

TECHNICAL UNIVERSITY OF CRETE  
School of Production Engineering & Management



DOCTORAL THESIS

---

**Development of computational tools, based on Radial  
Basis Functions and Differential Evolution, for the  
parametric design of aeroelastic systems**

---

**Giorgos STROFYLAS**

*A dissertation submitted in partial fulfillment of the  
requirements for the degree of Doctor of Philosophy  
(Ph.D.)*

Supervisor: **Prof. I.K. Nikolos**

Chania, April 2021

The present thesis is approved by the following jury:

**Advisory Committee:**

---

**Dr. Ioannis K. Nikolos (Supervisor)**

Professor, School of Production Engineering and Management,  
Technical University of Crete, Chania, Greece

**Dr. Spyridon Voutsinas (Member of Advisory Committee)**

Professor, School of Mechanical Engineering,  
National Technical University of Athens, Athens, Greece

**Dr. Kyriakos Giannakoglou (Member of Advisory  
Committee)**

Professor, School of Mechanical Engineering,  
National Technical University of Athens, Athens, Greece

**Thesis Examination Committee:**

---

**Dr. George Karatzas**

Professor, School of Environmental Engineering,  
Technical University of Crete, Chania, Greece

**Dr. Anargiros Delis**

Associate Professor, School of Production Engineering and Management,  
Technical University of Crete, Chania, Greece

**Dr. Fotios Kanellos**

Associate Professor, School of Electrical and Computer Engineering,  
Technical University of Crete, Chania, Greece

**Dr. George Arampatzis**

Assistant Professor, School of Production Engineering and Management,  
Technical University of Crete, Chania, Greece

# Abstract

Ph.D. Thesis

## **Development of computational tools, based on Radial Basis Functions and Differential Evolution, for the parametric design of aeroelastic systems**

by Giorgos STROFYLAS

In this study the development and validation of methodologies and computational tools, allowing for the effective design, optimization and numerical simulation of wind turbines is reported. Specifically, a software tool named "T4T" (Tools for Turbomachinery) is developed for the parametric design of the external surface and the internal geometry of horizontal-axis wind turbine blades, which is fully parametric and customizable, allowing the user for defining the internal blade structure, including shear webs. Moreover, the software can be used in an automated way (batch mode) to produce several candidate geometries in an optimization cycle, while retaining its topology unchanged. Regarding the use of an optimization methodology in the aforementioned wind turbine design loop, a parallel, synchronous/asynchronous, metamodel-assisted Differential Evolution algorithm is developed. Subsequently, a specialized surface reconstruction methodology is implemented, for the geometry definition of a wind turbine blade as a single Non-Uniform Rational B-Spline (NURBS) surface, from a target set of data points provided in the form of a surface triangular mesh. For the parameterization of the blade surface the dedicated blade geometry modelling software "T4T" is used. The shape reconstruction of the blade surface is formulated as an optimization procedure, which is realized with the aforementioned Differential Evolution algorithm. At next, a methodology for the deformation of computational grids, based on Radial Basis Functions (RBFs), is reported. Additionally, an approach for the acceleration of (RBF)-based mesh deformation procedure via the reduction of the surface points is developed, considering agglomeration of surface nodes' control volumes. It relies on the strategy followed by the corresponding multigrid methods, aiming to accelerate numerical solutions of fluid flow, radiative heat transfer, etc. Finally, a partitioned Fluid-Structure Interaction (FSI) methodology is developed, based on the RBFs Partition of Unity (PoU) method. With this approach the conservation of energy, momentum and force is ensured over the interface of the flow and structural grids as a result of the radial functions' properties. The use of the PoU methodology improves the efficiency of the data transfer procedure providing simultaneously a physical formulation of the force distribution.

# Περίληψη

Σκοπός της παρούσας Διδακτορικής Διατριβής είναι η ανάπτυξη και η επικύρωση μεθοδολογιών και υπολογιστικών εργαλείων, τα οποία επιτρέπουν τον αποτελεσματικό σχεδιασμό, τη βελτιστοποίηση και την αριθμητική προσομοίωση πτερυγώσεων ανεμογεννητριών. Συγκεκριμένα, αναπτύχθηκε ένα λογισμικό με το όνομα "T4T" (Tools for Turbomachinery - Εργαλεία για στροβιλομηχανές) για τον παραμετρικό σχεδιασμό τόσο της εξωτερικής επιφάνειας των πτερυγίων ανεμογεννητριών οριζοντίου άξονα όσο και της εσωτερικής τους δομής. Η όλη διαδικασία είναι πλήρως παραμετρική και επιτρέπει στον χρήστη να ορίσει πτερύγια διαφόρων τύπων. Επιπλέον, το λογισμικό παρέχει τη δυνατότητα αυτοματοποίησης των διαδικασιών του (batch mode) έτσι ώστε να μπορεί να χρησιμοποιηθεί αυτόματα σε έναν κύκλο βελτιστοποίησης και να παράγει διαφορετικές υποψήφιες γεωμετρίες, διατηρώντας παράλληλα αμετάβλητη την τοπολογία τους. Όσον αφορά στη μεθοδολογία βελτιστοποίησης, που χρησιμοποιείται στον προαναφερθέντα κύκλο σχεδιασμού των ανεμογεννητριών, αναπτύχθηκε ένας παράλληλος, σύγχρονος/ ασύγχρονος, Διαφορικός Εξελικτικός αλγόριθμος, που κάνει χρήση μεταμοντέλων (Differential Evolution (DE) algorithm). Στη συνέχεια, αναπτύχθηκε μια εξειδικευμένη μεθοδολογία ανακατασκευής επιφανείας πτερυγίων ανεμογεννητριών, από ένα σύνολο σημείων που παρέχονται με τη μορφή ενός τριγωνικού επιφανειακού πλέγματος. Η μέθοδος υπολογίζει όλες τις απαραίτητες παραμέτρους που απαιτούνται από το λογισμικό "T4T" για την περιγραφή της επιφάνειας του πτερυγίου ως μια μοναδική επιφάνεια NURBS. Στη συνέχεια παρουσιάζεται μια μεθοδολογία για την παραμόρφωση υπολογιστικών πλεγμάτων η οποία βασίζεται στη μέθοδο παρεμβολής με χρήση των ακτινικών συναρτήσεων βάσης (Radial Basis Functions - RBFs). Επιπλέον, αναπτύχθηκε μία προσέγγιση για την επιτάχυνση της διαδικασίας παραμόρφωσης πλεγμάτων με χρήση RBFs μέσω της τεχνικής μείωσης των επιφανειακών κόμβων, λαμβάνοντας υπόψη τη συσσωμάτωση (agglomeration) των γειτονικών όγκων ελέγχου. Βασίζεται στη στρατηγική που ακολουθείται από αντίστοιχες μεθόδους πολυπλέγματος (multigrid) που στοχεύουν στην επιτάχυνση της αριθμητικής επίλυσης προβλημάτων ροής ρευστού και μετάδοσης θερμότητας μέσω ακτινοβολίας κ.λπ. Τέλος, αναπτύχθηκε μία μεθοδολογία για την αριθμητική επίλυση προβλημάτων αλληλεπίδρασης ρευστού-στερεού (Fluid Structure Interaction - FSI), η οποία βασίζεται στη μέθοδο των RBFs και της τεχνικής του διαμερισμού της μονάδας (Partition of Unity - PoU). Με αυτήν την προσέγγιση, εξασφαλίζεται η αρχή διατήρησης της ενέργειας, της ορμής και της δύναμης κατά την μεταφορά της πληροφορίας μεταξύ των πλεγμάτων του ρευστού και του στερεού. Η χρήση της μεθοδολογίας PoU βελτιώνει την αποτελεσματικότητα της διαδικασίας μεταφοράς δεδομένων, παρέχοντας ταυτόχρονα μια φυσική διαμόρφωση της κατανομής των δυνάμεων.

# Acknowledgements

It's my pleasant duty to express my sincere regards and grateful obligation to all those who have contributed to the accomplishment of my research work.

First and foremost, I would like to express my deep gratitude to Professor Ioannis K. Nikolos for the scientific supervision of this thesis and for introducing me to the fascinating field of CFD. His academic excellence, valuable expertise and guidance, which was endless during these years, assisted in order to make this thesis an interesting blend of different methodologies and computational tools, applicable to a broad range of various specialized fields of engineering. I am deeply grateful for his inspiration, encouragement, kindness and ethics, which he showed during the entire period of my research. During the last seven years, he assisted me in many different ways, towards and beyond academic and I hope that he will continue being part of my professional and personal life.

