrix corresponds to the term involving the time derivative:

$$M_{ij} = \iiint_{\Omega} \psi_j \psi_i d\Omega.$$

- **Convection Matrix (C):** This matrix corresponds to the term involving the velocity vector  $\vec{u}$ :

$$C_{ij} = \iint_{\Omega} \vec{u} \psi_j \cdot \nabla \psi_i d\Omega.$$

- **Stiffness Matrix (K):** This matrix corresponds to the diffusion term

involving  $\Gamma$ :

$$K_{ij} = \iiint_{\Omega} \Gamma \nabla \psi_j \cdot \nabla \psi_i d\Omega.$$

- **Source Vector (F):** This vector corresponds to the source term  $\Phi_S$ :

$$F_i = \iiint_{\Omega} \Phi_S \psi_i d\Omega.$$

- **Boundary Flux Contributions (G):** These terms involve integrals over the boundary  $\partial\Omega$ :

$$G_i = \iint_{\partial\Omega} [\Gamma \nabla \phi_j \cdot \mathbf{n} - \vec{u} \phi_j \cdot \mathbf{n}] \psi_i dS.$$

Putting everything together, the Galerkin representation of the equation becomes a system of algebraic equations:

$$\mathbf{M} \frac{d\Phi}{dt} + \mathbf{C}\Phi + \mathbf{K}\Phi = \mathbf{F} + \mathbf{G}, \quad (2.3.3.1)$$

where  $\Phi$  is the vector of coefficients  $\phi_j$ , and  $\mathbf{M}$ ,  $\mathbf{C}$ ,  $\mathbf{K}$ ,  $\mathbf{F}$ , and  $\mathbf{G}$  are the matrices and vectors defined above.

The Galerkin method ensures that the solution  $\phi_h$  is as accurate as possible within the function space spanned by the basis functions  $\{\psi_j\}$ . The accuracy of the solution depends on the choice of basis functions, the discretization of the domain, and the integration of the resulting system of equations. This method has the property of local conservation, especially when the test functions are chosen carefully, which is critical for problems in fluid dynamics and other conservation laws. By choosing higher-order basis functions, the accuracy of the solution can be increased without significantly increasing the number of elements, although at the cost of more complex computations.

## 2.4 Comparison of Methods

The choice of discretization method depends on various factors, including the nature of the flow problem, the complexity of the geometry, the desired accuracy, and the available computational resources. For simple geometries and smooth flows, FDM can be a computationally efficient choice. FVM offers a balance of flexibility and conservation, making it suitable for a wide range of applications. FEM excels in handling complex geometries and adaptive mesh refinement, but it may come with increased

computational demands.

Feature	FDM	FVM	FEM
<b>Governing Equation Form</b>	Differential	Integral	Weak (Variational)
<b>Grid Type</b>	Structured	Structured/Unstructured	Structured/Unstructured
<b>Variable Location</b>	Grid points	Cell centers	Grid nodes (vertices)
<b>Conservation Properties</b>	Not inherently conservative	Inherently conservative	Depends on formulation
<b>Geometric Flexibility</b>	Limited	Moderate	High
<b>Computational Efficiency</b>	High	Moderate	Moderate to Low
<b>Mathematical Complexity</b>	Low	Moderate	High

Table 2.1: Comparison of Finite Difference Method FDM, FVM FVM, and Finite Element Method (FEM).

While FDM, FVM, and FEM are the most prevalent discretization methods in CFD, there are alternative approaches that do not rely on a grid-based representation of the flow domain. These include meshfree methods, such as the vortex method and the lattice Boltzmann method, which offer unique advantages in certain scenarios. In our PhD study the FEM method is used via COMSOL CFD module due to its geometric flexibility, while its computational efficiency is improved by the steady state form equations used (Michos et al., 2024).

In the context of CFD, approximations arising from discretization or turbulence modeling are often represented using equal signs (=) instead of approximation symbols ( $\approx$  or  $\sim$ ). This convention reflects the fact that these approximations are embedded within the numerical solution process and are treated as exact within the computational framework.

By thoroughly understanding the fundamental principles and trade-offs of various discretization methods, researchers and engineers can make informed decisions when selecting the most suitable approach for their specific CFD simulations. In some cases, it may be advantageous to apply Pareto optimization (Catthoor et al., 2020) to dynamically choose the best method for different regions of the computational domain.

For instance, in areas of the domain that are less complex or relatively flat, methods like the FDM or FVM might offer a balance of speed and accuracy. These methods can efficiently handle simpler regions without compromising the overall solution. Conversely, for more complex regions of the domain, where detailed accuracy is essential—such as areas with sharp gradients, intricate geometries, or challenging boundary conditions—the FEM could be selectively applied. This hybrid approach ensures that computational resources are used efficiently while maintaining high accuracy where it is needed most.

As the development of numerical algorithms and computational capa-

bilities continues to advance, the ability of CFD to simulate increasingly complex and realistic flow phenomena also expands, pushing the boundaries of what can be achieved in engineering and scientific research. The base CFD model presented in Michos et al., 2024 and in Part II is an example of testing the limits of CFD application in terms of speed accuracy and resilience in data scarcity.

---

## References

- Catthoor, F., Basten, T., Zompakis, N., Geilen, M., & Kjeldsberg, P. G. (2020). *System-scenario-based design principles and applications* (Vol. 1). Springer. <https://doi.org/10.1007/978-3-030-20343-6>
- COMSOL Multiphysics® (CFD). (2020). *COMSOL Multiphysics® CFD module user's guide, version 5.6*. COMSOL. <https://doc.comsol.com/5.6/doc/com.comsol.help.cfd/CFDModuleUsersGuide.pdf>
- COMSOL Multiphysics® (CYCLOPEDIA). (2017). COMSOL Multiphysics® CYCLOPEDIA [[Online]. Last accessed: 24 August 2024]. <https://www.comsol.com/multiphysics/finite-element-method>
- COMSOL Multiphysics® (Reference Manual). (2020). *COMSOL Multiphysics® Reference Manual, version 5.6*. COMSOL. [https://doc.comsol.com/5.5/doc/com.comsol.help.comsol/COMSOL\\_ReferenceManual.pdf](https://doc.comsol.com/5.5/doc/com.comsol.help.comsol/COMSOL_ReferenceManual.pdf)
- Dick, E. (2009). Introduction to finite element methods in computational fluid dynamics. In J. F. Wendt (Ed.), *Computational fluid dynamics* (pp. 235–274). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-85056-4\\_10](https://doi.org/10.1007/978-3-540-85056-4_10)
- Ferziger, J. H., & Perić, M. (2002). *Computational methods for fluid dynamics*. Springer. <https://doi.org/10.1007/978-3-642-56026-2>
- Ferziger, J. H., & Perić, M. (2020). *Computational methods for fluid dynamics* (4th). Springer. <https://doi.org/10.1007/978-3-319-99693-6>
- Kajishima, T., & Taira, K. (2016). *Computational fluid dynamics: Incompressible turbulent flows*. Springer. <https://doi.org/10.1007/978-3-319-45304-0>
- Michos, D., Catthoor, F., Foussekis, D., & Kazantzidis, A. (2024). A cfd model for spatial extrapolation of wind field over complex terrain—wi. sp. ex. *Energies*, 17(16), 4139. <https://doi.org/10.3390/en17164139>
- Wendt, J. (2008). *Computational fluid dynamics: An introduction*. Springer Berlin Heidelberg. <https://books.google.gr/books?id=IIUkql-HNbQC>
- Wilcox, D. C. (2006). *Turbulence modeling for cfd* (3rd) [Includes bibliography, index and Compact Disk.]. DCW Industries, Inc. <http://www.dcwindustries.com>
- Zemansky, M., & Dittman, R. (1997). *Heat and thermodynamics: An intermediate textbook*. McGraw-Hill. <https://books.google.gr/books?id=4vVAAQAAIAAJ>

## **Part II**

# **The CRES Experiment**

# **Chapter 3**

## **CFD Modeling in COMSOL Multiphysics®**

The creation of a robust CFD simulation is highly dependant on the careful selection, processing, and implementation of input data. This is particularly critical when modeling turbulent flow in the vicinity of a wind farm situated on complex terrain.

This chapter provides a fast step-by-step guide on modeling an incompressible turbulent steady airflow over a realistic terrain in COMSOL Multiphysics like the one studied in the scope of this PhD, based on COMSOL Multiphysics® (CFD), 2020 , COMSOL Multiphysics® (CYCLOPEDIA), 2017 and COMSOL Multiphysics® (Reference Manual), 2020. We will discuss the full CFD modeling process, giving a detailed view of the methodology presented in Michos et al., 2024, by integrating high-resolution terrain data, LIDAR-based wind velocity measurements, and time-averaged temperature data in COMSOL, to create a realistic and informative simulation environment.

In COMSOL Multiphysics, the term "stationary" is used to describe a study type or solver mode that seeks the steady-state solution of a physical system. This terminology aligns with the common CFD usage of "steady state," where it denotes a time-invariant flow, despite possible spatial variations. In this chapter we will also use the term "stationary" to align with COMSOL Multiphysics® (Reference Manual), 2020 for the demonstration of the modeling flow withing the software, remembering that it conceptually corresponds to the "steady-state" solutions discussed in CFD textbooks like Dick, 2009.

## 3.1 Geometry and Boolean operators

In CFD, the correct representation of the physical domain is crucial. To make this demonstration simple, we will create a complex terrain as shown in figure 3.1, representing the CRES Wind Farm in Lavrio, which is the site we used to test the models created in this PhD research. The terrain surface spans (horizontally) 2km in x and 2.5km in y direction. A high-resolution digital elevation model (DEM) captures the complex topography of the Lavrio site, serving as the foundation for defining the lower boundary of the computational domain. Imported into COMSOL as a CAD surface or mesh (figure 3.1a), this DEM delineates the interface between the solid terrain and the fluid airspace.

In COMSOL Multiphysics, the terrain and the expansive air volume above it are encapsulated within a single, unified computational domain. This approach streamlines the modeling process, ensuring that the interaction between the solid terrain and the fluid flow is captured seamlessly. COMSOL offers a powerful Boolean operation framework to achieve this in the geometry section.

A cube, often referred to as a bounding box, of preferred dimensions is created from the geometry section and serves as the initial geometric entity (figure 3.1b). This cube encompasses the entire region of interest, ensuring that the imported terrain is fully contained within its boundaries. The core of the terrain definition lies in a Boolean subtraction operation. This operation subtracts the CAD surface from the initial cube, effectively carving out the terrain and leaving behind the air volume above it (figure 3.1c). We can now delete the unwanted objects by deleting them (figure 3.1d).

Boolean operations are versatile tools in COMSOL's geometry toolbox, enabling the creation and manipulation of complex shapes through the combination or subtraction of solid objects. Beyond terrain modeling, they find applications in various engineering and scientific domains. COMSOL supports a range of Boolean operations, including unions (combining objects), intersections (finding overlapping regions), and differences (subtracting one object from another). These operations can be applied to diverse geometric entities like solids, surfaces, curves, and even imported CAD models. For meshing related reasons, presented in the next section, we divide the domain in two areas by creating a smaller cube and partitioning the domain with it. The resulting geometry, representing the airspace above the terrain, becomes the computational domain for the

fluid flow simulation, shown in the final geometrical representation of figure 3.1.

It is worth mentioning that boolean operations can be parameterized, allowing you to explore the impact of design changes or geometric variations on the simulation results. This capability is particularly valuable for optimization studies or sensitivity analyses. COMSOL seamlessly integrates Boolean operations into its modeling workflow. You can access these operations through the intuitive graphical user interface or leverage the built-in programming language for more intricate or automated geometry generation.

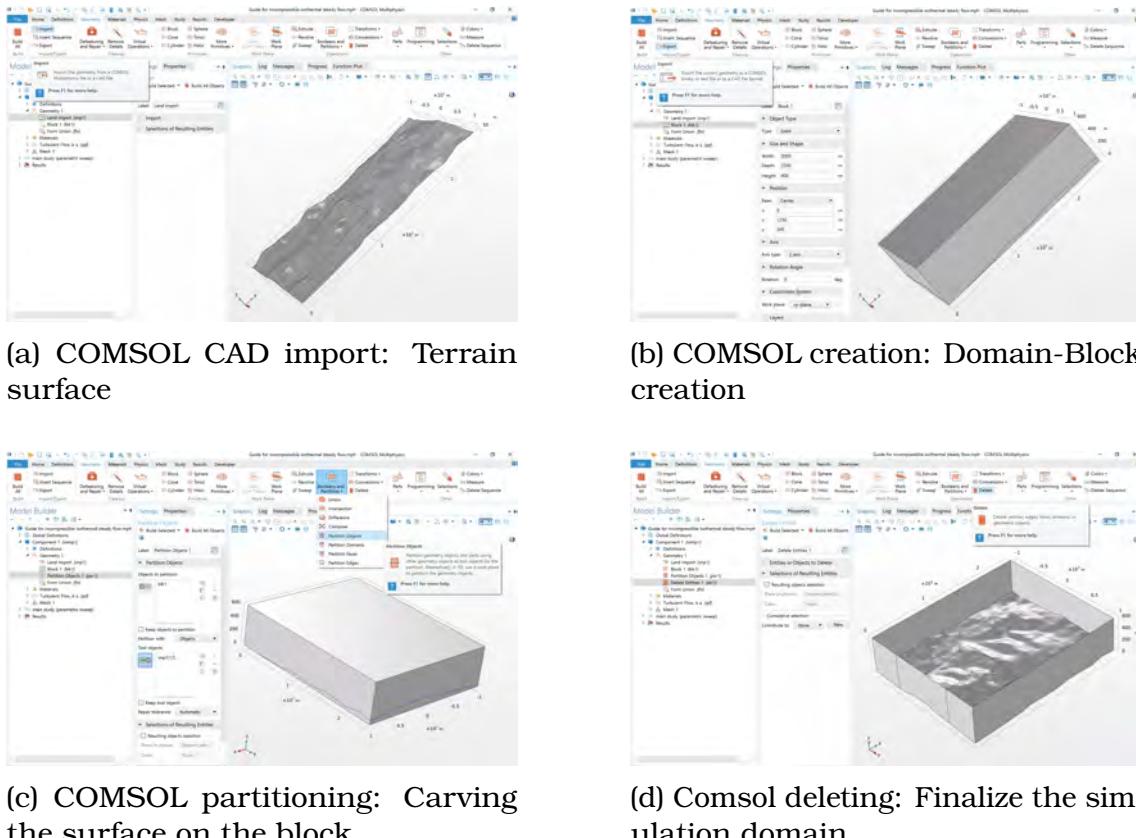


Figure 3.1: Setting up a Geometry for real simulations in COMSOL

## 3.2 Mesh

Terrain surfaces are rarely uniform; they exhibit a tapestry of features ranging from gentle slopes to abrupt cliffs. Similarly, the fluid flow over such terrain can experience localized complexities near obstacles or regions of sharp elevation changes. A non-uniform mesh allows for finer discretization in these critical areas, capturing intricate details and flow

patterns that would be missed with a uniform mesh. Conversely, regions with smoother terrain or less turbulent flow can be adequately represented with larger elements, thus optimizing computational resources.

The choice of mesh resolution directly impacts the accuracy and computational cost of a simulation. A very fine mesh with millions of elements can capture the terrain's topography with high precision but significantly increases computational demands. Conversely, a coarse mesh reduces computational cost by sacrificing detail in the terrain curvature, potentially affecting accuracy or convergence.

The Lavrio site, is characterized as "complex" in Bingöl et al., 2011, due to the presence of hills, valleys, and varying surface roughness. This topographical complexity can induce significant variations in wind flow patterns, including flow acceleration, separation, and turbulence generation. These complexities necessitate careful consideration in both measurement and modeling endeavors.

We can discretize our domain of interest through a non-uniform mesh composed of tetrahedral elements as shown in figure 3.2. These three-dimensional, triangular-based pyramids are favored for their flexibility in adapting to complex geometries. The non-uniform nature of the mesh signifies that the size of these tetrahedral elements varies strategically across the domain capturing the fine scale details where needed.

We want to simulate a wind field for evaluation purposes focusing on approximately 50 meters above the ground level and beyond. When simulating a wind field, especially at distances such as 50 meters Above Ground Level (AGL), the turbulence generated by micro-scale surface stress may not be critical. In such cases, using a mesh with extra coarse regions (figure 3.2a) and normal resolution meshed regions can be a practical choice (figure 3.2b), offering a balance between simulation speed and the level of detail required. This trade-off is particularly valuable for obtaining preliminary insights or conducting rapid design iterations, where the objective is to achieve sufficiently accurate results within a reasonable timeframe. This is the reason we partitioned our domain in two subdomains. The outer region represents the surroundings of Center For Renewable Energy Sources (CRES), while the internal domain represents the WF, where the WTs are located.

COMSOL offers a suite of advanced automatic mesh generation algorithms that streamline this process, even for complex geometries. Recognizing that different physical phenomena demand varying levels of mesh resolution, COMSOL offers physics-controlled meshing. This feature au-

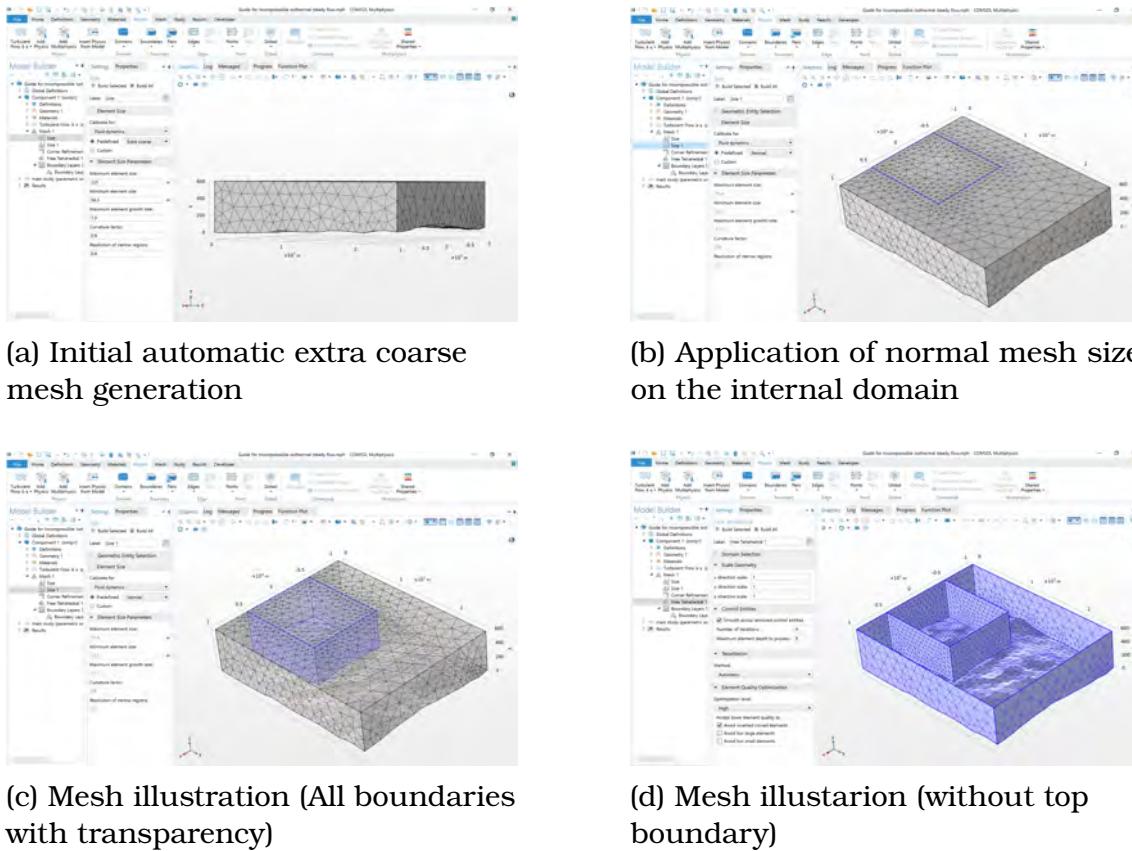


Figure 3.2: Meshing a Geometry in COMSOL

tomatically refines the mesh in regions where the solution is expected to exhibit steep gradients or rapid changes, ensuring accurate capture of boundary layers, shock waves, or other critical flow features.

When the solution's behavior is difficult to predict a priori, adaptive meshing is extremely useful. This iterative process involves solving the problem on an initial mesh, estimating errors, and then refining the mesh in areas with high errors. This cycle continues until the solution converges within specified tolerances, striking an optimal balance between accuracy and computational cost. To asses the quality of a generated mesh, COMSOL provides a range of mesh quality metrics, such as skewness, aspect ratio, and element size variation as shown in figures 3.2a and 3.2c. These metrics help you evaluate the suitability of the mesh for your simulation and identify areas that might require further refinement or improvement. The meshing of the terrain is depicted in figure 3.2d.

The foundation of any successful CFD simulation rests on a well-defined domain and an appropriate mesh selection. The combination of Boolean operations for precise geometry creation and automatic mesh generation algorithms for efficient discretization, empowers you to overcome complex terrain modeling challenges. By understanding the principles and capa-

bilities discussed in COMSOL Multiphysics® (Reference Manual), 2020, the generation of sufficiently accurate meshed domains in COMSOL is a relatively straightforward task.

### 3.3 Material Properties

Defining the fluid of our specific scenario as "air" within COMSOL is a fundamental step in setting up our simulations (figure 3.3). This selection triggers the software to access its extensive material database, where it retrieves essential fluid properties such as density, viscosity, specific heat, and thermal conductivity. These default values are based on standard conditions, providing a reliable starting point for accurate modeling. However, COMSOL also offers the flexibility to adjust these properties to match specific environmental conditions or experimental data, ensuring that simulations reflect real-world scenarios.

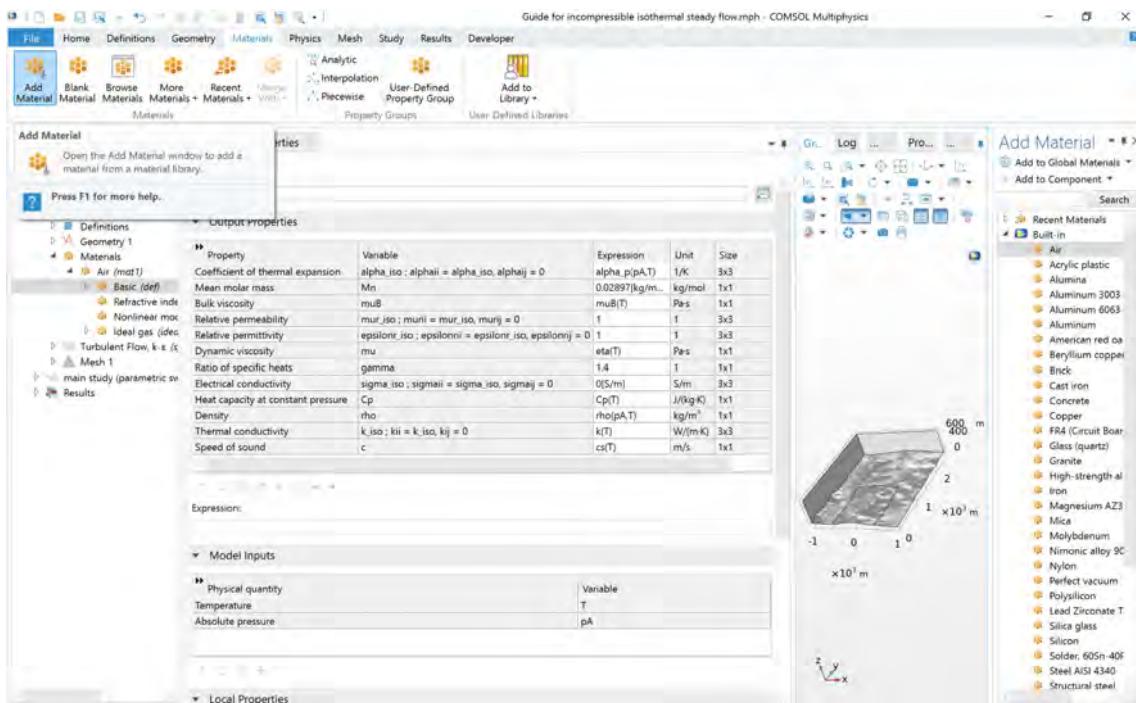


Figure 3.3: Defining the fluid properties in COMSOL

Fluid properties, such as density and viscosity, play a critical role in determining the behavior of the fluid within the simulation. Density influences the fluid's response to forces like gravity and acceleration, while viscosity affects its resistance to flow and the development of boundary layers. Specific heat and thermal conductivity are crucial for heat transfer simulations, dictating how the fluid's temperature changes and how

heat is distributed within and across the fluid's boundaries. These intrinsic properties are central to predicting accurate flow patterns, pressure distributions, and thermal behaviors in the simulated environment.

It is worth mentioning that COMSOL multiphysics framework is particularly powerful in scenarios where fluids interact with solid structures, such as air flowing around wind turbines. This framework allows for seamless coupling between different physics interfaces, enabling the exchange of information between fluid and solid domains. For instance, in fluid-structure interaction (FSI) simulations, the framework manages the transfer of material properties like thermal conductivity and the coupling of forces and displacements. This ensures that the interactions between the fluid and the structure are accurately represented, which is essential for realistic modeling of complex systems.

## 3.4 The 'Turbulent k-epsilon' Interface

Turbulence is a complex phenomenon in fluid dynamics, found in a broad spectrum of flow scales. These scales range from the largest, which are influenced by the geometry of the flow domain, to the smallest, rapidly fluctuating scales like the ones produced close to the surface by the contact with surface due to surface roughness like in the CRES complex terrain we are modeling. In fluid dynamics, the concepts of shear stress  $\tau$  and shear rate  $\gamma$  are fundamental for understanding how fluids behave under different conditions.

Shear stress measures the force per unit area exerted parallel to a surface within a fluid. It describes the force required to make one layer of fluid slide over an adjacent layer. Shear stress can be calculated using the following formula:

$$\tau = \frac{F}{A}$$

where  $F$  is the force applied parallel to the surface (in newtons, N), and  $A$  is the area over which the force is applied (in square meters, m<sup>2</sup>).

Shear rate quantifies how quickly adjacent layers of fluid move relative to each other. It measures the rate at which the fluid is deforming due to the applied shear stress. Mathematically, shear rate is defined as:

$$\gamma = \frac{du}{dy}$$

where  $du$  is the infinitesimal difference in velocity between the fluid layers,

$dy$  is the infinitesimal distance between the layers and  $\frac{du}{dy}$  is the velocity gradient perpendicular to the direction of shear. Shear rate is typically measured in reciprocal seconds ( $s^{-1}$ ), representing the rate at which layers of fluid are moving relative to each other. Specifically, it tells us how much velocity changes per unit of distance perpendicular to the direction of flow.

We can now define Dynamic viscosity ( $\mu$ ), that measures a fluid's resistance to shear or deformation and quantifies the internal friction within the fluid as it flows. It is defined as the ratio of shear stress to shear rate:

$$\mu = \frac{\tau}{\gamma}$$

In other words, dynamic viscosity indicates the amount of force required to deform the fluid relative to its shear rate.

Understanding these concepts is crucial for analyzing fluid behavior under various conditions and for applications in fluid dynamics. For example, to measure the tendency of a flow to transition to a turbulent state, particularly in isothermal conditions like the one we are modeling in our case, we use the Reynolds number:

$$Re = \frac{\rho U L}{\mu}$$

where  $\rho$  is the density,  $U$  is the characteristic velocity scale of the flow, and  $L$  are velocity and length scales of the flow.  $U$  represents the typical speed at which the fluid is moving. This could be the maximum velocity, average velocity, or a reference velocity depending on the specific flow situation.  $L$  is the characteristic length scale of the flow. It represents a significant physical dimension of the system, such as the diameter of a pipe, the length of a boundary layer, or the size of an object around which the fluid is flowing.

### 3.4.1 RANS Turbulence Modeling

The Navier-Stokes equations (continuity and momentum) can model turbulent flows, but doing so requires a very fine computational mesh due to the wide range of turbulence scales. A more efficient approach is to decompose flow variables into mean and fluctuating components, with the effects of fluctuations approximated using a turbulence closure model. This reduces computational costs compared to directly resolving all turbulence scales. Different turbulence closures, based on various

assumptions about turbulence, offer varying accuracy depending on the flow conditions.

In the RANS modeling approach (demonstrated by COMSOL Multiphysics® (CFD), 2020), assuming that the fluid is incompressible and Newtonian like in our case, the Navier-Stokes equations used have the form:

$$\begin{aligned}\rho \nabla \cdot u &= 0 \\ \rho \frac{\partial \vec{u}}{\partial t} + \rho(\vec{u} \cdot \nabla) \vec{u} &= \nabla [-p\mathbf{I} + \mu(\nabla \vec{u} + (\nabla \vec{u})^T)] + \mathbf{F}\end{aligned}$$

where  $p$  is the pressure and  $I$  is the identity matrix or tensor. A scalar variable of the flow  $\phi$  is decomposed into an averaged value  $\bar{\phi}$  and a fluctuating part  $\phi'$ :

$$\phi = \bar{\phi} + \phi'$$

where  $\phi$  can represent any scalar quantity of the flow, and is subject to spatial and temporal variations. Applying this Decomposition technique on the Navier-Stokes equation, gives the RANS equations:

$$\rho \nabla \cdot \bar{u} = 0$$

$$\rho \left( \frac{\partial \bar{u}}{\partial t} + \bar{u} \cdot \nabla \bar{u} \right) + \nabla \cdot (-\overline{\rho u' \otimes u'}) = \nabla [-\bar{p}I + \mu(\nabla \bar{u} + (\nabla \bar{u})^T)] + F$$

where  $\bar{u}$  is the averaged velocity field and  $\otimes$  is the outer vector product. The term  $u' \otimes u'$  accounts for the interaction between fluctuating components of the velocity field is known as the Reynolds stress tensor. This term reflects the impact of turbulence on the mean flow and is necessary to model small-scale turbulent fluctuations interact with one another. Specifically, the Reynolds stress tensor requires information about the correlations between velocity fluctuations in all three spatial directions. These correlations provide insight into how turbulent eddies and variations in velocity contribute to the overall stress and momentum transfer within the fluid. Assuming that turbulence has diffusive characteristics, it can be modeled as follows:

$$\rho (\overline{u' \otimes u'}) - \frac{2}{3} \rho k = \rho (\overline{u' \otimes u'}) - \frac{\rho}{3} \text{trace} (\overline{u' \otimes u'}) I = \mu_T (\nabla \bar{u} + (\nabla \bar{u})^T)$$

where  $\mu_T$  is the eddy viscosity, also known as the turbulent viscosity, and  $k$  is the turbulent kinetic energy. The term  $\frac{2}{3} \rho k$  accounts for isotropic

turbulence, and  $\frac{2}{3}\rho k$  reflects the mean pressure due to turbulence while the expression  $\mu_T (\nabla \bar{u} + (\nabla \bar{u})^T)$  denotes the contribution from turbulent viscosity (or eddy viscosity) to the stress within the fluid by modeling the diffusion of momentum resulting from turbulence, with  $\mu_T$  being the eddy viscosity.

For incompressible flows, the turbulence effects are often absorbed into the pressure term, effectively modifying the pressure field. The turbulence stress tensor  $-\rho u' \otimes u'$  is treated as part of the pressure gradient, simplifying the computation since it does not need to be explicitly modeled as a separate term in the equations. On the other hand, in compressible flows, the situation is different because the absolute pressure level becomes significant. Turbulence effects cannot be simply absorbed into the pressure term and must be explicitly included in the formulation, thus the turbulence stress tensor  $\rho(\bar{u}' \otimes \bar{u}')$  is not absorbed into the pressure term.

### 3.4.2 k-epsilon

The choice of the 'Turbulent k-epsilon' interface within the 'Single-Phase Flow' branch sets the foundation for our turbulent flow simulation. This interface uses a set of governing equations and modeling assumptions suitable for capturing the complexity of turbulent flows, which are characterized by chaotic fluctuations in velocity and pressure, in a wide range of length and time scales. Directly resolving all these fluctuations in a simulation is too computationally intensive for most practical applications. The 'k-epsilon' model is a widely used RANS approach, offering a computationally "viable" solution.

As we previously discussed, the RANS equations introduce the Reynolds stress tensor  $\tau$ , which represents the effect of turbulent fluctuations on the mean flow. In this COMSOL interface, it is defined as a tensor with elements:

$$\tau_{ij} = -\rho \bar{u}'_i \bar{u}'_j$$

where  $\bar{u}'_i \bar{u}'_j$  is the time-averaged product of the velocity fluctuations in the  $i$ -th and  $j$ -th directions. The indices  $i$  and  $j$  take values corresponding to the spatial dimensions of the flow. For example,  $i = 1, 2, 3$  could correspond to the  $x$ ,  $y$ , and  $z$  directions, respectively. Similarly,  $j = 1, 2, 3$  corresponds to the  $x$ ,  $y$ , and  $z$  directions. The term  $\bar{u}'_i \bar{u}'_j$  represents the correlation between the fluctuations of the velocity components in the  $i$ -th and  $j$ -th directions. When  $i = j$ , this term represents the normal stresses

(e.g.,  $\overline{u'_1 u'_1}$  corresponds to the fluctuations in the  $x$ -direction). When  $i \neq j$ , it represents the shear stresses (e.g.,  $\overline{u'_1 u'_2}$  corresponds to the correlation between fluctuations in the  $x$ - and  $y$ -directions). For example,  $\tau_{11}$  represents the Reynolds stress due to the fluctuations in the  $x$ -direction ( $u'_1$ ), while  $\tau_{12}$  represents the Reynolds stress due to the interaction between fluctuations in the  $x$ - and  $y$ -directions ( $u'_1$  and  $u'_2$ ).

The introduction of the Reynolds stress tensor adds new unknowns (the components  $\tau_{ij}$ ) that need to be solved. However, the original Navier-Stokes equations do not provide enough equations to solve for these new unknowns directly. This is known as the "closure problem" leading to more unknowns than equations. Turbulence models like the  $k-\varepsilon$  model provide additional equations to relate these Reynolds stresses to the mean flow variables, allowing the system to be closed and solved. The model owes its name to these equations, the turbulent kinetic energy equation  $k$  and the dissipation rate equation  $\varepsilon$ .

In turbulent flows, kinetic energy moves from large, energy-rich vortices to progressively smaller structures until it is fully converted into heat through viscous dissipation at the smallest scales. The dissipation rate of turbulent kinetic energy, represented by  $\epsilon$ , measures the rate at which this process of energy transfer and dissipation unfolds. This irreversible energy loss within turbulent flows, is associated with the smallest turbulence scales, referred to as Kolmogorov microscales. These microscales are where the turbulence is most intense, characterized by rapid rotation and significant viscous effects.  $\epsilon$  is directly related to the Kolmogorov length scale ( $\eta$ ), which characterizes the smallest eddies in the flow:

$$\eta = \left( \frac{\nu^3}{\epsilon} \right)^{\frac{1}{4}} \quad (3.4.2.1)$$

where  $\nu$  is the kinematic viscosity.

The dissipation rate  $\epsilon$  is a crucial variable in the  $k-\epsilon$  turbulence model, influencing the calculation of turbulent viscosity and the overall prediction of turbulent flow behavior.

After covering the basics of turbulence modeling, we can now get a look at the equations solved by this interface for transient incompressible flow solutions, according to COMSOL Multiphysics® (CFD), 2020, COMSOL Multiphysics® (Reference Manual), 2020 and COMSOL Multiphysics® (CYCLOPEDIA), 2017.

## Navier-Stokes equations:

- **Continuity Equation**

$$\rho \nabla \cdot \bar{u} = 0 \quad (3.4.2.2)$$

Here,  $\rho$  is fluid density and  $\bar{u}$  is time-averaged velocity vector

- **Momentum Equation**

$$\rho \frac{\partial \bar{u}}{\partial t} + \rho(\bar{u} \cdot \nabla) \bar{u} = \nabla \cdot [-\bar{p}\mathbf{I} + (\mu + \mu_T)(\nabla \bar{u} + (\nabla \bar{u})^T)] + \mathbf{F} \quad (3.4.2.3)$$

, where  $\bar{p}$  is Time-averaged pressure,  $\mu$  is Dynamic viscosity,  $\mu_T$  is Turbulent viscosity ( $\mu_T = \rho C_\mu \frac{k^2}{\epsilon}$ ),  $\mathbf{I}$  is Identity matrix,  $\mathbf{F}$  is External forces

## k- $\epsilon$ Model Equations:

- **Turbulent Kinetic Energy (k) Equation**

$$\rho \frac{\partial k}{\partial t} + \rho(\bar{u} \cdot \nabla) k = \nabla \cdot [(\mu + \mu_T/\sigma_k) \nabla k] + P_k - \rho \epsilon \quad (3.4.2.4)$$

where  $k$  is the turbulent kinetic energy,  $\mu$  is the dynamic viscosity,  $\mu_T$  is the turbulent viscosity (calculated as  $\mu_T = \rho C_\mu \frac{k^2}{\epsilon}$ ),  $\sigma_k$  is the turbulent Prandtl number for  $k$ , and  $P_k$  is the production of turbulent kinetic energy representing the rate at which turbulent kinetic energy is generated due to the interaction between the mean flow and the turbulent fluctuations. It signifies the transfer of energy from the mean flow to the turbulent eddies. The term  $\rho \epsilon$  represents the destruction or dissipation of turbulent kinetic energy. This equation describes the transport of  $\epsilon$  due to convection, diffusion, production, and destruction.

The equation for the production term  $P_k$  is given as:

$$P_k = \mu_T \left( \nabla \vec{u} : (\nabla \vec{u})^T + \frac{2}{3} (\nabla \cdot \vec{u})^2 I \right) - \frac{2}{3} \rho k \nabla \cdot \vec{u}$$

Here,  $\nabla \vec{u} : (\nabla \vec{u})^T$  represents the double inner product (or contraction) between the velocity gradient tensor and its transpose, quantifying the rate of strain in the fluid due to the mean flow.  $P_k$  accounts for the generation of turbulent kinetic energy due to two main mechanisms. The term  $\mu_T \{ \nabla \vec{u} : (\nabla \vec{u})^T \}$  captures the production of turbulent kinetic energy due to shear in the mean flow. Velocity gradients in the flow

cause stretching and deformation of turbulent eddies, leading to an increase in their kinetic energy. The term  $\frac{2}{3}\mu_T(\nabla \cdot \vec{u})^2 I$  represents the production due to dilatation or expansion/compression of the fluid. This term is usually negligible for incompressible flows where  $\nabla \cdot \vec{u} = 0$ . The last term  $-\frac{2}{3}\rho k \nabla \cdot \vec{u}$  is also related to dilatation effects and is typically zero for incompressible flows.

- **Dissipation Rate ( $\epsilon$ ) Equation**

$$\rho \frac{\partial \epsilon}{\partial t} + \rho(\bar{u} \cdot \nabla)\epsilon = \nabla \cdot [(\mu + \mu_T/\sigma_\epsilon) \nabla \epsilon] + C_{\epsilon 1} \frac{\epsilon}{k} P_k - C_{\epsilon 2} \frac{\rho \epsilon^2}{k} \quad (3.4.2.5)$$

, where  $\sigma_k, \sigma_\epsilon, C_{\epsilon 1}, C_{\epsilon 2}$  are model constants. Specifically,  $\sigma_k$  and  $\sigma_\epsilon$  are the Turbulent Prandtl Numbers for  $k$  and  $\epsilon$ , while  $C_{\epsilon 1}$  and  $C_{\epsilon 2}$  are the production and Destruction Coefficients.

The right selection  $C_{\epsilon 1}$  and  $C_{\epsilon 2}$  helps achieving a balanced turbulence model.  $C_{\epsilon 1}$  controls the generation of  $\epsilon$ , while  $C_{\epsilon 2}$  controls its destruction. The ratio  $C_{\epsilon 1}/C_{\epsilon 2}$  indirectly influences the turbulent viscosity ( $\mu_T$ ) and, consequently, the overall turbulence levels in the flow.

By selecting the 'Turbulent k-epsilon' interface, we are essentially instructing COMSOL to solve the RANS equations along with the k-epsilon model equations to predict the time-averaged flow behavior in our stationary turbulent flow scenario (figure 3.4).