I would like also to thank Professor Spyridon Voutsinas and Kyriakos Giannakoglou for their valuable help and support as co-advisors, as well as all the members of the examination committee, honouring me by accepting to participate to it.

Many thanks to Dr. Georgios N. Lygidakis, Dr. Sotirios S. Sarakinos and Mr. Stavros Leloudas, for all the work we did together and especially for sharing their knowledge and experience with me.

The completion of this thesis is also owed to my lovely wife Kallirroï, for her support, psychological contribution and confidence in me during all these years.

This work was funded by the European Union (European Social Fund - ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: THALES: Reinforcement of the interdisciplinary and/or inter-institutional research and innovation - Project: "Development of know-how on the aeroelastic analysis & design-optimization of wind turbines - WIND-FSI". Investing in knowledge society through the European Social Fund.

# Contents

<b>Abstract</b> .....	<b>iii</b>
<b>Περίληψη</b> .....	<b>iv</b>
<b>Acknowledgements</b> .....	<b>v</b>
<b>Contents</b> .....	<b>vi</b>
<b>Chapter 1: Introduction</b> .....	<b>1</b>
1.1 Objectives and Outline .....	1
1.2 Contributions .....	2
1.3 List of publications.....	3
1.4 Literature review .....	7
1.4.1 Design and surface reconstruction of wind turbine blades .....	7
1.4.2 Mesh deformation .....	12
1.4.3 Fluid Structure Interaction.....	16
<b>Chapter 2: Parametric design of wind turbine blades using T4T</b> .....	<b>19</b>
2.1 Introduction .....	19
2.2 Blade surface .....	20
2.3 Internal structure.....	24
2.3.1 Thickness Assignment.....	24
2.3.2 Shear Webs.....	28
2.4 Sample 3D blade design.....	30
2.4.1 Geometry definition.....	30
2.4.2 Mesh generation.....	32
2.5 Conclusions.....	35
<b>Chapter 3: The Parallel Asynchronous Differential Evolution Algorithm</b> .....	<b>36</b>
3.1 Introduction .....	36
3.2 The Differential Evolution Algorithm.....	36
3.3 The combined use of surrogate models .....	38
3.4 Parallel implementation.....	39
<b>Chapter 4: Reverse engineering of wind turbine blade surface using Differential Evolution</b> .....	<b>43</b>
4.1 Introduction .....	43
4.2 Blade Surface parametrization .....	43

4.3	Fitness function formulation .....	46
4.4	Validation.....	48
4.4.1	The 1 <sup>st</sup> test case .....	48
4.4.2	The 2 <sup>nd</sup> test case .....	51
4.5	Conclusions.....	56
<b>Chapter 5: Mesh Deformation .....</b>		<b>57</b>
5.1	Introduction .....	57
5.2	Application of Radial Basis Functions methodology to mesh deformation.....	57
5.2.1	Radial Basis Functions (RBFs) interpolation .....	57
5.2.2	The agglomeration strategy .....	60
5.2.3	Validation.....	67
5.3	Conclusions.....	98
<b>Chapter 6: Fluid Structure Interaction methodology .....</b>		<b>100</b>
6.1	Introduction .....	100
6.2	Coupling procedure.....	100
6.2.1	Data transfer scheme .....	101
6.2.2	Efficiency improvement - Partition of Unity (PoU) .....	103
6.3	Mesh deformation.....	103
6.4	Computational software.....	104
6.4.1	Flow Solver.....	104
6.4.2	Structural Solver.....	105
6.4.3	Coupling software.....	105
6.5	Numerical results.....	116
6.5.1	Commonwealth Advisory Aeronautical Research Council (CAARC).....	116
6.5.2	Bridge case .....	125
6.5.3	NREL offshore wind turbine .....	135
6.6	Conclusions.....	158
<b>Chapter 7: Conclusions and Future Work .....</b>		<b>159</b>
7.1	Concluding remarks .....	159
7.2	Further research .....	161
<b>Appendix A.....</b>		<b>162</b>
A.1	Free Form Deformation.....	162
A.2	Wind Turbine Blade.....	164
<b>Appendix B.....</b>		<b>168</b>
B.1	NASA Common Research Model (CRM) .....	168

B.2 Wind Turbine Blade.....	171
<b>References.....</b>	<b>175</b>

# List of Figures

FIGURE 1.1 (a) Infographic of the computational tools, developed within the TuboLab-TUC, for the design and optimization of wind turbine blades. ....	5
FIGURE 1.1 (b) Infographic of the computational tools, developed in this thesis, for the design and optimization of wind turbine blades. ....	6
FIGURE 2.1: Mean camber line and inscribed circles between pressure and suction sides.....	21
FIGURE 2.2: Modified trailing edge thickness.....	22
FIGURE 2.3: 3-dimensional blade along with its cross-sections.....	23
FIGURE 2.4: Blade surface sectioned in 10 regions. ....	24
FIGURE 2.5: Blade region split in 5 faces. ....	25
FIGURE 2.6: Different laminate thicknesses.....	26
FIGURE 2.7: Solid region with solid tail. ....	27
FIGURE 2.8: Solid tail formation for a blade section. ....	27
FIGURE 2.9: Definition of two shear webs. ....	28
FIGURE 2.10: Flowchart of the blade definition procedure. ....	29
FIGURE 2.11: NREL 5m <sub>rw</sub> blade surface.....	31
FIGURE 2.12: Cross-section of a solid blade with solid tail and 2 shear webs.....	32
FIGURE 2.13: Blade mesh with solid tetrahedral elements, focused on trailing edge. ....	33
FIGURE 2.14: Blade mesh with solid hex elements.....	33
FIGURE 2.15: Three first modes of a blade model with a fixed root.....	34
FIGURE 3.1: Flowchart of the asynchronous implementation of the parallel Differential Evolution algorithm.....	42
FIGURE 4.1: Airfoil construction procedure. ....	45
FIGURE 4.2: Mean camber line design parameters. ....	47
FIGURE 4.3: Illustration of the upper and lower permitted bounds for the blade angles, for the 1st test case. ....	48
FIGURE 4.4: (a) Target surface mesh; (b)-(d) Reconstructed surface. ....	49

FIGURE 4.5: The convergence history for the best and worst value of the fitness function: (a) no ANN; (b) both ANNs. (c) Comparison of the two approaches (best fitness with respect to exact evaluations). .....	50
FIGURE 4.6: Distribution of the surface error: (a)-(b) front and back views for the case without ANN; (c)-(d) front and back views for the surrogate-assisted case. ....	51
FIGURE 4.7: Upper and lower bounds for the control points of each section.....	51
FIGURE 4.8: Upper and lower permitted bounds for the chord of each section (the middle one is the target blade). ....	52
FIGURE 4.9: The convergence history for the best and worst value of the fitness function: (a) no ANN, (b) both ANNs. (c) Comparison of the two approaches (best fitness with respect to exact evaluations). .....	53
FIGURE 4.10: (a) Target surface mesh; (b)-(d) Reconstructed surface. ....	54
FIGURE 4.11: Distribution of the surface error: (a)-(b) front and back views for the case without ANN; (c)-(d) front and back views for both ANNs. ....	55
FIGURE 5.1: Generation of a supernode P via agglomeration.....	60
FIGURE 5.2: Control areas on the surface of the initial and agglomerated grids, representing a cubic domain. ....	62
FIGURE 5.3: Flowchart of the proposed agglomeration procedure. ....	63
FIGURE 5.4: Progress of the agglomeration front and fusion procedure. In the bottom-left illustration the seed nodes for the next agglomeration step are marked with an "S". ....	65
FIGURE 5.5: Schematic representation of supernodes coordinates' computation process. ....	66
FIGURE 5.6: Mesh density on aircraft and symmetry surfaces for the NASA CRM case. ....	67
FIGURE 5.7: Parabolic deformation of the NASA CRM aircraft wing (max deflection 5 m at wing tip). .....	68
FIGURE 5.8: Mesh density on the wing surface of the initial and agglomerated control area grids (NASA CRM case). ....	69
FIGURE 5.9: Super-areas' centers (yellow-coloured) at the coarsest grid (a) and their distribution over the wing surface (b) (NASA CRM case). ....	70
FIGURE 5.10: Distributions of computation time of the solution (a) and evaluation (b) stages, as well as of average (c) and maximum (d) wing surface errors for different agglomeration levels (NASA CRM case).....	72
FIGURE 5.11: Distribution of wing surface error for different numbers of agglomeration levels (CRM case).....	73
FIGURE 5.12: Convergence histories for density and velocity (left), and turbulent kinetic energy (right) for the deformed grids, extracted with (Level 4) and without (Initial) agglomeration of RBF centers (NASA CRM case). ....	75
FIGURE 5.13: Top view of aircraft surface dimensionless pressure contours of the deformed grids, derived with (left) and without (right) agglomeration of centers (NASA CRM case). ....	75

FIGURE 5.14: Distribution of pressure coefficient at different wing span-wise sections (a: 50.24%, b: 72.68%, c: 84.56% and d: 95.00%) of the deformed grids, obtained with (Level 4) and without (Initial) agglomeration of RBF centers (NASA CRM case).....	76
FIGURE 5.15: Deformation of the NASA CRM aircraft wing (max bending deflection 1.5 m at wing tip, max rotation of 10° at wing tip). .....	77
FIGURE 5.16: Super-areas' centers (yellow-coloured) at the coarsest grid, extracted via the approach with the loose constraints (NASA CRM case with additional torsional deformation). .....	78
FIGURE 5.17: Distribution of wing surface error for different constraint angles (CRM case with additional torsional deformation).....	79
FIGURE 5.18: Deformed mesh of NREL 5-MW turbine blade (a) and a close-up view at the tip region (b). .....	81
FIGURE 5.19: Mesh density on the blade surface of the initial (a-b) and agglomerated (c-f) surface grids (NREL case).....	82
FIGURE 5.20: Super-areas' centers (yellow-coloured) at the coarsest resolution (NREL case).....	83
FIGURE 5.21: Distributions of computation time of the solution (a) and evaluation (b) stages, as well as of average (c) and maximum (d) blade surface errors for different agglomeration levels (NREL case). .....	84
FIGURE 5.22: Distribution of blade surface error for different agglomeration levels (NREL case)....	85
FIGURE 5.23: Control areas of supernodes (a) and error distribution on front (b) and back (c) blade surface. ....	86
FIGURE 5.24: Parabolic deformation of aircraft wing (max deflection 6 m at wing tip) (DLR-F6 case). .....	87
FIGURE 5.25: Mesh density on the wing surface of the initial and agglomerated control area grids (DLR-F6 case).....	88
FIGURE 5.26: Super-areas' centers (yellow-colored) at the coarsest generated grid and their distribution on the wing surface (DLR-F6 case). .....	89
FIGURE 5.27: Distribution of (deformed wing) surface error at the fourth agglomeration level (top and bottom views) (DLR-F6 case). .....	90
FIGURE 5.28: Computation time of the solution and evaluation stages for different agglomeration levels (DLR-F6 case). .....	91
FIGURE 5.29: Convergence history curves for density, velocity in x-direction and turbulent kinetic energy of deformed grids (initial and 4th agglomeration level) (DLR-F6 case). .....	92
FIGURE 5.30: Contours of dimensionless pressure on the aircraft surface of deformed grids (initial and 4th agglomeration level) (DLR-F6 case).....	94
FIGURE 5.31: Distribution of pressure coefficient at wing span-wise section 63.8% of deformed grids (initial and 4th level) (DLR-F6 case).....	94
FIGURE 5.32: Distribution of pressure coefficient at wing span-wise section 84.7% of deformed grids (initial and 4th level) (DLR-F6 case).....	95

FIGURE 5.33: Distribution of wing surface error (top and bottom views) at the fourth agglomeration level (refined grid) (DLR-F6 case). .....	96
FIGURE 5.34: Computation time of the solution and evaluation stages for the refined grid (at each agglomeration level) (DLR-F6 case). .....	96
FIGURE 5.35: Contours of dimensionless pressure on the aircraft surface of deformed refined grids (initial and 4th level) (DLR-F6 case). .....	97
FIGURE 5.36: Distribution of pressure coefficient at wing span-wise section 84.7% of deformed refined grids (initial and 4th level) (DLR-F6 case). .....	98
FIGURE 6.1: Flow chart of the FSI coupling scheme. ....	107
FIGURE 6.2: Main window – “Mesh” tab. ....	109
FIGURE 6.3: Main window – “FSI” tab. ....	110
FIGURE 6.4: Hidden Symmetry and Up closures of the flow mesh. ....	111
FIGURE 6.5: (Top) Deformed structure as calculated by the structural solver. (Bottom) Deformed flow mesh boundary surface computed by RBF's PoU interpolation. ....	112
FIGURE 6.6: Defining agglomeration levels for surface mesh coarsening. ....	113
FIGURE 6.7: Mesh density on the cylindrical surface of the five agglomerated (a-e) control volume grids .....	113
FIGURE 6.8: Clipping plane -section of a flow mesh. ....	114
FIGURE 6.9: Pseudocolor representation of flow mesh boundary surface with colormap. ....	114
FIGURE 6.10: Vector presentation on the structural grid, illustrating the displacements of the corresponding nodes. ....	115
FIGURE 6.11: Sketch of CAARC test case setup. ....	116
FIGURE 6.12: Computational mesh around the CAARC standard tall building model. ....	116
FIGURE 6.13: Inlet velocity profile (left) and turbulence intensity profile (right) (CAARC case)...	117
FIGURE 6.14: Mean pressure coefficient on the CAARC building wall at $z=3/2H$ . ....	118
FIGURE 6.15: Mean pressure coefficient distribution on the CAARC building walls and surrounding “ground” .....	118
FIGURE 6.16: Comparison of force component distribution between CFD (left) and CSD (right) meshes (CAARC case). .....	120
FIGURE 6.17: Von Misses, $S_{XX}$ , $S_{YY}$ , $S_{ZZ}$ stresses of the CAARC standard tall building model. ....	121
FIGURE 6.18: Deformation of the structural grid (a) and corresponding deformation of the CFD boundary grid (b) (CAARC case). ....	122
FIGURE 6.19: Fine and coarser grids generated with the agglomeration procedure (CAARC case). .....	122
FIGURE 6.20: Before (a) and after (b) deformation states of the CFD grid (CAARC case). ....	123

FIGURE 6.21: Mean pressure coefficient and velocity streamlines around the deformed CFD grid (CAARC case).....	123
FIGURE 6.22: Realistic CAARC building model (structural members: green color, walls: red color). .....	124
FIGURE 6.23: Von Misses, $S_{xx}$ , $S_{yy}$ , $S_{zz}$ stresses for the CAARC realistic building model.....	125
FIGURE 6.24: General view of the geometry (bridge case).....	125
FIGURE 6.25: Flow (gray) and structural (yellow) computational grids (bridge case). ....	126
FIGURE 6.26: Pressure contours along with velocity streamlines at the YZ plane (bridge case)....	127
FIGURE 6.27: Pressure contours along with velocity streamlines at the XY plane, under the bridge deck (bridge case). ....	127
FIGURE 6.28: Pressure contours along with velocity streamlines at the XY plane at the bridge deck (bridge case). ....	128
FIGURE 6.29: Velocity streamlines over the bridges geometry. ....	129
FIGURE 6.30: Pressure contours on the solid surfaces (bridge case). ....	129
FIGURE 6.31: Force components' distribution at CFD (left) and CSD (right) meshes (bridge case). .....	130
FIGURE 6.32: Magnified deformation of the bridge under the wind load (bridge case).....	131
FIGURE 6.33: Computed von Misses stresses of the bridge (bridge case).....	132
FIGURE 6.34: Computed $S_{xx}$ , $S_{yy}$ and $S_{zz}$ stresses on the bridge (bridge case). ....	133
FIGURE 6.35: Control areas of the coarser grid generated with the agglomeration procedure (bridge case).....	134
FIGURE 6.36: Fluid (gray) and structural (yellow) computational grids prior (a) and after (b) the fluid mesh deformation step (bridge case). ....	135
FIGURE 6.37: The NREL-5MW Baseline Wind Turbine blade geometry. ....	137
FIGURE 6.38: General mesh view of the NREL blade. ....	138
FIGURE 6.39: Details of the mesh of the sections 1, 4, 8 (from left to right) (NREL case). ....	139
FIGURE 6.40: Tetrahedral C3D4 elements (Dhondt, 2014).....	140
FIGURE 6.41: Distributions of loads on $XX'$ axis at the blade during the first FSI iteration (front view): (a) coarse aerodynamic grid, (b) refined aerodynamic grid, (c) structural grid (NREL case). 142	142
FIGURE 6.42: Distributions of loads on $XX'$ axis at the blade during the first FSI iteration (back view): (a) coarse aerodynamic grid, (b) refined aerodynamic grid, (c) structural grid (NREL case).....	143
FIGURE 6.43: Distributions of loads on $YY'$ axis at the blade during the first FSI iteration (front view): (a) coarse aerodynamic grid, (b) refined aerodynamic grid, (c) structural grid (NREL case).....	144
FIGURE 6.44: Distributions of loads on $YY'$ axis at the blade during the first FSI iteration (back view): (a) coarse aerodynamic grid, (b) refined aerodynamic grid, (c) structural grid (NREL case).....	145