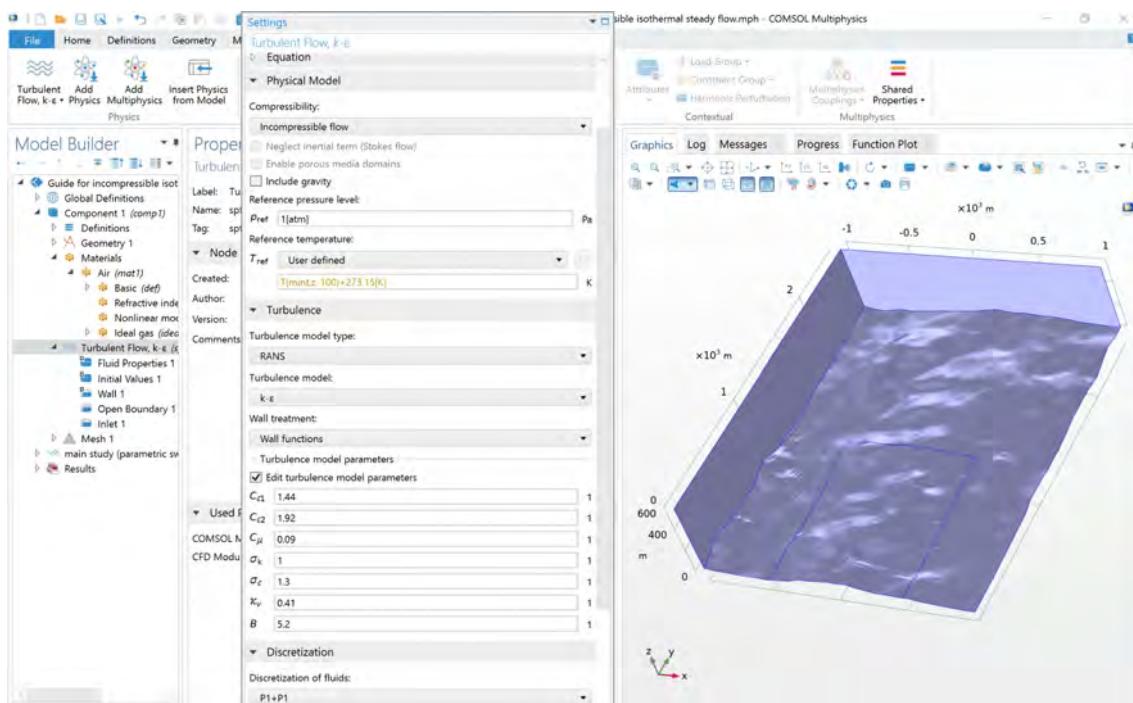


Figure 3.4: k-epsilon stationary study equations in COMSOL

### 3.4.3 Default Parameter Values

In occasions that measurements are not sufficient to derive the appropriate values, like in our example where measurements are missing close to the terrain surface (wall boundary) we can retain the default values for the 'k-epsilon' model parameters relying on the model's well-established calibration and validation. If the focus of the solutions is far from the surface of the boundary, this might be sufficient but still not desirable.

The 'k-epsilon' model involves several constants (e.g.,  $C_\mu$ ,  $\sigma_k$ ,  $\sigma_\epsilon$ ) that have been extensively tuned and tested against experimental data and can be found in COMSOL Multiphysics® (CFD), 2020 and are shown in figure 3.5. It's generally advisable to use these values when data availability is insufficient to help us understand if they are optimal for our flow. In certain specialized scenarios, where prior knowledge or experimental evidence suggesting that the default values might not be optimal, COMSOL allows the modification of these parameters based on your expertise.

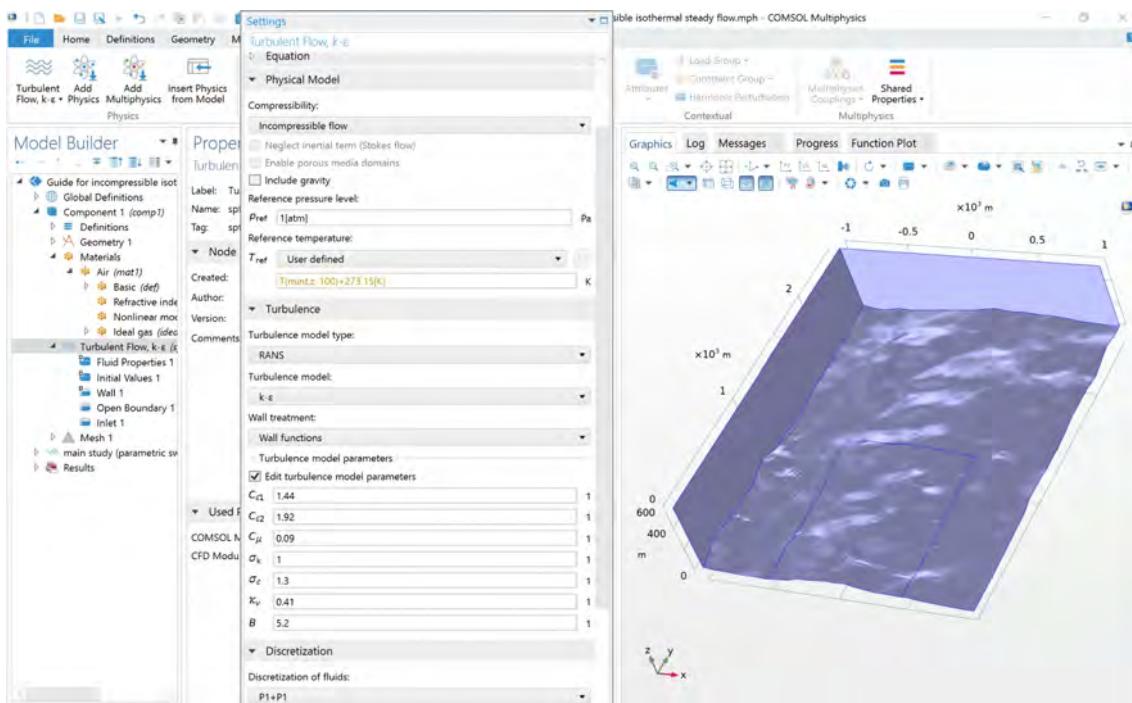


Figure 3.5: k-epsilon default turbulence parameters in COMSOL

### 3.4.4 Wall Functions

In turbulent flow simulations, accurately resolving the near-wall region, where viscous effects dominate, is crucial for predicting overall flow behavior and phenomena like drag and heat transfer. However, this demands significant computational resources due to the necessity of fine mesh reso-

lution within the thin boundary layer. To enhance computational efficiency without sacrificing accuracy, the k- $\epsilon$  turbulence model in COMSOL employs wall functions by default (figure 3.5).

The assumption that there is a gap between the wall and the computational domain is made. This gap can be related to the size of the mesh cells close to the wall. This gap can be explained in viscous units by obtaining the max of two arguments.

$$\delta_w^+ = \max\left(\frac{h \rho C_\mu^{1/4} \sqrt{k}}{\mu}, 11.06\right)$$

The first argument is derived from the log-law of the wall, a fundamental principle in turbulent boundary layer theory. It essentially estimates the dimensionless distance from the wall to the first computational node based on the local flow conditions (represented by  $k$ ) and mesh resolution  $h$ .

The second argument represents the dimensionless distance from the wall where the logarithmic layer would theoretically intersect with the viscous sublayer if there were no buffer layer in between. This value acts as a lower limit for  $\delta_w^+$ , ensuring that the wall functions remain mathematically valid and avoid singularities, regardless of the flow's Reynolds number.

$$\delta_w = \frac{\delta_w^+ \mu}{\rho u_\tau}$$

where friction velocity is modeled as:

$$u_\tau = \max(C_\mu^{1/4} \sqrt{k}, \frac{\|u\|}{u^+})$$

and

$$u^+ = \frac{1}{\kappa_v} \log(\delta_w^+) + B$$

here, we see the von Kármán constant ( $\kappa_v = 0.41$ ) and an other constant set by Comsol,  $B = 5.2$ . The two arguments for  $u_\tau$ . More information on the wall functions can also be found in Wilcox, 2006, in addition to COMSOL Multiphysics® (Reference Manual), 2020.

Wall functions are semi-empirical formulations connecting the fully turbulent flow in the core region and the viscous-dominated flow near the wall. They leverage the "law of the wall", which is an empirical relationship describing the mean velocity profile within the turbulent boundary layer. This law divides the near-wall region into three layers: the viscous sublayer, the buffer layer, and the log-law region.

### 3.4.5 P1+P1 Discretization

In our modeling approach we will keep the default P1+P1 discretization (figure 3.5), which is a finite element method (FEM) approach where both the velocity field  $u$  and the pressure field  $p$  are approximated using linear Lagrange elements (COMSOL Multiphysics® (Reference Manual), 2020). This implies a linear variation of both velocity and pressure within each element concerning the spatial coordinates.

#### Linear Lagrange Elements in a 3D Non-Uniform Tetrahedral Mesh

Once our mesh is generated, approximations can be introduced to the dependent variables in the simulation. For simplicity, let's consider a single dependent variable,  $u$ . The objective is to approximate  $u$  with a function described using a finite number of parameters, known as Degrees Of Freedom (DOF). Incorporating this approximation into the weak form of the governing equations leads to a system of equations for the DOF.

For a 3D mesh composed of tetrahedral elements like in our case. each tetrahedral element is defined by four nodes (vertices), and linear elements imply that within each tetrahedral element, the function  $u$  varies linearly. To uniquely define  $u$  within each element, the values of  $u$  at the nodes of the tetrahedron are required. For a single tetrahedron with nodes located at points  $(x_1, y_1, z_1)$ ,  $(x_2, y_2, z_2)$ ,  $(x_3, y_3, z_3)$ , and  $(x_4, y_4, z_4)$ , denote these values as  $U_1 = u(x_1, y_1, z_1)$ ,  $U_2 = u(x_2, y_2, z_2)$ ,  $U_3 = u(x_3, y_3, z_3)$ , and  $U_4 = u(x_4, y_4, z_4)$ . These values represent the degrees of freedom. The function  $u$  within the element can then be expressed as:

$$\begin{aligned} u(x, y, z) &= \sum_{i=1}^4 U_i \\ &= \phi_i(x, y, z) = U_1\phi_1(x, y, z) + U_2\phi_2(x, y, z) + U_3\phi_3(x, y, z) + U_4\phi_4(x, y, z), \end{aligned}$$

where  $\phi_i(x, y, z)$  are the basis (shape) functions. Each  $\phi_i(x, y, z)$  is linear within the tetrahedral element, equals 1 at the  $i$ -th node, and equals 0 at the other nodes. For our example:

$$\phi_1(x, y, z) = 1 - \xi_1 - \xi_2 - \xi_3,$$

$$\phi_2(x, y, z) = \xi_1,$$

$$\phi_3(x, y, z) = \xi_2,$$

$$\phi_4(x, y, z) = \xi_3,$$

where  $\xi_1, \xi_2, \xi_3$  are the local barycentric coordinates of the tetrahedral element.

In general, a finite element space is specified by defining a set of basis functions. For tetrahedral elements in 3D, these basis functions are typically defined in terms of local coordinates  $\xi_1, \xi_2, \xi_3$ , which parameterize the element. Consider a mesh element of dimension  $d$  in an  $n$ -dimensional geometry with spatial coordinates denoted as  $x_1, \dots, x_n$ . The standard  $d$ -dimensional simplex is given by:

$$\sum_{i=1}^d \xi_i \leq 1, \quad \xi_i \geq 0.$$

In 3D, this simplex is a tetrahedron. The global spatial coordinates  $x_i$  are defined as suitable linear (affine) functions of these local coordinates, mapping the standard simplex to its actual position, orientation, and shape in the computational domain.

When expressed in terms of local coordinates, the basis functions are simplified to a few basic shapes, termed shape functions. In the case of linear elements in 3D, any basis function on any mesh element is one of the forms described earlier.

### **The Ladyzhenskaya–Babuška–Brezzi (LBB) Condition**

In COMSOL Multiphysics, when using the P1+P1 element combination (linear Lagrange elements for both velocity and pressure) for incompressible turbulent flow with the  $k - \varepsilon$  turbulence model, the software employs stabilization techniques (discussed in study solver section) to address issues with the Ladyzhenskaya–Babuška–Brezzi (LBB) condition.

The LBB condition, also known as the inf-sup condition, is a crucial stability criterion for mixed finite element formulations, such as those employed for incompressible flow problems (Ferziger and Perić, 2020). It guarantees that the selected combination of velocity and pressure elements produces a well-posed and stable numerical problem, ensuring a sufficient relationship between the velocity and pressure fields to ovoid false pressure modes or checkerboarding patterns in the solution, standing as a pivotal stability criterion within the realm of mixed finite element formulations, particularly those employed for the numerical solution of incompressible flow problems governed by the Navier-Stokes equations (Wendt, 2008). Essentially, it serves as a test for the compatibility between the chosen finite element spaces for velocity and pressure, ensuring that their interplay

results in a well-posed and numerically stable discrete problem.

In the context of 3D incompressible flows, the LBB condition guarantees a sufficient relationship between the velocity and pressure fields, effectively precluding the emergence of spurious pressure modes or the undesirable "checkerboarding" patterns that can plague numerical solutions. The mathematical formulation of the LBB condition (Chen, 2021), used in the COMSOL Multiphysics documentation COMSOL Multiphysics® (Reference Manual), 2020, is given by:

$$\inf_{q_h \in Q_h} \sup_{v_h \in V_h} \frac{b(v_h, q_h)}{\|v_h\|_V \|q_h\|_Q} \geq \beta > 0$$

Here,  $V_h$  and  $Q_h$  denote the finite element spaces for the velocity and pressure fields, respectively. These spaces are constructed using basis functions defined on the 3D tetrahedral mesh elements that discretize the computational domain.  $\|\cdot\|_V$  and  $\|\cdot\|_Q$  signify the norms defined on the velocity and pressure spaces respectively and provide a measure of the magnitude or "size" of the velocity and pressure fields within their respective finite element spaces.  $\beta$  is a positive constant that remains independent of the mesh size. The existence of such a constant, bounded away from zero, ensures the stability and well-posedness of the discrete problem. Finally,  $b(v_h, q_h)$  represents the bilinear form that encapsulates the pressure-velocity coupling inherent in the weak formulation of the Navier-Stokes equations. For incompressible flows, this bilinear form typically takes the form:

$$b(v_h, q_h) = - \int_{\Omega} (\nabla \cdot v_h) q_h \, d\Omega$$

where  $\Omega$  represents the computational domain,  $\nabla \cdot v_h$  is the divergence of the velocity field, and the integral is taken over the entire domain.

In essence, the LBB condition mandates that for any non-zero pressure field  $q_h$ , there exists a velocity field  $v_h$  such that the bilinear form  $b(v_h, q_h)$  is sufficiently large compared to the product of the norms of  $v_h$  and  $q_h$ . This condition prevents the pressure field from exhibiting erratic oscillations or non-physical behavior, ensuring a numerically stable and convergent solution.

The choice of appropriate finite element spaces for velocity and pressure is crucial for satisfying the LBB condition. In the context of the  $k - \varepsilon$  turbulence model in COMSOL, the default P1+P1 discretization, which employs linear Lagrange elements for both velocity and pressure, is known

to fulfill this condition, rendering it a reliable and widely adopted choice for 3D incompressible flow simulations.

## Higher-Order Elements in 3D

In case we need to enhance accuracy, higher-order finite element spaces can be used(COMSOL Multiphysics® (CFD), 2020), such as quadratic elements. In this case, functions  $u$  within each element are second-order polynomials. To define such a function, additional nodes are introduced at midpoints of the edges of the tetrahedron, increasing the number of degrees of freedom. For example, a quadratic tetrahedral element will have nodes not only at the vertices but also at the midpoints of each edge, resulting in ten nodes per element. The function  $u(x, y, z)$  for quadratic elements is then expressed as:

$$u(x, y, z) = \sum_{i=1}^{10} U_i \phi_i(x, y, z),$$

where the shape functions  $\phi_i$  are quadratic polynomials that are specifically defined to be 1 at their respective node and 0 at all others.

## Conclusion

The P1+P1 discretization is a balanced choice (accuracy vs speed) for stationary k- $\epsilon$  turbulent incompressible flow simulations over complex terrain, such as the Lavrio wind farm study. This selection stems from its inherent simplicity, computational efficiency, and compliance with the stability condition(COMSOL Multiphysics® (Reference Manual), 2020). Linear elements, while not capturing the sharpest gradients, offer a practical balance of accuracy and computational cost, particularly valuable in large-scale simulations involving intricate terrain geometries. In the context of steady-state turbulent flow over complex terrain, where the primary focus lies in capturing the overall flow patterns and turbulence characteristics rather than minute flow details, the P1+P1 discretization, coupled with appropriate mesh refinement in critical regions, provides a reliable and computationally tractable framework for obtaining meaningful insights into the flow dynamics and their implications for wind energy assessment (Michos et al., 2024). Pareto optimization (Catthoor et al., 2020) can also be used to dynamically change the discretization by accounting the performance of different choices for different wind speeds and different terrain

complexities, leading to advanced subdomain discretization refinement.

### 3.4.6 Stationary Study implications

For our simulation modeling framework, we use the stationary study type to obtain steady-state solutions to tackle the computational time constraint problem, deriving in ultra-short-term operational purposes. In a steady-state simulation like in our example, the flow variables are assumed to be independent of time, leading to a simplified set of equations where time derivatives are zero (figure 3.4), resulting in a system of coupled, nonlinear partial differential equations that describe the equilibrium state of the flow. Steady-state simulations are valuable for analyzing scenarios where the flow has reached a stable configuration, such as the long-term behavior of wind flow over terrain or the equilibrium conditions in pipe flow.

In fluid dynamics and CFD, the terms "steady state" and "stationary" are often used interchangeably, but they have distinct meanings:

- **Steady state** refers to a flow where fluid properties (velocity, pressure, temperature, etc.) at any specific point in space remain constant over time. However, these properties can vary spatially across the flow field. This means that while the flow pattern is time-invariant, there may be spatial variations.
- **Stationary**, on the other hand, implies a flow that is not only time-invariant but also spatially uniform. In a strictly stationary flow, the fluid properties are the same at every point in the flow field, indicating no spatial or temporal variations.

In this PhD study we test the performance of CFD simulations under the steady-state hypothesis described in the next chapter and in Michos et al., 2024. In practical applications, truly stationary flows are rare. Most flows exhibit spatial variations even if they reach a steady state. Consequently, in CFD, "steady state" is the preferred term to describe flows that are time-invariant but may have spatial variations. "Stationary" is typically used in a more theoretical context to describe flows with no variations in space or time. Ferziger and Perić, 2002; Kajishima and Taira, 2016 both emphasize steady-state solutions, as these are the most common and relevant in practical CFD scenarios.

### 3.4.7 Isothermal Flow implications

In the context of simulating airflow in the vicinity of a wind farm situated on complex terrain, certain simplifying assumptions can be employed to streamline the computational process without compromising the fidelity of the results. One such assumption is the consideration of isothermal flow, wherein temperature variations within the fluid are deemed negligible.

By adopting an isothermal flow model, the energy equation, which governs the transport and conservation of energy, can be effectively decoupled from the system of equations. This simplification reduces the computational burden significantly, enabling faster and more efficient simulations, particularly when the primary focus lies in capturing the aerodynamic interactions between the wind flow and the terrain features.

In scenarios where temperature gradients are minimal or their impact on the flow dynamics is secondary, such as in wind farm simulations primarily concerned with wind velocity and turbulence characteristics, the isothermal flow assumption is often justified. However, it is crucial to recognize the limitations of this assumption. For instance, if buoyancy effects due to temperature differences play a significant role in the flow behavior, or if heat transfer between the wind and the terrain or wind turbines is of interest, the isothermal assumption might not be appropriate.

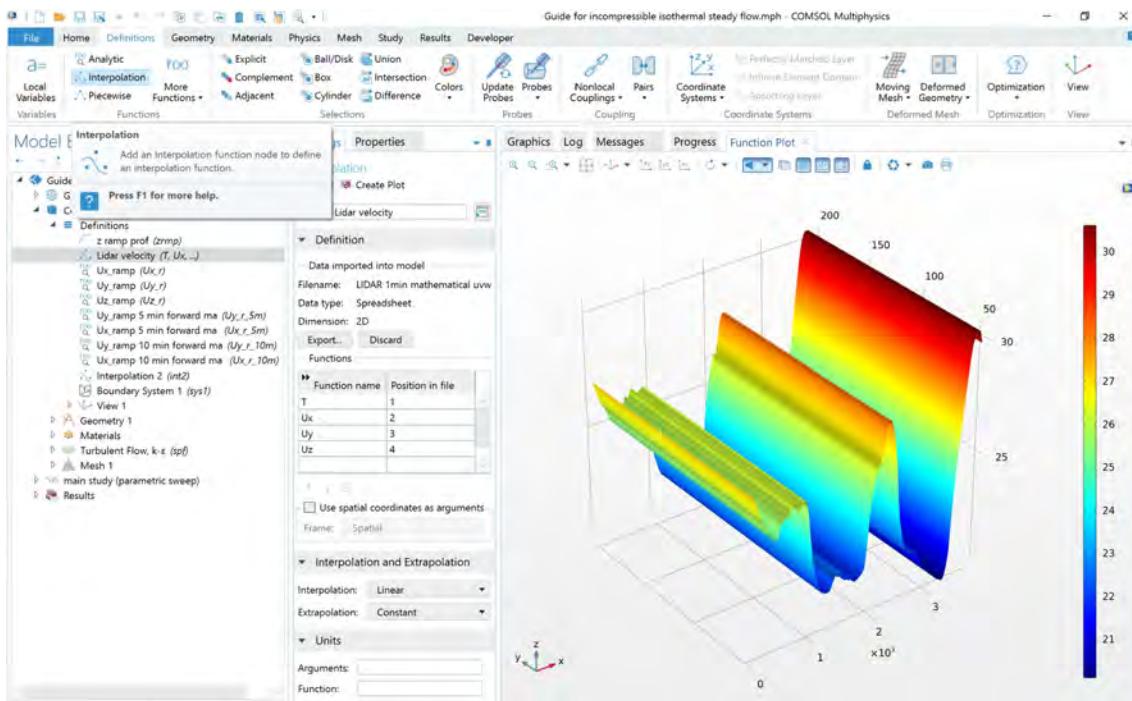


Figure 3.6: Temperature import via interpolation function in COMSOL

In the our case, where the primary concern is the steady-state turbulent

flow over complex terrain near a wind farm, the assumption of isothermal flow is reasonable. The focus is on capturing the intricate flow patterns and turbulence characteristics influenced by the terrain's topography and the presence of wind turbines. To account for the temporal variations in air temperature, scalar average temperature data 'T' is imported through an interpolation function (figure 3.6) in time-series format, with the time variable denoted as 'mint.' This allows the simulation to incorporate the effects of time-varying ambient temperature on the flow, albeit in a simplified manner.

The parameter 'mint' represents the timestamp or midpoint of the averaging interval for the temperature data. By systematically varying 'mint' within a parametric sweep, a series of steady-state solutions can be obtained, each reflecting the flow behavior at a different time step or averaging interval. This approach enables the investigation of how temporal changes in ambient temperature influence the flow patterns and turbulence characteristics in the vicinity of the wind farm.

It is important to acknowledge that the simplification of air temperature to a domain-averaged value might introduce minor inaccuracies in capturing localized thermal effects. However, for the specific focus of this study on the steady-state flow movement over the terrain, such inaccuracies are deemed acceptable.

However, it is crucial to recognize scenarios wherein this isothermal flow assumption may become less tenable. In particular, when the spatial extent of the simulated domain is considerably larger (spanning several kilometers) relative to the characteristic flow velocities and the represented time frame, or when significant portions of the terrain experience prolonged periods of shadowing, substantial surface temperature gradients can develop(Emeis, 2018). These gradients, in turn, can induce intricate interactions between the flow field and the underlying topography, potentially undermining the validity of the isothermal assumption. Mountainous areas, Coastal regions, urban environments and industrial areas are some examples of environments that the implications of this assumption need to be validated in terms of accuracy with high resolution real data in case the results of the simulations are not sufficient.

In conclusion, the application of the isothermal flow assumption, coupled with the incorporation of time-varying average temperature data, offers a practical and efficient approach to simulating turbulent flow over complex terrain in the vicinity of a wind farm. This simplification reduces computational complexity while preserving the essential flow features

relevant to the study's objectives.

## 3.5 Boundaries

The careful specification of boundary conditions, guided by the physical characteristics of domain, the prevailing wind direction, and the specific requirements of the simulation, is essential for obtaining reliable and insightful CFD results. By appropriately defining the wall, inlet, outlet, and open boundaries, researchers and engineers can create a virtual environment that faithfully captures the intricate interactions between the wind, terrain, and wind turbines, thereby contributing to the advancement of wind energy technology in complex terrain settings.

### 3.5.1 Wall Boundary

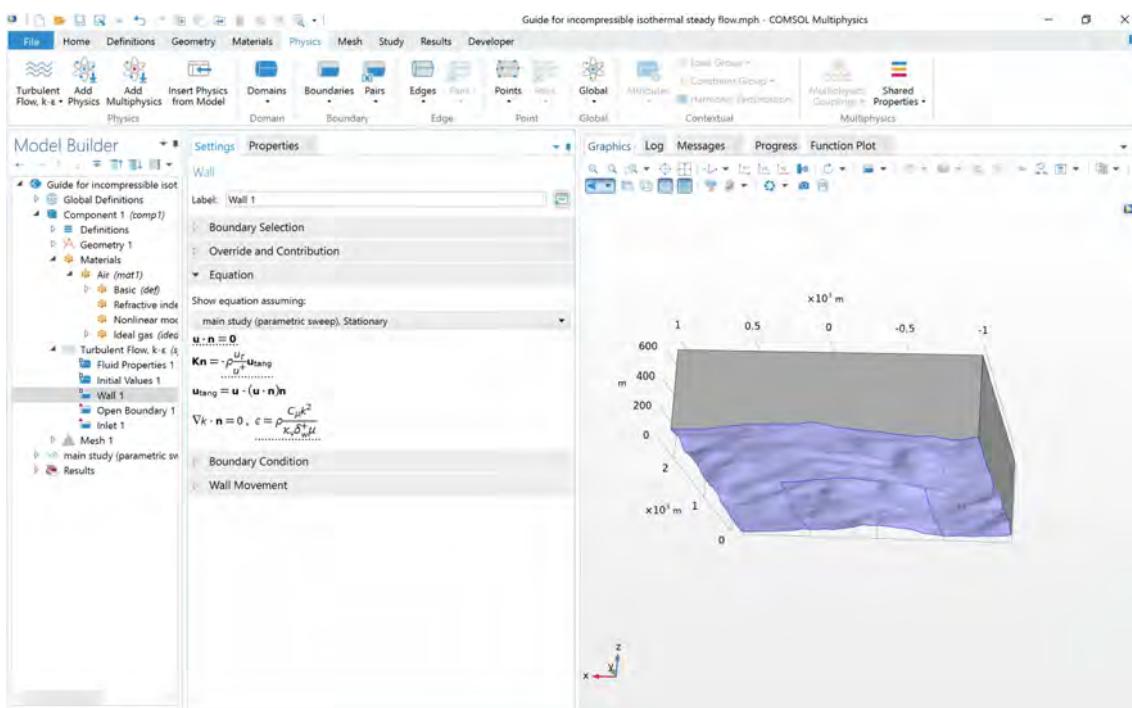


Figure 3.7: Wall boundary and k-epsilon equations in COMSOL

This type of boundary is employed to simulate the interaction of the fluid with solid objects, such as the terrain or wind turbine blades in the current simulation. The solid, no-slip surface of the terrain is represented by a wall boundary (figure 3.7). This boundary condition is implemented in COMSOL by applying it to the bottom boundary of the computational domain, which coincides with the terrain surface extracted from the digital

elevation model (DEM) . Physically, this enforces zero velocity at the terrain surface, simulating the adherence of the fluid to the solid boundary. Within the  $k-\epsilon$  turbulence model, wall functions may be employed to efficiently resolve the near-wall flow behavior. In some cases, a roughness parameter might be specified to account for the terrain's surface texture and its impact on flow resistance.

### 3.5.2 Inlet Boundary

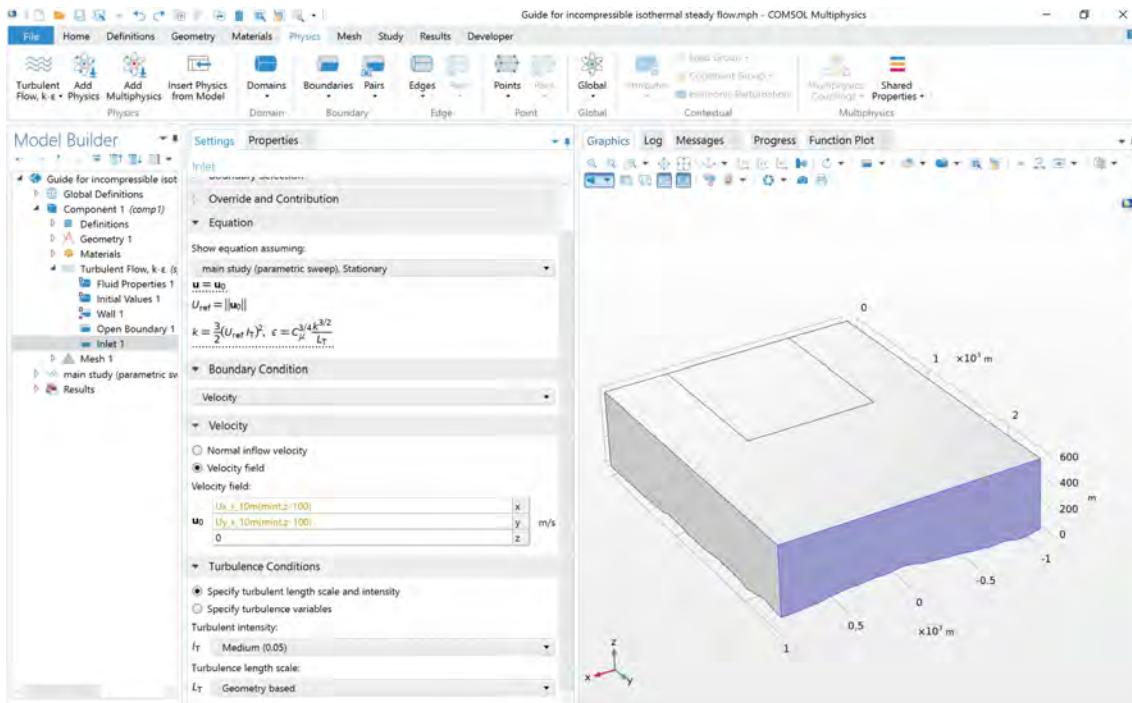


Figure 3.8: Inlet boundary,  $k$ -epsilon equations and variables in COMSOL

The inflow boundary, where the wind enters the computational domain, is defined as the inlet as shown in figure 3.8. This boundary condition is typically applied to the northern boundary of the domain, assuming a prevailing wind direction from the north. The inlet velocity field,  $U_i(\text{m/s})$ , derived from LIDAR measurements and extrapolated using the Wi.Sp.Ex model and power-law with ramp function, is imposed at this boundary. The inlet velocity profile should be carefully designed to ensure a smooth transition from the ground level to the extrapolated values at higher altitudes, mitigating potential convergence issues. The turbulence intensity and length scale at the inlet might also need to be specified, depending on the turbulence model used.

### 3.5.3 Open Boundaries

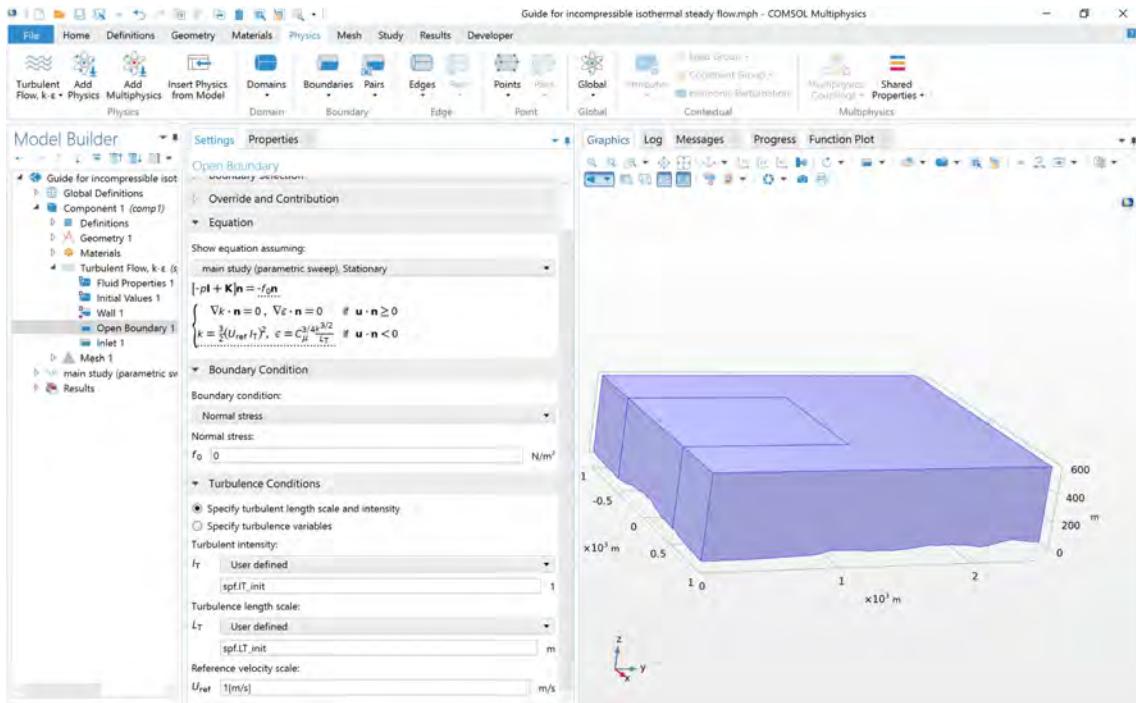


Figure 3.9: Open boundary, k-epsilon equations and variables in COMSOL

Open boundaries are employed to represent the far-field or ambient conditions surrounding the computational domain. They enable both inflow and outflow of fluid. To minimize artificial reflections or disturbances, open boundaries typically incorporate conditions that allow fluid to flow freely in and out. This can be achieved through implementations such as a stress-free condition like in our case. The flow direction at open boundaries is dictated by the pressure or velocity gradients within the flow field. Open boundaries are particularly suitable for simulating unbounded or infinite domains or when the precise conditions at the far-field boundaries remain ill-defined.

In the context of the wind farm simulation, the top, southern, eastern, and western boundaries are designated as open boundaries to accommodate potential irregular flow patterns arising from the complex terrain and the presence of wind turbines (figure 3.9). These boundaries allow for the free inflow and outflow of fluid while minimizing artificial constraints on the flow field. The no-slip condition at these open boundaries ensures that the fluid velocity is zero at the boundary, simulating the presence of a solid wall.

The choice to leave these boundaries open stems from the desire to accommodate irregular flow patterns that might arise due to the complex

terrain. The choice to leave the top and side boundaries open stems from the desire to allow the flow to develop freely without artificial constraints. The substantial distance between the wind turbine location and the western and eastern boundaries, coupled with the predominantly parallel flow direction relative to these boundaries, helps to minimize error propagation and ensure the accuracy of the simulation results in the region of interest.

### 3.5.4 Outlet Boundary

The outlet boundary, should typically be situated on the opposite side of inlet of the domain, representing the outflow where the fluid exits the computational domain. A constant pressure boundary condition, often set to atmospheric pressure, is commonly applied at the outlet, to establish a pressure gradient driving the flow. The flow direction at the outlet is predominantly outward, although some backflow might be permitted depending on specific settings. Ideally, the outlet is positioned sufficiently downstream from the region of interest to minimize its influence on the flow field in that area. Additionally, it should be adequately sized to facilitate the smooth exit of the fluid, preventing artificial flow recirculation or wave reflections.

In our case, however, the absence of measurements at the southern boundary precludes the establishment of a reliable outlet boundary condition. Consequently, the outlet is replaced with an open boundary, allowing the flow direction at the boundaries to be determined by the pressure or velocity gradients within the flow field. This approach leverages the inherent flexibility of open boundaries and the steady-state nature of the simulation to capture the outflow behavior without imposing artificial constraints.

## 3.6 Inlet and initial conditions

In conjunction with the inlet velocity, the initial conditions for both velocity and pressure fields within the computational domain are established. These initial values play a pivotal role in guiding the iterative solver towards a converged steady-state solution. A judicious choice of initial conditions, often informed by simplified approximations or prior knowledge, can significantly expedite the convergence process and enhance the stability of the numerical scheme.

### 3.6.1 Inlet Velocity

The inlet velocity field in Cartesian coordinates, denoted by  $U_i(\text{mint})$  where  $i$  indicates the x,y or z direction, is derived from measurements acquired by a wind LIDAR system stationed at the Lavrio site. The LIDAR measurements were collected at heights of 40, 54, 78, 100, 120, 140, 160, 180, and 200 meters AGL. The values are imported in a single file with  $T$  (figure 3.6) as time-series with constant extrapolation outside the boundaries and linear interpolation.

The potential underestimation of wind speeds and turbulence intensity by the LIDAR is particularly increased in scenarios with strong vertical wind shear or stable atmospheric conditions. These limitations underscore the importance of data validation and uncertainty quantification in CFD modeling.

It is important to note that the LIDAR wind speed measurements at the CRES site have errors ranging from 4% to 6%. These errors, likely arising from factors such as terrain slope, vegetation, and atmospheric stability, need to be considered when interpreting the simulation results and assessing their accuracy.

In many occasions, a WF may not be able to withstand the cost of buying and/or maintaining the equipment needed to collect large amounts of quality data, like in our case study. In the wind LIDAR measurements that were provided by a specific study with temporary use of the equipment by the CRES, measurements close to the ground level are missing. The quality in which the boundary conditions are specified, affect the simulation outcomes.

When modeling turbulent flow over complex terrain, where wind velocity profiles exhibit significant variations near the ground, it is critical to ensure a smooth and physically plausible transition from the ground level to the extrapolated velocity at the lowest measurement height. This transition can be achieved through various methods like the power law or an appropriate application of a ramp function. The model that we are trying to set up so far is able to extrapolate such measurements with sufficient accuracy and is computationally inexpensive, as presented in Michos et al., 2024. This is significant for operational purposes due to its speed and cost reduction.

The modelling approach we will use is based on EAS study that is explained in the WISPEX section. We can now proceed with the demonstration of the simplest way to extrapolate a vertical profile, the use of a COMSOL ramp function  $\mathcal{R}_{amp}$ . In COMSOL Multiphysics version 5.6,

the ramp function we use for inlet velocity extrapolation below the lowest LIDAR measurement height of  $z_{\mathcal{L}}^{\min} = 40$  m is defined as:

$$\mathcal{R}_{amp}(z, G, C, S, S_0, S_1, D_C),$$

where the variables are defined as follows:

- $z$ : The independent variable, typically representing the spatial coordinate or height above ground level in the context of wind profiles.
- $G$ : Represents the 'location' parameter, which specifies the center point of the transition zone where the ramp function smoothly blends the two input values.
- $C$ : Denotes the 'cutoff' parameter, which defines the extent of the transition zone on either side of the location parameter. A larger value of  $C$  leads to a wider transition zone.
- $S$ : Stands for the 'slope' of the transition, determining the steepness or gradient. A higher value of  $S$  results in a sharper transition, while a lower value creates a gentler transition.
- $S_0$ : Refers to the 'size of transition at start', defining the size of the transition zone at the beginning of the ramp.
- $S_1$ : Corresponds to the 'size of transition at end', specifying the size of the transition zone at the end of the ramp function.
- $D_C$ : Represents the 'number of continuous derivatives' parameter, which controls the continuity of the ramp function and its derivatives. A higher value ensures a smoother transition with continuous higher-order derivatives.

The velocity profile is then extrapolated:

$$U_z = \begin{cases} U_{\mathcal{L}} & \text{if } z_{\mathcal{L}}^{\min} \leq z, \\ U_{\mathcal{L}}^{\min} \cdot R_{amp} & \text{if } z_{\mathcal{L}}^{\min} > z, \end{cases}$$

In the context of inlet velocity extrapolation, this ramp function facilitates a transition from the velocity at the ground level ( $z = 0$ ) to the extrapolated velocity  $U_{\mathcal{L}}^{\min}$  at  $z_{\mathcal{L}}^{\min}$  m. For the specific case considered, the ramp function is implemented with parameter values of (-10, 1/30, 1, 0, 50, 2). These benchmark values have been selected to produce a simple

velocity profile near the ground, avoiding potential convergence issues that could arise from abrupt velocity changes.