FIGURE 6.45: Distributions of loads on ZZ' axis at the blade during the first FSI iteration (front view): (a) coarse aerodynamic grid, (b) refined aerodynamic grid, (c) structural grid (NREL case).....	146
FIGURE 6.46: Distributions of loads on ZZ' axis at the blade during the first FSI iteration (back view): (a) coarse aerodynamic grid, (b) refined aerodynamic grid, (c) structural grid (NREL case).....	147
FIGURE 6.47: Blade's deformation at YZ plane during the first FSI iteration (NREL case). ....	148
FIGURE 6.48: Blade's deformation at ZX plane during the first FSI iteration (NREL case). ....	148
FIGURE 6.49: Blade's deformation at XZ plane during the first FSI iteration (NREL case). ....	149
FIGURE 6.50: Blade's deformation at YX plane during the first FSI iteration (NREL case). ....	149
FIGURE 6.51: 3D Blade's deformation during the first FSI iteration (NREL case). ....	150
FIGURE 6.52: Distributions of the deformation at the blade during the first FSI iteration at XX' axis: (a) structural grid, (b) refined aerodynamic grid, (c) coarse aerodynamic grid (NREL case).....	151
FIGURE 6.53: Distributions of the deformation at the blade during the first FSI iteration at YY' axis: (a) structural grid, (b) refined aerodynamic grid, (c) coarse aerodynamic grid (NREL case).....	152
FIGURE 6.54: Distributions of the deformation at the blade during the first FSI iteration at ZZ' axis: (a) structural grid, (b) refined aerodynamic grid, (c) coarse aerodynamic grid (NREL case).....	153
FIGURE 6.55: Distributions of the velocities at the blade during the first FSI iteration at XX' axis: (a) structural grid, (b) refined aerodynamic grid, (c) coarse aerodynamic grid (NREL case). ....	154
FIGURE 6.56: Distributions of the velocities at the blade during the first FSI iteration at YY' axis: (a) structural grid, (b) refined aerodynamic grid, (c) coarse aerodynamic grid (NREL case). ....	155
FIGURE 6.57: Distributions of the velocities at the blade during the first FSI iteration at ZZ' axis: (a) structural grid, (b) refined aerodynamic grid, (c) coarse aerodynamic grid (NREL case). ....	156
FIGURE 6.58: Displacements of the blade tip along the Z axis (NREL case).....	157
FIGURE 6.59: Displacements of the blade tip along the X axis (NREL case). ....	157
FIGURE A.1: Initial grids (yellow – CFD grid), (grey – CSD grid) of NREL 5-MW turbine blade. ....	164
FIGURE A.2: Deformed grids (yellow – CFD grid), (grey – CSD grid) of NREL 5-MW turbine blade. ....	165
FIGURE A.3: Close-up view at the tip region of the deformed grids (yellow – CFD grid), (grey – CSD grid) of NREL 5-MW turbine blade. ....	166
FIGURE A.4: Deformed CSD grid of NREL 5-MW turbine blade. ....	167
FIGURE B.1: Comparison of the quality metrics: (a) aspect ratio, (b) skewness, (c) Jacobian and (d) orthogonality, for the un-deformed and the deformed grids, using agglomerated control areas (Level 4, NASA CRM case). ....	171
FIGURE B.2: Comparison of quality metrics: (a) aspect ratio, (b) skewness, (c) Jacobian and (d) orthogonality, for the un-deformed and the deformed grids, using agglomerated control areas (Level 4, NREL case). ....	174

# List of Tables

TABLE 2.1: Distributed blade aerodynamic properties.....	30
TABLE 2.2: Eigenvalues of the fixed root blade. ....	34
TABLE 4.1: List of free parameters for the definition of each cross section of a blade.....	46
TABLE 4.2: Wall-clock computation times and surface errors of the proposed surface reconstruction procedure, with and without surrogate assistance. ....	49
TABLE 4.3: Permitted range of “thickness distribution” for both sides of the mean camber line.....	52
TABLE 4.4: Wall-clock computation times and surface errors of the proposed surface reconstruction procedure, with and without surrogate assistance. ....	52
TABLE 5.1: Number of wing surface nodes at each agglomeration level (NASA CRM case).....	68
TABLE 5.2: Wall-clock computation time and surface errors of the RBF-mesh deformation procedure for different agglomeration levels (NASA CRM case). ....	71
TABLE 5.3: Number of wing surface nodes at each agglomeration level (NASA CRM case with additional torsional deformation).....	78
TABLE 5.4: Wall-clock computation time and surface errors of the RBF-mesh deformation procedure employing strict and loose constraints (NASA CRM case with additional torsional deformation). ...	78
TABLE 5.5: Number of surface nodes at each agglomeration level (NREL case). ....	80
TABLE 5.6: Computation time and surface errors for different agglomeration levels (NREL case)....	83
TABLE 5.7: Number of surface nodes at each agglomeration level employing the approach with the loose constraints (NREL case).....	86
TABLE 5.8: Computation time and surface errors employing the coarsest agglomeration level, extracted via the approach with the strict and loose constraints (NREL case). ....	87
TABLE 5.9: Surface errors for different agglomeration levels (DLR-F6 case). ....	89
TABLE 5.10: Wall-clock computation time for different agglomeration levels (DLR-F6 case). ....	90
TABLE 5.11: Surface errors of the refined grid for different agglomeration levels (DLR-F6 case).....	95
TABLE 5.12: Wall-clock computation time for different agglomeration levels (refined grid) (DLR-F6 case).....	96
TABLE 6.1: Flow field properties (CAARC case). ....	119
TABLE 6.2: Mechanical properties of the structure (CAARC case). ....	121
TABLE 6.3: Comparison of normalized maximum displacement (CAARC case).....	121
TABLE 6.4: Mechanical properties of the structure (realistic CAARC building model). ....	124

TABLE 6.5: Mechanical properties of the structure (bridge case).....	131
TABLE 6.6: Gross properties chosen for the NREL-5MW Baseline Wind Turbine (Jonkman et al., 2009).....	136
TABLE 6.7: Distributed Blade Aerodynamic and Geometrical Properties for the NREL blade (Charalampous et al., 2015).....	140
TABLE 6.8: NREL blade materials properties (NREL case).....	141
TABLE B.1: Aspect Ratio of the un-deformed and deformed grids, with (Level 4) and without agglomeration of RBF-centers (NASA CRM case).....	169
TABLE B.2: Skewness of the un-deformed and deformed grids, with (Level 4) and without agglomeration of RBF-centers (NASA CRM case).....	169
TABLE B.3: Jacobian of the un-deformed and deformed grids, with (Level 4) and without agglomeration of RBF-centers (NASA CRM case).....	170
TABLE B.4: Orthogonality of the un-deformed and deformed grids, with (Level 4) and without agglomeration of RBF-centers (NASA CRM case).....	170
TABLE B.5: Aspect Ratio of the un-deformed and deformed grids, with agglomeration (Level 4) of RBF-centers (NREL case).....	172
TABLE B.6: Skewness of the un-deformed and deformed grids, with agglomeration (Level 4) of RBF-centers (NREL case).....	172
TABLE B.7: Jacobian of the un-deformed and deformed grids, with agglomeration (Level 4) of RBF-centers (NREL case).....	173
TABLE B.8: Orthogonality of the un-deformed and deformed grids, with agglomeration (Level 4) of RBF-centers (NREL case).....	173

# List of Abbreviations

<b>Abbreviation/Term</b>	<b>Definition</b>
<b>ANN</b>	<b>Artificial Neural Network</b>
<b>CAD</b>	<b>Computer Aided Engineering</b>
<b>CFD</b>	<b>Computational Fluid Dynamics</b>
<b>CSD</b>	<b>Computational Structural Dynamics</b>
<b>DE</b>	<b>Differential Evolution</b>
<b>EA</b>	<b>Evolutionary Algorithm</b>
<b>FFD</b>	<b>Free Form Deformation</b>
<b>FSI</b>	<b>Fluid Structure Interaction</b>
<b>MLP</b>	<b>Multi-layer Perceptron</b>
<b>MPI</b>	<b>Message Passing Interface</b>
<b>NURBS</b>	<b>Non-Uniform Rational B-Spline</b>
<b>PDE</b>	<b>Partial Differential Equation</b>
<b>RBF</b>	<b>Radial Basis Function</b>
<b>RBFN</b>	<b>Radial Basis Functions Network</b>
<b>T4T</b>	<b>Tools for Turbomachinery</b>

*To my family*  
*Kallirroï and Aris*



# Chapter 1: Introduction

## 1.1 Objectives and Outline

The objective of this work is the development of new computational tools, towards the more effective design and optimization of wind turbine rotor blades. Considering the aforementioned goal, the following objectives were accomplished, while additional attention was directed towards the improvement of the accuracy and efficiency of the developed software:

- Development of a stand-alone geometry parameterization tool for the rotor blades, compatible with the FSI (Fluid-Structure Interaction) analysis, for use within the optimization procedure of wind turbine rotor blades.
- Development of design-optimization procedures based on heuristic methods.
- Development of an optimization-based “inverse” design methodology, used to extract the optimal blade geometry (the output of the optimization process) in geometry “formats” compatible with commercial CAD systems.
- Development of a fully automated computational procedure for the deformation of the computational grids, along with the deformed geometry.
- Development of a coupling methodology for implementing and applying multidisciplinary simulations.
- Integration of the above tools (as well as Computational Fluid Dynamics and Computational Structural Dynamics tools) into a unified computational platform.

The contents of this thesis are organized as follows. In Chapter 2 the development of a new version of the software tool named "T4T" (Tools for Turbomachinery), for the parametric design of turbomachinery and wind turbine blades, is presented. The software allows the user to design the outer surface of a blade either by specifying some physical parameters for each blade section, or by directly interpolating a surface through a cloud of points. The new/enhanced version of "T4T" software, introducing also the definition of internal blade structure (with shear webs) for wind turbines rotors, is fully parametric and customizable. The derivatives of the surface geometrical points, with respect to the design variables used in "T4T" can be numerically computed in order to be used with an adjoint optimization method.

Chapter 3 describes the development of a metamodel-assisted Parallel Differential Evolutionary (DE) algorithm. In particular, the DE algorithm was developed based on the panmictic approach, using a unique population that is distributed among the processors, with a Master-Slave architecture. The parallelization of the DE algorithm was performed using the Message Passing Interface (MPI). Furthermore, the utilization of surrogate models within the DE algorithm enhanced its performance by substituting the computationally time-consuming exact evaluations of the fitness functions with low-cost approximations.

Chapter 4 describes a shape reconstruction methodology, which is designed to make the resulting blade definition consistent with the geometry parameterization tool "T4T". The shape reconstruction is formulated as an optimization procedure, based on the parallel, metamodel-assisted differential evolution (DE) algorithm.

Chapter 5, contains the development and implementation of a mesh deformation procedure, based on Radial Basis Functions (RBFs). Additionally, a new methodology has been developed in order to alleviate the increased computational time requirements of the method. A reduced surface point selection procedure was developed, considering agglomeration of surface nodes' control volumes. It relies on the strategy followed by the corresponding multigrid methods, aiming to accelerate numerical solutions of fluid flow, radiative heat transfer, etc., problems. The developed merging procedure demonstrated to be a highly efficient scheme.

Chapter 6, describes the development of an FSI coupling shell, aiming to facilitate the interaction between an open-source CSD (Computational Structural Dynamics) and an in-house academic CFD (Computational Fluid Dynamics) code. The spatial coupling is achieved using Radial Basis Functions (RBFs) interpolation, which enables point-based interaction, needing therefore no information for connectivities and, consequently, allowing for the utilization of different types or even intersecting structural and flow grids. Moreover, in order to accelerate the exchange procedure, the Partition of Unity (PoU) approach was adopted, which regards the decomposition of the examined problem into several smaller ones, to be solved independently and hence more efficiently.

Finally, Chapter 7 contains some conclusions and information on ongoing and future work.

## 1.2 Contributions

The contribution of this Thesis is in the development of methodologies and computational tools for the synergetic design, optimization and simulation of wind turbine blades (*FIGURE 1.1*). However, their application is not limited to the aforementioned engineering field, as it will be revealed in the following Chapters. The main contributions of this Thesis are listed below:

- The development and evaluation of a novel methodology (and the corresponding software) for the RBF-based mesh deformation, highly accelerated using an agglomeration strategy. The developed methodology renders the RBF-based mesh deformation practically applicable in real-world problems. This is considered as the main contribution of this Thesis.
- The development and evaluation of a methodology (and the corresponding software) for Fluid-Structure Interaction (FSI) simulations, based on the RBFs Partition of Unity (PoU) method for the exchange of data between the interacting flow and structural domains and grids. The methodology is independent of the grid types and the analyses software used in the computational procedure, as it will be demonstrated in the following chapters.
- The introduction of new computational tools in the in-house "T4T" (Tools for Turbomachinery) blade design software, for the parametric design of the external surface and the internal geometry of wind turbine blades. The software can be a part of an automated design optimization procedure.
- The development of an asynchronous parallel version of the existing synchronous serial metamodel-assisted Differential Evolution algorithm. This methodology is

currently used with success (in the context of a different Ph.D. Thesis) for the design optimization of wind turbines (among other applications).

- The development and evaluation of an optimization-based methodology for the reverse-engineering of wind turbine blades, based on “T4T” and the developed DE algorithm.

### 1.3 List of publications

The aforementioned contributions have been published in fourteen peer-reviewed Journal and Conference papers, listed below.

#### Journal Publications

1. **G.A. Strofylas**, K.N. Porfyri, I.K. Nikolos, A.I. Delis, and M. Papageorgiou, “Using Synchronous and Asynchronous Parallel Differential Evolution for Calibrating a Second-order Traffic Flow Model”, *Advances in Engineering Software*, 125, pp. 1-18, 2018.
2. S.N. Leloudas, **G.A. Strofylas**, and I.K. Nikolos, “Constrained Airfoil Optimization Using the Area-Preserving Free-Form Deformation”, *Aircraft Engineering and Aerospace Technology*, 90(6), pp. 914-926, 2018.
3. **G.A. Strofylas**, G.N. Lygidakis, I.K. Nikolos, “An Agglomeration Strategy for Accelerating RBF-Based Mesh Deformation”, *Advances in Engineering Software*, 107, pp. 13-37, 2017.