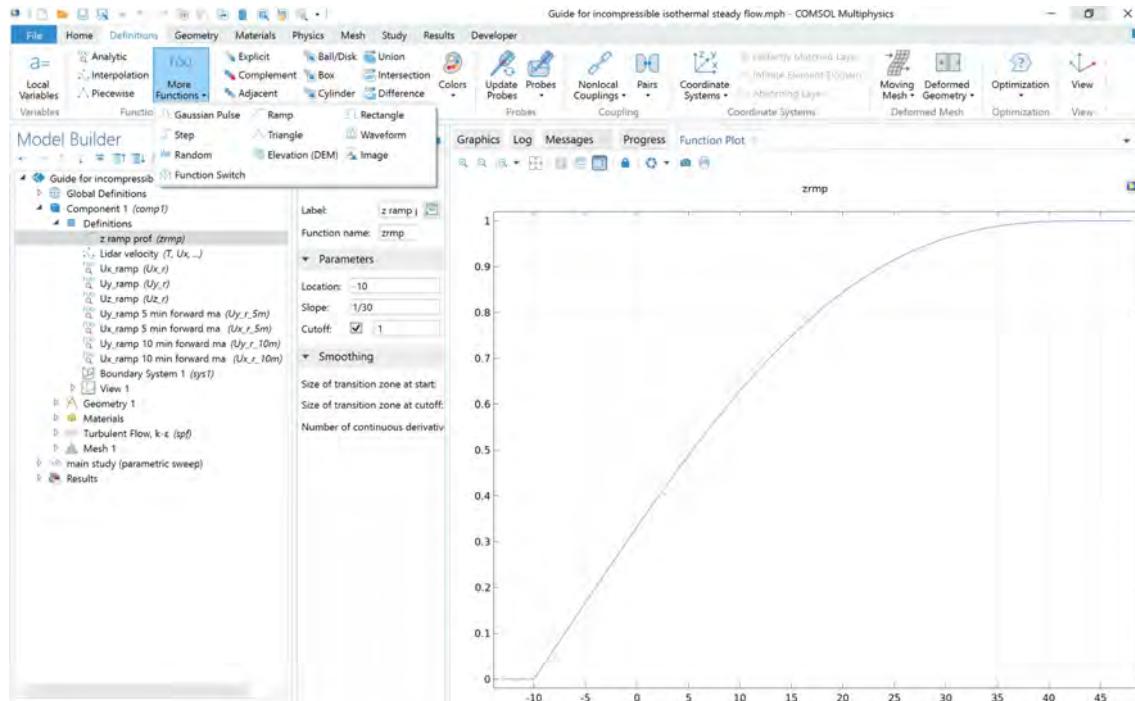


Figure 3.10: Ramp function in COMSOL

The application of the ramp function, combined with accurately specified boundary and initial conditions, is crucial in constructing robust and informative CFD simulations of turbulent flow over complex terrain in transient solution. On the other hand in stationary simulations like the one we model, the uncertainty is high due to lack of sufficient data, while our purpose is not to converge to the solutions of velocity, but its gradients (details in WISPEX). Such simulations allow researchers and engineers to explore the intricate flow patterns, turbulence characteristics, and potential impacts on wind turbine performance, and development of ultra short term forecasting models like WEEL, which is the scope of this dissertation, thereby contributing to the advancement of wind energy technology in challenging environments.

For heights above the highest LIDAR measurement (200 m), the wind velocity is assumed to remain constant. The reason we keep the top boundary of the domain to 600 meters is the elimination of errors near the top boundary. The time-averaged and extrapolated wind velocity data (figure 3.11) allow the dynamic adjustment of the inlet boundary condition based on the 'mint' parameter during parametric sweeps.

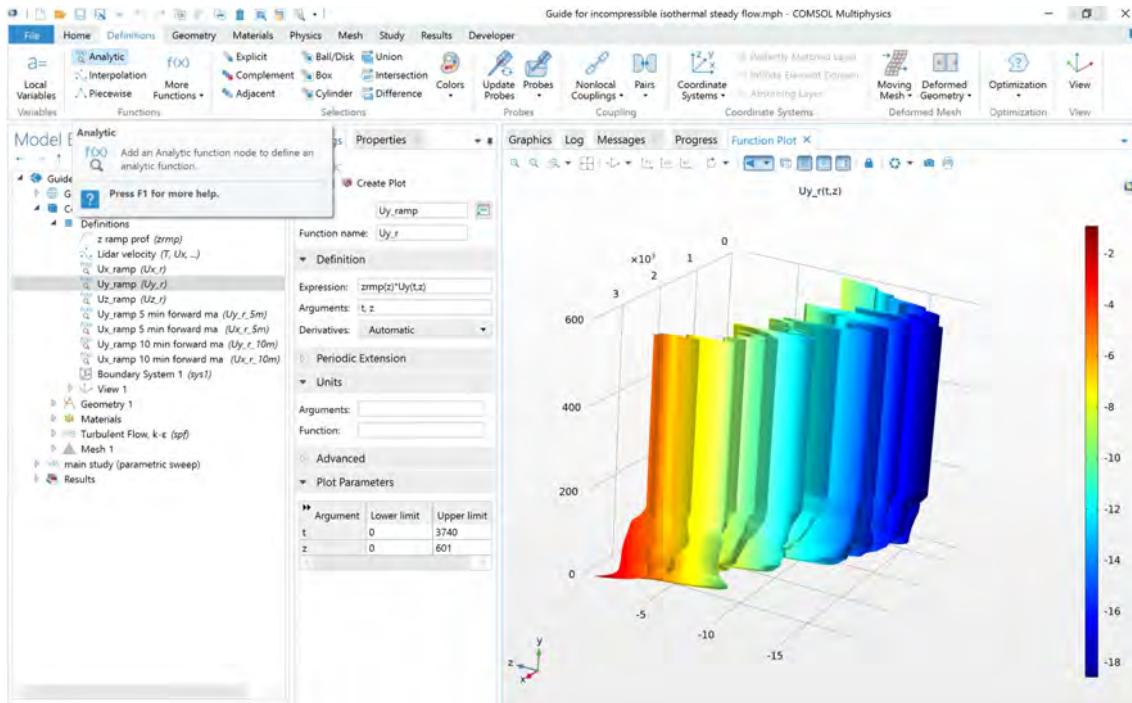


Figure 3.11: Extrapolated y-component wind speed in COMSOL (negative values in y direction indicate a wind direction toward the south – dominant direction)

### 3.6.2 Initial Conditions

In the context of the steady-state turbulent flow simulations presented in Michos et al., 2024, the initial conditions for velocity and pressure fields within the computational domain are dynamically assigned based on the time-averaged LIDAR measurements. Specifically, the initial velocity field is set to the manipulated wind LIDAR velocity data of the previous section,  $U_i(\text{mint})$ , corresponding to the initial value of the time parameter `mint`. This ensures that the simulation starts with a flow field that is consistent with the averaged wind conditions at the LIDAR location of the simulated time period. The initial pressure field is assumed to be uniform over the whole domain.

The use of time-averaged LIDAR measurements to define the initial conditions enhances the realism of the simulation by providing a starting point that is closely aligned with the actual wind conditions at the site. This approach can also contribute to faster convergence of the numerical solver, as the initial guess is closer to the expected steady-state solution.

It's important to note that while the initial conditions influence the early stages of the simulation, their impact diminishes as the solver iterates towards the steady-state solution. However, a carefull choice of initial

conditions can still play a role in improving the overall efficiency and stability of the simulation.

### 3.6.3 Conclusion

The accurate specification of inlet velocity and initial conditions, informed by real-world measurements, forms the foundation of a realistic and informative CFD simulation of turbulent flow in the vicinity of a wind farm situated on complex terrain. By incorporating these essential boundary and initial values, researchers and engineers can gain valuable insights into the intricate flow patterns, turbulence characteristics, and potential impacts on wind turbine performance, ultimately contributing to the advancement of wind energy technology in challenging environments (Emeis, 2018; Wendt, 2008).

The dynamic specification of initial conditions based on time-averaged LIDAR measurements, coupled with an accurate representation of inlet velocity, forms a crucial aspect of the CFD modeling framework employed in the CRES study as presented by Michos et al., 2024. This approach enables the simulation to capture the dynamic interactions between the wind and the terrain, despite the scarcity of available measurements, providing valuable insights into the flow dynamics and their implications for wind energy harvesting in complex terrain environments.

## 3.7 Parametric sweeps

Investigating the temporal evolution of turbulent flow over complex terrains for operational purposes via CFD necessitates innovative approaches. COMSOL Multiphysics, with its robust solver framework and parametric study capabilities, offers a powerful solution through the strategic implementation of parametric sweeps within a 'Stationary' study.

The core principle lies in leveraging a parameter representing the timestamp of the averaging interval for the flow data, in our case, it is named '*mint*'. By systematically varying '*mint*' within a predefined range, one can obtain multiple steady-state solutions, each corresponding to a distinct time-step or averaging interval. While each solution inherently captures a static flow field, the collective ensemble of solutions from the sweep reveals the dynamic evolution of the turbulent flow over time.

To implement this approach in COMSOL, the parameter '*mint*' is first

defined and assigned an initial value. Subsequently, the inlet conditions (velocity and temperature) and initial conditions (velocity and temperature fields) are expressed as functions of 'mint,' enabling their dynamic adjustment during the sweep. A parametric sweep is then configured within the 'Stationary' study, specifying the range and increment for 'mint.'

Upon execution, COMSOL solves the steady-state turbulent flow problem for each 'mint' value, generating a series of solutions that depict the flow's behavior at different time-steps. Analyzing these solutions unveils the dynamic nature of the flow through the evolving inlet conditions, solution states and the resulting flow patterns, pressure distribution, and turbulence characteristics.

It is imperative to recognize that this approach rests on the assumption of steady-state flow within each averaging interval. Consequently, if the flow exhibits significant transient behavior within these intervals, the accuracy of the results might be compromised. Additionally, ensuring convergence for each solution within the parametric sweep is vital, necessitating careful monitoring and potential adjustments to solver settings or mesh refinement.

Despite its inherent steady-state nature, the parametric sweep approach offers an efficient means of exploring the temporal dynamics of turbulent flow over complex terrain. It enables the investigation of how varying inlet conditions and initial states influence the flow's evolution, enhancing understanding of intricate flow phenomena.

In conclusion, parametric sweeps in COMSOL, coupled with appropriate data handling through the integrated functions and an awareness of inherent assumptions, provide a valuable tool for researchers and engineers to study the complexities of time-dependent turbulent flow behavior over intricate terrains in a simple, fast and consistent way. They empower the extraction of meaningful insights from a series of steady-state solutions, bridging the gap between "static snapshots" and the dynamic nature of fluid flow.

## 3.8 Study Solver

COMSOL Multiphysics provides a versatile solver framework for addressing diverse CFD challenges. In the context of turbulent flow simulations using the  $k-\epsilon$  model, two primary solver approaches are available: the Fully Coupled solver and the Segregated solver.

The Fully Coupled solver treats the governing equations—continuity, momentum, turbulent kinetic energy ( $k$ ), and its dissipation rate ( $\epsilon$ )—as a unified system, solving them simultaneously. This approach ensures strong coupling between the variables and often employs the damped Newton method, an iterative technique known for its robustness and quadratic convergence rate. The damped Newton method iteratively solves the system of nonlinear equations represented by:

$$J(U^n)\Delta U = -R(U^n)$$

where  $J(U^n)$  is the Jacobian matrix,  $\Delta U$  is the solution update, and  $R(U^n)$  is the residual vector, all evaluated at the current solution  $U^n$ .

In contrast, the Segregated solver decouples the governing equations, solving them sequentially and iterating between them until convergence is achieved. While generally more memory-efficient, the Segregated solver might require stabilization techniques for strongly nonlinear problems. For  $k$ - $\epsilon$  turbulent stationary studies, COMSOL employs pseudo-time stepping by default within the Segregated solver.

Pseudo-time stepping introduces an artificial time-like variable into the steady-state equations we use in our model (figure 3.12), transforming the problem into a pseudo-transient one. This facilitates a controlled march towards the steady-state solution, particularly beneficial for turbulent flows where nonlinearities can hinder convergence. The modified governing equations incorporate a pseudo-time derivative term:

$$\frac{\partial U}{\partial t^*} + R(U) = 0$$

where  $t^*$  represents pseudo-time,  $U$  is the vector of unknowns (velocity, pressure,  $k$ ,  $\epsilon$ ), and  $R(U)$  is the residual vector. Discretization in pseudo-time, often using the backward Euler scheme, leads to an iterative solution update process.

While pseudo-time stepping is the default for the Segregated solver in turbulent flow simulations, COMSOL's flexible framework allows users to customize solver settings. This includes the ability to disable pseudo-time stepping or adjust its parameters, such as the pseudo-time step size, to influence convergence behavior and computational efficiency.

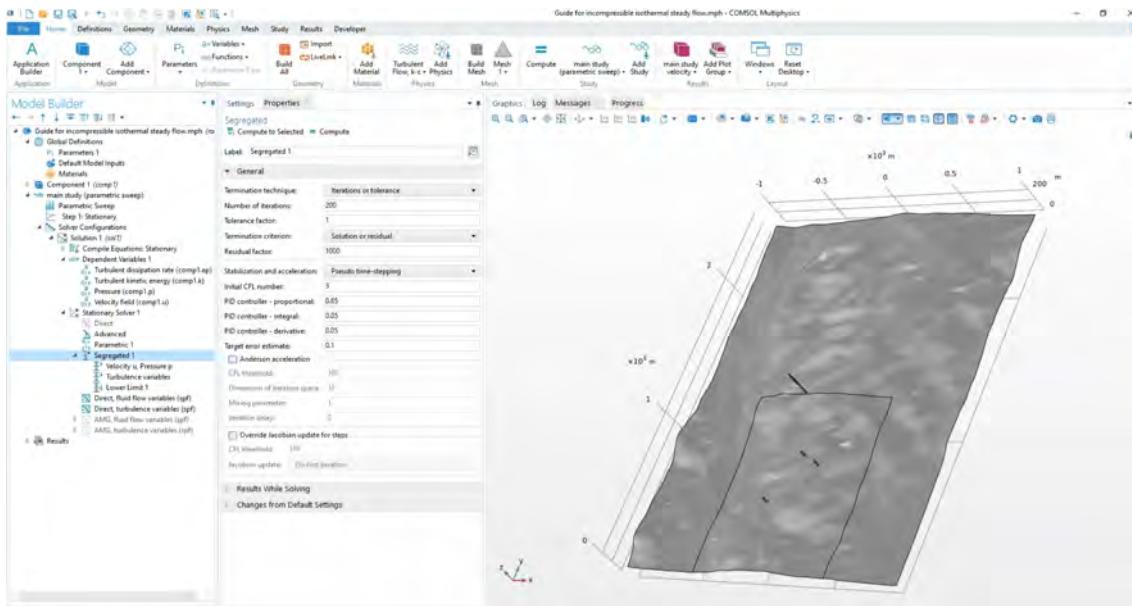


Figure 3.12: Segregated solver configuration in COMSOL

Successful CFD simulations require not only an appropriate solver but also a clear understanding of convergence criteria and termination mechanisms. Convergence signifies that the numerical solution has stabilized, with further iterations yielding negligible changes within a specified tolerance, indicating that the solver has successfully approximated the true solution to the governing equations within a specified tolerance. COMSOL offers termination criteria based on residuals, variable changes, or a maximum number of iterations.

- **Residuals:** Residuals measure the imbalance in the governing equations. When the residuals fall below a predefined tolerance, the solver considers the solution converged.
- **Variable Change:** This criterion monitors the change in the solution variables between successive iterations. If the changes are sufficiently small, convergence is declared.
- **Number of Iterations:** You can also specify a maximum number of iterations. The solver will terminate even if other convergence criteria are not met. This is useful in cases where fast testing simulation must be done to understand the proper set up for termination set ups or to terminate a solution when the other selection measures converge to a fluctuating state in the limits of convergence criteria completion after a known number of iterations

Monitoring the convergence process by tracking these metrics is crucial for assessing solution quality and identifying potential issues.

Understanding solver configurations, pseudo-time stepping, convergence criteria, and termination mechanisms is essential for effective CFD simulations in COMSOL. By leveraging the software's flexibility and control, users can tailor simulations to specific needs, ensuring both accuracy and efficiency in capturing the complexities of turbulent flows. In our case, where we test the limits of CFD to obtain fast simulations, this is crucial to obtain results with sufficient accuracy.

When dealing with stationary incompressible flow problems, particularly those involving the Reynolds-Averaged Navier-Stokes (RANS) equations with the  $k - \varepsilon$  turbulence model, the Ladyzhenskaya–Babuška–Brezzi (LBB) condition is a significant concern. This condition pertains to the stability and accuracy of the numerical solution, especially when using equal-order interpolation for velocity and pressure (P1+P1 elements). COMSOL addresses this issue using a segregated solver with the PARDISO direct solver and a linear direct solver approach.

The segregated solver approach handles equations separately, as demonstrated in figures 3.13. This reduces the coupling complexity and allows each set of equations to be handled more robustly and efficiently. By breaking down the problem into smaller subsystems, this method reduces the complexity and size of each solve step, allowing COMSOL to handle the lack of LBB stability more effectively. The segregated solver alternates between solving for velocity and pressure (fluid flow variables) and solving for turbulence variables ( $k$  and  $\varepsilon$ ).

Linear Direct Solver with PARDISO is used for the fluid flow variables (velocity and pressure), to handle the linear systems resulting from the discretization of the Navier-Stokes equations. Each segregated step for velocity and pressure uses a nonlinear solver based on the Constant Newton method. This involves solving the linearized system at each Newton iteration, using the direct solver to ensure robust handling of the system despite potential LBB issues.

For the turbulence variables, the segregated solver uses PARDISO again, tailored to the specific structure of the  $k-\varepsilon$  equations. Similar to the fluid flow step, the turbulence variables are solved using a nonlinear solver with the Constant Newton method, with each Newton step requiring a linear solve by PARDISO.

The nonlinear solver refines the solution iteratively, ensuring that the turbulence variables are accurately resolved in relation to the evolving velocity and pressure fields.

The segregated solver iterates between the velocity-pressure and turbu-

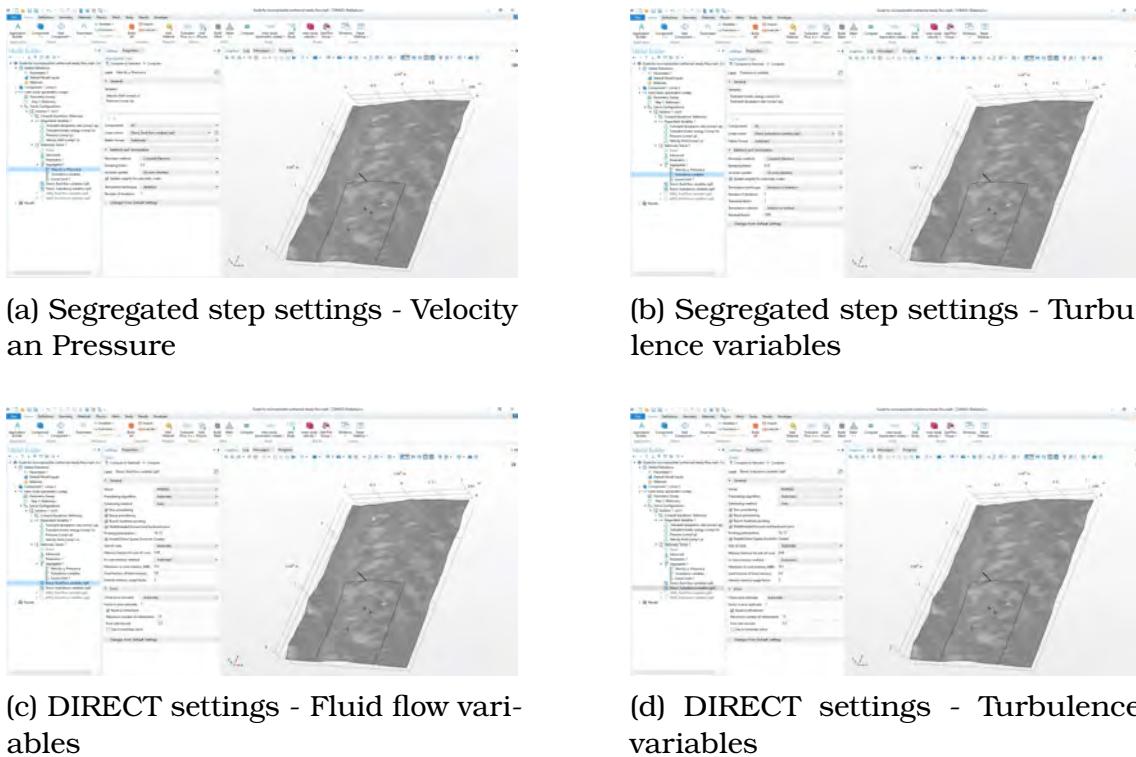


Figure 3.13: Segregated Solver configurations in comsol

lence variable steps. After each set of solves, the segregated solver checks for convergence based on the residuals of both subsystems. This iterative approach allows the solver to stabilize the coupled solution incrementally, reducing the impacts of the LBB condition violation by ensuring each variable set is handled with stabilization and direct solution methods.

The segregated approach inherently allows more flexible stabilization adjustments, targeting each subsystem separately rather than relying on a fully coupled, potentially unstable approach. By solving each subsystem with PARDISO and refining iteratively using the Constant Newton method, COMSOL effectively bypasses the difficulties of the LBB condition, leveraging the robustness of the direct solver to stabilize the pressure field within the segregated framework.

Outside of the segregated solver structure, COMSOL configures the PARDISO direct solver specifically for both fluid flow variables and turbulence variables. This ensures consistent handling of all variables with high numerical stability. The PARDISO solver's ability to manage large, sparse systems directly helps address the potential instabilities associated with the P1+P1 elements, providing a reliable solution pathway.

## 3.9 Conclusion

This chapter presented a comprehensive guide on modeling an incompressible turbulent steady airflow over realistic terrain using COMSOL Multiphysics. We covered the entire modeling process, including domain creation, meshing, physics settings, boundary conditions, and solver configurations. By following these steps, you should be able to set up and run similar simulations effectively. The Physics and Study Settings section lays the groundwork for the CFD simulation in COMSOL. By carefully selecting the appropriate physics interface, study type, turbulence model, discretization, and other settings, we can tailor the simulation to accurately capture the desired flow behavior while optimizing computational efficiency.

It is worth mentioning that this demonstration highlights the significance of understanding the physics behind a CFD modeling process. The choice of a stationary  $k-\varepsilon$  turbulence model for simulating incompressible turbulent flow over complex terrain, as exemplified in the Lavrio wind farm study, is well-founded and justified by several factors.

Firstly, the  $k-\varepsilon$  model, being a Reynolds-Averaged Navier-Stokes (RANS) approach, offers a computationally efficient solution for capturing the time-averaged flow characteristics in turbulent regimes. By solving for mean flow variables and modeling the effects of turbulence through transport equations for turbulent kinetic energy ( $k$ ) and its dissipation rate ( $\varepsilon$ ), the  $k-\varepsilon$  model bypasses the need to resolve the full spectrum of turbulent fluctuations, thereby significantly reducing computational demands. This efficiency is particularly crucial in simulations involving complex terrain, where the intricate topography can lead to intricate flow patterns and increased computational complexity.

Secondly, the assumption of steady-state flow, coupled with the time-averaged nature of the LIDAR-derived wind data, aligns well with the  $k-\varepsilon$  model's formulation. The model's focus on predicting mean flow behavior rather than instantaneous fluctuations renders it suitable for scenarios where the primary interest lies in understanding the overall wind flow patterns and their impact on wind turbine performance, rather than capturing transient flow phenomena.

Furthermore, the  $k-\varepsilon$  model's inherent ability to handle complex geometries and its widespread validation across a broad range of flow scenarios lend credence to its applicability in the Lavrio case study. The model's robustness and relative ease of implementation make it a practical choice for simulating flow over intricate terrain, where flow separation, recirculation,

and turbulence generation are prevalent.

While the  $k-\epsilon$  model does have limitations, such as its potential inaccuracies in predicting flows with strong streamline curvature or adverse pressure gradients, its strengths in computational efficiency, applicability to complex geometries, and focus on mean flow behavior coupled with isothermal assumption, make it a suitable and sufficient choice for the present study. The integration of time-averaged LIDAR data further enhances the model's ability to capture the temporal variability of the wind field, offering valuable insights into the flow dynamics in the vicinity of the Lavrio wind farm.

While the  $k$ -epsilon model is widely used, it has limitations, particularly in flows with strong streamline curvature, separation, or swirling. COMSOL offers other turbulence models (e.g.,  $k$ -omega, Spalart-Allmaras) that might be more suitable for specific flow regimes. Always refer to the COMSOL documentation (COMSOL Multiphysics® (CFD), 2020 , COMSOL Multiphysics® (CYCLOPEDIA), 2017 and COMSOL Multiphysics® (Reference Manual), 2020) for detailed information on any topic regarding the available interfaces, like the 'Turbulent  $k$ -epsilon' interface and specific features and settings used in your simulation.

---

## References

- Bingöl, F., Mann, J., & Foussekis, D. (2011). 2.3 lidar in complex terrain. *Advancements in Wind Energy Metrology—UPWIND 1A2*. 3, 11. <https://orbit.dtu.dk/en/publications/advancements-in-wind-energy-metrology-upwind-1a23>
- Catthoor, F., Basten, T., Zompakis, N., Geilen, M., & Kjeldsberg, P. G. (2020). *System-scenario-based design principles and applications* (Vol. 1). Springer. <https://doi.org/10.1007/978-3-030-20343-6>
- Chen, L. (2021). Inf-sup conditions for operator equations. *Lecture Notes*, <https://www.math.uci.edu/~chenlong/226/infSupOperator.pdf>.
- COMSOL Multiphysics® (CFD). (2020). *COMSOL Multiphysics® CFD module user's guide, version 5.6*. COMSOL. <https://doc.comsol.com/5.6/doc/com.comsol.help.cfd/CFDModuleUsersGuide.pdf>
- COMSOL Multiphysics® (CYCLOPEDIA). (2017). COMSOL Multiphysics® CYCLOPEDIA [[Online]. Last accessed: 24 August 2024]. <https://www.comsol.com/multiphysics/finite-element-method>
- COMSOL Multiphysics® (Reference Manual). (2020). *COMSOL Multiphysics® Reference Manual, version 5.6*. COMSOL. [https://doc.comsol.com/5.5/doc/com.comsol.help.comsol/COMSOL\\_ReferenceManual.pdf](https://doc.comsol.com/5.5/doc/com.comsol.help.comsol/COMSOL_ReferenceManual.pdf)
- Dick, E. (2009). Introduction to finite element methods in computational fluid dynamics. In J. F. Wendt (Ed.), *Computational fluid dynamics* (pp. 235–274). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-85056-4\\_10](https://doi.org/10.1007/978-3-540-85056-4_10)
- Emeis, S. (2018). *Wind energy meteorology: Atmospheric physics for wind power generation*. Springer. <https://doi.org/10.1007/978-3-642-30523-8>
- Ferziger, J. H., & Perić, M. (2002). *Computational methods for fluid dynamics*. Springer. <https://doi.org/10.1007/978-3-642-56026-2>
- Ferziger, J. H., & Perić, M. (2020). *Computational methods for fluid dynamics* (4th). Springer. <https://doi.org/10.1007/978-3-319-99693-6>
- Kajishima, T., & Taira, K. (2016). *Computational fluid dynamics: Incompressible turbulent flows*. Springer. <https://doi.org/10.1007/978-3-319-45304-0>
- Michos, D., Catthoor, F., Foussekis, D., & Kazantzidis, A. (2024). A cfd model for spatial extrapolation of wind field over complex terrain—wi. sp. ex. *Energies*, 17(16), 4139. <https://doi.org/10.3390/en17164139>

- 
- Wendt, J. (2008). *Computational fluid dynamics: An introduction*. Springer Berlin Heidelberg. <https://books.google.gr/books?id=IIUkqI-HNbQC>
- Wilcox, D. C. (2006). *Turbulence modeling for cfd* (3rd) [Includes bibliography, index and Compact Disk.]. DCW Industries, Inc. <http://www.dcwindustries.com>

# **Chapter 4**

## **Ultra Short Term Wind Power Forecasting**

Wind energy forecasting is crucial for optimizing energy management by improving the use and storage of diverse energy sources and enabling rapid responses to extreme conditions (Quiñones et al., 2023). In energy sectors characterized by dynamic demand and storage capabilities, predicting energy production over short time horizons is essential for the effective implementation of dynamic pricing systems based on smart grids, similar to those used for longer time frames (Manfren et al., 2011; Nallolla et al., 2023; Polimeni et al., 2021). This growing demand for ultra-short-term wind energy forecasting is driven by the need to minimize power losses, stabilize electricity prices, and achieve cost savings through improved energy storage and management.

Accurate and reliable wind energy forecasting plays a pivotal role in integrating wind power into the grid and ensuring stability. This is especially critical in ultra-short-term forecasts, which focus on predictions within a 30-minute window, as they help to optimize energy management strategies by leveraging the benefits of precise forecasts. Failures in this area can have widespread repercussions, including disruptions to industrial operations, financial losses, and impaired communication systems, which can ultimately hinder economic growth and job creation in affected communities. In critical settings, such as hospitals, power outages pose even greater risks, disrupting life-sustaining equipment and emergency services.

Predicting the Energy Yield (E-yield) of a WT is particularly challenging due to the chaotic nature of wind movement over short timescales and in complex terrains. Wind forecasting models are generally categorized into

---

statistical models and physics-based models. Statistical models, including Artificial Neural Network (ANN), are valued for their fast and accurate predictions suitable for operational use (Dai et al., 2023; Wei et al., 2023). These models can directly predict the E-yield or wind speed, as seen in B. Yang et al., 2021. However, they require extensive datasets for training and must be tailored to specific locations, making them less adaptable to changing conditions. Established WFs with minimal infrastructural changes can provide the necessary data, but new WFs need to start measurement campaigns well before the installation of the first turbine. Furthermore, these models need to be retrained when site changes occur, such as new turbines or nearby structures, which is time-consuming and can disrupt operations.

In contrast, physics-based models, like CFD models like the ones presented by Sezer-Uzol and Long, 2006; Wu and Porté-Agel, 2011; B. Yan and Li, 2016; S. Yan et al., 2018 and Valldecabres et al., 2018, offer detailed estimations of wind fields and are less affected by environmental changes compared to statistical models. Wind flows are highly sensitive to terrain features and obstacles, particularly near buildings and urban areas (Xiaoxia et al., 2022; Zhou et al., 2021). Even minor terrain modifications, such as constructing a small shed near a WT, can significantly impact the wind field (Michos et al., 2024a). The complex interactions between three-dimensional wind flows and turbine blades mean that E-yield can be sensitive to terrain changes, especially with elevation fluctuations. However, the high computational demands of CFD models, which often involve long computation times for time-dependent simulations, limit their application for ultra-short-term forecasting where rapid predictions are needed.

This PhD research introduces an innovative physics-based model designed for ultra-short-term WT power forecasting. The presented model can be separated in two parts, the first which is a wind spatial extrapolation and forecasting model that leverages CFD principles with stationary equations to address the unique challenges posed by complex terrains, and the second part which is the power forecasting model that takes advantage of a novel, yet simple method to detect the power production latency of a given WT, accounting for persistence in wind conditions.

By spatially extrapolating limited wind LIDAR measurements, the CFD model named WiSpEx, is able to generate high-resolution input datasets that enhance the accuracy of ultra-short-term forecasts (Baile and Muzy, 2023), even in complex environments. Recognizing the impact of complex

terrain on LIDAR measurement accuracy Bingöl et al., 2011; Kim et al., 2016; Kogaki et al., 2020, this research also examines the model's forecasting performance under various conditions through simulations conducted over the CRES site.

The final part of our modeling approach involves WEEL that handles the conversion of simulated wind speeds into power through a continuous power curve approximation, introducing a novel method for detecting power extraction latency in WTs. This approach can empower small, low-cost WFs to optimize their existing setups while enabling further research through the use of physics-based synthetic datasets.

By prioritizing the physics of wind behavior, this approach reduces dependency on terrain-specific adjustments, allowing the model to retain the accuracy of traditional physics-based forecasts at a lower computational cost. This enables fast and reliable predictions, promoting broader operational use of CFD in wind forecasting and encouraging the development of hybrid models that combine the strengths of statistical and physics-based approaches. Such hybrid models could potentially overcome the limitations of each individual method, delivering more accurate and reliable ultra-short-term wind energy forecasts and fostering further innovation in the field.

## 4.1 Experimental setup

In this study, an attempt to test the limits of CFD simulations to recreate a given wind field over complex a terrain from scarce data, in a sufficiently fast and accurate way, was made. This research yielded the WiSpEx. The CFD modeling approach demonstrated in the previous chapter is the core of WiSpEx which is incorporated in WEEL to create ultra short term Power forecasts, which is the final product of this PhD. The model we present is created as a benchmark for similar future attempts. Being a physics based model is advantageous because it's performance is analogous to the data quality, the modeling approach and the hardware used, while history shows that a strong positive trend is driving all of them to an interesting future.

In many occasions, a WF may not be able to withstand the cost of buying and/or maintaining the equipment needed to collect large amounts of quality data, like in our case study. The CRES site, situated in the complex terrain of Lavrio, Greece, presents a unique challenge for wind

resource assessment, as already discussed in the previous chapter. Wind LIDAR measurements were provided by a specific wind LIDAR campaign conducted at CRES between October 22nd and 24th, 2010, with temporary use of the equipment. Wind measurements were obtained at nine different heights (40m, 54m, 78m, 100m, 120m, 140m, 160m, 180m, and 200m).

The potential underestimation of wind speeds and turbulence intensity by the LIDAR is affected by vertical wind shear and atmospheric stability. These limitations underscore the importance of data validation and uncertainty quantification in CFD modeling. It is important to note that the LIDAR wind speed measurements at the CRES site have errors ranging from 4% to 6%, as stated by Bingöl et al., 2011. These errors, likely arising from factors such as terrain slope, vegetation, and atmospheric stability, need to be considered when interpreting the simulation results and assessing their accuracy.

The wind LIDAR measurements, while valuable, lack data close to the ground level. This gap in data, coupled with the inherent errors in LIDAR measurements, underscores the challenges in accurately specifying boundary conditions for CFD simulations. The quality of these boundary conditions directly impacts the reliability and accuracy of simulation outcomes (Michos et al., 2024a).

To address these challenges, the CFD benchmark version of WiSpEx demonstrated, is designed to spatially extrapolate wind fields in under 3 minutes, even with limited wind and temperature data. The benchmark model achieves Mean Absolute Percentage Error (MAPE) below 10% for the estimations of wind speed at hub height of the WTs installed in CRES, making it a valuable tool for generating high-resolution wind datasets when critical information is missing.

During the LIDAR campaign, horizontal wind speed measurements taken at hub height from WTs seen in 4.1 were available. These WTs employ anemometers (either cup or ultrasonic) mounted directly on their hubs to measure wind speed. However, the WT's presence influences the airflow, necessitating online corrections to the measured wind speed data. Manufacturers achieve this by utilizing pre-existing calibration data gathered from a separate mast, positioned 2-3 rotor diameters away at the WT's hub height. Power output of two WTs (Enercon and NEG) located at the WF that were available at the time window we used from the LIDAR campaign took place, were provided by CRES. Figure 4.2 illustrates the placement of the LIDAR, WTs, and the surrounding terrain.

	<b>E40-500</b>	<b>V47-660 kW</b>	<b>NM 750/48</b>
<b>Rotor</b>			
<b>Diameter:</b>	40 m	47 m	48.2 m
<b>Area swept:</b>	1275 m <sup>2</sup>	1735 m <sup>2</sup>	1824 m <sup>2</sup>
<b>Speed of revolution:</b>	variable, 18 - 38 rpm	28.5 rpm	22/14 rpm
<b>Number of blades:</b>	3	3	3
<b>Length of blades:</b>	19 m		
<b>Blade material:</b>	fiberglass (reinforced epoxy) with integral lightning protection		
<b>Tip speed:</b>	38 - 80 m/s		
<b>Type:</b>	upwind rotor with active pitch control	Upwind rotor	Upwind Rotor
<b>Tower</b>			
<b>Type:</b>			Conical, steel, painted
<b>Hub height(approx.)</b>	44m	45m	45m
<b>Generator</b>			
<b>Type:</b>	direct-driven ENERCON ring generator (with drive train)	Asynchronous with OptiSlip®	Asynchronous, 4/6 pole
<b>Name plate rating:</b>	500 kW	660 kW	750/200 kW
<b>Nominal Voltage:</b>		690 V	690 V
<b>Nominal frequency:</b>	50 Hz	50 Hz	50 Hz
<b>Hub:</b>	Rigid		
<b>Cooling:</b>			Liquid-cooled with pump
<b>Operational data</b>			
<b>Nominal output:</b>	500 kW	660 kW	750 kW
<b>Cut-in wind speed:</b>	2.5 m/s	4 m/s	4 m/s
<b>Cut-out wind speed:</b>		25 m/s	25 m/s
<b>Nominal wind speed:</b>	12 m/s	15 m/s	16 m/s
<b>Power regulation:</b>	Variable Speed + Pitch	Pitch/OptiSlip®	Stall

Figure 4.1: WT information from CRES, 2024.

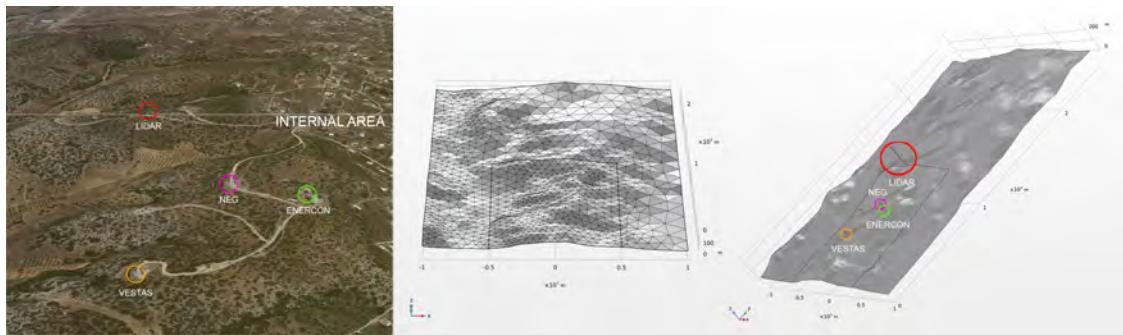


Figure 4.2: CRES satelite view Google Maps, 2024 with LIDAR and WT locations (**left**), meshed geometry (center), and geometry with LIDAR and WT locations (**right**), Michos et al., 2024a

The 3D representation of the CRES site and its vicinity, described by a domain volume of approximately  $2 \text{ km} \times 2.5 \text{ km} \times 0.6 \text{ km}$  is created within COMSOL as described in the previous chapter. We have created a subdomain enclosed by the initial, encompassing the locations of the WTs and the LIDAR, which is positioned at the inlet boundary plane of WiSpEx simulations conducted for the inner area. The terrain within the domain is complex, with an average elevation of 66m and a standard deviation of 33m. The maximum elevation reaches 136m, while the minimum is near sea level as stated by Michos et al., 2024a.

## 4.2 Impact of Obstacles and Simplifications on Wind Flow

### 4.2.1 Obstacles

In our study the WTs are not inserted in the simulations to decrease the computational time of the simulations. We conducted Simulations to evaluate the effects of different obstacles and simplifications on wind flow. For instance, a non-rotating WT of similar size to those in our case study can produce a wake extending up to 270 m. We can observe the differences in wind speed magnitude before and after the introduction of a non-rotating WT in Figure 4.3.

## 4.2. IMPACT OF OBSTACLES AND SIMPLIFICATIONS ON WIND FLOW

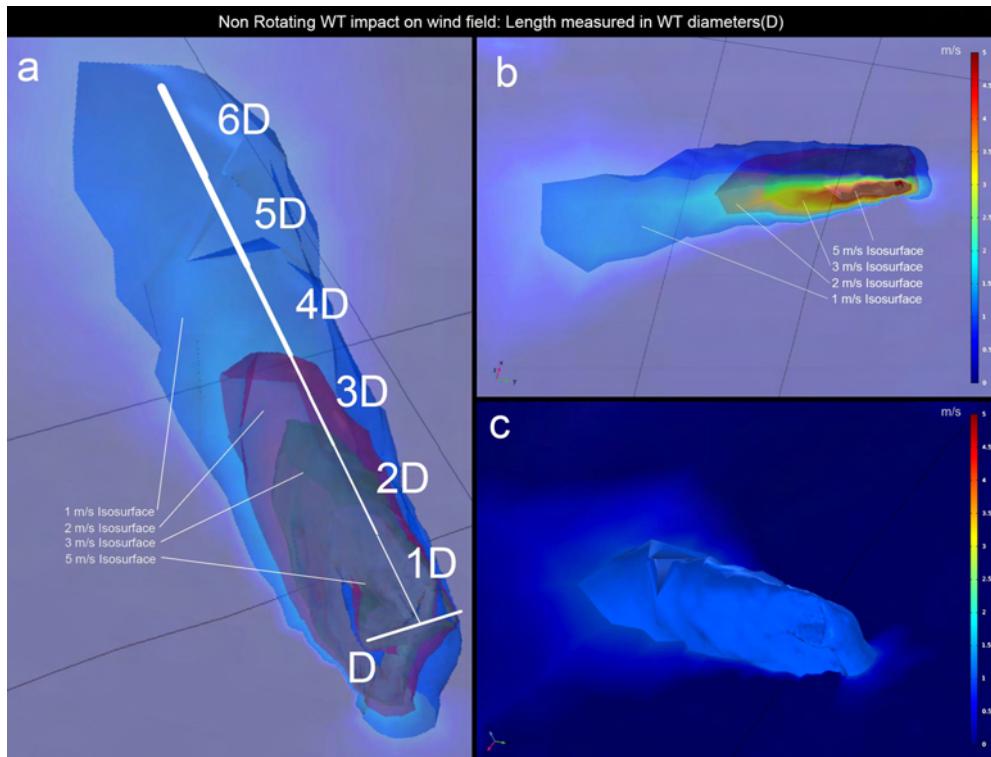


Figure 4.3: Effect of a non-rotating WT on wind speed magnitude (m/s). Iso-surfaces represent the wind speed difference before and after installing a non-rotating WT. **(a)** Length of iso-surfaces (in WT diameters) showing varying wind speed differences (with transparency), **(b)** view of these iso-surfaces from below (with transparency), **(c)** 1 m/s wind speed magnitude difference iso-surface (opaque), Michos et al., 2024a.