#### Publications in Conference Proceedings

4. S.N. Leloudas, G.N. Lygidakis, **G.A. Strofylas**, I.K. Nikolos, “Aerodynamic Shape Optimization of Diffuser Augmented Wind Turbine Shrouds Using Asynchronous Differential Evolution”, *Proceedings of the ASME International Mechanical Engineering Congress and Exposition, IMECE 2018*, 9-15 November, 2018, Pittsburgh, PA, USA, Paper No. IMECE2018-86820.
5. S.N. Leloudas, **G.A. Strofylas**, I.K. Nikolos, “Aerodynamic shape optimization of Diffuser Augmented Wind Turbine blades using asynchronous parallel Differential Evolution”, *9th GRACM International Congress on Computational Mechanics*, Chania, 4-6 June, 2018.
6. **G.A. Strofylas**, K.N. Porfyri, I.K. Nikolos, A. I. Delis, M. Papageorgiou, “Calibrating a Traffic Flow Model with Parallel Differential Evolution”, *Proceedings of the Fifth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*, P. Ivanyi, B.H.V. Topping and G. Varady (Editors), Civil-Comp Press, Stirlingshire, Scotland, 2017.
7. **G.A. Strofylas**, G.I. Mazanakis, S.S. Sarakinos, G.N. Lygidakis, I.K. Nikolos, "On the use of improved Radial Basis Functions methods in Fluid-Structure Interaction simulations", *Proceedings of the ASME 2016 International Mechanical Engineering Conference & Exposition, IMECE2016*, Nov. 11-17, Phoenix, Arizona, USA, Paper No. IMECE2016-66412.
8. **G.A. Strofylas**, G.I. Mazanakis, S.S. Sarakinos, G.N. Lygidakis, I.K. Nikolos, "Using improved Radial Basis Functions methods for fluid-structure coupling and mesh deformation", in M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (Eds.),

Proceedings of the ECCOMAS Congress 2016, VII European Congress on Computational Methods in Applied Science and Engineering, Crete Island, Greece, 5-10 June, 2016.

9. **G.A. Strofylas**, G.N. Lygidakis, I.K. Nikolos, "Accelerating RBF-based mesh deformation by implementing an agglomeration strategy", Proceedings of the ASME 2015 International Mechanical Engineering Conference & Exposition, IMECE2015, Nov. 13-19, Houston, Texas, USA, Paper No. IMECE2015-50902.
10. S.N. Leloudas, **G.A. Strofylas**, I.K. Nikolos, "Airfoil optimization using area-preserving free-form deformation", Proceedings of the ASME 2015 International Mechanical Engineering Conference & Exposition, IMECE2015, Nov. 13-19, Houston, Texas, USA, Paper No. IMECE2015-50904.
11. **G.A. Strofylas**, I.K. Nikolos, "Reverse engineering of wind turbine blade surface using differential evolution", Proceedings of the 4th International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering, Prague, Czech Republic, 1-4 Sept., 2015.
12. K.G. Charalampous, **G.A. Strofylas**, G.I. Mazanakis, I.K. Nikolos, "Wind turbine blade parametric design using Grasshopper<sup>R</sup>", 8th GRACM International Conference on Computational Mechanics, 12-15 July, 2015, Volos, Greece.
13. **G.A. Strofylas**, G.I. Mazanakis, I.K. Nikolos, "Wind turbine blade structure parameterization using T4T", Proceedings of the ASME 2014 International Mechanical Engineering Conference & Exposition, IMECE2014, Nov. 14-20, Montreal, Quebec, Canada, Paper No. IMECE2014-39674.
14. **G. Strofylas**, I.K. Nikolos, "Wind turbine blade design using T4T", 10th HSTAM International Congress on Mechanics, Chania, Crete, Greece, 25-27 May, 2013.

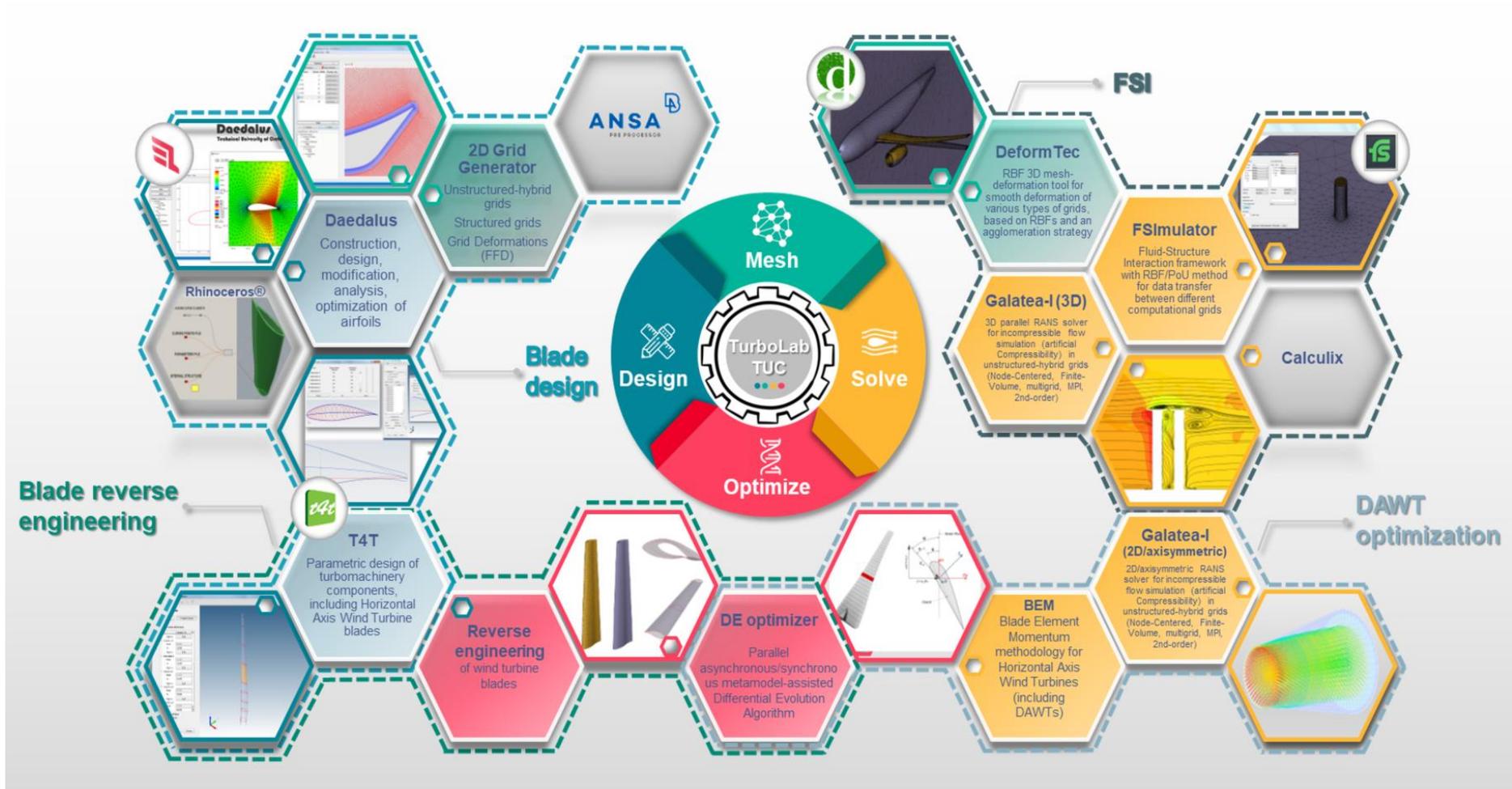


FIGURE 1.1 (a) Infographic of the computational tools, developed within the TurboLab-TUC, for the design and optimization of wind turbine blades.