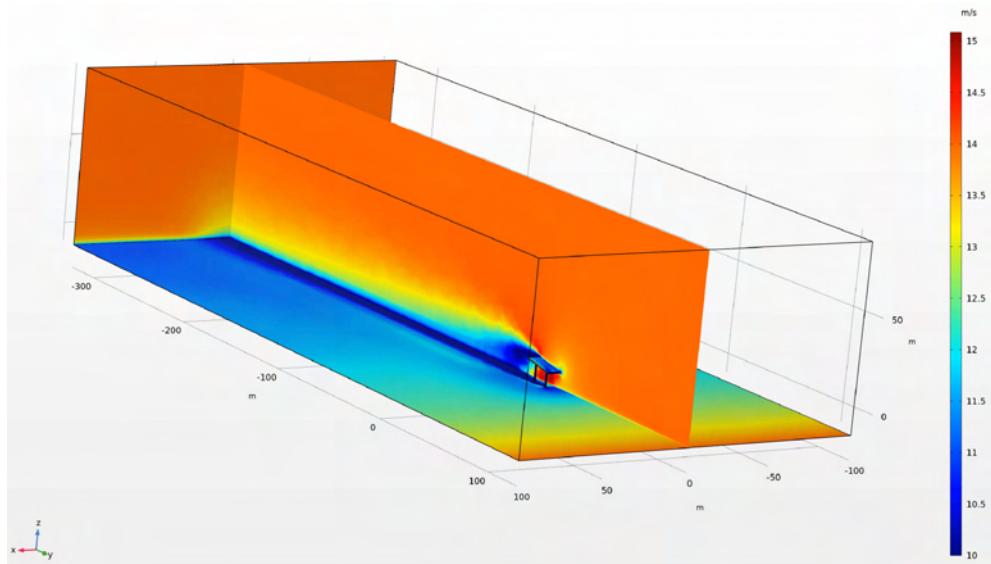


Figure 4.4: Impact of a small shed on the wind field over flat terrain. The presence of the structure causes alterations in wind speed that can propagate significantly in all directions (Michos et al., 2024a).

The iso-surfaces at 1, 2, 3, and 5 m/s show the extent of the wake introduced by the geometry of the construction. The 1 m/s iso-surface stretches approximately 250 m downstream of the WT—equivalent to six times the rotor diameter—and spans 50–80 m in width, despite the WT being static and non-rotating. Figure 4.4 demonstrates how a small shed affects the wind field over flat terrain. A similar structure placed 200 m away from a WT could influence the flow in all directions, potentially extending its effect to the rotor area. These simulations reveal how various obstacles can significantly alter wind flow behavior over large areas.

### 4.2.2 CFD modeling

The WiSpEx CFD simulations were performed using COMSOL Multiphysics 5.6, chosen for its robust capabilities in multi-physics simulations and its flexibility in customizing models and geometries to meet specific requirements as shown in the previous chapter. The WiSpEx approach for this study includes three primary components: the Internal Area Simulation (IAS), the EAS and the Extrapolated-Internal Area Simulation (E-IAS). The IAS is created to validate the performance of WiSpEx in the E-IAS, which uses the results of the EAS to extrapolate inlet conditions.

The  $k-\epsilon$  turbulent flow model is employed for the simulations due to its computational efficiency and comparable performance to other turbulent models. This model solves the Reynolds-averaged Navier-Stokes (RANS) equations for momentum conservation and the continuity equation for mass conservation. Turbulence effects are modeled using the standard two-equation  $k-\epsilon$  model as shown in the previous chapter.

A total of 375 simulations were conducted to evaluate WiSpEx's performance. A fixed inlet position was chosen due to time constraints and the number of simulations. The maximum absolute deviation of measured wind direction with respect to the normal vector to the inlet plane was approximately 30°. The presence of WTs and small buildings was excluded from the simulations to simplify the models mesh, acknowledging the potential impact of this simplification on the results as previously discussed.

In our case study, we divided the profile vertically into three sections ( $z < 40 \text{ m}$ ,  $40 \text{ m} \leq z \leq 200 \text{ m}$ , and  $z > 200 \text{ m}$ ). Missing data for  $z < 40 \text{ m}$  and for  $z > 200 \text{ m}$  were the sole reason for such a division. The simulations were further streamlined by assuming isothermal incompressible flow.

For the geometrical discretization of the simulation domain, a non-

## 4.2. IMPACT OF OBSTACLES AND SIMPLIFICATIONS ON WIND FLOW

uniform tetrahedral mesh was employed, dividing the domain into two distinct regions: the inner region that is used for the IAS and E-IAS, which contains the instruments and WTs, and the outer region, designated for capturing flow characteristics via the EAS. The meshing strategy was influenced by the low resolution of the inlet data, computational limitations, and assumptions made at the inlet.

Following the approach by Michos et al., 2024a, the domain was discretized using an unstructured tetrahedral mesh, as illustrated in Figure 4.5. This mesh type effectively represents terrain anomalies without excessively increasing the number of elements. An automatic normal-size mesh was applied to the inner region, resulting in a maximum element size of 73.5 m and a minimum element size of 21.9 m. In contrast, an automatic extra coarse mesh was used for the outer region, yielding a maximum element size of 219 m and a minimum element size of 54.8 m.

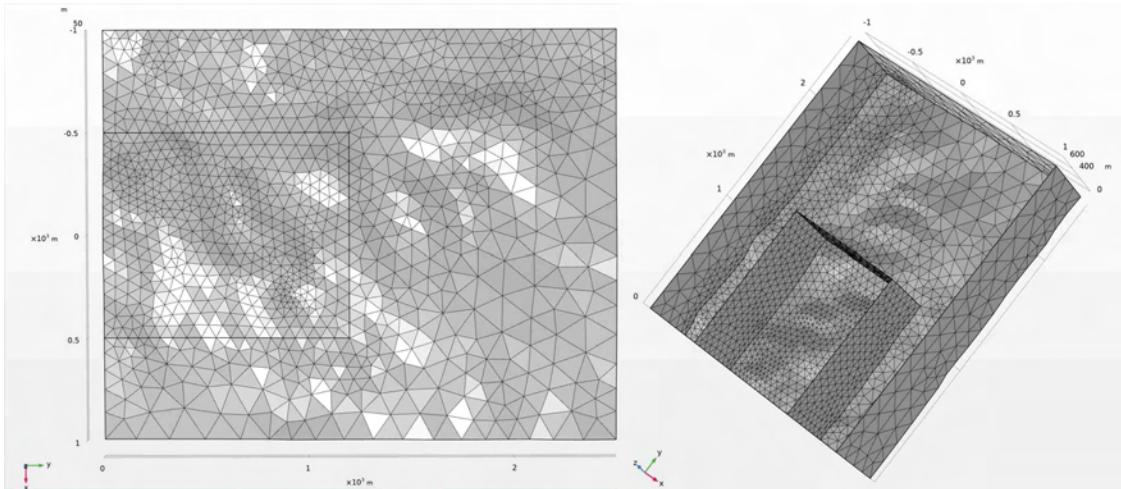


Figure 4.5: Geometry meshing of the simulation domain. The inner region features a finer mesh to capture detailed flow characteristics near the WTs, while the outer region has a coarser mesh, optimized for computational efficiency (Michos et al., 2024a).

The choice of mesh density was driven by the need to balance accuracy and computational efficiency. The inner region, where the WF is located, features a denser mesh to accurately capture the localized flow dynamics around the WTs. In the outer region, the mesh is coarser, which aids in capturing the broader flow characteristics without unnecessary computational expense. A uniformly dense mesh in the outer region would not have significantly improved the accuracy of results in the inner region, as a non-uniform coarse mesh that gradually becomes denser towards the inner region already provides sufficient detail. The domain height was set to 0.6 km, based on findings from Shen et al., 2019. This setup considers

the inlet assumption that wind speed remains constant along the inlet height for the IAS and EAS, which combined with the complex terrain, can disrupt simulation convergence and propagate errors, potentially resulting in unsafe results. The simulated flow can experience strong shocks at the initial interaction with the terrain due to the homogeneity of initial conditions over complex topography. A gradual transition from the coarser outer mesh to the denser inner mesh helps the simulated flow to absorb these shocks, smoothing the impact of surface anomalies.

The distance from the inlet area to the inner region is over 1 km, more than five times the height difference between the lowest and highest terrain altitudes, allowing sufficient space for flow development. The top boundary of the domain, positioned over 450 m above the highest terrain point, has the coarsest mesh, as the flow at this altitude is expected to be less turbulent, reducing the need for finer resolution.

### 4.3 Wi.Sp.Ex.

WiSpEx operates under the Michos et al., 2024a hypothesis that steady-state solutions of CFD equations can effectively represent the flow over complex, uncontrolled environments under specific conditions, stated as follows:

**The temporal evolution of airflow within a volume can be represented by a series of stationary states. As the volume increases, a stationary flow should effectively represent the average flow over a longer duration. Conversely, as the mean velocity of the airflow within the volume increases, a stationary flow should represent the averaged flow over a shorter duration.**

This assumption is rooted in the principles of calculus used in CFD and finite-element theory. The core idea is that the temporal evolution of airflow within a volume (CFD discretized) can be approximated by a series of steady (stationary) states. As the volume increases or the mean velocity of the airflow within the volume rises, a stationary flow should represent the averaged flow over a longer or shorter duration, respectively.

The EAS aims to physically extrapolate the wind profile both vertically and horizontally over the E-IAS inlet plane. To achieve this, a 2D inlet plane wind field was generated, assuming uniform velocity along the X-axis of the inlet and zero vertical velocity component. A ramp function was

used to extrapolate LIDAR wind measurements vertically below 40m, while constant extrapolation was used above 200m.

Once the EAS results were obtained, a 3D matrix containing the simulated wind velocity vectors at the inlet of the examined area was created. A deviation matrix was then calculated to capture the difference between the EAS simulated LIDAR velocity and the actual LIDAR measurements. Finally, a new matrix was generated to represent the extrapolated inlet wind field, combining the deviation matrix and the extrapolated LIDAR measurements. The vertical velocity component for the E-IAS inlet field was set to zero, similar to the IAS and EAS. The E-IAS simulation was then run over the area of interest, which is approximately 1/4 the volume of the expanded area.

#### 4.3.1 IAS and EAS

Both IAS and EAS are based on the same assumptions and used the same data as inlet conditions. The inlet of the expanded area was located 1.3 km away from the LIDAR, which was positioned at the IAS inlet. The average distance of the WTs from the IAS inlet was 566 m (figure 4.6).

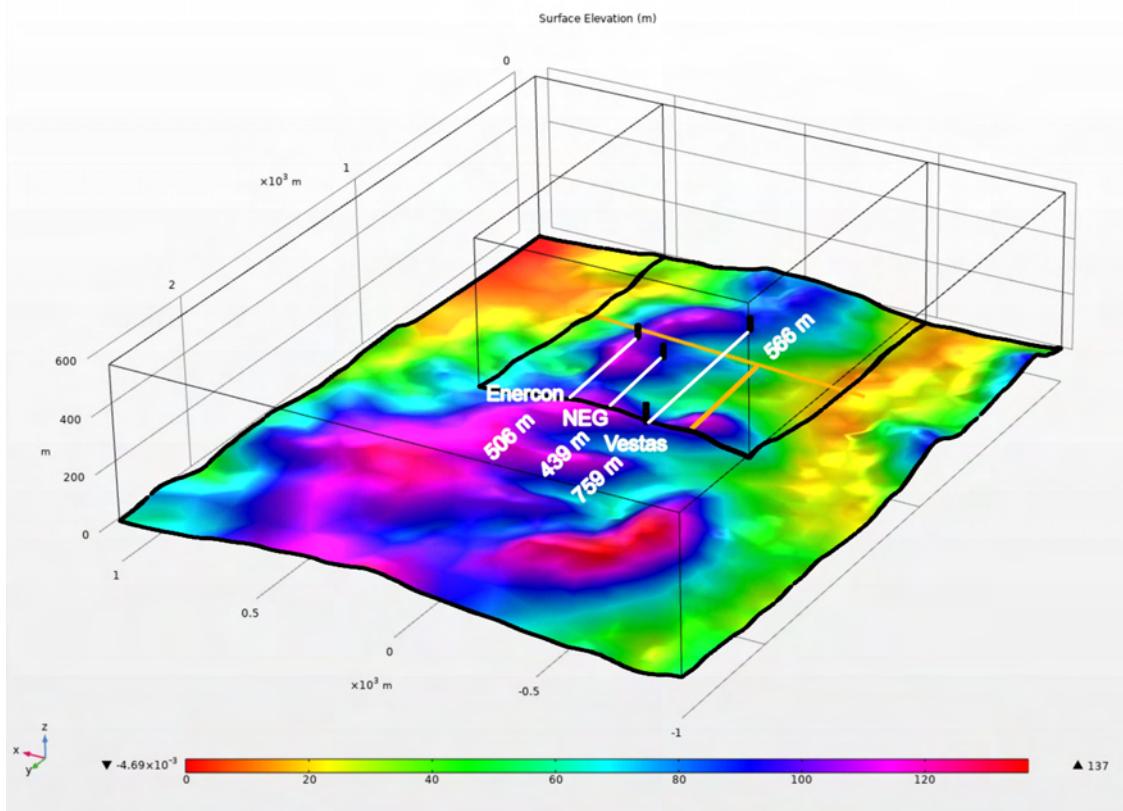


Figure 4.6: Elevation map of CRES location used for the simulations and distances between LIDAR and WTs(Michos et al., 2024a).

A ramp function ( $U_{ramp}(z)$ ,  $z \in \mathbb{R}$ ) from the COMSOL 5.6 interface was used to extrapolate LIDAR wind measurements ( $U_{LIDAR}(x, z, t_n)$  vertically below 40 m (parameters: location = -10, slope = 1/30, cutoff = 1; smoothing parameters: size of transition zone at start = 0, size of transition zone at cutoff = 50, number of continuous derivatives = 2). The ramp function graph is demonstrated in the previous chapter and in Figure 4.7.

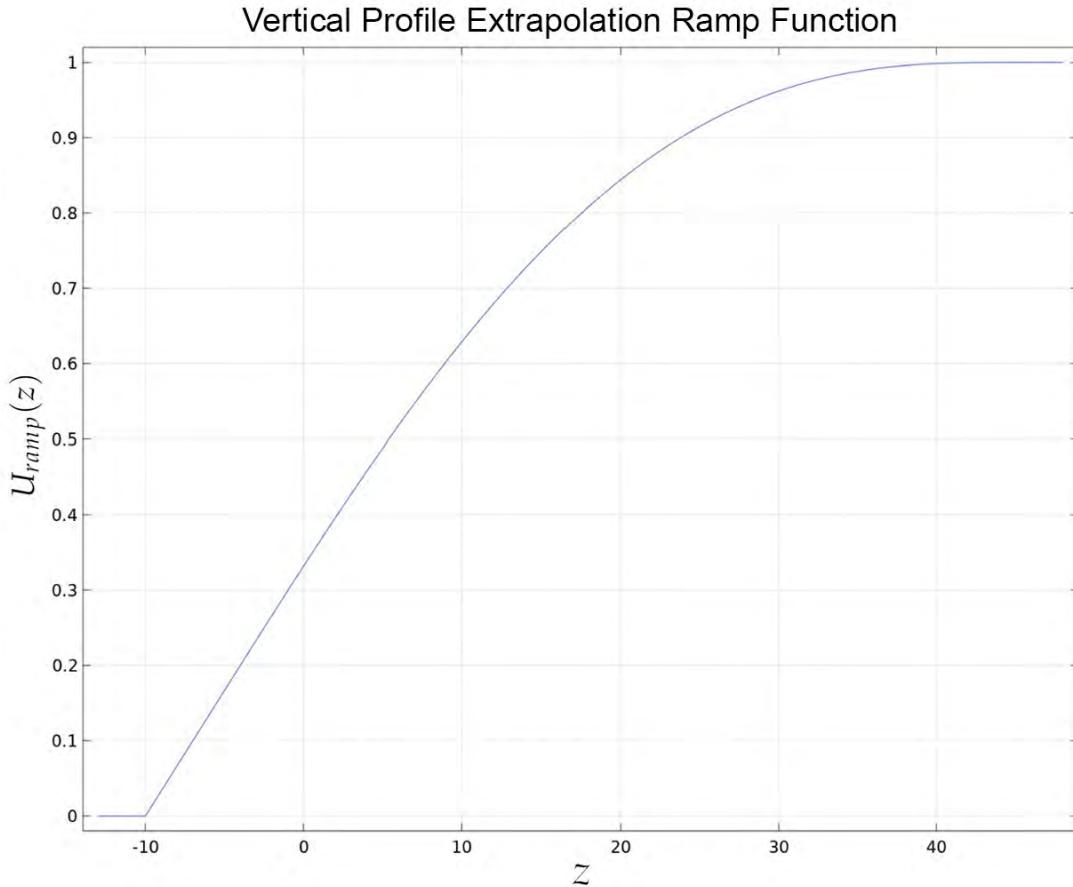


Figure 4.7: Vertical profile extrapolation ramp function  $U_{ramp}(z)$ .  $z = 0$  represents the ground level at the location of the LIDAR.  $U_{ramp}(z) = 1$  for  $z \geq 40$  m (First LIDAR measurement height), Michos et al., 2024a.

In the absence of data below or above a certain height, researchers must adapt the ramp function to their specific case. We chose the simplest approach due to this uncertainty, to demonstrate the performance of WiSpEx. Constant extrapolation was used above 200 m. The EAS goal was to physically extrapolate the wind profile vertically and horizontally over the E-IAS(/IAS) inlet plane. To generate the 2D inlet plane wind field ( $U_{inlet}(x, z, t_n)$ ), the velocity along the X-axis of the inlet was assumed to be uniform, and the vertical velocity component was set to zero. The mathematical formulation of our inlet and initial conditions is :

If

$$U_{LIDAR}(x, z, t_n) = (u_{x-LIDAR}, u_{y-LIDAR}, u_{z-LIDAR})(x, z, t_n), n \in [0, 375] = C \subset \mathbb{N}$$

and

$$U_{inlet}(x, z, t_n) = (u_{x-inlet}, u_{y-inlet}, u_{z-inlet})(x, z, t_n), n \in C$$

, then  $\forall n \in C$ :

$$\begin{aligned} \partial U_{inlet}(x, z, t_n) / \partial x &= (0, 0, 0) \\ u_{z-inlet}(x, z, t_n) &= 0 \end{aligned} \quad (4.3.1.1)$$

and

$$\begin{aligned} U_{inlet}(x, z, t_n) &= U_{ramp}(z) * U_{LIDAR}(40, t_n), z < 40 \\ U_{inlet}(x, z, t_n) &= U_{LIDAR}(z, t_n), 40 \leq z \leq 200 \\ U_{inlet}(x, z, t_n) &= U_{ramp}(z) * U_{LIDAR}(200, t_n), 200 < z \end{aligned} \quad (4.3.1.2)$$

The assumption that the averaged wind flow would have similar behaviour if the terrain was homogeneous matches wind LIDAR measurement assumptions above complex terrain Kogaki et al., 2020. The aim of EAS simulations was to capture the wind flow characteristics at the inner area's inlet (wind velocity and direction distribution along the 2D inlet).

### 4.3.2 E-IAS

After the EAS results were created, a 3D matrix that contains the simulated wind velocity vectors  $U_{EAS}(x, z, t_n) = (U_{x-EAS}, U_{y-EAS}, U_{z-EAS})(x, z, t_n)$  at the inlet of the examined area of our study (where the LIDAR is located) was created. A deviation matrix was created for  $0 \text{ m} \leq z \leq 600 \text{ m}$  and  $-500 \text{ m} \leq x \leq 500 \text{ m}$ :

$$U_{dev}(x, z, t_n) = U_{EAS}(x, z, t_n) - U_{LIDAR-sim}(z, t_n) \quad (4.3.2.1)$$

where  $U_{LIDAR-sim}$  represents EAS simulated LIDAR velocity at measurement heights, extrapolated with constant extrapolation below 40 m and above 200 m. To extrapolate the LIDAR measurements along the inlet plane, we created a new matrix that contains the extrapolated inlet wind field:

$$U_{LIDAR-ex}(x, z, t_n) = U_{dev}(x, z, t_n) + U_{LIDAR*}(z, t_n) \quad (4.3.2.2)$$

Here,  $U_{\text{LIDAR}^*}$  represents LIDAR measurements extrapolated with constant extrapolation below 40 m and above 200 m. The vertical velocity component is set once again to zero for the E-IAS inlet field, similar to IAS and EAS. The E-IAS simulation is run over the area of interest (internal domain), which is approximately one-fourth the volume of the expanded area.

### 4.3.3 Results

In this section, we will discuss the results from IAS, EAS, and E-IAS and compare their performance after being corrected with linear regression, in terms of Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE) and MAPE:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (4.3.3.1)$$

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4.3.3.2)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (4.3.3.3)$$

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| 100\% \quad (4.3.3.4)$$

where  $\hat{y}_i$  is the forecasted value and  $y_i$  is the measured and  $y_i - \hat{y}_i$  is the absolute error.

#### IAS

Table 4.1: IAS – WT Corrected Velocity Magnitude Statistics

	<b>Enercon</b>	<b>NEG</b>	<b>Vestas</b>
MAE [m/s]	0.88	0.76	0.56
MSE [m <sup>2</sup> /s <sup>2</sup> ]	1.29	0.98	0.6
RMSE [m/s]	1.14	0.99	0.77
MAPE [%]	8.26	7.94	7.09

The MAE of wind velocity estimation at the hub height is 0.88 m/s for the Enercon WT, 0.76 m/s for the NEG WT, and 0.56 m/s for the

Vestas WT. The corresponding MAPE values are 8.26%, 7.94%, and 7.09%, respectively, as summarized in Table 4.1. These variations in error metrics can be attributed to the positional relationship between the WT locations and the LIDAR measurements, as illustrated in Figure 4.2.

Figures 4.8–4.11 present the velocity magnitude plots alongside their respective linear regression scatter plots for the Enercon, NEG, and Vestas WTs. An examination of the velocity magnitudes figures, indicates that wind direction (shown via the intensity of the west-east component of the Simulated velocity by the green line) affects the accuracy of the model.

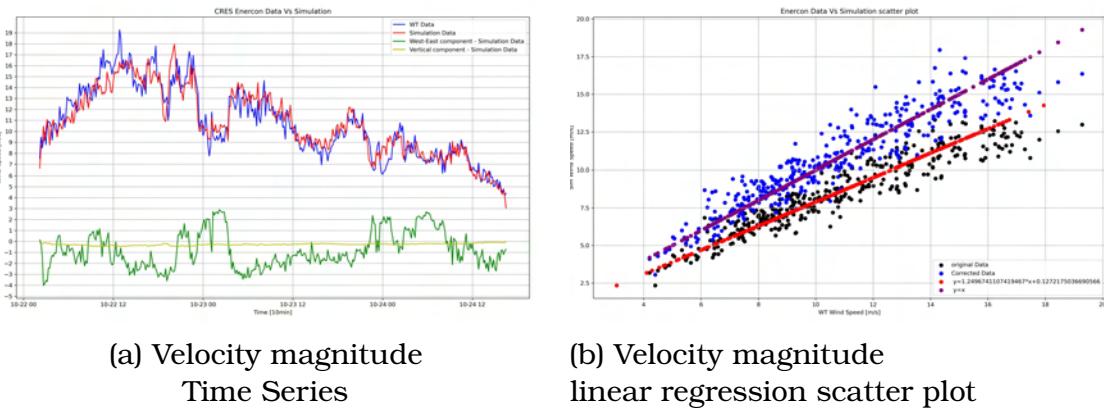


Figure 4.8: ENERCON IAS results(Michos et al., 2024a).

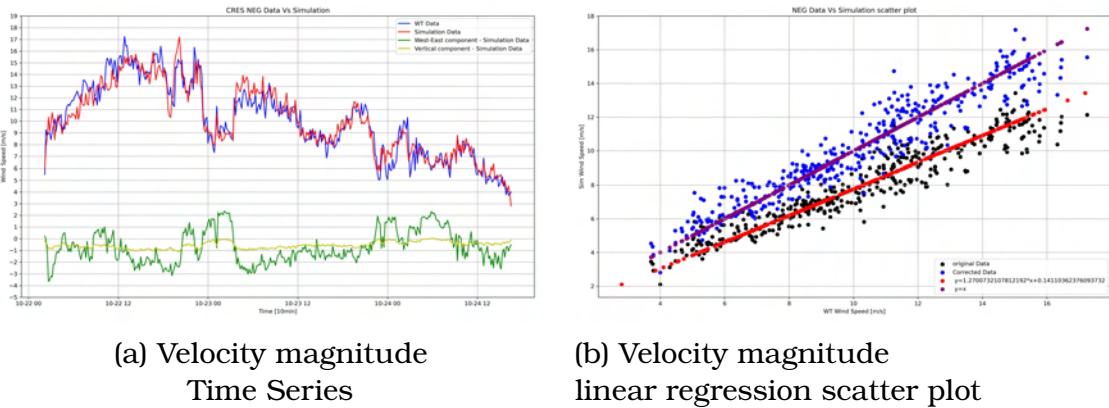


Figure 4.9: NEG IAS results(Michos et al., 2024a).

The observed biases seen in figures 4.8b, 4.9b are influenced by the assumption of a homogeneous velocity profile along the horizontal axis of the inlet and the extrapolation technique. The homogeneity assumption introduces bias dependency on wind speed because it fails to account for terrain slopes in front of the WT locations. This omission leads to inaccurate inlet conditions that do not fully capture the flow dynamics over the complex terrain. In figures 4.10, we can see the differences in the wind direction distribution for LIDAR and Enercon locations, underscoring the effect of terrain curvature on the the flow.

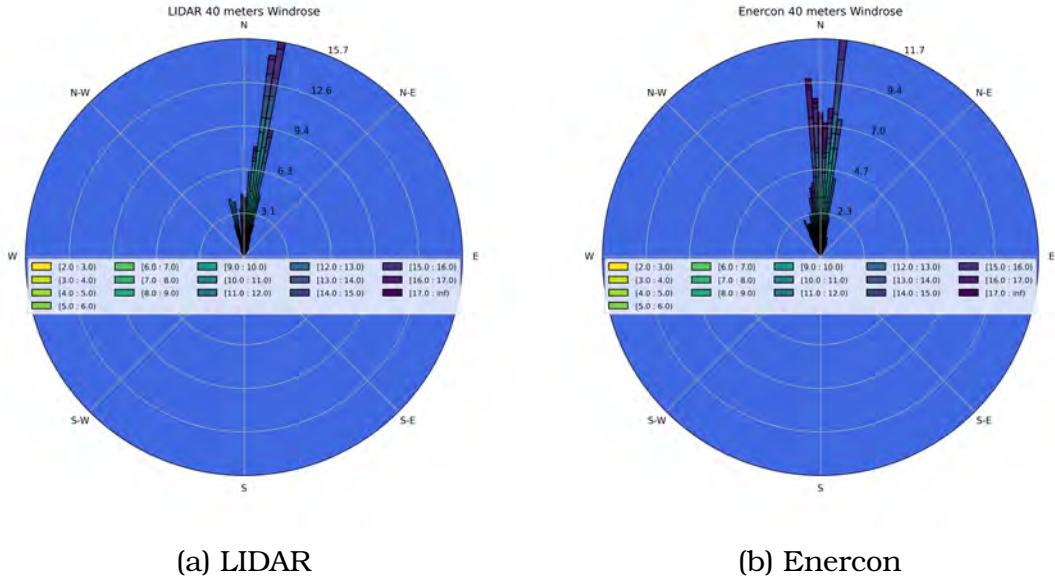


Figure 4.10: Windroses.

It is worth noting that the Vestas WT, positioned downstream of the LIDAR, benefits from more accurate flow representation, thereby resulting in lower MAE and MAPE values. Additionally, different types of biases are observed across varying wind speeds, as seen in figure 4.11b. These discrepancies may be linked to calibration issues or inaccuracies in the reported wind output values, especially since the Vestas WT is oriented vertically relative to the flow direction and is not producing power. These factors likely contribute to inconsistencies in velocity measurements across different operational conditions.

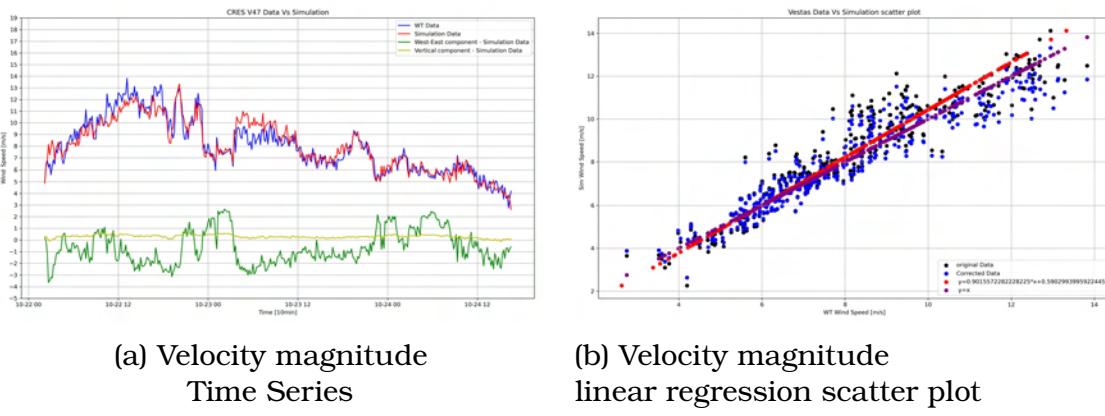


Figure 4.11: VESTAS IAS results(Michos et al., 2024a).

The RMSE values of 1.14 m/s for Enercon, 0.99 m/s for NEG, and 0.77 m/s for Vestas, together with the MSE values, further underscore the accuracy of the simulations in representing the wind velocity, regarding the assumptions of the IAS. Lower MAE and RMSE values indicate a closer

match between the predicted and observed velocities, highlighting the robustness of the model at the Vestas WT location. The RMSE specifically measures the standard deviation of prediction errors, providing insights into the magnitude of errors present, while the MAE provides a direct measure of average error magnitude, and the MAPE expresses this error relative to the observed values in percentage terms. It is worth noting that higher MAPE values are observed at low speeds.

## EAS

Table 4.2: EAS – LIDAR Corrected Velocity Magnitude Statistics

	40m	54m	78m	100m	120m	140m	160m	180m	200m
MAE[m/s]	0.41	0.37	0.30	0.22	0.17	0.13	0.10	0.09	0.11
MSE[m <sup>2</sup> /s <sup>2</sup> ]	0.30	0.23	0.15	0.08	0.05	0.03	0.02	0.01	0.02
RMSE[m/s]	0.55	0.48	0.39	0.29	0.22	0.16	0.12	0.11	0.13
MAPE[%]	4.28	3.70	2.81	2.05	1.56	1.17	0.86	0.76	1.00

Table 4.2 presents the MAE values for LIDAR wind velocity magnitude estimations across all available heights, ranging from 0.09 m/s to 0.41 m/s, with corresponding MAPE values between 0.76% and 4.28%. Notably, for 90% of all LIDAR level estimations, the absolute errors remain below 0.7 m/s, and the absolute percentage errors are kept under 10%. The higher errors observed during abrupt shifts in wind speed or direction, commonly referred to as ramp events, reflect the challenges of capturing such transient phenomena.

Table 4.3: EAS – WT Corrected Velocity Magnitude Statistics

	Enercon	NEG	Vestas
MAE [m/s]	0.85	0.85	0.67
MSE [m <sup>2</sup> /s <sup>2</sup> ]	1.20	1.13	0.77
RMSE [m/s]	1.10	1.06	0.88
MAPE [%]	7.78	8.78	8.41

At the Enercon WT's hub height, the MAE for wind velocity estimation is 0.85 m/s, aligning closely with the MAE observed for the NEG WT. In contrast, the Vestas WT demonstrates a lower MAE of 0.67 m/s. The corresponding MAPE values are 7.78% for Enercon, 8.78% for NEG, and 8.41% for Vestas, as detailed in Table 4.3. The improved performance for Enercon WT can be attributed to its position relative to the LIDAR, resulting in more accurate flow characteristics at its location compared

to the other two WT due to the inlet offset combined with the terrain characteristics (4.6). On the other hand, the terrain complexity along the paths that air flow streams to reach NEG and Vestas WTs, increases the error magnitude and variance. Specifically, in the case of Vestas WT, this error change can also derive from wakes of nearby objects, like the Vestas WT and a meteorological Mast.

The MSE and RMSE values offer further insights into the model's performance. For the Enercon WT, the RMSE is 1.10 m/s, while for the NEG WT it is 1.06 m/s, and for the Vestas WT, it is the lowest at 0.88 m/s. These metrics are of the wind velocity estimations, particularly for the Vestas WT, where the errors are minimized due to better alignment of the wind flow characteristics with the model assumptions. The performance statistics of the EAS simulation is worse than that of the IAS due to the assumptions about the offset placement of the inlet boundary.

It is worth noting that the improvement of the ENERCON WT's results can also be attributed to the increased offset distance allowing the unstructured mesh to better absorb the initial flow shock due to terrain characteristics, thereby stabilizing the simulated flow before it reaches the IAS inlet. This configuration not only reduces the magnitude of errors but also eliminates the bias dependency on wind speed, leading to more reliable simulation outcomes.

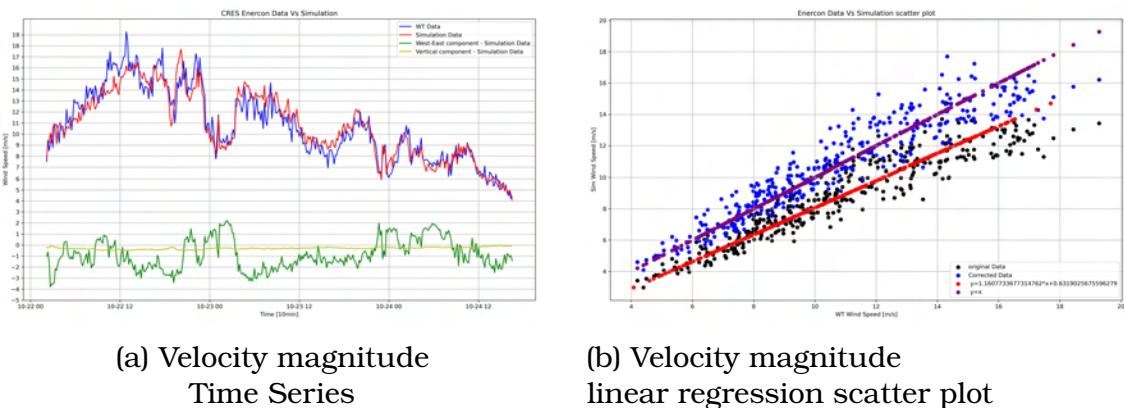


Figure 4.12: Enercon EAS results(Michos et al., 2024a).

Figures 4.12b and 4.13b demonstrate that the biases in the raw simulation results are less dependent on the velocity magnitude, indicating that the EAS effectively captures the influence of terrain complexity on flow behavior, despite the simplified extrapolation of our benchmark model and the coarse mesh applied over the terrain in contrast with the performance statistics of NEG WT. This bias magnitude and wind speed dependency reduction underscores the ability CFD steady state simulations to capture

terrain induced variations, which is the focus of the EAS simulations.

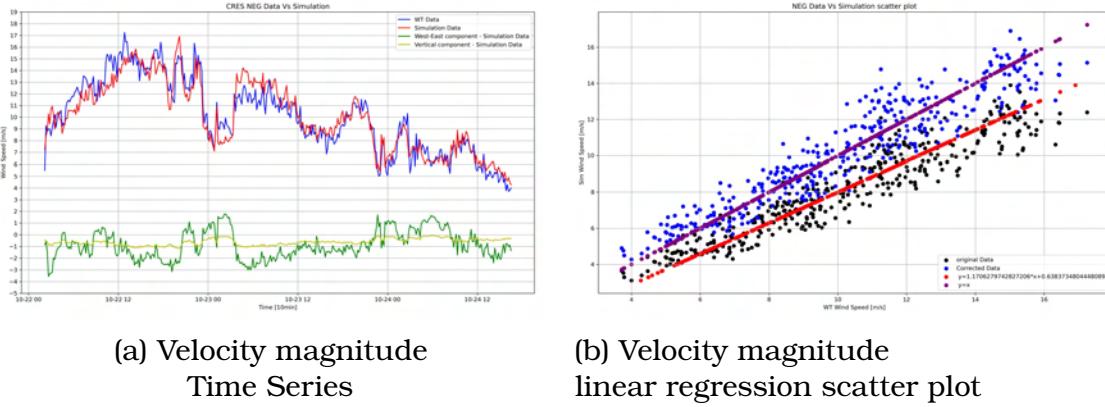


Figure 4.13: NEG EAS results(Michos et al., 2024a).

The Vestas WT, positioned downstream of the LIDAR, benefits from flow conditions that are more accurately represented by the model due to the favorable initial conditions. This leads to the lowest error rates observed among the evaluated WTs. However, it's important to acknowledge that the error metrics for the Vestas WT aren't superior to those of the IAS due to the aforementioned reasons. The observed error behaviors underscore the already mentioned implications deriving from insufficient datasets, as well as the strength of CFD usage in wind speed extrapolation.

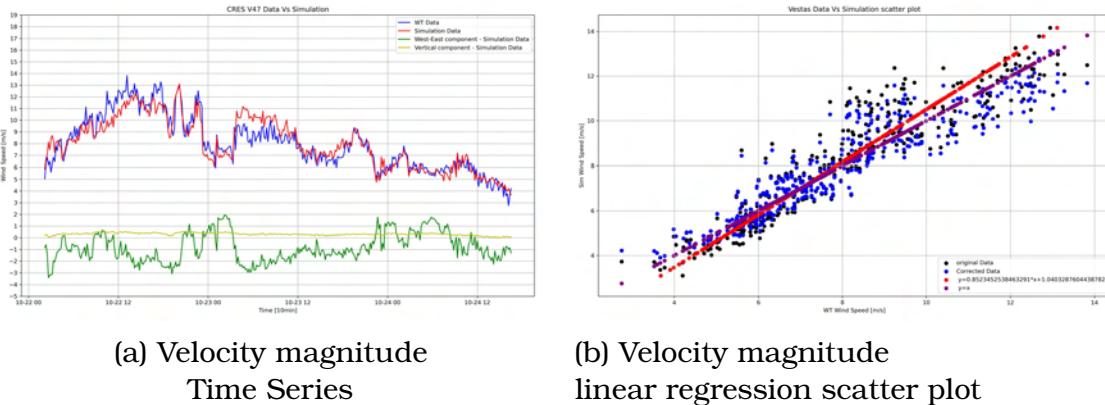


Figure 4.14: Vestas EAS results(Michos et al., 2024a).

Overall, EAS LIDAR estimations exhibit strong accuracy despite the study's simplifications, and the absence of measurements near ground level, where turbulence effects are pronounced, contributes to higher error margins, as seen in figure 4.15. The use of an extrapolation method for wind fields below 40 m significantly affects accuracy in lower altitudes, particularly from ground level up to 54 m. As wind speed estimations are made at higher heights, the errors generally diminish. However, at 200 m—the

highest measurement point—estimation errors increase, highlighting the limitations of relying on extrapolation in place of direct measurements. Since measurements below 40 m are critical for our analysis, the incorporation of additional low-level data points would likely enhance EAS accuracy in representing the wind flow behavior influenced by terrain-induced turbulence.

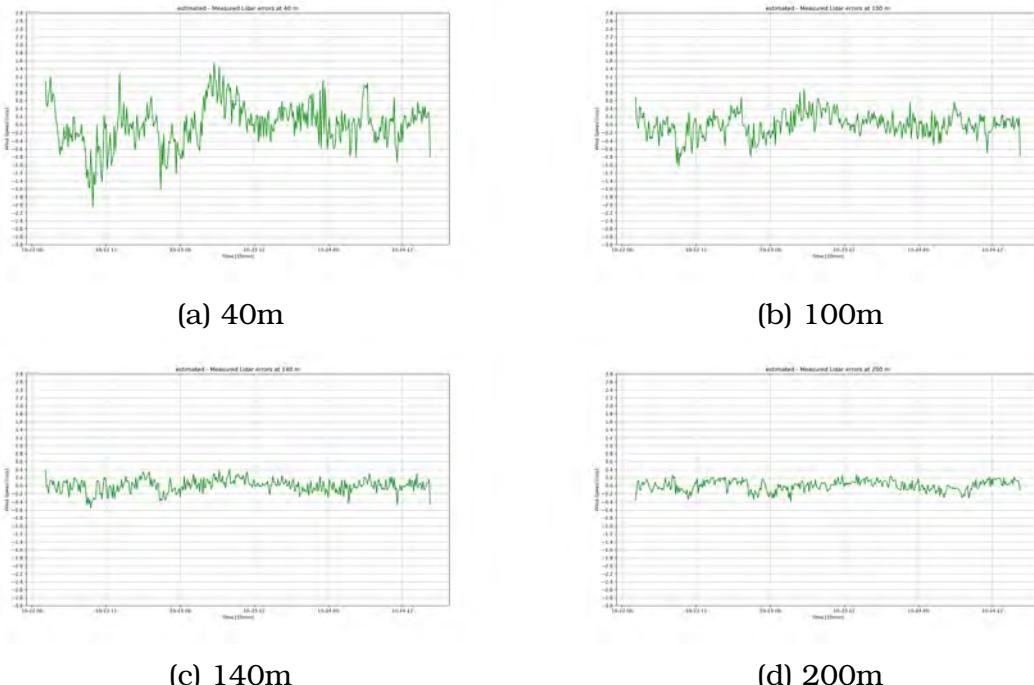


Figure 4.15: EAS - Corrected LIDAR estimation errors at different heights.

### E-IAS

Table 4.4: E-IAS – WT Corrected Velocity Magnitude Statistics

	<b>Enercon</b>	<b>NEG</b>	<b>Vestas</b>
MAE [m/s]	0.81	0.76	0.56
MSE [ $m^2/s^2$ ]	1.08	0.93	0.58
RMSE [m/s]	1.04	0.96	0.76
MAPE [%]	7.62	7.98	7.08

The E-IAS simulations, producing the final output of the WiSpEx model, demonstrate the ability to transform limited LIDAR measurements into a detailed wind field dataset. As shown in Table 4.4, the MAE for wind velocity estimation is 0.81 m/s for the Enercon WT, 0.76 m/s for the NEG WT, and 0.56 m/s for the Vestas WT. The corresponding MAPE values

are 7.62%, 7.98%, and 7.08%, respectively. These results indicate an improvement over previous simulations, highlighting the model's efficiency in extrapolating sparse measurements into a comprehensive high-resolution dataset. Specifically, the model successfully reconstructs a 3D wind field at 29,928 locations using only nine vertical measurement points per time step, achieving this transformation in less than two minutes. This capability provides a 3325-fold increase in data resolution, which is particularly valuable for enhancing ultra-short-term wind forecasting applications.

Figures 4.16–4.18 provide a detailed view of the velocity magnitude plots and corresponding linear regression scatter plots for each WT. These visualizations demonstrate that extrapolating One-dimensional (1D) measurements into a 2D inlet wind field has effectively reduced the dependency of errors on wind speed for the Enercon and NEG WT simulations.

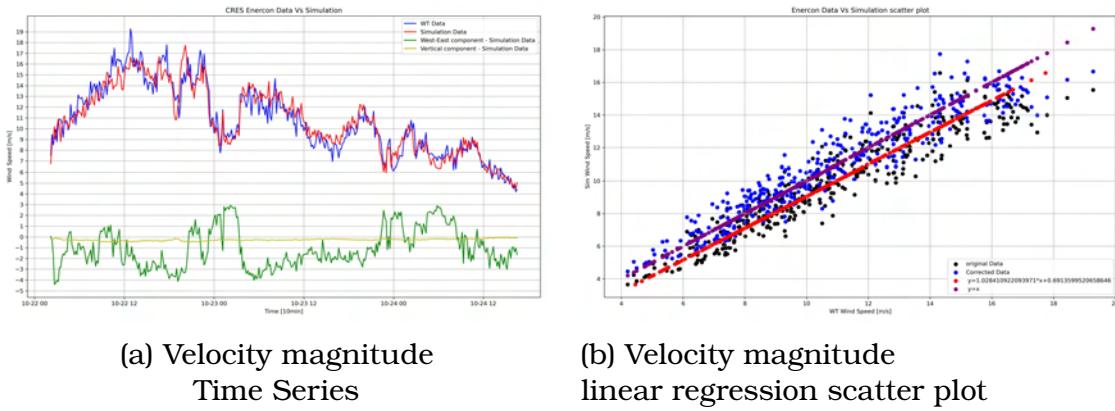


Figure 4.16: ENERCON E-IAS Results(Michos et al., 2024a).

This reduction is a promising outcome, considering the E-IAS objective to improve velocity estimates at locations not immediately downstream of the LIDAR. Notably, the benchmark  $k - \epsilon$  version of WiSpEx tends to underestimate wind speeds, a pattern observable across the datasets.

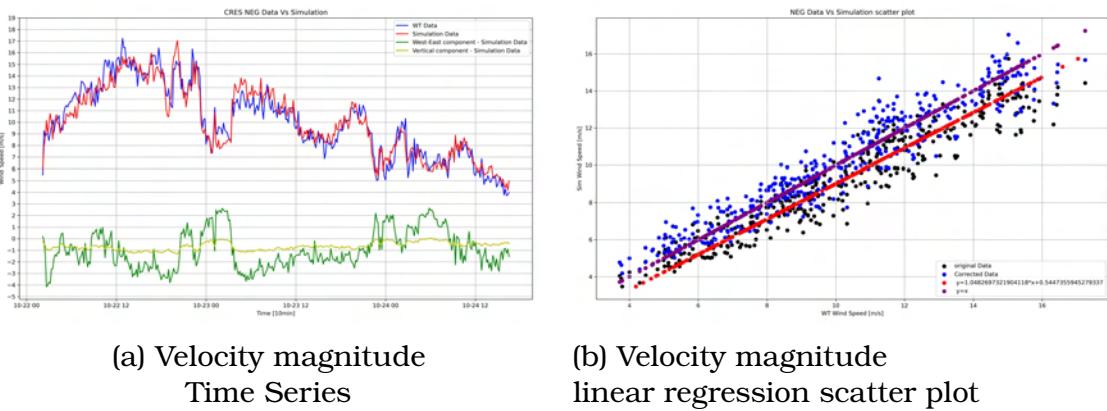


Figure 4.17: NEG E-IAS Results(Michos et al., 2024a).

For the Vestas WT, depicted in Figure 4.18, a noticeable dependency of errors on wind speed magnitude persists, similar to that in the IAS and EAS. This error is particularly evident at higher wind speeds, where the E-IAS model exhibits an abrupt shift in bias. A detailed inspection of Figure 4.18b reveals four distinct sectors of error behavior, delineated within the ranges 0-5 m/s, 5-8 m/s, 8-10 m/s, and 10-14 m/s. The error patterns suggest potential intrinsic issues with measurement accuracy, particularly at high wind speeds or presence of wakes.

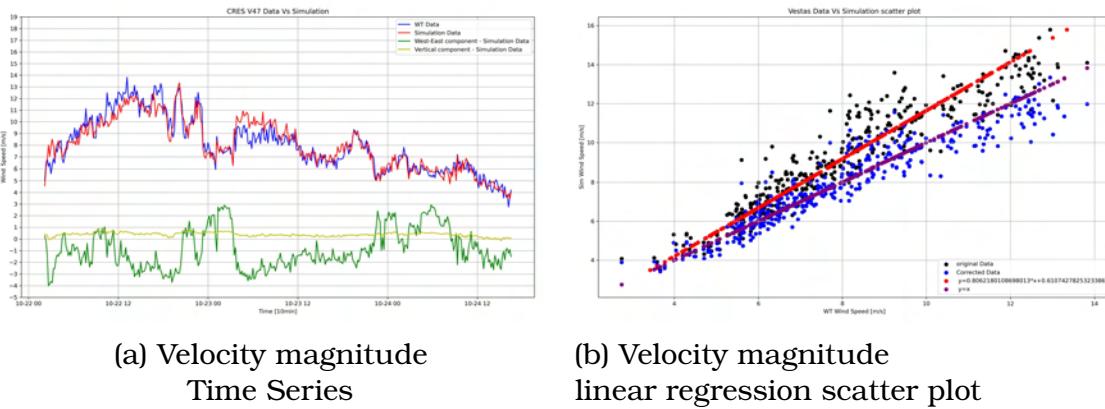


Figure 4.18: VESTAS E-IAS Results(Michos et al., 2024a).

The MSE highlights how the Enercon WT experiences slightly higher average discrepancies compared to the other WTs, with values of 1.08 for Enercon, 0.93 for NEG, and 0.58 for Vestas. The RMSE shows a similar trend, with values of 1.04, 0.96, and 0.76, respectively. Lower RMSE values for the Vestas WT indicate more consistent and accurate velocity predictions, underscoring the benefit of its downstream location relative to the LIDAR, where flow characteristics are less affected by terrain-induced disturbances.

The initial conditions for E-IAS are derived from EAS, which introduces an inherent dependency on EAS simplifications and leads to computational times more than twice as long as those for IAS. Despite these dependencies, E-IAS outperforms both EAS and IAS due to the absence of measurement offsets and the elimination of abrupt flow changes at the inlet plane. This is achieved by incorporating curvature effects directly into the flow via physics-based extrapolation through EAS, rather than relying on the simpler ramp functions used in IAS and EAS. The results strongly suggest that increasing the number of measurement locations at the inlet would further enhance the accuracy of the model.

Despite the inherent simplifications and the absence of high-resolution

measurements Ingenhorst et al., 2021; Xiaoxia et al., 2022, the E-IAS model consistently estimates wind speed at WT hub heights with relatively small errors, with 70% of absolute errors falling below 1 m/s and absolute percentage errors under 10%. Additionally, 50% of absolute errors are less than 0.71 m/s, and corresponding percentage errors are under 7.6%. The largest errors typically occur at high wind speeds, while higher percentage errors are more frequent at lower wind speeds. The highest errors are observed during extreme changes in wind speed and direction, highlighting the model's sensitivity to rapid variations in flow conditions.

#### **4.3.4 Assessing Wi.Sp.Ex.'s forecastability**

The computational time required for a single IAS simulation averages 43 seconds, with EAS requiring 61 seconds. However, E-IAS simulations exceed 100 seconds due to the additional calculations needed for extrapolation and data loading. Nonetheless, the E-IAS model's ability to generate high-resolution wind field data makes it highly suitable for use in ultra-short-term forecasting models. By creating extrapolation look-up tables, the E-IAS framework can be seamlessly integrated into forecasting applications, improving accuracy without significantly increasing computational demands. Furthermore, statistical models can be easily updated following changes in terrain by reconstructing wind fields without requiring new measurement datasets.

While direct comparisons between different model types are generally discouraged, we evaluate WiSpEx's performance against state-of-the-art forecasting models from the literature to gain a comprehensive understanding of its capabilities. This approach aims to determine whether WiSpEx can be adapted and expanded for ultra-short-term wind power generation forecasting, with the intention of integrating it into WEEL. Since the core physics-based principles of WiSpEx will remain consistent for this new application, we do not expect significant changes in its behavior.

It is important to note that the positioning of the inlet relative to the WTs affects the phase accuracy of simulated wind speeds. Specifically, increasing the distance between the inlet and the WTs leads to greater phase errors due to the time required for an air parcel to travel this distance. Consequently, placing measurement instruments far from a WF suggests that WiSpEx's extrapolations could serve effectively as forecasts.

In Wang and Yang, 2021, the basic statistics of state-of-the-art forecasting models are presented, providing a benchmark for evaluating our

model's performance. At the first forecasting step, the MAPE of our model falls within the middle of the MAPE range of these models, which is promising given the simplifications implemented in our simulations. However, at the second forecasting step, the MAPE of all models significantly increases, ranging between 8.09% and 14.89%. Notably, the model proposed by Wang and Yang, 2021 almost doubles its MAPE to 8.09%. By the third forecasting step, their model establishes the lower limit for MAPE at 11.73%, further escalating to exceed 40%.

This behavior across all models highlights the challenges of predicting such a highly variable atmospheric condition over shorter time frames. Despite these challenges, our model demonstrates the potential to produce accurate estimations. The ability to generate 3D high-resolution datasets from 1D measurements with low errors is advantageous for any forecasting model, including those compared in the study. Creating reliable extrapolated wind datasets is critical for all stages of a WF, from selecting optimal locations for WTs to ensuring efficient operation of the WF and optimizing smart grid performance.

## 4.4 W.E.E.L.

Every WT extracts Energy from wind based on it's engineered design. The momentum of a WT temporarily delays it from being synchronized with its optimal rotational frequency for a given change in wind speed. This results in a power production latency(or lag) which we call Wind Energy Extraction Latency (Michos et al., 2024b). Turbulence and strong gradients affect this latency. Our model, WEEL, named after the it's purpose, is created to unveil and exploit a WT's "power lag" to create, improve or extend (in time) Ultra-Short-Term Power Production forecasts throught the conversion of wind speed to power by reducing the errors produced by this physical process in a fast and simple way.

### 4.4.1 Model Description

WEEL captures the inherent lag between wind speed and power output with simple, fast, data driven statistical methods. Instead of relying on simple averaging of wind speed data, which can lead to a loss of valuable information regarding trends and short-term fluctuations, we propose utilizing the highest resolution  $\Delta t$  averaged measurements  $u_t^R$  (available or interpolated) and Moving Average (MA)s instead of simple averaging. This

approach helps maintain the original sample length while preserving key trend characteristics (Michos et al., 2024b).

The first part of WEEL is the Power Curve model. The Power Curve's accuracy is the primary source of errors. The Power curves  $\mathcal{P}(u)$  ( $u$  is the scalar velocity magnitude) can be produced by continuous functions approximated to real data, by averaging to speed bins (wind speed ranges), or be obtained by manufacturers. Most complex models include meteorological variables and account turbulence, focusing on how atmospheric conditions affect the power output of a WT as presented by Saint-Drenan et al., 2020. Tailoring a curve to suit realistic needs is always important in any type and scope of modeling approach.

The second part of the model is the identification of the latency. The detection of latency is called Wind Optimal Shifted Value (ShiVa) identification. Once we define the Power forecasting step(= average window for Power measurements  $P_t^R$ ),  $\Delta_{Pt} = m\Delta t, m \in \mathbb{N}$  and select the the averaging window of the wind measurements ,  $\Delta_{Wt} = n\Delta t, n \in \mathbb{N}$ , where  $m > n$ , we calculate the  $n$ -MAs of wind speed:

$$u_t \equiv u_t^n = \frac{1}{n} \sum_{t-n+1}^t u_t^R \quad (4.4.1.1)$$

and the  $m$ -MAs of Power:

$$P_t \equiv P_t^m = \frac{1}{m} \sum_{t-m+1}^t P_t^R \quad (4.4.1.2)$$

The wind measurements  $u_t$  are then shifted iteratively by  $\Delta t$  creating a series of Shifted Series (SS) and the errors over  $i$ -th iteration are calculated. For our specific case  $i \in \mathbb{N}$  and we will symbolize the shift of  $u$  by  $i$  time steps as  $u_t||_i$ , and the shift of  $\mathcal{P}(u_t||_i)$  as  $\mathcal{P}_t||_i(u) \equiv \mathcal{P}_t||_i$ . We define this type of objects as SS series and in general cases we will symbolize them as  $\mathcal{S}_t||_{\mathbb{S}} \subseteq \mathbb{R} \times \mathbb{R}$ . The error series per iteration is obtained by:

$$\mathcal{E}_{t,i} = \mathcal{P}(u_t||_i) - P_t \quad (4.4.1.3)$$

This type of objects are the SS Error series and in general occasions they are notated as  $\mathcal{E}_{t,\mathbb{S}} \subseteq \mathbb{R} \times \mathbb{R}$  , where  $m$  denotes the working average frame and  $n$  denotes the shifted by  $\mathbb{S}$  averaged frame. We can now obtain the mean error from:

$$\mathcal{M}_{\mathcal{E}_{t,i}} = \frac{1}{|\mathcal{E}_{t,i}|} \sum_t \mathcal{E}_{t,i} \quad (4.4.1.4)$$

, where  $|\mathcal{E}_{t,i}|$  represents the number of elements in the series and  $\mathcal{M}_{\mathcal{E}_{t,i}} \equiv \frac{m}{n} \mathcal{E}_{t,i} \subseteq \mathbb{R}$  is the mean Shifted Scalar error series calculated for the  $\mathcal{P}_t \|_i$  power from wind SS. In a similar fashion, we can obtain any statistical measure we want to apply. The optimal ShiVa we seek is the value  $i_{opt}$  for wind speed  $u_t \|_{i_{opt}}$  that minimizes  $\mathcal{M}_{\mathcal{E}_{t,i}}$ .

Assume a general Power Shift Series error of Series  $\mathcal{E}_{t,\$}^m \subseteq \mathbb{R} \times \mathbb{R}$  with a general Shifted scalar error series  $G_{\mathcal{E}_{t,\$}}^m \subseteq \mathbb{R}$ . The optimal ShiVa  $\mathbb{S}_{opt}$  is defined by the Michos Optimal Shift Value Hypothesis (MOSVaH) as :

$$\exists i_{opt} \in (0, n) \subset \mathbb{R}, \text{ for which } G_{\mathcal{E}_{t,i_{opt}}}^m = \min_{i \in \mathbb{Z}} \frac{G_{\mathcal{E}_{t,i}}^m}{n}$$

In our case, where  $i \in \mathbb{N}$ , the identification of the  $i_{opt}$  value minimizing  $\mathcal{M}_{\mathcal{E}_{t,i}}$  or a combination of error functions is a simple minimization identification of m statistical metrics or sets of statistical metrics respectively.

When the optimal ShiVa is found, WEEL searches for the maximum ShiVa  $i_{max}$  which represents the maximum shift we can do to approximate the error of the unsifted series. Now that we got a grasp of SS basics, we can further simplify the symbols. let  $\mathcal{E}_i$  be the simple SS Error series and  $\mathcal{M}_i$  the respective mean Shifted Scalar error series . The maximum ShiVa  $i_{max}$  is defined:

$$\exists i_{max} \in [n, m + n] \subset \mathbb{R}, \text{ for which } \mathcal{M}_{i_{max}} = \mathcal{M}_0$$

In numerical solving, where continuous variables are never available, the simplest method is to select the best approximation. WEEL can directly produce ultra-high-time-frame forecasts quickly and directly by applying a power curve transformation to WEEL's maximum ShiVa shifted wind speed series to obtain  $\mathcal{P}(u_t \|_{i_{max}})$ . WEEL uses the ShiVas to stabilize the error from power conversion and/or extend the forecasting horizon of a forecasting module. In our case, it utilizes WiSpEx, assuming that there is a phase error due to air travel distance from the inlet of the simulations to the WTs (Michos et al., 2024b). Such a scenario is possible when measurement locations are far away from the WT vicinity, like in the case of conically scanning LIDARs. WEEL's task flow can be seen in figure 4.19.

## W.E.E.L Diagram

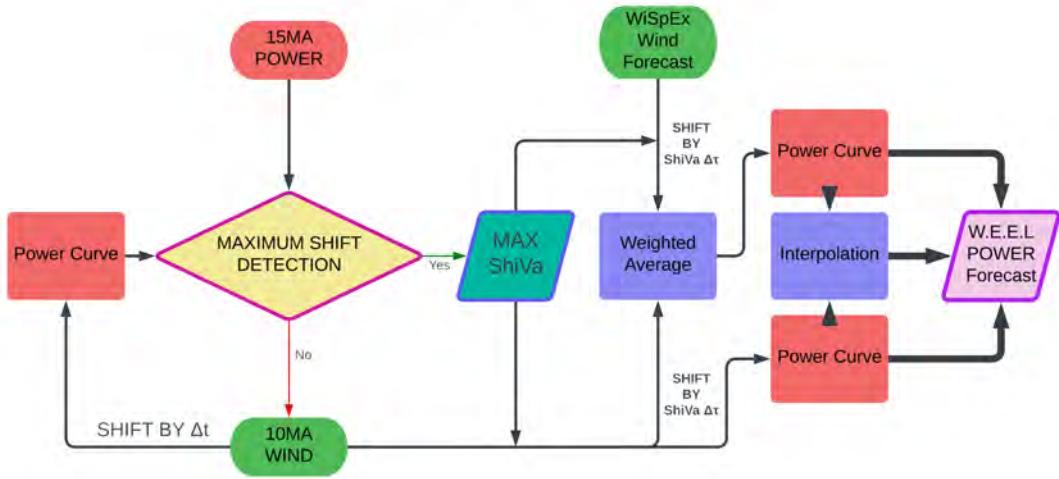


Figure 4.19: WEEL diagram (Michos et al., 2024b)

### 4.4.2 Power curve

The conversion of wind speed to power is typically accomplished by generating a power curve  $P_{fit}$  from empirical data or using the manufacturer's specified power curve. The accuracy of the fitted power function and the standard deviation of the error significantly influence the performance of WEEL. To assess the performance of WEEL under realistic conditions (Michos et al., 2024b), we consider a scenario where the function approximating the power curve of a WT captures the primary characteristics of power production but may not fully account for behavior at the extremes of the WT's operational range. For this purpose, we employ least squares optimization (Virtanen et al., 2020) in python to approximate the power curve with  $P_{fit}$ , which is based on the input variable  $x$  (e.g., wind speed) and several parameters.

Least squares optimization is a fundamental technique for fitting a model to data by minimizing the sum of squared differences between observed values and the values predicted by the model. The objective is

to find the model parameters that make the predicted values as close as possible to the actual observations.

Given a set of observed data points  $(x_i, y_i)$ , where  $x_i$  denotes the input variable (e.g., wind speed) and  $y_i$  denotes the output variable (e.g., power), we aim to fit a model function  $f(x, \theta)$  to these data points. Here,  $\theta$  represents the parameters of the model. The function  $f(x, \theta)$  predicts the output  $y$  for a given input  $x$ . To evaluate the fit of the model, we use the Sum of Squared Errors (SSE), which is defined as:

$$\text{SSE} = \sum_{i=1}^n [y_i - f(x_i, \theta)]^2$$

where  $n$  is the number of data points,  $y_i$  is the observed value corresponding to  $x_i$ , and  $f(x_i, \theta)$  is the model's predicted value for  $x_i$ . The SSE quantifies the total squared deviation of the model's predictions from the observed data, with a smaller SSE indicating a better fit.

The optimization process begins with an initial estimate of the parameters  $\theta$ . The model then calculates predicted values  $f(x_i, \theta)$  for each data point  $x_i$ , and the SSE is computed from these predictions and the observed values  $y_i$ . The parameters  $\theta$  are iteratively adjusted to minimize the SSE, often using algorithms like gradient descent. This process continues until the SSE is minimized or converges to a stable value. The resulting parameters  $\theta$  minimize the SSE, providing the best possible fit of the model to the data.

In our specific application, the power model is defined by the function:

$$P_{fit}(x, a, k, b, v, q, c) = a + \left( \frac{k - a}{(c + q \exp(-bx))^{\frac{1}{v}}} \right) \quad (4.4.2.1)$$

where  $x$  represents the input variable (e.g., wind speed),  $a$  is the lower asymptote of the power curve,  $k$  is the upper asymptote,  $b$  controls the rate of exponential decay,  $v$  influences the shape of the curve, and  $q$  scales the exponential term (details can be found in APPENDIX). The constant  $c$  is set to 1 to normalize the base of the exponential component (Michos et al., 2024b), simplifying the function to :

$$P_{fit}(x, a, k, b, v, q) = a + \left( \frac{k - a}{(1 + q \exp(-bx))^{\frac{1}{v}}} \right),$$

To find the optimal parameter values, the `curve_fit` function from Python's optimization library is employed. This function performs non-linear least squares optimization, adjusting the parameters of the power model to minimize the SSE between the observed WT power data and the model's predictions. By fitting the function to the WT power data, `curve_fit` generates a set of parameter values that best represent the observed power output as a function of wind speed. This approach ensures that the model closely aligns with real-world data, allowing for accurate power output predictions based on varying wind conditions.

### **Enercon**

For the first WT (Enercon), the optimal parameters for the power function were determined using the least squares fitting procedure described earlier. The function  $P_{fit}$  with  $c = 1$ , is simplified by normalizing the base of the exponential term. The fitting process utilized wind speed data ( $x$ ) and corresponding power output data ( $y$ ), both of which were pre-processed to remove any missing values and ensure alignment between the datasets. The curve fitting was performed using the Python function `curve_fit`, and the results of this optimization provided the following parameter estimates:

$$\begin{aligned}a &= -53.35 \text{ kW}, \\k &= 514.48 \text{ kW}, \\b &= 0.6625, \\v &= 2.173, \\q &= 2107.87.\end{aligned}$$

These values represent the coefficients that best fit the observed data for the Enercon WT. Specifically, the value of parameter  $a$  corresponds to the lower asymptote, indicating the minimum power output level as wind speed approaches zero is violating power limitations. The upper asymptote, represented by  $k$ , gives a slightly higher maximum power output level as wind speed increases to the WT's rated value than the manufacturer. The value of  $a$  is a concern that we will discuss at NEG WT power estimations.

The covariance matrix associated with the parameter estimates, which reflects the uncertainties and correlations between the parameters, is given by:

$$\text{Cov}([a, k, b, v, q]^T) = \begin{bmatrix} 25.23 & 2.07 & -0.047 & -0.554 & -1724.48 \\ 2.07 & 0.900 & -0.0097 & -0.0732 & -305.86 \\ -0.047 & -0.0097 & 0.000166 & 0.00144 & 5.54 \\ -0.554 & -0.0732 & 0.00144 & 0.0144 & 50.10 \\ -1724.48 & -305.86 & 5.54 & 50.10 & 187222.69 \end{bmatrix}$$

This matrix provides insights into the precision of the estimated parameters, with smaller variances indicating more precise estimates. Additionally, off-diagonal elements represent the covariances between pairs of parameters, showing how changes in one parameter may affect the others (details can be found in APPENDIX).

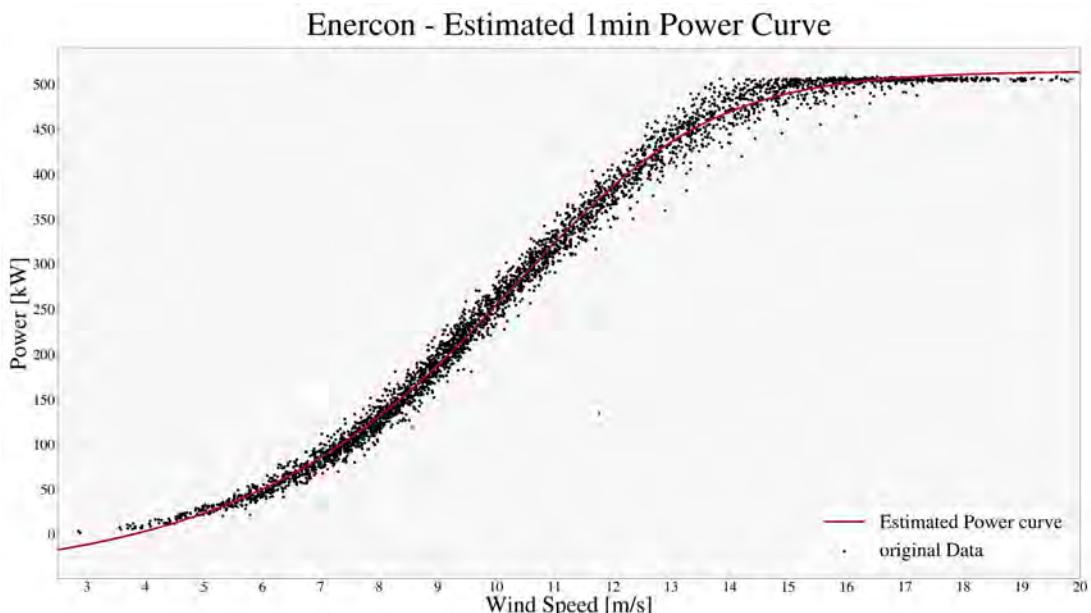


Figure 4.20: Enercon Power curve scatterplot

The fitting process ensures that the resulting power function closely approximates the actual power output of the WT across the observed range of wind speeds, capturing the key operational characteristics of the Enercon WT. This model serves as a simple but reliable tool for estimating optimal ShiVa by approximating the power output based on varying wind conditions, which is crucial for optimizing the performance and energy efficiency of wind energy systems.

## NEG

Continuing with the analysis for the second WT (NEG), the optimal parameters for the power function were determined using the same least squares fitting approach. By setting  $c = 1$  to normalize the exponential term, the model function  $P_{fit}$  was fitted to the observed data of wind speed and corresponding power output for the NEG WT. The estimated parameters for the NEG WT are as follows:

$$a = -153.66 \text{ kW},$$

$$k = 700.96 \text{ kW},$$

$$b = 0.7242,$$

$$v = 3.169,$$

$$q = 3337.68.$$

In this context,  $a$  denotes the lower asymptote, representing the minimum power output at low wind speeds. The upper asymptote  $k$  indicates the maximum power output achievable by the WT. The rate of exponential decay is captured by  $b$ , which affects how rapidly the power output increases as wind speed rises. The parameter  $v$  controls the curvature of the power response, and  $q$  scales the exponential component's influence. The associated covariance matrix, reflecting the precision and interaction of the parameter estimates, is:

$$\text{Cov}([a, k, b, v, q]^T) = \begin{bmatrix} 95.01 & 5.38 & -0.106 & -1.554 & -5559.35 \\ 5.38 & 1.461 & -0.0138 & -0.133 & -636.73 \\ -0.106 & -0.0138 & 0.000206 & 0.00227 & 10.01 \\ -1.554 & -0.133 & 0.00227 & 0.0288 & 114.55 \\ -5559.35 & -636.73 & 10.01 & 114.55 & 490881.42 \end{bmatrix}$$

It is worth noting that the lower Power Asymptote parameter  $a$ , is in both cases below zero, violating Energy value limits. In our final step where the curves will be computed from 10 minute MA wind speed values and 15 minute MA Power values, the violation can be fixed by limiting values to zero as a minimum an preferred way. In cases where the dataset range does not produce power outputs that violate the zero-limit, the power curve can be used without limitations.

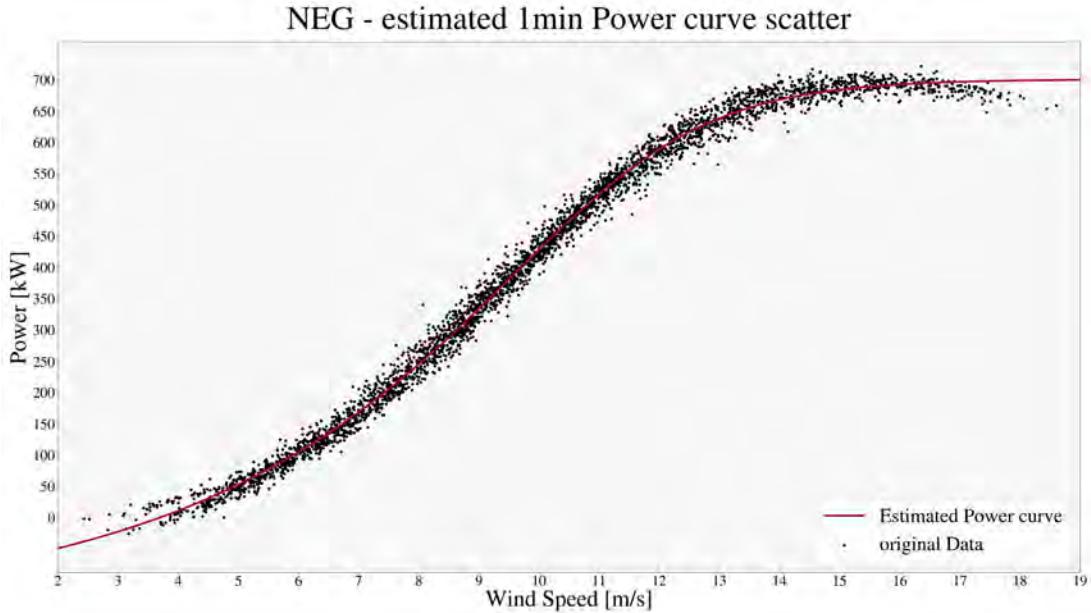


Figure 4.21: Enercon Power curve scatterplot

This fitting approach ensures that the derived power function accurately reflects the NEG WT's behavior under varying wind conditions, making it a useful tool for modeling power output based on wind speed. Understanding the problem that is being modeled is essential for optimizing WT performance and enhancing the overall efficiency of wind energy production systems.

#### 4.4.3 optimal and maximum Shift detection

For our experiment (Michos et al., 2024b), we focus on 15-minute power production forecasting due to its significance in providing WFs a reasonable timeframe for operational adjustments and offering energy traders an opportunity to strategize. This focus defines the averaging window for power production. We employ a 10-minute averaging window for the wind speed MAs, matching the already extracted 10 min wind speed WiSpEx results. 10-minute MAs are both slightly faster and more volatile, a characteristic that can be particularly advantageous during ramp events.

Our validation process utilizes WTs' 10-minute MAs of wind speed measurements and 15-minute power measurements of 1-minute resolution to pinpoint the maximum ShiVa offset of wind speed series that minimizes the standard deviation with respect to their fitted power curve. By employing different averaging window sizes for wind speed and power calculations we exploit the inherent trade-off between volatility and trend robustness

(Michos et al., 2024b). The 10-minute MA wind speed, while exhibiting higher volatility, captures trend changes more rapidly. Conversely, the 15-minute MA power output smooths out fluctuations but requires a slightly longer duration to reflect the underlying trend.

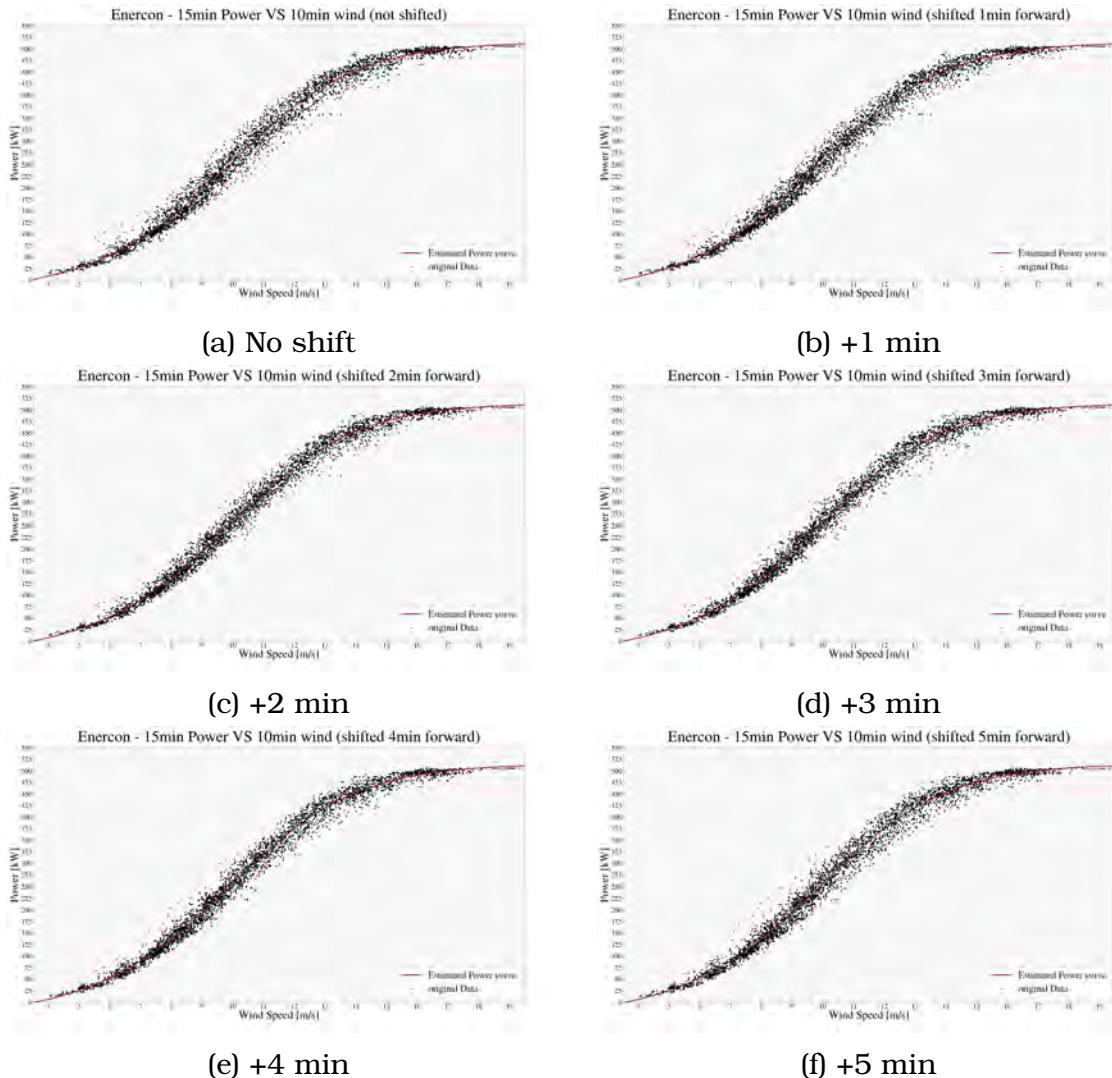


Figure 4.22: Enercon 15min Power Curves from shifted 10min wind measurements (a,d,f presented in Michos et al., 2024b)

By identifying the WT "power lag" via visual scatter plot inspection (figures 4.22 and 4.23) and different error metrics (table 4.5 and 4.6), the volatility inherent in the wind speed data can be effectively mitigated. This is achieved by shifting the wind speed series forward by the determined optimal or maximum ShiVa. Once the optimal ShiVa is established, the estimation of 15-minute power production with a max ShiVa-minute lead time is possible and the final power curve can be selected. This methodology

proves particularly valuable for capturing power production during rapid changes in wind speed like ramp events, without incurring a significant computational burden.

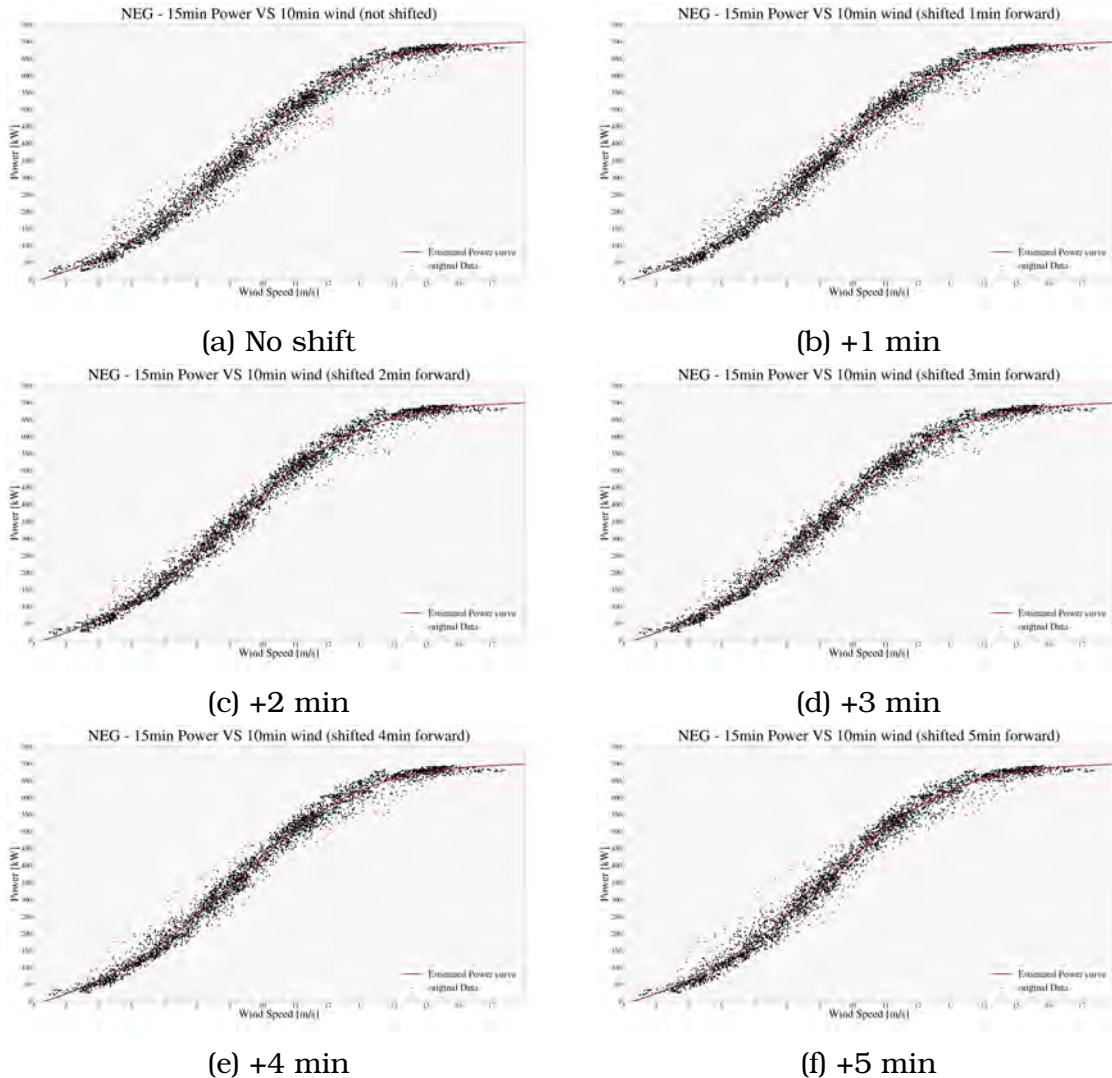


Figure 4.23: NEG 15min Power Curves from shifted 10min wind measurements (a,d,f presented in Michos et al., 2024b)

The utilization of 10-minute MAs for wind speed series and 15-minute MAs for wind power output effectively exposes the inherent "power lag", as depicted in Figures 4.22 and 4.23. These figures give a strong indication of the possible optimal shift, as the graphs for both WTs pinpoint that the values at 2 and 3 minute shift align better with the approximated power curve.