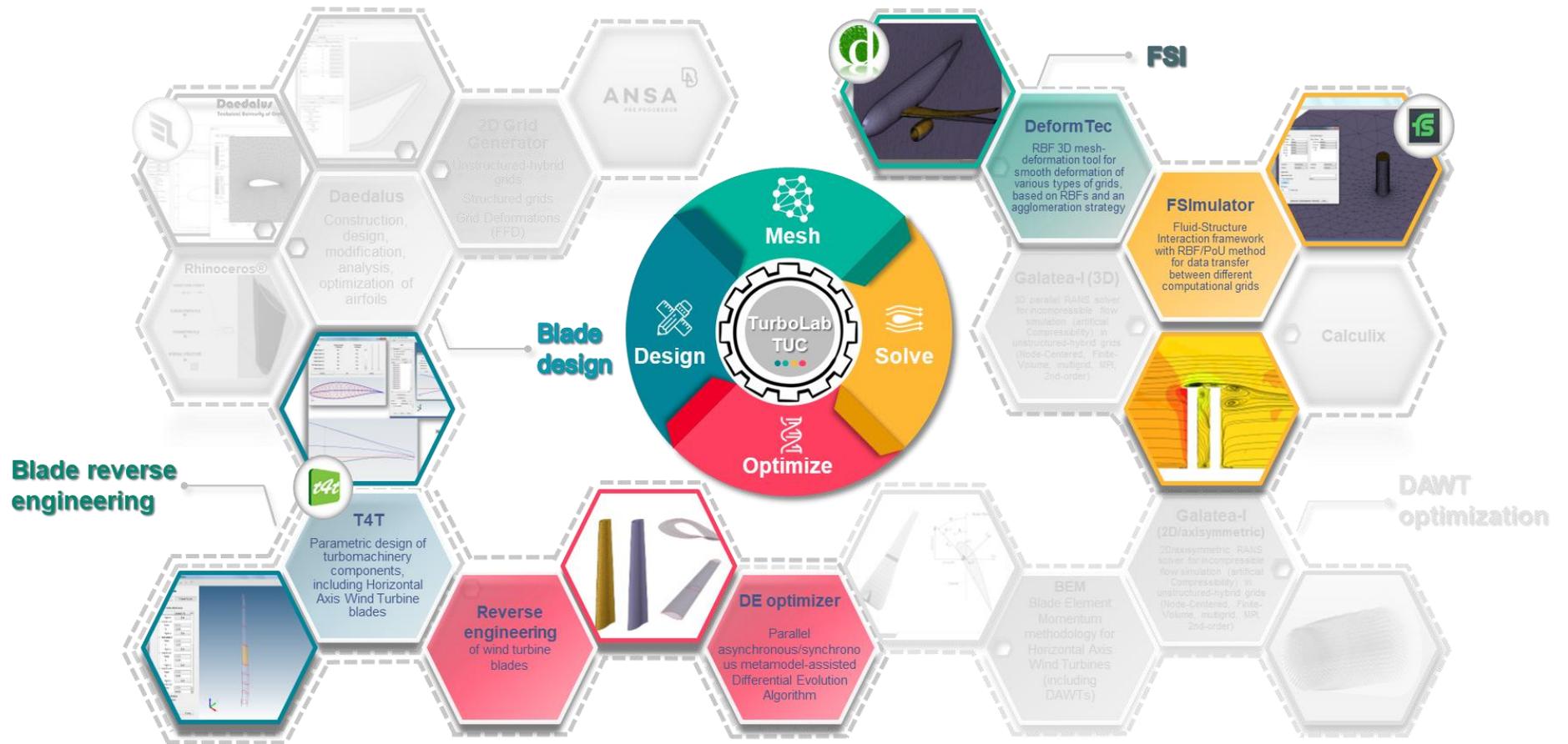


FIGURE 1.2 (b) Infographic of the computational tools, developed in this thesis, for the design and optimization of wind turbine blades.

## 1.4 Literature review

### 1.4.1 Design and surface reconstruction of wind turbine blades

#### 1.4.1.1 Parametric design

The design and manufacturing of optimized and cost-effective wind turbines requires reliable and advanced design tools that can model the dynamics and response of wind turbines in a comprehensive and fully integrated way. In particular, wind turbine blades design being very demanding is one of the most important aspects of wind turbine technology ("Gurit," 2013; Singh *et al.*, 2013). Designing wind turbine blades has to do with balancing aerodynamic performance with structural integrity. However, the requirements of contemporary wind turbine blade design have changed impressively over the last decades (Broek, 1982; Mandell *et al.*, 1992; Veers *et al.*, 1993; Samborsky & Mandell, 1996; Inomata *et al.*, 1999; Kong *et al.*, 2005, 2006; Yang *et al.*, 2013; Yang & Sun, 2013). The scale of the blades has increased dramatically as well as the corresponding cost (Ackermann & Söder, 2002; İlkılıç *et al.*, 2011; Leung & Yang, 2012). This has resulted in more attention being given to blade design and analysis.

All available computational methodologies are currently used for the design, analysis and optimization of wind turbine blades, such as FEM (Finite Element Method), CFD (Computational Fluid Dynamics), automated mesh generation and mesh deformation methodologies, and various types of optimization methods. Moreover, the geometry parameterization and definition is a complicated and demanding procedure that requires special attention, as the correct and efficient geometry parameterization is a vital prerequisite for an effective optimization procedure.

The airfoils used to construct wind turbine blades are standard and non-standard ones, with NACA type airfoils being widely used (Strofyas & Nikolos, 2013). Different methodologies have been reported in the open literature for the design and geometrical definition of wind turbines (Kamoun *et al.*, 2006; Henriques *et al.*, 2009; Zhang *et al.*, 2012; Liu *et al.*, 2013; Wang *et al.*, 2013). Moreover, extended effort has been spent to produce detailed experimental data and use them in parallel with numerical simulations (Kooijman *et al.*, 2003; Jørgensen *et al.*, 2004; Kong *et al.*, 2005; Jensen *et al.*, 2006; Jensen, 2008; Henriques *et al.*, 2009; Jonkman *et al.*, 2009; Malhotra, 2010; Malhotra *et al.*, 2012; Yang & Sun, 2013; Abd-Elhady *et al.*, 2014; Dervilis *et al.*, 2014).

The NREL (Jonkman *et al.*, 2009) (National Renewable Energy Laboratory) 5 MW Fast aeroelastic model project has played an important role in the recent research on this subject. It was based on the blades of DOWEC study, while the 62.6 m DOWEC's blade (Bulder *et al.*, 2001; Kooijman *et al.*, 2003) was truncated down to 61.5 m. The analysis of the blade was completed with the use of aero-servo-elastic FAST (Jonkman & Buhl Jr., 2005) code with AeroDyn routines (Laino & Hansen, 2002) and MSC.ADAMS code with A2AD (Hansen & Laino, 1998). Based on the above NREL study other related works have been developed (Cox & Echtermeyer, 2012; Lee, Lee, *et al.*, 2012; Lee, Park, *et al.*, 2012; Sayed *et al.*, 2012; Bumsuk *et al.*, 2013).

(Bumsuk *et al.*, 2013) developed a software tool for the optimum shape design of multi-MW wind turbine blades and the analysis of their performance; it includes aerodynamic shape design, performance analysis, pitch torque analysis and shape optimization for wind turbine blades. In order to verify the accuracy of the performance analysis results of the proposed software, the 5 MW blade designed by NREL was used as a reference, and the results were compared to the corresponding ones of the commercial software GH-Bladed (Bossanyi, 2003).

In the afore-mentioned software tool, after defining initial design parameters, such as blade diameter, rated speed, design tip speed ratio etc., baseline blade shape is generated by calculating chord length and twist angle at each calculation point set by the designer.

A concept for the design of a straight symmetrical blade for a small-scale vertical-axis wind turbine, using beam theories for analytical modeling and the commercial software ANSYS 11.0 ("ANSYS," 2013) for numerical modeling, was presented by Hameed and Afaq (Hameed & Afaq, 2013). Furthermore, Pérez-Arribas and Trejo-Vargas have presented a computer-based method for modeling the blades of horizontal axis turbines using B-spline curves and surfaces (Pérez-Arribas & Trejo-Vargas, 2012). The method begins with the creation of a 3D set of offsets that constitute the rotor blades based on a 2D definition of the airfoils, normally utilized in the design of different stations along the rotor blade. It uses geometrical parameters, such as the skew and rake or coning distribution. The method highlights the design of the blade's leading edge, which has a significant impact on the properties of the rotor, and separately models the trailing edge of the blades; a blade surface with a minimum number of control points is produced, which fits the provided data points below a prescribed tolerance.

The aim of QBLADE project (Marten & Wendler, 2013) is to provide an open source turbine calculation software that is seamlessly integrated into XFOIL, an airfoil design and analysis tool. The motivation is to create a one-solution software for the design and aerodynamic computation of wind turbine blades. This software, besides other features, contains a blade design module, which enables the user to define the blade shape in detail; the geometry of the 3D blade can be exported at standard "stl" format. Various types of commercial software are available, which can be used for the design of wind turbine blades. FOCUS6 includes a 3D structural blade modeler that allows users to interactively model rotor blades in detail; the interactive 3D visualization provides direct feedback of the design modifications. Blade definition is utilized by putting profiles in 3D space, defining lines and materials and finally specifying between which lines and profiles the section of material needs to be placed. Material definitions include layer thicknesses and actual material properties. Blade data can be viewed and exported both graphically and numerically; this includes export as a thick shell element mesh with full layup data, for analyses with various FEM software, as well as standard geometry exchange formats (STEP/IGES). Sandia's team (Griffith & Ashwill, 2011; Resor, 2013) developed a design tool for wind blades, named NuMAD (Berg & Resor, 2012), which manages all blade information, including airfoil databases, materials and material placement, to enable efficient creation of models. It is a stand-alone, user-friendly, Matlab-based ("Mathworks," n.d.) graphical pre-processor for the ANSYS FEM software. NuMAD provides an intuitive interface for defining the outer blade geometry, shear web locations, materials and stack placement in the blade. The output from NuMAD is a sequence of ANSYS APDL commands, used to create the FE model in ANSYS; it can export many shell-type elements but not solid ones in its current version. NTUA's team developed a parametric modeler for turbomachinery blade rows, named GMTurbo (Tsiakas *et al.*, 2016). The software can be used to build, a wide range of 3D blade shapes parametrically using metal and other angles, while it has been successfully combined with evolutionary based optimization tools, for shape optimization (Tsiakas *et al.*, 2013, 2015).