An inspection of tables 4.5 and 4.6 validates the illustration of the scatter plots, giving us plenty of measures that can be used to get a data

driven decision. As depicted in both tables, the optimal ShiVa for both the Enercon and NEG WTs is +2 minutes, demonstrating the lowest MAE, MSE, and RMSE values, coupled with the highest  $R^2$  values at this shift.

Table 4.5: WEEL - Enercon shift stats (part of table presented in Michos et al., 2024b)

Metric	+0 min shift	+1 min shift	+2 min shift	+3 min shift	+4 min shift	+5 min shift
MAE	12.961	11.577	10.838	10.675	11.129	12.353
MSE	293.660	231.483	204.463	201.226	220.992	274.580
RMSE	17.136	15.214	14.299	14.185	14.865	16.570
MAPE	6.342	5.664	5.317	5.225	5.416	5.988
$R^2$	0.986	0.989	0.990	0.990	0.990	0.987

Furthermore, the maximum allowable shift, defined as the largest future shift that maintains the closest metric values to the unshifted dataset, is consistently observed at +5 minutes for both WTs. This suggests that a 5-minute shift is within the acceptable range of the WEEL assumption, providing a balance between capturing the power lag effect and maintaining forecast accuracy.

Interestingly, the Enercon WT appears to exhibit a larger "power lag" compared to the NEG WT. This discrepancy might be attributed to differences in their sizes, aerodynamic and engineering design, turbulence, as well as potential errors in power curve modeling. The power curve produced errors will be discussed in the next section while the other possible error sources must be investigated by the WT manufacturers.

Table 4.6: WEEL - NEG shift stats (part of table presented in Michos et al., 2024b)

Metric	+0 min shift	+1 min shift	+2 min shift	+3 min shift	+4 min shift	+5 min shift
MAE	18.385	16.930	16.384	16.490	17.179	18.653
MSE	614.732	511.373	470.061	472.260	518.871	629.909
RMSE	24.793	22.613	21.680	21.731	22.778	25.097
MAPE	7.561	6.872	6.601	6.652	6.988	7.74
$R^2$	0.985	0.987	0.988	0.988	0.987	0.985

Finally, WEEL uses WiSpEx to extrapolate LIDAR wind speed measurement to the WTs' hub height and get the 10 min averaged wind speed forecasts (Michos et al., 2024b). The resolution of WiSpEx wind speed series is enhanced with simple linear interpolation to 5 minutes while the resolution of the max ShiVa shifted 10MA values is downgraded to 5 minutes. The ShiVa shifted WiSpEx 10 min ahead wind speed can be directly

converted to Power production via the aforementioned approximation for each WT power curve. The 10-minute wind speed forecast derived from CFD simulations, coupled with the identified maximum shift, serves as the basis for an initial estimate of the 15-minute power production.

In its simplest form, the final WEEL 15 minutes ahead Power forecast is obtained by averaging the +5 shifted 10 min measured wind speed and the WiSpEx 10-minute CFD forecast series(shifted by +15min). By averaging these two series, we effectively incorporate the momentum of the WT and possible persistence, resulting in WEEL forecasted 10 minutes averaged wind speed. The WEEL generated wind speed is converted to 15min-Power by applying the fitted power curve to obtain the final 15 minutes ahead forecasted Power output for the WTs. This version of WEEL could be identified as a hybrid CFD-momentum model which accounts for wind focused persistence (Michos et al., 2024b).

#### 4.4.4 Results and Discussion

In this section, we present and analyze the results obtained from WEEL model (Michos et al., 2024b). Many ultra-short-term wind Power forecasting models are being published the last few years and almost all of them are statistical models like the ones presented by L. Liu et al., 2023; Niu et al., 2022; Xiang et al., 2022 and Zhang et al., 2021. Most of the models available present their results in terms of error reduction and not with direct error evaluations. Most of the times scientists use Symmetric Mean Absolute Percentage Error (SMAPE) instead of MAPE to reduce extreme error significance and normalized values like Normalized Root Mean Square Error (NRMSE) and Normalized Mean Absolute Error (NMAE) to generalize model behavior:

$$\text{NMAE} = \frac{\text{MAE}}{P_{\text{rated}}} \quad (4.4.4.1)$$

$$\text{NRMSE} = \frac{\text{RMSE}}{P_{\text{rated}}} \quad (4.4.4.2)$$

$$\text{SMAPE} = \frac{1}{N} \sum_{i=1}^N \frac{|y_i - \hat{y}_i|}{\frac{|y_i| + |\hat{y}_i|}{2}} 100\% \quad (4.4.4.3)$$

,where  $P_{\text{rated}}$  is the rated or max power of a given WT,  $y_i$  is the forecast value and  $\hat{y}_i$  is the actual value.

## Enercon

### A. Error Analysis

Table 4.7 provides a detailed overview of how well the WEEL method performs in the prediction of power output at different time steps. A key takeaway is the consistently low MAPE, which stays below 10% for all prediction horizons, indicating sufficient accuracy.

Table 4.7: Enercon 15 minute Power Forecast Errors stats for WEEL at different Time Steps (part of table presented in Michos et al., 2024b)

Method	15min MA (Measurements)	W.E.E.L. (10min MA)	Interpolated (from adjacent steps)	W.E.E.L. (Wi.Sp.Ex.)	W.E.E.L. (Wi.Sp.Ex.)
MAE [kW]	5.06	12.23	12.96	17.81	18.43
MSE [kW <sup>2</sup> ]	45.18	271.15	294.81	532.48	589.55
RMSE [kW]	6.72	16.47	17.17	23.08	24.28
MAPE [%]	2.4	5.91	6.1	8.44	8.84
SMAPE [%]	2.39	5.88	5.99	8.18	8.55
time step	live	+5 min	+10 min	+15 min	+20 min

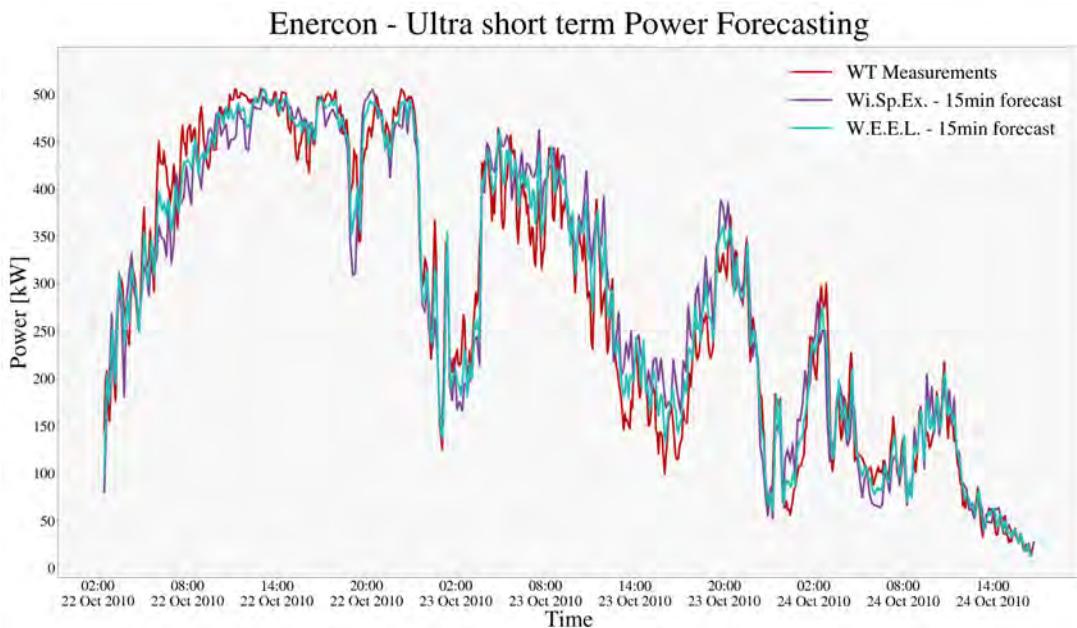


Figure 4.24: Time series of 15-minute power forecasts for the Enercon WT (Michos et al., 2024b).

The table also highlights the performance of WEEL for the first 15min forecast step, as visually represented in Figure 4.24 and accompanied by the direct WiSpEx equivalent forecast, assuming a 10min phase latency due to airflow travel distance, shifted by the maximum ShiVa. This figure

emphasizes specific areas where power curve conversion errors tend to be higher (shown by the red line), which affects the overall accuracy of the model's predictions.

Figures 4.25 and 4.26 provide a deeper look into the errors associated with power prediction. Figure 4.25 showcases the errors stemming from the power curve itself, while Figure 4.26 compares the errors of the WiSpEx and WEEL. These figures suggest that we can further understand and reduce some of the power curve produced errors. This could be done by creating more accurate models that combine advanced mathematical techniques, high-quality meteorological data, and the fundamental physics related to WTs, similarly to the findings of Saint-Drenan et al., 2020. For instance, the figures reveal that power conversion errors tend to increase at higher wind speeds, indicating a potential area for model refinement.

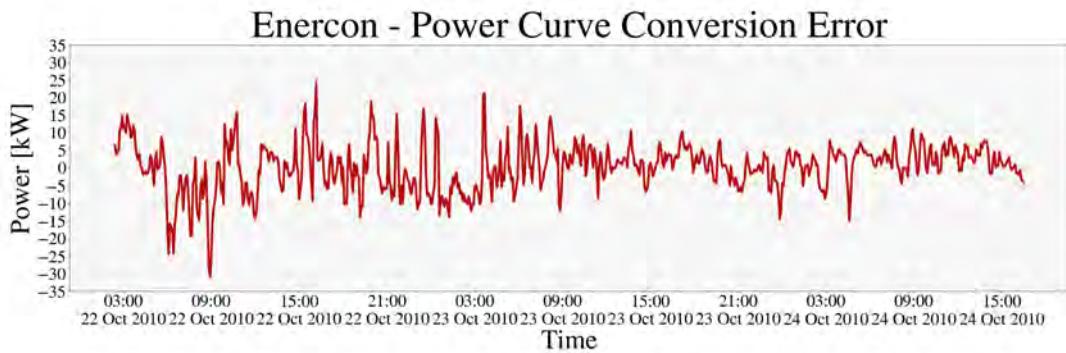


Figure 4.25: Errors in 15-minute power conversion from measured wind speed for the Enercon WT (Michos et al., 2024b).

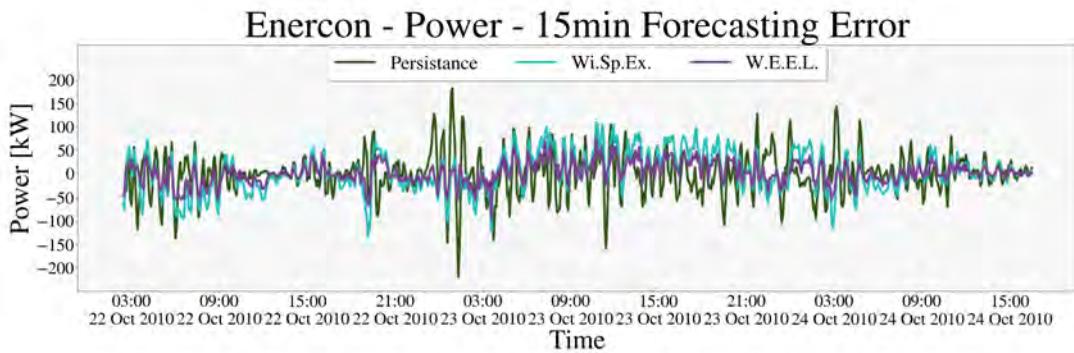


Figure 4.26: Comparison of 15-minute power forecast errors between WiSpEx and WEEL for the Enercon WT (Michos et al., 2024b).

A closer examination of the forecasting results reveals intriguing patterns in the error distribution. High horizontal wind speeds with positive gradients (increasing wind speeds) tend to lead to underestimation of power

output. In contrast, negative gradients (decreasing wind speeds) often result in overestimation. On the other hand, low wind speed conditions generally exhibit more stable and predictable behavior, resulting in reduced bias in power predictions.

The variability of errors is significantly impacted by rapid changes in airflow, particularly during ramp events where wind speed increases or decreases sharply. Interestingly, the highest error variance is observed at mid-range wind speeds. This is likely due to the complex interplay of wind speed magnitude and turbulence across various scales in this regime.

Figures 4.27 and 4.28 visually illustrate the absolute and absolute percentage errors for WiSpEx and WEEL, respectively. These figures clearly demonstrate that the power estimation error is influenced by wind speed, with a noticeable decrease in error at both nominal (rated) and low wind speeds. Notably, Figure 4.28 underscores the increase in MAPE at lower wind speeds. This behavior is expected, as even small absolute errors can lead to large percentage errors when the actual power output is low, highlighting the challenges of accurate power prediction in low wind energy regimes. This phenomenon is also connected to the inherent physical constraints imposed by the fundamental concept of energy.

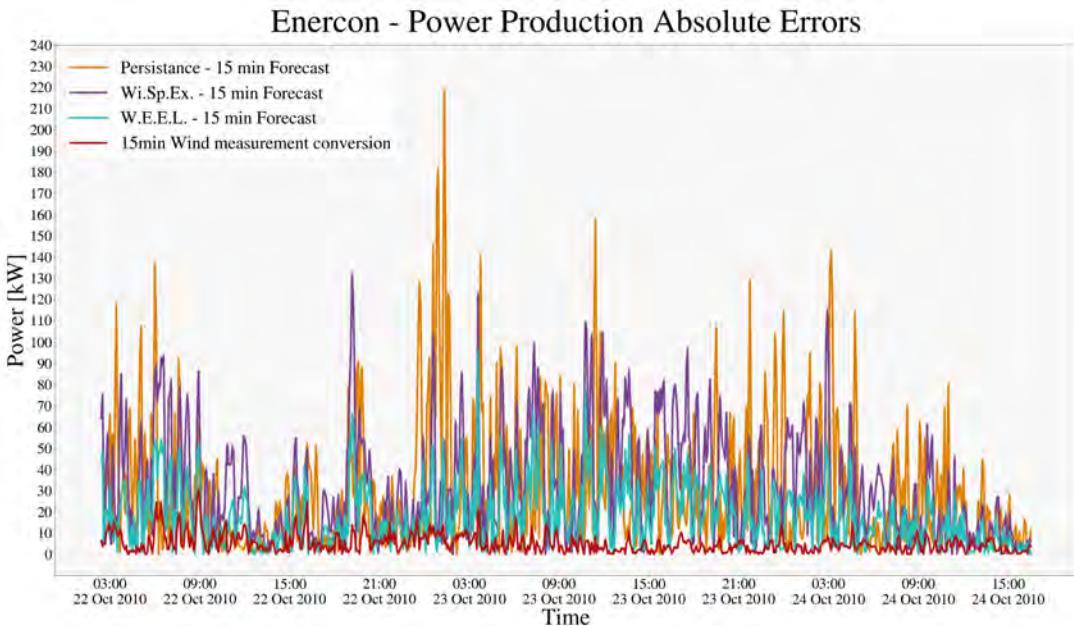


Figure 4.27: Absolute errors (Michos et al., 2024b)

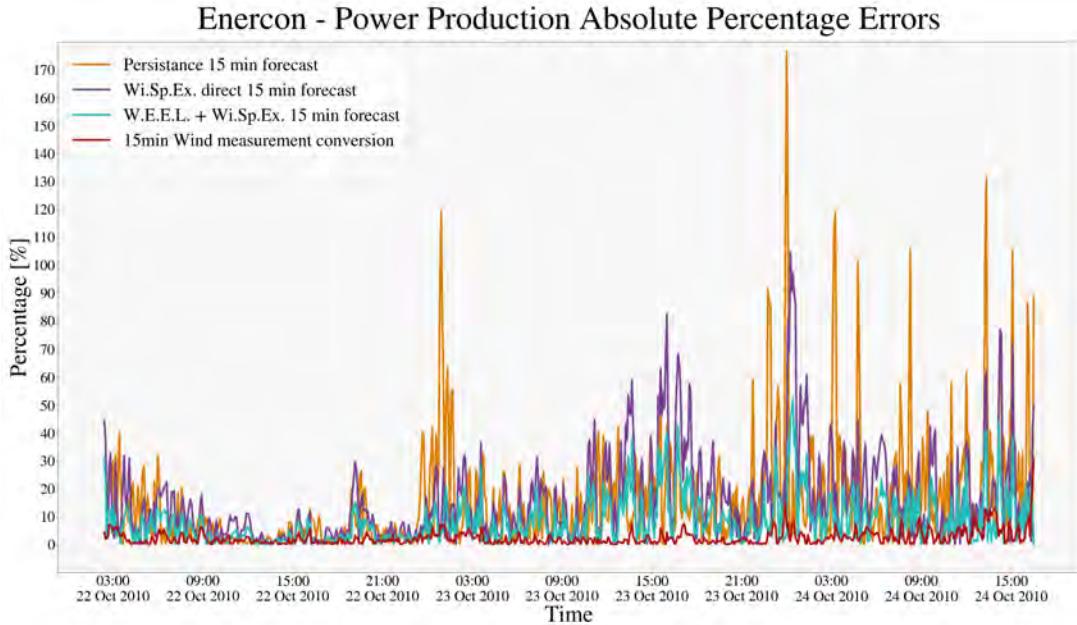


Figure 4.28: Absolute Percentage errors (Michos et al., 2024b)

Table 4.8 provides the summary statistics for different types of forecasting methods grouped by their forecasting steps. By accounting for persistence, WEEL coupled with WiSpEx outperforms all the other models according to all metrics. Table 4.9 gives insights on the absolute error behaviour, underscoring the error reduction induced by the usage of WEEL and the usage of ShiVa.

Both MAPE and SMAPE are widely used metrics for evaluating the accuracy of forecasting models, particularly in the context of time series data. They express the average percentage deviation between the predicted values and the actual values, thus providing a readily interpretable measure of forecast accuracy. However, a key distinction lies in their handling of low or zero actual values. MAPE becomes undefined or highly sensitive to outliers when actual values approach zero, potentially leading to misleading interpretations of model performance. SMAPE addresses this limitation by introducing a symmetric formulation that remains well-behaved even in the presence of low or zero actual values.

In the context of wind power forecasting, where power output can be low or even zero during certain periods, SMAPE offers a more robust and reliable metric for evaluating model performance. It mitigates the potential distortions that MAPE might exhibit in such scenarios. While both MAPE and SMAPE have their merits, the choice between them depends on the specific characteristics of the data and the potential presence of low or

zero values. In cases where such values are prevalent, SMAPE is often preferred due to its greater resilience and stability.

A comparative analysis of the MAPE values for the 15-minute ahead forecast reveals the superior performance of WEEL, with a MAPE of 8.44%. This is in contrast to the persistence forecasting model, which exhibits a MAPE of 16.01% (almost double). The significantly lower MAPE value associated with WEEL underscores its enhanced accuracy in providing power predictions compared to the persistence model. While there is a slight increase in MAPE compared to the 5-minute power latency forecast (5.91%), WEEL's 15-minute ahead forecast offers a valuable advantage for grid management and energy trading operations. The ability to accurately predict power output 15 minutes in advance allows for more effective planning and optimization of grid resources, ultimately contributing to improved grid stability and efficiency.

Tables 4.11 and 4.10 provide a comprehensive summary of the SMAPE and the conventional MAPE for the Enercon WT, respectively. The WT under consideration is strategically located at the site illustrated in Figure 4.2. These tables meticulously document absolute percentage error behaviour across different forecasting lead times and methodologies, unveils several key findings.

Further reinforcing WEEL's robust performance, the SMAPE, as presented in Table 4.10, exhibits a similar trend with MAPE. For the 15-minute ahead forecast, WEEL achieves an SMAPE of 8.18%, notably lower than the persistence model's SMAPE of 15.03%. This consistency across both MAPE and SMAPE metrics further solidifies WEEL's superior accuracy and reliability in predicting wind power generation, even in scenarios with potentially low or zero power output.

## **B. Evaluation of WEEL at different forecasting steps**

Examining the distribution of errors provides further evidence of WEEL's enhanced performance. For the 15-minute ahead forecast, Table 4.9 shows that WEEL's 90th percentile absolute error is 39.29 kW, significantly lower than the persistence model's 71.30 kW. This indicates that WEEL is more effective at limiting large errors, a critical factor for grid stability and energy trading. Similarly, Table 4.10 reveals that WEEL's 90th percentile MAPE is 19.04%, compared to 33.61% for persistence. The corresponding SMAPE values at the 90th percentile are 18.60% for WEEL and 34.57% for persistence.

The 5-minute ahead forecast, obtained directly by WEEL, leveraging a 10-minute wind speed magnitude MA, demonstrates a MAE of 12.23 kW. This result is notably better than the persistence forecast's MAE of 15.32 kW, establishing WEEL's effectiveness at this short-term horizon. The effectiveness of WEEL is further underscored by its lower RMSE of 16.47 kW, compared to persistence's 21.21 kW.

As we extend the prediction window to 10 minutes, WEEL, predicts the Power production by converting wind speed interpolated values from the adjacent time windows, maintaining its advantage over the persistence model. WEEL's MAE of 12.97 kW and RMSE of 17.18 kW are considerably lower than the persistence forecast's 24.85 kW and 34.47 kW, respectively. Additionally, combining WEEL with WiSpEx at the 10-minute mark yields a MAE of 17.75 kW and an RMSE of 22.96 kW. Although slightly higher than WEEL with wind speed interpolation, these values still represent a significant improvement over the persistence forecast.

At the critical 15-minute forecasting horizon, WEEL continues to deliver the best performance. It achieves a MAE of 17.81 kW and an RMSE of 23.08 kW, once again outperforming the persistence model's MAE of 31.82 kW and RMSE of 44.03 kW. This trend of superior accuracy is maintained even at the 20-minute forecast horizon, with WEEL's MAE of 18.43 kW and RMSE of 24.28 kW being substantially lower than the persistence model's 34.62 kW and 48.20 kW.

The 15-minute time step is particularly important due to its practical relevance for grid operators and energy traders, offering a balance between short-term precision and actionable lead time for decision-making. In this context, WEEL coupled with WiSpEx emerges as the clear front-runner, achieving an impressive MAPE of 8.44%. This translates to an average forecast deviation of just 8.44% from the actual power output. Further analysis of the percentiles in Table 4.9 shows that even at the 90th percentile, the absolute error is contained within 39.29 kW. This implies that 90% of the forecasts exhibit an error margin less than 39.29 kW, reinforcing the model's reliability.

A consistency observed throughout WEEL's performance across all time steps, is its ability to outperform persistence forecasting. This is a noteworthy accomplishment, as persistence, which simply assumes future power output will mirror the current output, serves as a common benchmark for comparison. WEEL's ability to not only surpass persistence but also enhance the specifically employed WiSpEx model underscores its considerable value as a forecasting tool.

Table 4.8: Enercon 15 minute Power Forecast Errors stats for Various Forecast Methods and Time Steps

Method	15min MA	10min MA	W.E.E.L. (Direct)	Persistence	W.E.E.L. Power	W.E.E.L. Wind	W.E.E.L. Interpolated	W.E.E.L. Interpolated (Wi.Sp.Ex.)	W.E.E.L. (Wi.Sp.Ex.)	W.E.E.L. (Wi.Sp.Ex.)	W.E.E.L. (Wi.Sp.Ex.)	W.E.E.L. (Wi.Sp.Ex.)
MAE [kW]	5.06	12.85	12.23	15.32	12.96	12.97	17.75	32.46	24.85	17.81	31.58	18.43
MSE[kW <sup>2</sup> ]	45.18	291.6	271.15	450.05	294.81	295.3	527.35	1752.01	1188.44	532.48	1643.36	589.55
RMSE [kW]	6.72	17.08	16.47	21.21	17.17	22.96	41.86	34.47	23.08	40.54	44.03	24.28
MAPE [%]	2.4	6.31	5.91	7.41	6.1	8.35	15.66	12.2	8.44	15.36	16.01	8.84
NMAE	0.0101	0.0257	0.0245	0.0306	0.0259	0.0260	0.0355	0.0497	0.0356	0.0632	0.0636	15.96
NRMSE	0.0134	0.0342	0.0330	0.0424	0.0343	0.0344	0.0459	0.0837	0.0462	0.0811	0.0881	0.0651
SMAPE [%]	2.39	6.35	5.88	7.29	5.99	6.0	8.07	14.84	11.72	8.18	14.51	15.03
time step	live	live	+5 min	+5 min	+10 min	+10 min	+10 min	+10 min	+10 min	+15 min	+15 min	+20 min
												+20 min
												+20 min

Table 4.9: Enercon 15 minute Power Forecast Absolute Errors [kW] for Various Forecast Methods and Time Steps

Metric	15min MA	10min MA	W.E.E.L. (Direct)	Persistence	W.E.E.L. Power	W.E.E.L. Wind	W.E.E.L. Interpolated	W.E.E.L. Interpolated (Wi.Sp.Ex.)	W.E.E.L. (Wi.Sp.Ex.)	W.E.E.L. (Wi.Sp.Ex.)	W.E.E.L. (Wi.Sp.Ex.)	W.E.E.L. (Wi.Sp.Ex.)	
Mean	5.06	12.85	12.23	15.32	12.96	12.97	17.75	32.46	24.85	17.81	31.58	18.43	
Std	4.42	11.25	11.03	14.68	11.27	11.28	14.58	26.45	23.92	14.69	25.44	30.46	
Min	0.02	0.16	0.01	0.01	0.03	0.04	0.04	0.09	0.00	0.01	0.06	0.05	
10%	0.57	1.74	1.67	1.36	1.84	1.81	2.18	4.22	2.23	2.28	4.34	3.75	
20%	1.41	3.57	3.20	3.16	3.47	3.54	4.86	9.73	5.68	4.34	8.23	7.36	
30%	2.22	5.20	5.02	5.48	5.22	5.10	7.28	13.99	9.39	7.41	13.03	11.63	
40%	3.16	7.07	6.67	7.85	7.22	7.40	10.51	19.48	12.96	10.86	18.83	16.41	
50%	3.99	9.58	9.24	10.72	9.81	9.82	14.37	25.90	17.48	14.52	26.08	22.90	
75%	7.02	18.68	17.42	22.33	18.57	18.52	26.68	47.59	36.56	26.35	48.35	43.61	
90%	10.70	28.01	26.60	36.49	28.37	28.17	39.37	70.90	55.13	39.29	70.47	71.30	
95%	13.77	34.85	33.95	46.59	35.16	35.33	45.28	83.35	73.36	47.41	78.66	90.99	
Max	30.84	64.21	69.85	81.39	83.11	83.11	77.24	146.03	156.88	96.38	132.05	218.79	88.75
Time step	live	live	+5 min	+5 min	+10 min	+10 min	+10 min	+10 min	+10 min	+15 min	+15 min	+20 min	+20 min
												+20 min	+20 min
												+20 min	+20 min

Table 4.10: Enercon 15 minute Power Forecast Absolute Percentage Errors [%] for Various Forecast Methods and Time Steps

Metric	15min MA	10min MA	W.E.E.L. (Direct)	W.E.E.L. Power Interpolated	Wind Interpolated (Wi.Sp.Ex.)	W.E.E.L. Wind Interpolated (Wi.Sp.Ex.)	W.E.E.L. Wi.Sp.Ex. Persistence	W.E.E.L. Wi.Sp.Ex. Persistence	W.E.E.L. Wi.Sp.Ex. Persistence	W.E.E.L. Wi.Sp.Ex. Persistence	W.E.E.L. (Wi.Sp.Ex.)
<b>mean</b>	2.4	6.31	5.91	7.41	6.11	8.35	15.66	12.2	8.44	15.36	16.01
<b>std</b>	2.65	6.57	6.17	8.16	6.14	8.24	16.23	14.41	8.34	15.77	8.93
<b>min</b>	0.01	0.04	0.01	0.0	0.01	0.01	0.03	0.0	0.02	0.03	0.01
<b>10%</b>	0.25	0.6	0.49	0.6	0.75	0.76	1.7	0.93	0.87	1.14	0.87
<b>20%</b>	0.55	1.13	1.2	1.17	1.28	1.32	1.65	3.54	2.06	2.91	1.8
<b>30%</b>	0.85	1.95	1.89	2.13	1.98	2.77	5.41	3.55	2.89	4.89	2.82
<b>40%</b>	1.18	3.03	2.72	3.21	2.95	4.18	7.81	5.54	4.22	7.8	7.27
<b>50%</b>	1.54	4.14	3.8	4.65	4.15	6.01	10.82	7.41	5.83	10.75	9.6
<b>75%</b>	3.17	9.19	8.31	10.4	8.56	11.73	21.44	17.51	11.86	21.28	12.39
<b>90%</b>	5.64	14.89	13.95	16.83	13.94	14.01	18.68	35.82	27.23	19.04	35.68
<b>95%</b>	7.13	19.04	18.35	22.79	19.03	19.08	25.5	35.32	24.11	44.88	33.61
<b>max</b>	30.51	59.5	44.18	76.79	40.03	60.13	114.65	128.08	52.56	104.59	176.67
<b>Time step</b>	live	live	+5 min	+5 min	+10 min	+10 min	+10 min	+10 min	+15 min	+15 min	+20 min
											+20 min
											+20 min

Table 4.11: Enercon 15 minute Power Forecast Symmetrical Absolute Percentage Errors [%] for Various Forecast Methods and Time Steps

Metric	15min MA	10min MA	W.E.E.L. (Direct)	W.E.E.L. Power Interpolated	Wind Interpolated (Wi.Sp.Ex.)	W.E.E.L. Wind Interpolated (Wi.Sp.Ex.)	W.E.E.L. Wi.Sp.Ex. Persistence	W.E.E.L. Wi.Sp.Ex. Persistence	W.E.E.L. Wi.Sp.Ex. Persistence	W.E.E.L. Wi.Sp.Ex. Persistence	W.E.E.L. (Wi.Sp.Ex.)
<b>mean</b>	2.39	6.35	5.88	7.29	5.99	6.0	8.07	14.84	11.72	8.18	14.51
<b>std</b>	2.73	7.07	6.22	7.63	5.87	5.89	7.45	13.77	12.27	7.64	13.29
<b>min</b>	0.01	0.04	0.01	0.0	0.01	0.01	0.03	0.0	0.02	0.0	0.03
<b>10%</b>	0.25	0.6	0.49	0.75	0.76	0.76	1.7	0.92	0.87	1.14	1.14
<b>20%</b>	0.55	1.12	1.21	1.17	1.28	1.33	1.66	3.56	2.07	1.8	2.7
<b>30%</b>	0.86	1.95	1.89	2.11	1.98	1.97	2.81	5.5	3.57	2.9	4.93
<b>40%</b>	1.19	3.08	2.7	3.24	2.95	4.19	7.77	5.48	4.18	7.86	7.27
<b>50%</b>	1.54	4.15	3.83	4.68	4.13	4.17	6.05	10.75	7.46	5.75	10.81
<b>75%</b>	3.18	9.09	8.23	10.38	8.39	8.53	11.87	21.11	17.15	11.73	20.67
<b>90%</b>	5.58	14.86	13.34	17.19	13.62	13.61	18.36	34.76	27.47	18.6	33.78
<b>95%</b>	6.95	18.11	17.12	23.28	18.61	18.63	23.0	43.64	34.18	23.19	47.16
<b>max</b>	36.01	84.69	56.7	55.49	34.95	34.82	46.23	72.87	78.08	41.71	68.68
<b>Time step</b>	live	live	+5 min	+5 min	+10 min	+10 min	+10 min	+10 min	+15 min	+15 min	+20 min
											+20 min
											+20 min

## NEG

### A. Error Analysis

Table 4.12 provides a detailed overview of the performance of the WEEL method in predicting the power output of the NEG WT at various time steps. Similar to the Enercon WT, WEEL exhibits consistently low SMAPE, all remaining below 10% for all prediction horizons.

Table 4.12: NEG 15 minute Power Forecast Errors stats for WEEL at different Time Steps (part of table presented in Michos et al., 2024b)

Method	15min MA (Measurements)	W.E.E.L. (10min MA)	Interpolated (from adjacent steps)	W.E.E.L. (Wi.Sp.Ex.)	W.E.E.L. (Wi.Sp.Ex.)
MAE [kW]	8.15	18.48	18.99	24.93	26.13
MSE [kW <sup>2</sup> ]	121.24	603.92	611.47	1039.3	1158.82
RMSE [kW]	11.01	24.57	24.73	32.24	34.04
MAPE [%]	3.86	7.67	7.95	10.25	10.76
SMAPE [%]	3.71	7.53	7.44	9.26	9.75
time step	live	+5 min	+10 min	+15 min	+20 min

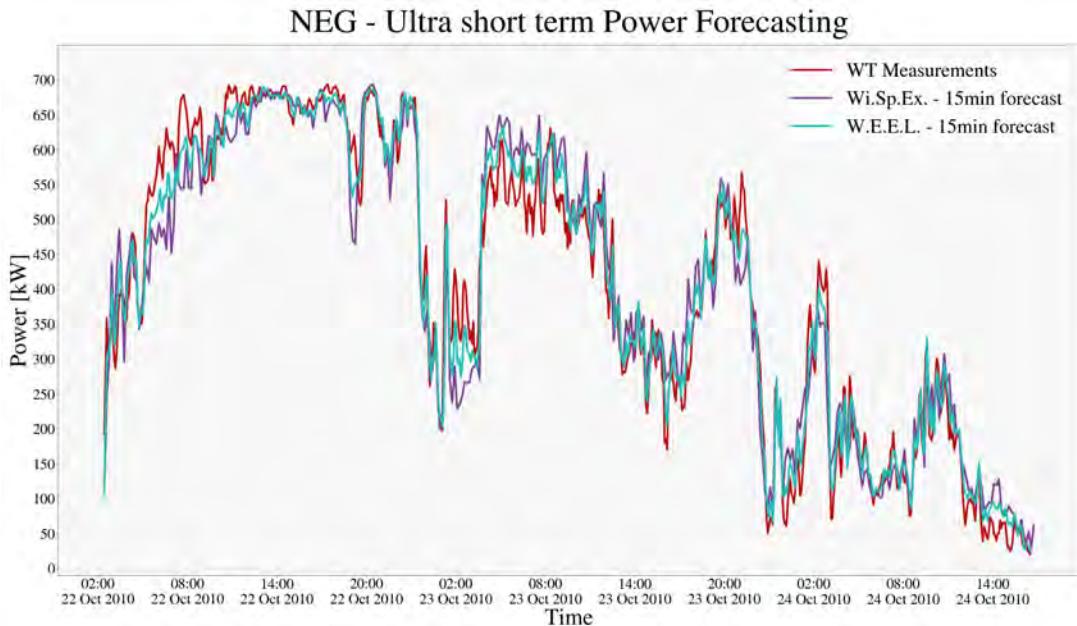


Figure 4.29: Time series of 15-minute power forecasts for the NEG WT (Michos et al., 2024b).

The table also highlights WEEL's performance for the crucial first 15-minute forecast step, which is visually represented in Figure 4.29. This figure, along with the direct WiSpEx equivalent forecasts (assuming a 10-minute phase latency due to airflow travel distance), pinpoints areas

within the power curve where conversion errors tend to be higher (shown by the red line), impacting the model's overall predictive accuracy.

Figures 4.30 and 4.31 offer a more granular view of the errors involved in power prediction. Figure 4.30 focuses on the errors inherent to the power curve itself, while Figure 4.31 provides a direct comparison between the errors of WiSpEx and WEEL. These figures suggest potential avenues for refining power curve models to enhance accuracy. For instance, they reveal that power conversion errors, particularly those associated with the power curve, tend to escalate at higher wind speeds, indicating a promising area for future model improvements.

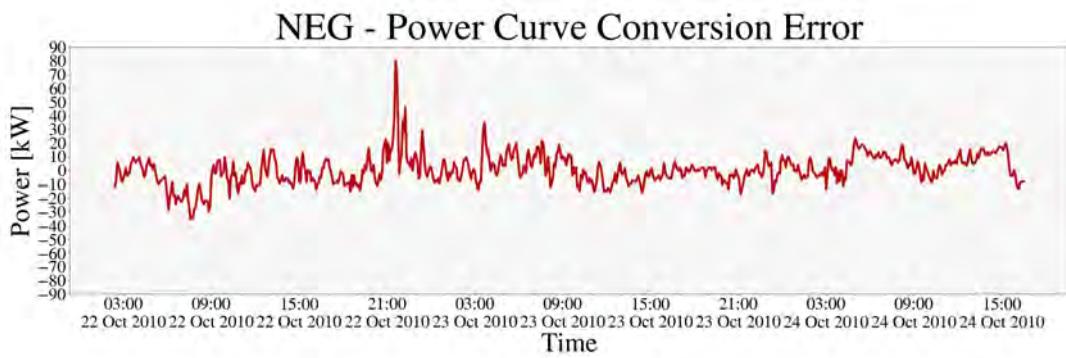


Figure 4.30: Errors in 15-minute power conversion from measured wind speed for the NEG WT (Michos et al., 2024b).

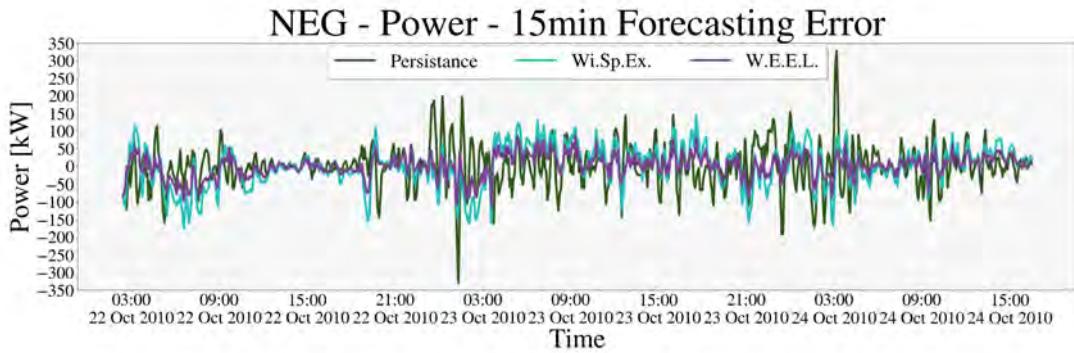


Figure 4.31: Comparison of 15-minute power forecast errors between WiSpEx and WEEL for the NEG WT (Michos et al., 2024b).

A closer examination of the forecasting results reveals intriguing patterns in the error distribution, similar to that of Enercon's. The differences observed are caused by the power curve approximation. The variability of errors is also significantly impacted by ramp events, and the highest error variance is observed at mid-range wind speeds for the same reasons with Enercon

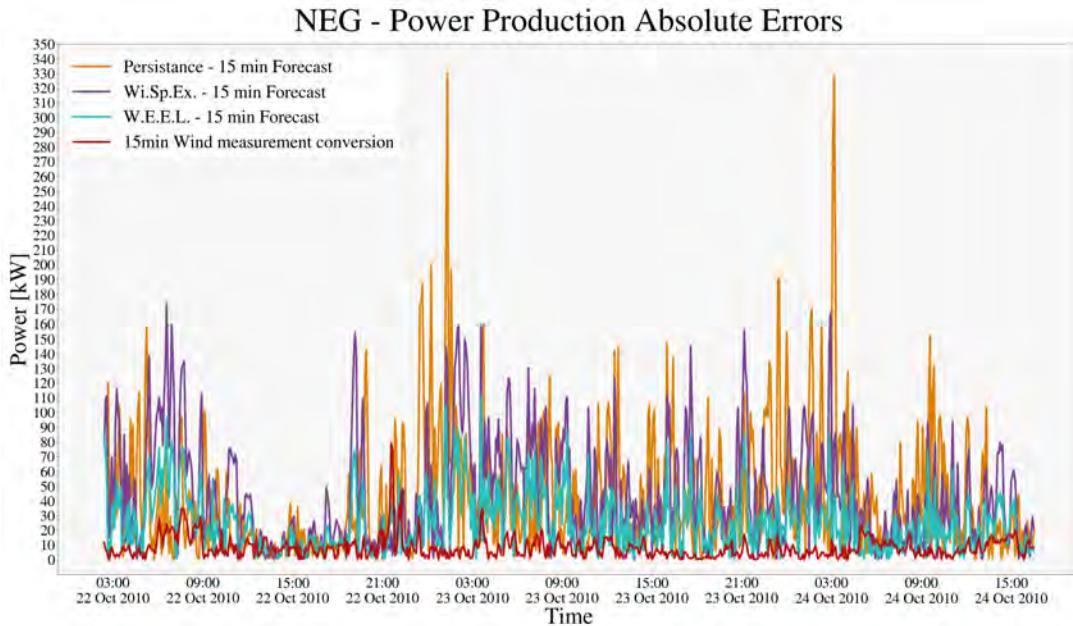


Figure 4.32: Absolute errors (Michos et al., 2024b)

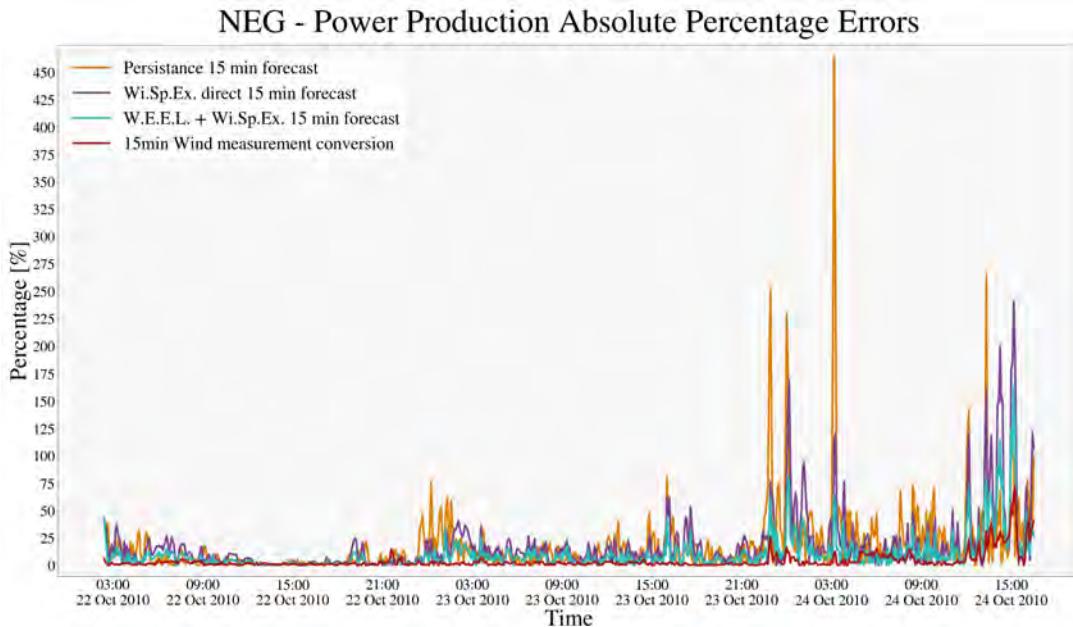


Figure 4.33: Absolute Percentage errors (Michos et al., 2024b)

Figures 4.32 and 4.33 visually illustrate the absolute and absolute percentage errors for WiSpEx and WEEL, respectively. These figures clearly demonstrate the influences by wind speed. Both figure illustrations come in alignment with figures 4.27 and 4.28.

Tables 4.13 and 4.14 allows us to compare the performance of WEEL against the baseline persistence model and WiSpEx, and provide insights for the distribution of errors across different percentiles, strengthening

WEEL's reliability and its ability to handle extreme events. Similar to the Enercon analysis, the 15-minute forecast horizon holds particular significance for real-world applications. WEEL, combined with WiSpEx, demonstrates its quality at this time step with a MAPE of 10.25% and an SMAPE of 9.26%. These figures indicate that, on average, WEEL's forecasts deviate from the actual power output by just over 10% on average, showcasing its strong predictive capability. Furthermore, an analysis of the quantiles in Table 4.14 reveals that even at the 90th percentile, the absolute error is limited to 54.12 kW, suggesting that 90% of WEEL's forecasts have an error margin below this value, demonstrating the model's reliability in providing power predictions.

Tables 4.16 and 4.15 present absolute and symmetrical absolute percentage errors, respectively, offering further insights into the model's forecasting accuracy, similarly to the Enercon's results. By examining these tables, we can gain insights into the distributions for MAPE and SMAPE across various forecasting methods and time steps. Both MAPE and SMAPE demonstrate WEEL's consistent superiority over the persistence model, particularly at longer forecast horizons. Furthermore, the increase in MAPE and SMAPE at lower wind speeds, evident in Figure 4.33, highlights the effects of power conversion error that is inherited from the curve approximation itself. The percentile analysis of MAPE and SMAPE further strengthens the case for WEEL's reliability. For instance, at the 15-minute horizon, WEEL's 90th percentile MAPE is 20.53%, significantly lower than persistence's 42.34%. Similarly, the 90th percentile SMAPE for WEEL is 19.82%, compared to 39.93% for persistence. These results suggest that WEEL is not only accurate on average but also consistently produces reliable forecasts even in challenging scenarios.

## B. Evaluation of WEEL at different forecasting steps

At the 5 minute ahead step, WEEL, utilizing a 10-minute MA for wind speed magnitude, exhibits a MAE of 18.48 kW, showcasing its improved accuracy compared to the persistence forecast's MAE of 20.75 kW. This advantage is further emphasized by WEEL's lower RMSE of 24.57 kW compared to persistence's 29.04 kW. The MAPE of WEEL is only 6.58% smaller compared to persistence but this is expected due to the small step we are evaluating. This percentage error reduction might seem significant but in a WF with many WTs this can represent an amount of Power that may be crucial for power storage optimization.

---

At the next step (+10 minutes), WEEL, with wind speed interpolation, continues to outperform persistence. Its MAE of 19.03 kW and RMSE of 24.78 kW are considerably lower than persistence's 32.84 kW and 46.85 kW, respectively. When coupled with WiSpEx, WEEL's performance at the 10-minute horizon yields a MAE of 24.8 kW and an RMSE of 32.08 kW. Though slightly higher than WEEL with wind speed interpolation, these values still represent a significant improvement over the persistence forecast.

WEEL's performance remains successful at longer prediction horizons. At 15 minutes, WEEL with WiSpEx (accounting for wind persistence) achieves a MAE of 24.93 kW and an RMSE of 32.24 kW, surpassing persistence's 42.75 kW and 60.3 kW. The MAPE of persistence is almost 77% higher than that of WEEL, while the MAPE of the forecast directly from WiSpEx is almost 72% higher than that of WEEL, aligning to the results of Enercon WT once again.

This trend continues at the 20-minute horizon, where WEEL's MAE of 26.13 kW and RMSE of 34.04 kW are notably lower than persistence's 47.57 kW and 67.24 kW. The MAPE of persistence is almost 92% higher than WEEL's MAPE, close to the respective (persistence's) 100% MAPE increase for Enercon WT. The results for NEG WT align with the results for the Enercon WT, showing the consistency of the method for different types of WTs across different time-steps. These results are promising and highlight the importance of maximum ShiVa identification.

Table 4.13: NEG 15 minute Power Forecast Errors stats for Various Forecast Methods and Time Steps

Method	15min MA	10min MA	W.E.E.L. (Direct)	W.E.E.L. Power Interpolated	W.E.E.L. Wind Interpolated	W.E.E.L. Interpolated (Wi.Sp.Ex.)	W.E.E.L. Wi.Sp.Ex. Persistence (Wi.Sp.Ex.)	W.E.E.L. Wi.Sp.Ex. Persistence (Wi.Sp.Ex.)	W.E.E.L. Wi.Sp.Ex. Persistence (Wi.Sp.Ex.)
MAE [kW]	8.15	18.28	18.48	20.75	18.99	19.03	24.8	44.65	24.93
MSE[kW]	121.24	584.41	603.92	843.16	611.47	613.83	1029.35	3375.76	3176.67
RMSE [kW]	11.01	24.17	24.57	29.04	24.73	24.78	32.08	58.1	46.85
MAPE [%]	3.86	7.45	7.67	8.21	7.95	7.98	10.14	18.24	13.65
NMAE	0.0109	0.0244	0.0246	0.0277	0.0253	0.0254	0.0331	0.0595	0.0438
NRMSE	0.0148	0.0322	0.0328	0.0387	0.0329	0.0330	0.0428	0.0775	0.0626
SMAPE [%]	3.71	7.34	7.53	8.0	7.44	7.45	9.14	15.78	12.68
time step	live	live	+5 min	+5 min	+10 min	+10 min	+10 min	+10 min	+10 min

Table 4.14: NEG 15 minute Power Forecast Absolute Errors [kW] for Various Forecast Methods and Time Steps

Metric	15min MA	10min MA	W.E.E.L. (Direct)	W.E.E.L. Power Interpolated	W.E.E.L. Wind Interpolated	W.E.E.L. Interpolated (Wi.Sp.Ex.)	W.E.E.L. Wi.Sp.Ex. Persistence (Wi.Sp.Ex.)	W.E.E.L. Wi.Sp.Ex. Persistence (Wi.Sp.Ex.)	W.E.E.L. Wi.Sp.Ex. Persistence (Wi.Sp.Ex.)
mean	8.15	18.28	18.48	20.75	18.99	19.03	24.8	44.65	24.93
std	7.41	15.83	16.21	20.32	15.85	15.88	20.36	37.21	33.44
min	0.01	0.0	0.0	0.11	0.01	0.07	0.02	0.14	0.01
10%	1.29	2.61	2.6	3.39	2.58	2.54	3.09	5.84	3.86
20%	2.61	5.84	5.54	4.94	5.64	5.61	7.14	10.7	7.37
30%	3.8	8.2	8.39	7.33	8.67	8.7	10.46	17.86	11.25
40%	5.09	11.18	10.96	10.13	11.66	11.67	14.12	25.88	15.85
50%	6.75	14.37	13.82	14.55	15.08	15.13	19.32	34.66	22.06
75%	10.82	24.84	25.68	29.0	26.1	26.27	35.69	67.41	46.66
90%	16.13	37.46	39.53	46.7	41.47	41.95	53.6	96.11	70.37
95%	20.15	44.61	51.73	61.46	51.06	49.79	67.22	116.29	96.22
max	79.72	117.18	137.07	138.08	121.0	120.7	115.97	213.07	251.44
time step	live	live	+5 min	+5 min	+10 min	+10 min	+10 min	+15 min	+15 min

Table 4.15: NEG 15 minute Power Forecast Absolute Percentage Errors [%] for Various Forecast Methods and Time Steps

Metric	15min MA	10min MA	W.E.E.L.	Persistence	W.E.E.L.	Wind	W.E.E.L.	Wi.Sp.Ex.	Persistence	W.E.E.L.	Wi.Sp.Ex.	Persistence	W.E.E.L.	Wi.Sp.Ex.	Persistence
	(Direct)		Interpolated	Power	Interpolated	(Wi.Sp.Ex.)		(Wi.Sp.Ex.)		(Wi.Sp.Ex.)		(Wi.Sp.Ex.)		(Wi.Sp.Ex.)	
<b>mean</b>	3.86	7.45	7.67	8.21	7.95	7.98	10.14	18.24	13.65	10.25	17.71	18.14	10.76	18.62	20.64
<b>std</b>	7.32	10.07	10.41	11.53	11.98	12.08	15.84	27.45	22.66	15.89	27.24	32.91	16.27	27.81	38.93
<b>min</b>	0.0	0.0	0.0	0.03	0.0	0.01	0.01	0.02	0.0	0.02	0.03	0.0	0.02	0.0	0.01
<b>10%</b>	0.35	0.67	0.63	0.54	0.59	0.66	0.79	1.26	0.91	0.88	1.24	1.15	0.92	1.52	1.27
<b>20%</b>	0.62	1.29	1.32	1.08	1.31	1.27	1.62	2.61	1.73	1.81	2.46	2.29	1.76	2.77	2.39
<b>30%</b>	0.92	1.88	1.92	1.8	2.01	2.09	2.59	4.9	2.75	2.63	4.51	3.79	2.71	4.78	4.41
<b>40%</b>	1.27	2.69	2.7	3.02	3.06	3.04	3.96	7.77	4.54	4.07	6.94	5.96	4.18	7.53	6.86
<b>50%</b>	1.67	3.78	3.85	4.19	4.27	4.27	5.63	10.74	6.57	5.75	10.31	8.81	6.04	10.45	9.67
<b>75%</b>	3.26	8.99	9.55	9.82	9.15	8.87	12.0	20.81	15.88	11.23	19.79	20.97	11.99	21.34	23.31
<b>90%</b>	8.56	18.16	19.21	20.51	17.14	17.18	20.19	39.1	32.97	20.53	37.77	42.34	22.92	39.74	46.35
<b>95%</b>	15.36	26.88	27.06	29.04	31.92	32.03	36.93	61.11	46.15	37.3	60.62	56.84	39.09	67.13	63.79
<b>max</b>	71.93	85.56	94.5	108.37	128.48	129.32	156.98	225.06	303.87	164.14	240.71	464.35	171.37	256.65	507.76
<b>Time step</b>	live	live	+5 min	+5 min	+10 min	+10 min	+10 min	+10 min	+10 min	+15 min	+15 min	+20 min	+20 min	+20 min	+20 min

Table 4.16: NEG 15 minute Power Forecast Symmetrical Absolute Percentage Errors [%] for Various Forecast Methods and Time Steps

Metric	15min MA	10min MA	W.E.E.L.	Persistence	W.E.E.L.	Wind	W.E.E.L.	Wi.Sp.Ex.	Persistence	W.E.E.L.	Wi.Sp.Ex.	Persistence	W.E.E.L.	Wi.Sp.Ex.	Persistence
	(Direct)		Interpolated	Power	Interpolated	(Wi.Sp.Ex.)		(Wi.Sp.Ex.)		(Wi.Sp.Ex.)		(Wi.Sp.Ex.)		(Wi.Sp.Ex.)	
<b>mean</b>	3.71	7.34	7.53	8.0	7.44	7.45	9.14	15.78	12.68	9.26	15.28	16.12	9.75	16.13	17.83
<b>std</b>	6.74	10.05	10.26	10.34	9.8	9.84	11.44	17.35	16.47	11.62	17.28	20.18	12.08	17.97	21.92
<b>min</b>	0.0	0.0	0.0	0.03	0.0	0.01	0.01	0.02	0.0	0.02	0.03	0.02	0.0	0.02	0.01
<b>10%</b>	0.35	0.67	0.63	0.54	0.6	0.66	0.79	1.26	0.91	0.88	1.24	1.16	0.92	1.52	1.27
<b>20%</b>	0.62	1.3	1.32	1.08	1.3	1.28	1.62	2.65	1.72	1.79	2.45	2.28	1.78	2.76	2.41
<b>30%</b>	0.92	1.9	1.92	1.8	2.02	2.1	2.62	4.84	2.78	2.64	4.54	3.76	2.73	4.74	4.41
<b>40%</b>	1.28	2.72	2.99	3.07	3.05	3.09	7.68	4.48	4.08	7.12	5.99	4.24	7.35	6.86	
<b>50%</b>	1.67	3.8	3.87	4.25	4.27	4.29	5.59	10.65	6.54	5.82	10.25	8.67	6.0	10.43	9.8
<b>75%</b>	3.29	9.11	9.14	10.01	8.99	8.99	11.84	20.59	15.92	11.43	19.97	21.11	12.12	21.55	23.68
<b>90%</b>	8.21	17.14	18.74	20.0	16.86	16.93	19.29	36.55	32.82	19.82	36.01	39.93	21.53	38.56	45.53
<b>95%</b>	14.26	25.56	25.73	29.52	28.38	28.45	31.73	52.22	43.48	32.44	48.62	57.74	34.67	54.79	62.5
<b>max</b>	52.91	95.67	94.57	79.78	78.23	78.54	87.95	105.9	122.19	90.15	109.24	139.79	92.29	112.4	143.48
<b>Time step</b>	live	live	+5 min	+5 min	+10 min	+10 min	+10 min	+10 min	+10 min	+15 min	+15 min	+20 min	+20 min	+20 min	+20 min

## Comparison and Insights

In this section, we will evaluate WEEL's performance by comparing its results across the two WTs and contrasting it with statistical and hybrid modeling approaches. We will also delve into insights on the ultra-short-term forecasting landscape, aiming to inspire the scientific community toward more diverse collaborations. It's essential to achieve the parallelization of the processes created by the practical "know-how" focused on swift problem-solving and the theoretical "know-what" that delves slowly into the unexplored core of a problem, if we want to adapt to the ongoing future demands.

Comparing the two WTs, the Enercon WT consistently shows lower MAPE values compared to the NEG WT across all forecasting methods (Michos et al., 2024b). This discrepancy is mainly induced by the Power curve selection, the differences in the WTs' locations, interactions with the terrain, and inherent WT characteristics. Both WTs show that power estimation errors depend on wind speed and direction, suggesting potential for further accuracy improvements through correlation methods. Persistence forecasting serves as a baseline for evaluating physics based forecasting models. WEEL coupled with WiSpEx consistently outperforms persistence for both WTs, demonstrating its superior ability to model the dynamic nature of wind power generation.

Data-driven models leverage vast amounts of historical wind power and meteorological data to identify patterns and correlations. These models, particularly those based on deep learning architectures like LSTM, CNN, and their hybrids, have demonstrated impressive accuracy in capturing the complex dynamics of wind power generation. Their ability to learn from data makes them highly adaptable to specific locations and capable of capturing site-specific nuances. As evidenced in the works of Niu et al., 2022 and L. Liu et al., 2023, data-driven models often achieve MAPE values in the range of 3-5% for one-step-ahead forecasts (5-15 minutes averaging periods). Their impressive accuracy comes from their excellent capacity to capture complex patterns and changes over time in the data. However, their performance tends to degrade as the prediction horizon extends. Longer-term forecasts rely on extrapolating patterns from historical data, which becomes increasingly challenging due to the chaotic nature of wind.

Physics-based models like the demonstrated version of WEEL (Michos et al., 2024b), conversely, can leverage Numerical Weather Prediction (NWP) data and physical principles governing wind flow. They offer a more in-

terpretable approach, as forecasts are rooted in the understanding of atmospheric processes. These models are particularly valuable in scenarios with limited historical wind power data, relying instead on readily available meteorological information. Physics-based models demonstrate good performance in the first few steps, often achieving MAPE values between 5-10% like in our case. This initial accuracy stems from their direct link to measurements or "early step" NWP data and the physical understanding of wind dynamics. However, their reliance on NWP forecasts to expand the forecasting horizon also presents a vulnerability. Errors and uncertainties in NWP models can directly propagate into wind power predictions, especially as the forecast horizon increases to a time zone where the spatial scale and the turbulence scales are not entirely captured by the resolution of either NWP or CFD models.

The evolution of errors in both model types has interesting behaviour. Data-driven models typically excel in the immediate future, capturing short-term fluctuations with impressive precision. However, their accuracy tends to decline as predictions extend into longer horizons (more than 9 steps ahead) like in Niu et al., 2022 L. Liu et al., 2023, Xiang et al., 2022 and Zhang et al., 2021, where the MAPE values may pass 60%. Physics-based models, while slightly less accurate in the very short term, can maintain reasonable performance over longer horizons due to their foundation in meteorological understanding.

This complementary behavior highlights the necessity of both approaches in a comprehensive wind power forecasting system. Data-driven models are invaluable for real-time operations and short-term decision-making, where capturing immediate fluctuations is critical. Physics-based models, are widely used to provide valuable insights into longer-term trends and can serve as a reliable backup when historical data is scarce or unreliable. This is why the attempt to create a physics based ultra-short-term forecasting model that has the ability to extrapolate data like WiSpEx is a difficult and valuable task. A significant advantage of physics based model is their capability to directly adapt to any environment and changes that happen by human intervention. On the other hand statistical models need training on new data for new locations and in cases where human interventions changes the characteristics in the vicinity of a WT.

Recognizing the strengths and limitations of each approach, researchers are increasingly exploring hybrid models that combine data-driven(statistical) and physics-based components ,like NWP forecasts (Chang et al., 2024; Chen et al., 2014; Du, 2023; He et al., 2022; Shirzadi

et al., 2023; Yakoub et al., 2023; M. Yang et al., 2023 and C. Liu et al., 2023). These hybrid models aim to leverage the short-term accuracy of data-driven techniques with the longer-term stability of physics-based approaches like NWP. By integrating both paradigms, they seek to achieve superior overall forecasting performance across various time horizons. Our fast modeling approach is an attempt to bridge the scale gap that is between mesoscale and microscale, especially in Urban areas and their surroundings, as NWP models' horizontal resolution is usually 1x1 km or more (up to several kilometers). Ultimately, neither approach is universally superior. Their coexistence and potential integration into hybrid models hold the key to unlocking the full potential of wind power forecasting. As research in this field progresses, we can anticipate even more sophisticated models capable of accurately predicting wind power generation across various time scales, facilitating the seamless integration of this renewable resource into our energy systems.

#### **4.4.5 conclusion**

Wind power, is inherently volatile and unpredictable. Accurate forecasting, particularly in the ultra-short-term range (minutes to hours), is paramount for effective grid integration and resource management. Both data-driven models and physics-based models that are used for ultra short term forecasting have their strengths and weaknesses, influencing their performance and suitability for specific scenarios.

The foundation of WEEL's success lies in ShiVa detection by the application of MOSVaH, which intelligently selects optimal shifted MAs. This adaptive approach enables WEEL to effectively handle the dynamic nature of wind power generation, thus ensuring consistent accuracy across various time horizons.

A huge advantage of WEEL stems from its physics-based approach by it's coupling with WiSpEx, which accounts for complex wind flow, terrain interactions, and WT characteristics. Furthermore, WEEL's 15-minute ahead forecasts are more valuable than the 5-minute forecasts, enhancing its utility for real-time grid management. If needed, the 15 minute averaged Power forecast of 5 minute resolution can be easily converted to 5 minute averaged power with a 3-step ahead forecasting horizon.

The promising results from power estimations based on modeled power curves, despite conversion errors, suggest that WEEL can further improve with refinements in modeling and error correction. In conclusion, WEEL

and WiSpEx are promising tools for ultra-short-term wind power forecasting, particularly in complex terrains. Its approach, computational efficiency, and adaptability make it a strong candidate for enhancing wind energy integration into the grid.

In conclusion, for those who have journeyed through the entirety of this thesis, I think this is a great multi-dimensional,  $D^n$  space to offer a small token of appreciation for your valuable time. To the readers that have just skipped most of the script, I give you the same gift, but unfortunately your perception of the  $D^n$  space is an  $m < n$ , projected  $D^m$  space. The first two persons to have read this book (this was added after the final revision) are my two professors **Andreas Kazantzidis**(Andreas) and **Francky Catthoor**(Francky). Francky's primary focus is in know how, while Andreas' primary focus is on know what. Their understanding of the opposite view is "well cultivated" in both. As someone naturally inclined toward theoretical perspectives, their guidance helped me broaden my horizons. Andreas, my Supervisor had the know how and know what of my personality, which led him showing Francky the know when to find the know how during this PhD. As this journey concludes, I extend my heartfelt gratitude to both for allowing me to test my ideas against their challenges, which ultimately led to the validation of WEEL. Their mentorship not only pushed me to refine my work but also unlocked new facets of our own personalities. WEEL, a simple tool born from a method that could potentially extend its natural application to any real-life problem, from personal behavior modeling and physics to financial economics and engineering, stands as a testament to their invaluable contributions under my oath to serve science for the best out of humanity.

---

## References

- Baile, R., & Muzy, J.-F. (2023). Leveraging data from nearby stations to improve short-term wind speed forecasts. *Energy*, 263, 125–644. <https://doi.org/10.1016/j.energy.2022.125644>
- Bingöl, F., Mann, J., & Foussekis, D. (2011). 2.3 lidar in complex terrain. *Advancements in Wind Energy Metrology—UPWIND 1A2*. 3, 11. <https://orbit.dtu.dk/en/publications/advancements-in-wind-energy-metrology-upwind-1a23>
- Chang, Y., Yang, H., Chen, Y., Zhou, M., Yang, H., Wang, Y., & Zhang, Y. (2024). A hybrid model for long-term wind power forecasting utilizing nwp subsequence correction and multi-scale deep learning regression methods. *IEEE Transactions on Sustainable Energy*, 15(1), 263–275. <https://doi.org/10.1109/TSTE.2023.3283242>
- Chen, N., Qian, Z., Nabney, I. T., & Meng, X. (2014). Wind power forecasts using gaussian processes and numerical weather prediction. *IEEE Transactions on Power Systems*, 29(2), 656–665. <https://doi.org/10.1109/TPWRS.2013.2282366>
- CRES. (2024). Technical specs of the WTs [[Online]. Last accessed: 24 July 2024]. [http://www.creswindfarm.gr/site1/tech\\_specs.htm](http://www.creswindfarm.gr/site1/tech_specs.htm)
- Dai, X., Liu, G.-P., & Hu, W. (2023). An online-learning-enabled self-attention-based model for ultra-short-term wind power forecasting. *Energy*, 272, 127173. <https://doi.org/10.1016/j.energy.2023.127173>
- Du, P. (2023). Ensemble machine learning-based wind forecasting to combine nwp output with data from weather stations. In *Renewable energy integration for bulk power systems: Ercot and the texas interconnection* (pp. 263–281). Springer International Publishing. [https://doi.org/10.1007/978-3-031-28639-1\\_10](https://doi.org/10.1007/978-3-031-28639-1_10)
- Google Maps. (2024). Center of renewable energy sources and saving (cres) [[Online] Last accessed 24 July 2024]. <https://goo.gl/maps/atzVNhWghePz5HXH7>
- He, B., Ye, L., Pei, M., Lu, P., Dai, B., Li, Z., & Wang, K. (2022). A combined model for short-term wind power forecasting based on the analysis of numerical weather prediction data. *Energy Reports*, 8, 929–939. <https://doi.org/10.1016/j.egyr.2021.10.102>
- Ingenhorst, C., Jacobs, G., Stößel, L., Schelenz, R., & Juretzki, B. (2021). Method for airborne measurement of the spatial wind speed distri-

- 
- bution above complex terrain. *Wind Energy Science*, 6(2), 427–440. <https://doi.org/10.5194/wes-6-427-2021>
- Kim, D., Kim, T., Oh, G., Huh, J., & Ko, K. (2016). A comparison of ground-based lidar and met mast wind measurements for wind resource assessment over various terrain conditions. *Journal of Wind Engineering and Industrial Aerodynamics*, 158, 109–121. <https://doi.org/10.1016/j.jweia.2016.09.011>
- Kogaki, T., Sakurai, K., Shimada, S., Kawabata, H., Otake, Y., Kondo, K., & Fujita, E. (2020). Field measurements of wind characteristics using lidar on a wind farm with downwind turbines installed in a complex terrain region. *Energies*, 13(19), 5135. <https://doi.org/10.3390/en13195135>
- Liu, C., Zhang, X., Mei, S., Zhou, Q., & Fan, H. (2023). Series-wise attention network for wind power forecasting considering temporal lag of numerical weather prediction. *Applied Energy*, 336, 120815. <https://doi.org/10.1016/j.apenergy.2023.120815>
- Liu, L., Liu, J., Ye, Y., Liu, H., Chen, K., Li, D., Dong, X., & Sun, M. (2023). Ultra-short-term wind power forecasting based on deep bayesian model with uncertainty. *Renewable Energy*, 205, 598–607. <https://doi.org/10.1016/j.renene.2023.01.038>
- Manfren, M., Caputo, P., & Costa, G. (2011). Paradigm shift in urban energy systems through distributed generation: Methods and models. *Applied energy*, 88(4), 1032–1048. <https://doi.org/10.1016/j.apenergy.2010.10.018>
- Michos, D., Catthoor, F., Foussekis, D., & Kazantzidis, A. (2024a). A cfd model for spatial extrapolation of wind field over complex terrain—wi. sp. ex. *Energies*, 17(16), 4139. <https://doi.org/10.3390/en17164139>
- Michos, D., Catthoor, F., Foussekis, D., & Kazantzidis, A. (2024b). Ultra-short-term wind power forecasting in complex terrain: A physics-based approach. *Energies*, 17(21), 5493. <https://doi.org/10.3390/en17215493>
- Nallolla, C. A., P, V., Chittathuru, D., & Padmanaban, S. (2023). Multi-objective optimization algorithms for a hybrid ac/dc microgrid using res: A comprehensive review. *Electronics*, 12(4), 1062. <https://doi.org/10.3390/electronics12041062>
- Niu, D., Sun, L., Yu, M., & Wang, K. (2022). Point and interval forecasting of ultra-short-term wind power based on a data-driven method and

- 
- hybrid deep learning model. *Energy*, 254, 124384. <https://doi.org/10.1016/j.energy.2022.124384>
- Polimeni, S., Nespoli, A., Leva, S., Valenti, G., & Manzolini, G. (2021). Implementation of different pv forecast approaches in a multigood microgrid: Modeling and experimental results. *Processes*, 9(2), 323. <https://doi.org/10.3390/pr9020323>
- Quiñones, J. J., Pineda, L. R., Ostanek, J., & Castillo, L. (2023). Towards smart energy management for community microgrids: Leveraging deep learning in probabilistic forecasting of renewable energy sources. *Energy Conversion and Management*, 293, 117440. <https://doi.org/10.1016/j.enconman.2023.117440>
- Saint-Drenan, Y.-M., Besseau, R., Jansen, M., Staffell, I., Troccoli, A., Dubus, L., Schmidt, J., Gruber, K., Simões, S. G., & Heier, S. (2020). A parametric model for wind turbine power curves incorporating environmental conditions. *Renewable Energy*, 157, 754–768. <https://doi.org/10.1016/j.renene.2020.04.123>
- Sezer-Uzol, N., & Long, L. (2006). 3-d time-accurate cfd simulations of wind turbine rotor flow fields. *American Institute of Aeronautics and Astronautics*, 394. <https://doi.org/10.2514/6.2006-394>
- Shen, L., Wei, C., Cai, C., & Liu, X. (2019). Influence of calculation domain size on numerical simulation results for complex terrain wind fields. *Journal of Engineering Science & Technology Review*, 12(2). <https://doi.org/10.25103/jestr.122.09>
- Shirzadi, N., Nasiri, F., Menon, R. P., Monsalvete, P., Kaifel, A., & Eicker, U. (2023). Smart urban wind power forecasting: Integrating weibull distribution, recurrent neural networks, and numerical weather prediction. *Energies*, 16(17). <https://doi.org/10.3390/en16176208>
- Valldecabres, L., Peña, A., Courtney, M., von Bremen, L., & Kühn, M. (2018). Very short-term forecast of near-coastal flow using scanning lidars. *Wind energy science*, 3(1), 313–327. <https://doi.org/10.5194/wes-3-313-2018>
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Aldcroft, T., Mayorov, N., Jones, E., Smith, R., Colbert, S., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261–272. <https://doi.org/10.1038/s41592-019-0686-2>

- 
- Wang, J., & Yang, Z. (2021). Ultra-short-term wind speed forecasting using an optimized artificial intelligence algorithm. *Renewable Energy*, 171, 1418–1435. <https://doi.org/10.1016/j.renene.2021.03.020>
- Wei, J., Wu, X., Yang, T., & Jiao, R. (2023). Ultra-short-term forecasting of wind power based on multi-task learning and lstm. *International Journal of Electrical Power & Energy Systems*, 149, 109073. <https://doi.org/10.1016/j.ijepes.2023.109073>
- Wu, Y.-T., & Porté-Agel, F. (2011). Large-eddy simulation of wind-turbine wakes: Evaluation of turbine parametrisations. *Boundary-layer meteorology*, 138(3), 345–366. <https://doi.org/10.1007/s10546-010-9569-x>
- Xiang, L., Liu, J., Yang, X., Hu, A., & Su, H. (2022). Ultra-short term wind power prediction applying a novel model named satcn-lstm. *Energy Conversion and Management*, 252, 115036. <https://doi.org/10.1016/j.enconman.2021.115036>
- Xiaoxia, G., Luqing, L., Shaohai, Z., Xiaoxun, Z., Haiying, S., Hongxing, Y., Yu, W., & Hao, L. (2022). Lidar-based observation and derivation of large-scale wind turbine's wake expansion model downstream of a hill. *Energy*, 259, 125051. <https://doi.org/10.1016/j.energy.2022.125051>
- Yakoub, G., Mathew, S., & Leal, J. (2023). Intelligent estimation of wind farm performance with direct and indirect ‘point’ forecasting approaches integrating several nwp models. *Energy*, 263, 125893. <https://doi.org/10.1016/j.energy.2022.125893>
- Yan, B., & Li, Q. (2016). Coupled on-site measurement/cfd based approach for high-resolution wind resource assessment over complex terrains. *Energy Conversion and Management*, 117, 351–366. <https://doi.org/10.1016/j.enconman.2016.02.076>
- Yan, S., Shi, S., Chen, X., Wang, X., Mao, L., & Liu, X. (2018). Numerical simulations of flow interactions between steep hill terrain and large scale wind turbine. *Energy*, 151, 740–747. <https://doi.org/10.1016/j.energy.2017.12.075>
- Yang, B., Zhong, L., Wang, J., Shu, H., Zhang, X., Yu, T., & Sun, L. (2021). State-of-the-art one-stop handbook on wind forecasting technologies: An overview of classifications, methodologies, and analysis. *Journal of Cleaner Production*, 283, 124628. <https://doi.org/10.1016/j.jclepro.2020.124628>

- 
- Yang, M., Wang, D., & Zhang, W. (2023). A short-term wind power prediction method based on dynamic and static feature fusion mining. *Energy*, 280, 128226. <https://doi.org/10.1016/j.energy.2023.128226>
- Zhang, Y., Han, J., Pan, G., Xu, Y., & Wang, F. (2021). A multi-stage predicting methodology based on data decomposition and error correction for ultra-short-term wind energy prediction. *Journal of Cleaner Production*, 292, 125981. <https://doi.org/10.1016/j.jclepro.2021.125981>
- Zhou, L., Hu, G., Tse, K. T., & He, X. (2021). Twisted-wind effect on the flow field of tall building. *Journal of Wind Engineering and Industrial Aerodynamics*, 218, 104778. <https://doi.org/10.1016/j.jweia.2021.104778>

# **Chapter 5**

## **Conclusion & Future Work suggestions**

The utilization of CFD simulations for ultra short term wind energy forecasting has been explored through two distinct yet interconnected methodologies: WiSpEx and WEEL. These methodologies, while rooted in different principles, converge towards the common goal of enhancing the accuracy and reliability of wind power predictions, ultimately facilitating improved grid management and energy trading operations.

### **5.1 WiSpEx: Unlocking the Power of Spatial Extrapolation**

WiSpEx emerges as a potent solution to the persistent challenge of data scarcity in wind energy forecasting. By harnessing the computational power of CFD simulations, WiSpEx can generate high-resolution wind field reconstructions from a limited set of measurement points. This transformative capability significantly amplifies the available data, proving especially valuable for ultra-short-term forecasting, where rapid and precise predictions are critical for efficient grid management. The presented model showcases this potential by reconstructing a 3D wind field at an impressive 29,928 estimated locations using only 9 vertical measurement points per time step, all within a remarkably short timeframe of under 2 minutes. This translates to an extraordinary 3325-fold increase in data resolution, underscoring WiSpEx's capacity to extract meaningful insights even from sparse data.

Beyond its role in data augmentation, WiSpEx acts as a critical bridge

---

between large-scale weather patterns and the localized wind dynamics at specific WF sites. By enabling a deeper understanding of how macroscale atmospheric phenomena influence microscale wind behavior at the turbine level, WiSpEx aids in the development and refinement of NWP. This symbiotic relationship between WiSpEx and NWP models promises improved forecasting of wind resources, benefiting both the wind energy industry and meteorological science.

The computational efficiency of WiSpEx further solidifies its practicality. The ability to generate extensive datasets from limited measurements within minutes, positions it as a readily deployable solution for real-world applications. The average computational time for a single IAS is a mere 43 seconds, while the EAS requires 61 seconds. Even the more computationally demanding E-IAS, which incorporates extrapolation calculations and dataset loading, completes in just over 100 seconds. This rapid processing capability is crucial for time-sensitive applications like ultra-short-term forecasting, where swift decision-making is often necessary to optimize grid operations and ensure stability. In cases where proper positioning of measurement instruments is performed and high resolution inlet data are available for the inlet, the EAS extrapolation is not necessary, leading to a single simulation per timestep.

WiSpEx's potential extends beyond wind energy forecasting. In meteorology, its ability to generate high-resolution wind fields can contribute to more accurate weather forecasts and timely warnings for severe weather events. By providing detailed insights into localized wind patterns, WiSpEx can support meteorologists in predicting the onset and intensity of storms, thereby enhancing public safety and enabling proactive disaster preparedness measures.

Within the broader energy sector, WiSpEx can be instrumental in optimizing the placement and operation of WTs. Accurately mapping the wind resource potential across complex terrains allows for strategic turbine positioning to harness the most favorable wind conditions. Additionally, WiSpEx can aid in real-time adjustments of turbine operations to adapt to changing wind patterns, ensuring optimal performance and minimizing wear and tear on the turbines.

Ongoing advancements in remote sensing technologies, especially scanning long-range lidars, hold the promise of further enhancing WiSpEx's performance. By providing the model with even more detailed and precise input data, these technologies can unlock new levels of accuracy and reliability in wind field reconstructions. The advent of scanning lidars,

---

capable of capturing wind data not just vertically but also at an angle aligned with the flow, offers particularly exciting prospects. This capability provides invaluable insights into the spatial and temporal evolution of wind patterns, enabling WiSpEx to generate more realistic and predictive wind field representations.

Moreover, WiSpEx can be instrumental in assessing the suitability of potential WF sites. By simulating wind patterns over complex terrains, it can furnish valuable insights into the anticipated energy yield and any potential challenges associated with different locations. Furthermore, in scenarios where direct measurements are limited or entirely unavailable, the datasets generated by WiSpEx can be leveraged to train statistical and AI-based forecasting models. This capability is particularly beneficial in the early stages of WF development or in regions with sparse meteorological infrastructure, significantly expediting the development and deployment of effective wind energy forecasting solutions.

## **5.2 WEEL: Power latency identification "made simple"**

Building upon the foundation established by WiSpEx, WEEL introduces a critical refinement: the explicit consideration of the inherent delay, or "Wind Energy Extraction Latency", between changes in wind speed and the corresponding power output from a WT. This nuanced understanding of the temporal dynamics of wind energy extraction proves instrumental in enhancing the accuracy of short-term wind power forecasts, a crucial aspect of effective grid management and energy trading.

Our comprehensive analysis reveals WEEL's exceptional performance. In the practically significant 15-minute ahead forecast, WEEL, when coupled with WiSpEx, consistently surpasses both the persistence model and WiSpEx alone. This superior performance is evident in the significantly lower MAPE and SMAPE values observed for WEEL. For the 15-minute horizon, WEEL achieves an impressive MAPE of 8.44% for the Enercon turbine and 10.25% for the NEG turbine, indicating an average forecast deviation of less than 10% from the actual power output. In stark contrast, the persistence model's MAPE values are nearly twice as high, underscoring the substantial improvement in accuracy offered by the WEEL methodology. The SMAPE for both WTs is clearly below 10%, which is the goal set for the benchmark version of the model presented in this book.

---

WEEL's advantage is particularly pronounced in scenarios with low or zero power output. The traditional MAPEmetric can be susceptible to distortions in these cases due to its sensitivity to small absolute errors. SMAPE, with its symmetric formulation, mitigates this issue and provides a more robust evaluation of model performance. The consistently lower SMAPE values observed for WEEL further emphasize its reliability and accuracy, even in challenging wind conditions.

A striking feature of WEEL is its ability to deliver impressive forecasting capabilities even when faced with inherent assumptions and simplifications. This resilience underscores the robustness of the methodology and its capacity to adapt to the dynamic and often unpredictable nature of wind power generation. Its consistent outperformance of the persistence model, a common benchmark in forecasting, further highlights WEEL's value as a powerful predictive tool.

The core of WEEL's success lies in optimal ShiVa detection, which intelligently selects optimal shifted moving averages of different averaging lengths to account for the power lag phenomenon. This adaptive approach enables WEEL to capture the temporal dynamics of wind power generation, ensuring consistent accuracy across various time horizons. By aligning wind speed data with the actual power output, WEEL effectively bridges the gap between meteorological observations and power generation reality.

### **5.3 Implications and Future Directions**

While WiSpEx presents a powerful paradigm for wind data extrapolation, acknowledging its limitations and areas for further refinement is crucial. The model's accuracy is inherently tied to the quality and resolution of the input data. Although it demonstrates impressive performance, the current implementation could be further enhanced by incorporating higher-resolution measurements, especially near the ground level where complex terrain features and surface roughness can significantly influence wind flow patterns. Integrating diverse vertical profile extrapolation methods for lidar measurements and including other meteorological variables such as temperature, pressure, and humidity can lead to the creation of more realistic simulations, offering a deeper understanding of the model's behavior under diverse atmospheric conditions.

It's important to highlight the potential of WiSpEx for generating accurate forecasts by strategically positioning the inlet sufficiently far away

---

from the location of interest. This leverages the inherent time lag, or phase error, associated with the travel of air parcels from the inlet to the WF. By carefully calibrating the inlet location and accounting for the phase error, WiSpEx can provide valuable predictions of future wind conditions at the WF site, further extending its forecasting capabilities. As the E-IAS study indicates, higher-resolution measurements could not only increase the accuracy of the model but also mitigate extreme errors. The results from the E-IAS model can be used to create extrapolation look-up tables, which can then be integrated into ultra-short-term forecasting models, thereby increasing their accuracy without incurring a significant computational overhead. Moreover, statistical models can be retrained after terrain changes by reconstructing any given wind field without the need for new datasets, offering a degree of adaptability and resilience to evolving environmental conditions.

By addressing these limitations and continuously pushing the boundaries of innovation with the assist of Pareto optimization techniques for dynamic modeling of flow behavior, WiSpEx is poised to maintain its position at the forefront of wind energy forecasting, unlocking the full potential of wind power as a clean and sustainable energy source, especially when coupled with WEEL.

WEEL's enhanced forecasting accuracy implications are extensive. For grid operators, the ability to anticipate wind power fluctuations with greater precision empowers them to implement smarter and more efficient grid management strategies. This, in turn, mitigates the challenges associated with the inherent variability of wind power, contributing to improved grid stability and reliability. By anticipating valleys and peaks in wind power generation, operators can proactively adjust turbine settings, schedule maintenance activities, and manage energy storage systems more effectively, leading to increased energy production, reduced downtime, and improved overall operational efficiency.

WEEL's generated forecasts can also catalyze the development of more effective energy trading mechanisms, enabling market participants to make informed decisions based on reliable predictions of wind power availability. This fosters increased confidence in wind energy as a viable and predictable source of electricity and profitability, encouraging its wider adoption and integration into the energy mix.

Beyond these immediate benefits, WEEL has the potential to accelerate the transition toward a cleaner and more sustainable energy future. By enhancing the predictability and reliability of wind power, it can contribute

---

to a more stable and resilient power grid, enabling greater penetration of renewable energy sources and reducing reliance on fossil fuels.

While this study provides compelling evidence of WEEL's effectiveness, plenty of room for future research and development remain. Further investigations into WEEL's performance across diverse geographical locations and WF configurations are suggested. Analyzing its capabilities in varying terrains and wind regimes can yield deeper insights into its adaptability and potential limitations.

Exploring WEEL's application at the individual turbine level is another promising direction. Predicting the power output of each turbine separately and then aggregating the results could further reduce forecasting errors, especially in large WFs with heterogeneous wind conditions.

Integrating WEEL with advanced remote sensing technologies, such as scanning long-range lidars, can unlock even greater levels of accuracy. Leveraging the high-resolution wind data captured by these technologies allows WEEL to refine its understanding of wind flow dynamics and generate more precise power predictions.

Finally, investigating the potential of combining WEEL with other state-of-the-art forecasting models, such as those based on machine learning or artificial intelligence, could lead to the development of hybrid forecasting approaches that harness the strengths of both physics-based and data-driven techniques.

# **Part III**

# **APPENDIX**

# Appendix A

## Linear Algebra

### A.1 An Overview of Tensors in Linear Algebra and Physics

Tensors play a crucial role in both linear algebra and physics, providing a unified framework for describing and analyzing various quantities. In Linear Algebra, a tensor is a multi-dimensional array of numerical values that transforms according to certain rules under a change of coordinates. The number of indices required to specify an element of a tensor determines its order (or rank).

- **Scalar (Zero-order tensor):** A single number, no indices (e.g., temperature, mass).
- **Vector (First-order tensor):** A list of numbers, one index (e.g., velocity  $\vec{v}$  with components  $v_i$ ).
- **Matrix (Second-order tensor):** A rectangular array of numbers, two indices (e.g., a linear transformation  $A$  with components  $A_{ij}$ ).
- **Higher-order tensors:** Arrays with more than two indices (e.g., a third-order tensor  $T_{ijk}$  requires three indices).

#### Operations

- **Addition:** Tensors of the same order can be added by adding their corresponding components.
- **Multiplication:** Tensors can be multiplied using operations like the dot product (contracting one index) or the outer product (combining indices).

- 
- **Transformation:** Under a change of basis, the components of a tensor transform in a specific way that preserves the tensor's intrinsic geometric properties.

In Linear Algebra Tensors can represent linear transformations, rotations, scaling, and other geometric transformations. For example, a matrix (second-order tensor) can stretch or rotate a vector in space. They allow for expressing physical laws in a way that is independent of the coordinate system, making them essential in fields like differential geometry and general relativity. In physics, tensors are used to describe physical quantities that have a directional dependence or that operate in multi-dimensional space.

- **Scalars (0th-order tensors):** Quantities with no direction, such as energy or mass.
- **Vectors (1st-order tensors):** Quantities with magnitude and direction, such as force or velocity.
- **Second-order tensors:** Describe more complex interactions, such as stress, strain, and moment of inertia.

Many physical laws are naturally expressed in terms of tensors because they remain valid under any coordinate transformation. This property makes tensors especially useful in general relativity, where the equations describing gravity (Einstein's field equations) involve tensors. The tensorial nature of physical laws ensures that they are invariant under changes in the observer's perspective, a fundamental requirement for the consistency of physical theories.

Some examples are the **Stress Tensor** ( $\sigma_{ij}$ ), that describes the internal forces in a material. Each component  $\sigma_{ij}$  represents the force per unit area acting in the  $i$ -direction on a surface with a normal in the  $j$ -direction, and the **Inertia Tensor** that describes how mass is distributed in a rigid body and how it resists rotational motion.

In Summary, Tensors are generalizations of scalars, vectors, and matrices, with components that transform according to specific rules under a change of coordinates. In linear algebra, tensors represent multi-linear relationships and can be manipulated through operations like addition, multiplication, and transformation. In physics, tensors describe quantities that have directional dependencies and are essential for expressing physical laws that are independent of the observer's frame of reference. Tensors

---

provide a powerful language for describing and analyzing both abstract mathematical relationships and concrete physical phenomena, making them indispensable in advanced mathematics and theoretical physics.

## A.2 Tensor operations

### A.2.1 Scalar/Inner Product (Dot Product)

- **Definition:** The inner product (or dot product) of two vectors results in a scalar. It measures the degree to which two vectors point in the same direction.
- **Notation:**  $\mathbf{a} \cdot \mathbf{b}$
- **Formula:** For vectors  $\mathbf{a} = (a_1, a_2, a_3)$  and  $\mathbf{b} = (b_1, b_2, b_3)$ ,

$$\mathbf{a} \cdot \mathbf{b} = a_1b_1 + a_2b_2 + a_3b_3$$

- **Properties:**

- Commutative:  $\mathbf{a} \cdot \mathbf{b} = \mathbf{b} \cdot \mathbf{a}$
- Distributive:  $\mathbf{a} \cdot (\mathbf{b} + \mathbf{c}) = \mathbf{a} \cdot \mathbf{b} + \mathbf{a} \cdot \mathbf{c}$
- Scalar Multiplication:  $(k\mathbf{a}) \cdot \mathbf{b} = k(\mathbf{a} \cdot \mathbf{b})$

- **Use:** Measures the projection of one vector onto another, the angle between vectors, and is used in calculations involving orthogonality and vector norms. It is useful for finding projections, angles, and in various optimization problems.

### A.2.2 Outer Product (Tensor Product)

- **Definition:** The outer product (or tensor product) of two vectors results in a matrix (or higher-dimensional tensor). It combines each component of one vector with each component of another vector.
- **Notation:**  $\mathbf{u} \otimes \mathbf{v}$
- **Formula:** For vectors  $\mathbf{u} = (u_1, u_2)$  and  $\mathbf{v} = (v_1, v_2)$ ,

$$\mathbf{u} \otimes \mathbf{v} = \begin{pmatrix} u_1v_1 & u_1v_2 \\ u_2v_1 & u_2v_2 \end{pmatrix}$$

---

- **Properties:**

- Not Commutative:  $\mathbf{u} \otimes \mathbf{v} \neq \mathbf{v} \otimes \mathbf{u}$
- Associative with Scalars:  $k(\mathbf{u} \otimes \mathbf{v}) = (k\mathbf{u}) \otimes \mathbf{v} = \mathbf{u} \otimes (k\mathbf{v})$
- **Use:** Constructs matrices and tensors, used in various applications like quantum mechanics, machine learning, and multidimensional data representation.

### A.2.3 Vector Product (Cross Product)

- **Definition:** The vector product (or cross product) of two vectors results in a third vector that is perpendicular to the plane defined by the first two vectors. It measures the area of the parallelogram spanned by the vectors.
- **Notation:**  $\mathbf{a} \times \mathbf{b}$
- **Formula:** For vectors  $\mathbf{a} = (a_1, a_2, a_3)$  and  $\mathbf{b} = (b_1, b_2, b_3)$ ,

$$\mathbf{a} \times \mathbf{b} = \begin{pmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{pmatrix}$$

- **Properties:**

- Anticommutative:  $\mathbf{a} \times \mathbf{b} = -(\mathbf{b} \times \mathbf{a})$
- Distributive:  $\mathbf{a} \times (\mathbf{b} + \mathbf{c}) = (\mathbf{a} \times \mathbf{b}) + (\mathbf{a} \times \mathbf{c})$
- Scalar Multiplication:  $(k\mathbf{a}) \times \mathbf{b} = k(\mathbf{a} \times \mathbf{b})$
- **Use:** Finds a vector orthogonal to two given vectors, used in calculating areas, torques, and in 3D graphics.

### A.2.4 Summary of fundamental Tensor operations

Operation	Definition	Result	Notation	Formula
Scalar (Inner) Product	Scalar product of two vectors	Scalar	$\mathbf{a} \cdot \mathbf{b}$	$\mathbf{a} \cdot \mathbf{b} = a_1b_1 + a_2b_2 + a_3b_3$
Outer Product	Tensor product of two vectors	Matrix (or tensor)	$\mathbf{u} \otimes \mathbf{v}$	$\mathbf{u} \otimes \mathbf{v} = \mathbf{u}\mathbf{v}^T = \begin{pmatrix} u_1v_1 & u_1v_2 & u_1v_3 \\ u_2v_1 & u_2v_2 & u_2v_3 \\ u_3v_1 & u_3v_2 & u_3v_3 \end{pmatrix}$
Vector Product	Cross product of two vectors	Vector	$\mathbf{a} \times \mathbf{b}$	$\mathbf{a} \times \mathbf{b} = \begin{pmatrix} a_2b_3 - a_3b_2 \\ a_3b_1 - a_1b_3 \\ a_1b_2 - a_2b_1 \end{pmatrix}$

---

### A.2.5 Tensor Contraction $T : S$

**Tensors:** A tensor is a mathematical object that can be thought of as a generalized matrix. In this context,  $T$  and  $S$  can be second-rank tensors, meaning they can be represented as matrices with components  $T_{ij}$  and  $S_{ij}$ .

**Contraction:** The operation  $T : S$  denotes the contraction of these two tensors. In index notation, this contraction is written as:

$$T : S = T_{ij}S_{ji}$$

This means that you multiply corresponding components of the tensors and sum over the repeated indices  $i$  and  $j$ . It results in a scalar quantity.

For example, if  $T$  and  $S$  are  $3 \times 3$  matrices, then:

$$T : S = T_{11}S_{11} + T_{12}S_{21} + T_{13}S_{31} + T_{21}S_{12} + T_{22}S_{22} + T_{23}S_{32} + T_{31}S_{13} + T_{32}S_{23} + T_{33}S_{33}$$

This contraction operation is important in deriving scalar quantities from tensor equations, particularly in the context of energy conservation.

## Determinant of a $3 \times 3$ Matrix

Given a  $3 \times 3$  matrix:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

the determinant of matrix  $A$ , denoted as  $\det(A)$  or  $|A|$ , is calculated as follows:

$$\det(A) = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix}$$

where each  $2 \times 2$  matrix is a minor of  $A$ .

## Step-by-Step Computation

1. \*\*Compute the minors\*\*:

---

- The minor of  $a_{11}$  is:

$$M_{11} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} = a_{22}a_{33} - a_{23}a_{32}$$

- The minor of  $a_{12}$  is:

$$M_{12} = \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} = a_{21}a_{33} - a_{23}a_{31}$$

- The minor of  $a_{13}$  is:

$$M_{13} = \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} = a_{21}a_{32} - a_{22}a_{31}$$

2. \*\*Apply the cofactor signs\*\*:

- For  $a_{11}$ , the sign is positive. - For  $a_{12}$ , the sign is negative. - For  $a_{13}$ , the sign is positive.

3. \*\*Combine the results\*\*:

$$\det(A) = a_{11} \cdot M_{11} - a_{12} \cdot M_{12} + a_{13} \cdot M_{13}$$

Substituting the minors:

$$\det(A) = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

# Appendix B

## Power Curve

### B.1 Explanation of WEEL's POWER CURVE

The equation used for approximating the power curve of a WT – based on real data – in WEEL is:

$$P_{watt}(x, a, k, b, v, q, c) = a + \left( \frac{k - a}{(c + q \exp(-bx))^{\frac{1}{v}}} \right) \quad (\text{B.1.0.1})$$

Where:

#### Parameter $a$ (Lower Asymptote):

- **Role:** The parameter  $a$  represents the lower asymptote of the power curve, which is the minimum power output value that the function approaches as wind speed decreases.
- **Effect:** This parameter sets the baseline power output when wind speed is very low. If  $a$  is negative, it can indicate a loss or a state where the power output is below zero, potentially modeling power consumption or inefficiencies at very low wind speeds.

#### Parameter $k$ (Upper Asymptote):

- **Role:** The parameter  $k$  defines the upper asymptote of the power curve, representing the maximum power output the turbine can achieve under optimal conditions.
- **Effect:**  $k$  sets the upper limit for the power curve. As wind speed increases and the turbine reaches its rated capacity, the power output

---

levels off near this upper asymptote. Thus,  $k$  is critical in modeling the rated power of the wind turbine.

#### **Parameter $b$ (Rate of Exponential Decay):**

- **Role:** The parameter  $b$  controls how rapidly the power output changes with respect to the wind speed. Specifically, it influences the decay rate of the exponential term  $\exp(-bx)$ .
- **Effect:** A larger  $b$  value causes the exponential term to decay more quickly, leading to a steeper initial increase in power output as wind speed rises. A smaller  $b$  value results in a more gradual increase, spreading the power gain over a broader wind speed range.

#### **Parameter $v$ (Shape Control):**

- **Role:** The parameter  $v$  adjusts the shape of the curve by determining the curvature of the transition between lower and upper power output.
- **Effect:** A larger  $v$  value makes the transition more gradual, resulting in a flatter curve. This means that power output increases more slowly as wind speed increases. A smaller  $v$  makes the curve steeper, with a sharper increase in power at certain wind speeds.

#### **Parameter $q$ (Scaling Factor for the Exponential Term):**

- **Role:** The parameter  $q$  scales the exponential decay term, shifting the wind speed range where significant changes in power output occur.
- **Effect:** A larger  $q$  moves the impact of the exponential term to higher wind speeds, delaying the significant rise in power output. Conversely, a smaller  $q$  value brings the effect of the exponential term into play at lower wind speeds, accelerating the increase in power output.

#### **Parameter $c$ (Base Adjustment of the Exponential Term):**

- **Role:** The parameter  $c$  is a constant added to the exponential term, which adjusts the baseline of the exponential function.
- **Effect:** By setting  $c = 1$ , the function simplifies the base of the exponential term, effectively normalizing the curve. This normalization helps maintain the consistency and scale of the exponential term's effect across different wind speeds, aiding in smoother curve fitting and interpretation.

---

## B.2 Complementary information on approximation

Together, these parameters  $a$ ,  $k$ ,  $b$ ,  $v$ ,  $q$ , and  $c$  enable the power function to capture the full range of a wind turbine's performance, from low wind speeds where power output is minimal, to the rated capacity at optimal wind speeds, and beyond to where power output stabilizes or decreases. The model thus provides a flexible representation of the wind turbine's power curve under various operating conditions.

In our case, the constant  $c$  is set to 1 to normalize the base of the exponential component, simplifying the function, yielding:

$$P_{\text{watt}}(x, a, k, b, v, q) = a + \left( \frac{k - a}{(1 + q \exp(-bx))^{\frac{1}{v}}} \right),$$

In the context of power curve approximation,  $\hat{\theta}$  represents the vector of estimated parameters from a given model. Specifically, it encapsulates all the parameters estimated in the power curve fit:

$$\hat{\theta} = [a, k, b, v, q]^T$$

When denoting the covariance matrix,  $\hat{\theta}$  is used to indicate that the matrix corresponds to the variances and covariances of the parameter estimates within this vector. This notation succinctly represents the idea that we refer to the statistical properties (variances and covariances) of the estimated parameters as a collective group.

$$\text{Cov}(\hat{\theta}) = \begin{bmatrix} \text{Var}(a) & \text{Cov}(a, k) & \text{Cov}(a, b) & \text{Cov}(a, v) & \text{Cov}(a, q) \\ \text{Cov}(k, a) & \text{Var}(k) & \text{Cov}(k, b) & \text{Cov}(k, v) & \text{Cov}(k, q) \\ \text{Cov}(b, a) & \text{Cov}(b, k) & \text{Var}(b) & \text{Cov}(b, v) & \text{Cov}(b, q) \\ \text{Cov}(v, a) & \text{Cov}(v, k) & \text{Cov}(v, b) & \text{Var}(v) & \text{Cov}(v, q) \\ \text{Cov}(q, a) & \text{Cov}(q, k) & \text{Cov}(q, b) & \text{Cov}(q, v) & \text{Var}(q) \end{bmatrix}$$

In Python, when using curve fitting methods to approximate functions from datasets, such as the `scipy.optimize.curve_fit`, the covariance matrix of the estimated parameters is often provided as part of the output. The calculation of covariance and variance of the estimated parameters in the context of curve fitting, are demonstrated below:

---

## Curve Fitting with `scipy.optimize.curve_fit`

When a curve is fitted with the use of `curve_fit`, it returns two values: the estimated parameters and the covariance matrix of these parameters. The parameters  $\hat{\theta}$  are estimated by minimizing the difference between the observed data and the model's predicted values. The covariance matrix  $C$  is estimated based on the residuals of the fit. In Python, this matrix is calculated using:

$$C = (J^\top J)^{-1} \cdot \sigma^2$$

where  $J$  is the Jacobian matrix of the model function with respect to the parameters and  $\sigma^2$  is an estimate of the variance of the residuals (errors).

The variance of each parameter estimate is given by the diagonal elements of the covariance matrix  $C$ . That is, the variance of the  $i$ -th parameter  $\hat{\theta}_i$  is  $\text{Var}(\hat{\theta}_i) = C_{ii}$ . The covariance between the  $i$ -th and  $j$ -th parameters is given by the off-diagonal elements  $\text{Cov}(\hat{\theta}_i, \hat{\theta}_j) = C_{ij}$ . It is worth noting that diagonal elements of the covariance matrix  $C$  represent the variances of the estimated parameters, off-diagonal elements represent the covariances between pairs of parameters and the standard deviations of the parameters are the square roots of the variances. This approach allows the quantification of the uncertainty in your parameter estimates, providing insights into how precisely each parameter has been estimated based on the data.

The Jacobian matrix  $J$  is a key component in curve fitting and many other numerical methods. It represents the first-order partial derivatives of a vector-valued function with respect to its parameters. In the context of curve fitting, it captures how changes in the parameters affect the model's predictions. For a model function  $f(x; \theta)$ , the Jacobian matrix  $J$  is defined as:

$$J = \begin{bmatrix} \frac{\partial f(x_1; \theta)}{\partial \theta_1} & \frac{\partial f(x_1; \theta)}{\partial \theta_2} & \dots & \frac{\partial f(x_1; \theta)}{\partial \theta_n} \\ \frac{\partial f(x_2; \theta)}{\partial \theta_1} & \frac{\partial f(x_2; \theta)}{\partial \theta_2} & \dots & \frac{\partial f(x_2; \theta)}{\partial \theta_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f(x_m; \theta)}{\partial \theta_1} & \frac{\partial f(x_m; \theta)}{\partial \theta_2} & \dots & \frac{\partial f(x_m; \theta)}{\partial \theta_n} \end{bmatrix}$$

, where  $x$  is the independent variable,  $\theta = [\theta_1, \theta_2, \dots, \theta_n]$  is the vector of parameters to be estimated and  $f(x; \theta)$  is the function that predicts the dependent variable  $y$ . Each row corresponds to a different observation

---

$x_i$  and each column corresponds to the partial derivative of the model function with respect to one of the parameters  $\theta_j$ .

# Bibliography

- Baile, R., & Muzy, J.-F. (2023). Leveraging data from nearby stations to improve short-term wind speed forecasts. *Energy*, 263, 125–644. <https://doi.org/10.1016/j.energy.2022.125644>
- Bazilevs, Y., Hsu, M.-C., Kiendl, J., Wüchner, R., & Bletzinger, K.-U. (2011). 3d simulation of wind turbine rotors at full scale. part ii: Fluid–structure interaction modeling with composite blades. *International Journal for numerical methods in fluids*, 65(1-3), 236–253. <https://doi.org/10.1002/fld.2454>
- Bazilevs, Y., Hsu, M.-C., Akkerman, I., Wright, S., Takizawa, K., Henicke, B., Spielman, T., & Tezduyar, T. (2011). 3d simulation of wind turbine rotors at full scale. part i: Geometry modeling and aerodynamics. *International journal for numerical methods in fluids*, 65(1-3), 207–235. <https://doi.org/10.1002/fld.2400>
- Beaucage, P., Brower, M. C., & Tensen, J. (2014). Evaluation of four numerical wind flow models for wind resource mapping. *Wind Energy*, 17(2), 197–208. <https://doi.org/10.1002/we.1568>
- Bilal, M., Birkelund, Y., Homola, M., & Virk, M. S. (2016). Wind over complex terrain–microscale modelling with two types of mesoscale winds at nygårdsfjell. *Renewable Energy*, 99, 647–653. <https://doi.org/10.1016/j.renene.2016.07.042>
- Bingöl, F., Mann, J., & Foussekis, D. (2011). 2.3 lidar in complex terrain. *Advancements in Wind Energy Metrology–UPWIND 1A2*. 3, 11. <https://orbit.dtu.dk/en/publications/advancements-in-wind-energy-metrology-upwind-1a23>
- Castellani, F., Astolfi, D., Mana, M., Burlando, M., Meißner, C., & Piccioni, E. (2016). Wind power forecasting techniques in complex terrain: Ann vs. ann-cfd hybrid approach. *Journal of Physics: Conference Series*, 753(8), 082002. <https://doi.org/10.1088/1742-6596/753/8/082002>

- 
- Catthoor, F., Basten, T., Zompakis, N., Geilen, M., & Kjeldsberg, P. G. (2020). *System-scenario-based design principles and applications* (Vol. 1). Springer. <https://doi.org/10.1007/978-3-030-20343-6>
- Chang, Y., Yang, H., Chen, Y., Zhou, M., Yang, H., Wang, Y., & Zhang, Y. (2024). A hybrid model for long-term wind power forecasting utilizing nwp subsequence correction and multi-scale deep learning regression methods. *IEEE Transactions on Sustainable Energy*, 15(1), 263–275. <https://doi.org/10.1109/TSTE.2023.3283242>
- Chen, H. (2024). A novel wind model downscaling with statistical regression and forecast for the cleaner energy. *Journal of Cleaner Production*, 434, 140217. <https://doi.org/10.1016/j.jclepro.2023.140217>
- Chen, L. (2021). Inf-sup conditions for operator equations. *Lecture Notes*, <https://www.math.uci.edu/~chenlong/226/inf-supOperator.pdf>.
- Chen, N., Qian, Z., Nabney, I. T., & Meng, X. (2014). Wind power forecasts using gaussian processes and numerical weather prediction. *IEEE Transactions on Power Systems*, 29(2), 656–665. <https://doi.org/10.1109/TPWRS.2013.2282366>
- Choukulkar, A., Pichugina, Y., Clack, C. T., Calhoun, R., Banta, R., Brewer, A., & Hardesty, M. (2016). A new formulation for rotor equivalent wind speed for wind resource assessment and wind power forecasting. *Wind Energy*, 19(8), 1439–1452. <https://doi.org/10.1002/we.1929>
- COMSOL Multiphysics® (CFD). (2020). *COMSOL Multiphysics® CFD module user's guide, version 5.6*. COMSOL. <https://doc.comsol.com/5.6/doc/com.comsol.help.cfd/CFDModuleUsersGuide.pdf>
- COMSOL Multiphysics® (CYCLOPEDIA). (2017). COMSOL Multiphysics® CYCLOPEDIA [[Online]. Last accessed: 24 August 2024]. <https://www.comsol.com/multiphysics/finite-element-method>
- COMSOL Multiphysics® (Reference Manual). (2020). *COMSOL Multiphysics® Reference Manual, version 5.6*. COMSOL. [https://doc.comsol.com/5.5/doc/com.comsol.help.comsol/COMSOL\\_ReferenceManual.pdf](https://doc.comsol.com/5.5/doc/com.comsol.help.comsol/COMSOL_ReferenceManual.pdf)
- CRES. (2024). Technical specs of the WTs [[Online]. Last accessed: 24 July 2024]. [http://www.creswindfarm.gr/site1/tech\\_specs.htm](http://www.creswindfarm.gr/site1/tech_specs.htm)
- Dai, X., Liu, G.-P., & Hu, W. (2023). An online-learning-enabled self-attention-based model for ultra-short-term wind power forecasting. *Energy*, 272, 127173. <https://doi.org/10.1016/j.energy.2023.127173>
- Dick, E. (2009). Introduction to finite element methods in computational fluid dynamics. In J. F. Wendt (Ed.), *Computational fluid dynamics*

- 
- (pp. 235–274). Springer Berlin Heidelberg. [https://doi.org/10.1007/978-3-540-85056-4\\_10](https://doi.org/10.1007/978-3-540-85056-4_10)
- Du, P. (2023). Ensemble machine learning-based wind forecasting to combine nwp output with data from weather stations. In *Renewable energy integration for bulk power systems: Ercot and the texas interconnection* (pp. 263–281). Springer International Publishing. [https://doi.org/10.1007/978-3-031-28639-1\\_10](https://doi.org/10.1007/978-3-031-28639-1_10)
- Emeis, S. (2018). *Wind energy meteorology: Atmospheric physics for wind power generation*. Springer. <https://doi.org/10.1007/978-3-642-30523-8>
- Fadaeinedjad, R., Moallem, M., & Moschopoulos, G. (2008). Simulation of a wind turbine with doubly fed induction generator by fast and simulink. *IEEE Transactions on energy conversion*, 23(2), 690–700. <https://doi.org/10.1109/TEC.2007.914307>
- Feng, J., & Shen, W. Z. (2015). Modelling wind for wind farm layout optimization using joint distribution of wind speed and wind direction. *Energies*, 8(4), 3075–3092. <https://doi.org/10.3390/en8043075>
- Ferziger, J. H., & Perić, M. (2002). *Computational methods for fluid dynamics*. Springer. <https://doi.org/10.1007/978-3-642-56026-2>
- Ferziger, J. H., & Perić, M. (2020). *Computational methods for fluid dynamics* (4th). Springer. <https://doi.org/10.1007/978-3-319-99693-6>
- Giebel, G., Kariniotakis, G., & Brownsword, R. (2003). The state-of-the-art in short-term prediction of wind power-from a danish perspective. *Proceedings of the 4th International Workshop on Large-Scale Integration of Wind Power and Transmission Networks for Offshore Wind Farms*, 21–23. <https://minesparis-psl.hal.science/hal-00529338>
- Google Maps. (2024). Center of renewable energy sources and saving (cres) [[Online] Last accessed 24 July 2024]. <https://goo.gl/maps/atzVNhWghePz5HXH7>
- He, B., Ye, L., Pei, M., Lu, P., Dai, B., Li, Z., & Wang, K. (2022). A combined model for short-term wind power forecasting based on the analysis of numerical weather prediction data. *Energy Reports*, 8, 929–939. <https://doi.org/10.1016/j.egyr.2021.10.102>
- Ingenhorst, C., Jacobs, G., Stößel, L., Schelenz, R., & Juretzki, B. (2021). Method for airborne measurement of the spatial wind speed distribution above complex terrain. *Wind Energy Science*, 6(2), 427–440. <https://doi.org/10.5194/wes-6-427-2021>

- 
- Kajishima, T., & Taira, K. (2016). *Computational fluid dynamics: Incompressible turbulent flows*. Springer. <https://doi.org/10.1007/978-3-319-45304-0>
- Kavari, G., Tahani, M., & Mirhosseini, M. (2019). Wind shear effect on aerodynamic performance and energy production of horizontal axis wind turbines with developing blade element momentum theory. *Journal of Cleaner Production*, 219, 368–376. <https://doi.org/10.1016/j.jclepro.2019.02.073>
- Kim, D., Kim, T., Oh, G., Huh, J., & Ko, K. (2016). A comparison of ground-based lidar and met mast wind measurements for wind resource assessment over various terrain conditions. *Journal of Wind Engineering and Industrial Aerodynamics*, 158, 109–121. <https://doi.org/10.1016/j.jweia.2016.09.011>
- Kogaki, T., Sakurai, K., Shimada, S., Kawabata, H., Otake, Y., Kondo, K., & Fujita, E. (2020). Field measurements of wind characteristics using lidar on a wind farm with downwind turbines installed in a complex terrain region. *Energies*, 13(19), 5135. <https://doi.org/10.3390/en13195135>
- Liu, C., Zhang, X., Mei, S., Zhou, Q., & Fan, H. (2023). Series-wise attention network for wind power forecasting considering temporal lag of numerical weather prediction. *Applied Energy*, 336, 120815. <https://doi.org/10.1016/j.apenergy.2023.120815>
- Liu, L., Liu, J., Ye, Y., Liu, H., Chen, K., Li, D., Dong, X., & Sun, M. (2023). Ultra-short-term wind power forecasting based on deep bayesian model with uncertainty. *Renewable Energy*, 205, 598–607. <https://doi.org/10.1016/j.renene.2023.01.038>
- Manfren, M., Caputo, P., & Costa, G. (2011). Paradigm shift in urban energy systems through distributed generation: Methods and models. *Applied energy*, 88(4), 1032–1048. <https://doi.org/10.1016/j.apenergy.2010.10.018>
- Manyonge, A., Manyala, R., Onyango, F., & Shichika, J. (2012). Mathematical modelling of wind turbine in a wind energy conversion system: Power coefficient analysis. *Applied Mathematical Sciences*, 6, 4527–4536. <https://api.semanticscholar.org/CorpusID:14491844>
- Marten, D., Lennie, M., Pechlivanoglou, G., Nayeri, C. N., & Paschereit, C. O. (2016). Implementation, optimization, and validation of a non-linear lifting line-free vortex wake module within the wind turbine simulation code qblade. *Journal of Engineering for Gas Turbines and Power*, 138(7). <https://doi.org/10.1115/1.4031872>

- 
- Michos, D., Catthoor, F., Foussekis, D., & Kazantzidis, A. (2024a). A cfd model for spatial extrapolation of wind field over complex terrain—wi. sp. ex. *Energies*, 17(16), 4139. <https://doi.org/10.3390/en17164139>
- Michos, D., Catthoor, F., Foussekis, D., & Kazantzidis, A. (2024b). Ultra-short-term wind power forecasting in complex terrain: A physics-based approach. *Energies*, 17(21), 5493. <https://doi.org/10.3390/en17215493>
- Monfared, M., Kojabadi, H. M., & Rastegar, H. (2008). Static and dynamic wind turbine simulator using a converter controlled dc motor. *Renewable Energy*, 33(5), 906–913. <https://doi.org/10.1016/j.renene.2007.06.007>
- Nallolla, C. A., P, V., Chittathuru, D., & Padmanaban, S. (2023). Multi-objective optimization algorithms for a hybrid ac/dc microgrid using res: A comprehensive review. *Electronics*, 12(4), 1062. <https://doi.org/10.3390/electronics12041062>
- Nathan, J., Forsting, A. M., Troldborg, N., & Masson, C. (2017). Comparison of openfoam and ellipsys3d actuator line methods with (new) mexico results. *Journal of Physics: Conference Series*, 854(1), 012033. <https://doi.org/10.1088/1742-6596/854/1/012033>
- Niu, D., Sun, L., Yu, M., & Wang, K. (2022). Point and interval forecasting of ultra-short-term wind power based on a data-driven method and hybrid deep learning model. *Energy*, 254, 124384. <https://doi.org/10.1016/j.energy.2022.124384>
- Polimeni, S., Nespoli, A., Leva, S., Valenti, G., & Manzolini, G. (2021). Implementation of different pv forecast approaches in a multigood microgrid: Modeling and experimental results. *Processes*, 9(2), 323. <https://doi.org/10.3390/pr9020323>
- Quiñones, J. J., Pineda, L. R., Ostanek, J., & Castillo, L. (2023). Towards smart energy management for community microgrids: Leveraging deep learning in probabilistic forecasting of renewable energy sources. *Energy Conversion and Management*, 293, 117440. <https://doi.org/10.1016/j.enconman.2023.117440>
- Regodeseves, P. G., & Morros, C. S. (2020). Unsteady numerical investigation of the full geometry of a horizontal axis wind turbine: Flow through the rotor and wake. *Energy*, 202, 117674. <https://doi.org/10.1016/j.energy.2020.117674>
- Réthoré, P.-E., Sørensen, N., Zahle, F., Bechmann, A., & Madsen, H. (2011). Mexico wind tunnel and wind turbine modelled in cfd. *29th AIAA*

- 
- Applied Aerodynamics Conference*, 1–10. <https://doi.org/10.2514/6.2011-3373>
- Saheb-Koussa, D., Haddadi, M., Belhamel, M., et al. (2012). Modeling and simulation of windgenerator with fixed speed wind turbine under matlab-simulink. *Energy Procedia*, 18, 701–708. <https://doi.org/10.1016/j.egypro.2012.05.085>
- Saint-Drenan, Y.-M., Besseau, R., Jansen, M., Staffell, I., Troccoli, A., Dubus, L., Schmidt, J., Gruber, K., Simões, S. G., & Heier, S. (2020). A parametric model for wind turbine power curves incorporating environmental conditions. *Renewable Energy*, 157, 754–768. <https://doi.org/10.1016/j.renene.2020.04.123>
- Sezer-Uzol, N., & Long, L. (2006). 3-d time-accurate cfd simulations of wind turbine rotor flow fields. *American Institute of Aeronautics and Astronautics*, 394. <https://doi.org/10.2514/6.2006-394>
- Shen, L., Wei, C., Cai, C., & Liu, X. (2019). Influence of calculation domain size on numerical simulation results for complex terrain wind fields. *Journal of Engineering Science & Technology Review*, 12(2). <https://doi.org/10.25103/jestr.122.09>
- Shirzadi, N., Nasiri, F., Menon, R. P., Monsalvete, P., Kaifel, A., & Eicker, U. (2023). Smart urban wind power forecasting: Integrating weibull distribution, recurrent neural networks, and numerical weather prediction. *Energies*, 16(17). <https://doi.org/10.3390/en16176208>
- Shourangiz-Haghghi, A., Haghnegahdar, M. A., Wang, L., Mussetta, M., Kolios, A., & Lander, M. (2020). State of the art in the optimisation of wind turbine performance using cfd. *Archives of Computational Methods in Engineering*, 27(2), 413–431. <https://doi.org/10.1007/s11831-019-09316-0>
- Song, D., Yang, Y., Zheng, S., Tang, W., Yang, J., Su, M., Yang, X., & Joo, Y. H. (2019). Capacity factor estimation of variable-speed wind turbines considering the coupled influence of the qn-curve and the air density. *Energy*, 183, 1049–1060. <https://doi.org/10.1016/j.energy.2019.07.018>
- Tahir, A., Elgabaili, M., Rajab, Z., Buaossa, N., Khalil, A., & Mohamed, F. (2019). Optimization of small wind turbine blades using improved blade element momentum theory. *Wind Engineering*, 43(3), 299–310. <https://doi.org/10.1177/0309524X18791395>
- Thompson, J. F., Warsi, Z. U., & Mastin, C. W. (1985). *Numerical grid generation: Foundations and applications*. Elsevier North-Holland, Inc.

- 
- Tian, Z. (2021). Modes decomposition forecasting approach for ultra-short-term wind speed. *Applied Soft Computing*, 105, 107303. <https://doi.org/10.1016/j.asoc.2021.107303>
- Valdecabres, L., Peña, A., Courtney, M., von Bremen, L., & Kühn, M. (2018). Very short-term forecast of near-coastal flow using scanning lidars. *Wind energy science*, 3(1), 313–327. <https://doi.org/10.5194/wes-3-313-2018>
- Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Aldcroft, T., Mayorov, N., Jones, E., Smith, R., Colbert, S., et al. (2020). SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3), 261–272. <https://doi.org/10.1038/s41592-019-0686-2>
- Wang, J., & Yang, Z. (2021). Ultra-short-term wind speed forecasting using an optimized artificial intelligence algorithm. *Renewable Energy*, 171, 1418–1435. <https://doi.org/10.1016/j.renene.2021.03.020>
- Wei, J., Wu, X., Yang, T., & Jiao, R. (2023). Ultra-short-term forecasting of wind power based on multi-task learning and lstm. *International Journal of Electrical Power & Energy Systems*, 149, 109073. <https://doi.org/10.1016/j.ijepes.2023.109073>
- Wendt, J. (2008). *Computational fluid dynamics: An introduction*. Springer Berlin Heidelberg. <https://books.google.gr/books?id=IIUkqlI-HNbQC>
- Wharton, S., Bulaevskaya, V., Irons, Z., Qualley, G., Newman, J., & Miller, W. (2015). *Wind power curve modeling using statistical models: An investigation of atmospheric input variables at a flat and complex terrain wind farm* (tech. rep.). Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States). <https://doi.org/10.2172/1223839>
- Wilcox, D. C. (2006). *Turbulence modeling for cfd* (3rd) [Includes bibliography, index and Compact Disk.]. DCW Industries, Inc. <http://www.dcwindustries.com>
- Wu, Y.-T., & Porté-Agel, F. (2011). Large-eddy simulation of wind-turbine wakes: Evaluation of turbine parametrisations. *Boundary-layer meteorology*, 138(3), 345–366. <https://doi.org/10.1007/s10546-010-9569-x>
- Xiang, L., Liu, J., Yang, X., Hu, A., & Su, H. (2022). Ultra-short term wind power prediction applying a novel model named satcn-lstm. *Energy Conversion and Management*, 252, 115036. <https://doi.org/10.1016/j.enconman.2021.115036>

- 
- Xiaoxia, G., Luqing, L., Shaohai, Z., Xiaoxun, Z., Haiying, S., Hongxing, Y., Yu, W., & Hao, L. (2022). Lidar-based observation and derivation of large-scale wind turbine's wake expansion model downstream of a hill. *Energy*, 259, 125051. <https://doi.org/10.1016/j.energy.2022.125051>
- Yakoub, G., Mathew, S., & Leal, J. (2023). Intelligent estimation of wind farm performance with direct and indirect 'point' forecasting approaches integrating several nwp models. *Energy*, 263, 125893. <https://doi.org/10.1016/j.energy.2022.125893>
- Yan, B., & Li, Q. (2016). Coupled on-site measurement/cfd based approach for high-resolution wind resource assessment over complex terrains. *Energy Conversion and Management*, 117, 351–366. <https://doi.org/10.1016/j.enconman.2016.02.076>
- Yan, S., Shi, S., Chen, X., Wang, X., Mao, L., & Liu, X. (2018). Numerical simulations of flow interactions between steep hill terrain and large scale wind turbine. *Energy*, 151, 740–747. <https://doi.org/10.1016/j.energy.2017.12.075>
- Yang, B., Zhong, L., Wang, J., Shu, H., Zhang, X., Yu, T., & Sun, L. (2021). State-of-the-art one-stop handbook on wind forecasting technologies: An overview of classifications, methodologies, and analysis. *Journal of Cleaner Production*, 283, 124628. <https://doi.org/10.1016/j.jclepro.2020.124628>
- Yang, M., Wang, D., & Zhang, W. (2023). A short-term wind power prediction method based on dynamic and static feature fusion mining. *Energy*, 280, 128226. <https://doi.org/10.1016/j.energy.2023.128226>
- Zemansky, M., & Dittman, R. (1997). *Heat and thermodynamics: An intermediate textbook*. McGraw-Hill. <https://books.google.gr/books?id=4vVAAQAAIAAJ>
- Zhang, Y., Han, J., Pan, G., Xu, Y., & Wang, F. (2021). A multi-stage predicting methodology based on data decomposition and error correction for ultra-short-term wind energy prediction. *Journal of Cleaner Production*, 292, 125981. <https://doi.org/10.1016/j.jclepro.2021.125981>
- Zhou, L., Hu, G., Tse, K. T., & He, X. (2021). Twisted-wind effect on the flow field of tall building. *Journal of Wind Engineering and Industrial Aerodynamics*, 218, 104778. <https://doi.org/10.1016/j.jweia.2021.104778>