#### **1.4.1.2 Surface reconstruction**

Surface reconstruction, being a very challenging problem, is a key element of reverse engineering methodologies and finds a variety of interesting applications in different

disciplines. Shape reconstruction is defined as the construction of a compact mathematical representation of the shape under consideration, using partial information about it, usually in the form of discrete points in the 3D geometric space.

The aim of shape reconstruction is to build a mathematical (parametric) model that approximates an unknown surface from a large amount of data points. Such points can be obtained through a scanning procedure (Savio *et al.*, 2007) or from other sources (such as the result of a shape optimization procedure), forming either two-dimensional slices of a three-dimensional shape, or the iso-parametric curves on a surface, or being scattered (unorganized) clouds of data points. The resulting parametric model of the reconstruction procedure is usually easier and computationally cheaper to modify, leading in this way to a reduction of its processing costs.

For complicated geometries, described with large amount of discrete points, the problem of geometry reconstruction depends a) on the proper choice of the approximation function and b) on the selection of the appropriate parameters for the description of the surface model (parameter tuning). Surface fitting methods are mainly based on the least-squares approximation scheme and the procedure starts with an estimation of the parameters describing the geometry, while the application of the fitting procedure tries to find the values of such parameters, which produce a geometry that closely approximates the given data points. The basic requirements of surface fitting are the following (Weiss *et al.*, 2002): a) the surface should approximate each data point within a given tolerance; b) it should be aesthetically pleasing and predictable; c) the surface should extend in a reasonable way beyond the boundaries of the point cloud and over regions where no data is available; d) it is desirable to obtain a non-redundant surface representation automatically, which has a reasonable number of control points.

The problem of shape reconstruction can be modelled as an optimization problem, containing two main tasks, a) surface parameterization and b) surface fitting; the resulting optimization problem is usually high-dimensional and non-linear, making its solution a very challenging task. Different methodologies have been reported in the open literature, attempting to the development of effective, efficient and robust algorithms that can be used for solving the shape reconstruction problem. Among them, a special category of methodologies utilize techniques categorized in the artificial intelligence (AI) scientific field, such as evolutionary algorithms (EAs), artificial neural networks (ANNs), particle swarm optimization (PSO) techniques, artificial immune systems (AIS), etc. Such methodologies, have proven their effectiveness and usefulness in many different technological fields, especially for the solution of demanding and high-dimensional optimization problems. The application of such methodologies to the shape reconstruction problem is neither a trivial nor a straightforward task. Moreover, population based methodologies (such as EAs, PSO, AIS, etc.) ask for considerable computational resources, and, therefore, cannot be used for real-time applications, in general. Nevertheless, such methodologies can provide alternative and very effective solutions to the shape reconstruction problem.

#### 1.4.1.2.1 *Geometry representation methods*

Four different surface representation methodologies are usually utilized, having as a basis a given set of data points; they provide a different level of accuracy in the representation of the required surface. The first refers to the construction of a polygonal mesh, using as vertices either all the given data points, or less points instead (Albersmann *et al.*, 1998; Yu, 1999; Goński & Obuchowicz, 2007). This is not an actual parameterization of the surface, and a

large set of data is needed for its representation. Constructive solid geometry (CSG) uses elementary geometrical entities (primitives) through Boolean operations to produce more complicated shapes, utilizing a binary tree representation (Weinert *et al.*, 2001). The use of parametric free-form surfaces is the most powerful surface reconstruction technique, with B-splines and NURBS (Farin, 1992; Piegł & Tiller, 1995; Patrikalakis & Maekawa, 2002) being the most used parametric representations. They have a sufficient number of tuning parameters, which provide the ability of producing accurate reconstructions of the shapes under consideration, with a controlled smoothness. However, their parameter estimation is a very difficult problem, especially for NURBS surfaces, and many different methodologies have been developed to this end, some of them based on artificial intelligence (AI) techniques (Piegł & Tiller, 1987; Laurent-Gengoux & Mekhilef, 1993; Ma & Kruth, 1995, 1998; Barhak & Fischer, 2002; Weiss *et al.*, 2002; Beielstein *et al.*, 2003; Krause *et al.*, 2003; Wagner *et al.*, 2007; Gálvez *et al.*, 2012; Gálvez & Iglesias, 2012; Xie *et al.*, 2012). The fourth methodology uses implicit surfaces, such as artificial neural networks (ANNs) or radial basis functions (RBFs), which enable a compact mathematical description of the surface and very fast operations. However, special procedures are required for the tessellation and rendering of such surface representations (Kojekine *et al.*, 2004; Xiaojun Wu *et al.*, 2005; Yumer & Kara, 2012).

The available data set usually cannot be reconstructed into a single 3D free-form surface; a segmentation procedure should be carried out first (dividing the point cloud into proper subsets), to produce a higher level representation of the shape, in the form of a set of parametric surfaces (Várady *et al.*, 1997). For a NURBS surface reconstruction, the parameters to be defined are: a) the number of control points in each parametric direction, along with their coordinates and weights, b) the knot vectors in the two parametric directions, c) the degrees in the two directions.

#### 1.4.1.2.2 Surface fitting

Albersmann *et al.* (Albersmann *et al.*, 1998) proposed two different artificial intelligence techniques, namely a simulated annealing (SA) algorithm and an evolutionary multi-individual strategy, for surface reconstruction by means of triangulation of digitized 3D scattered data points. The proposed EAs are applied to four different smoothness criteria, with the appropriate fitness functions, for the triangulation of sparse point data sets. Simulation results demonstrated that each criterion lead to different solutions, while the gained quality depends on the adopted criterion, the parameter settings and the local curvature properties of the surface.

An evolutionary algorithm is combined with a recursive subdivision scheme in (Goiński & Obuchowicz, 2007) for the construction of an iso-surface from an input set of unorganized 3D data points. The population of the EA consists of meshes, which iteratively approximate the given data points, starting from an initial population of deformed cubes (having only 24 triangles), which bound the object to be reconstructed. In each generation, a closer approximation of the object is obtained, through the subdivision scheme, and the utilized evolutionary procedure. The evolutionary procedure is described by the following steps, which are iteratively repeated: a) selection (tournament selection or the best individual); b) subdivision process; c) mutation and/or recombination. For mesh refinement, two schemes are used; the ternary subdivision scheme and the isotropic template. For the mutation operator, triangles with most obtuse angles are mutated by randomly selecting new points from a nearest neighbourhood. In the recombination operator, two individuals randomly

exchange points of appropriate triangles (with the constraints that the two individuals should have the same number of vertices and faces).

Few methodologies have been presented so far for surface reconstruction that use NURBS parameterization and evolutionary techniques. The reason is that NURBS surface parameterization is very challenging, as it involves a large number of parameters and the corresponding non-linear optimization problem becomes a high-dimensional one (Gálvez & Iglesias, 2012; Xie *et al.*, 2012).

The neural network self-organizing map (SOM) method was utilized in (Barhak & Fischer, 2002) for the creation of a 3D parametric surface, for a given set of data points; the method is used for both parameterization and surface fitting. The main advantage of the proposed SOM method is that it automatically detects the orientation of the grid and the position of the sub-boundaries. Through an adaptive process, the neural network grid converges to the given data points. The following stages are used by the proposed algorithm: a) construction of the 2D/3D parametric grid by SOM neural network parameterization methods; b) parameterization of the sampled data with respect to the parametric grid; c) construction of an initial parametric base surface, using the computed parameterization of the given data points; d) projection of the data points on the base surface, and recalculation of their parameterization; e) approximating the data points to a B-spline surface; f) optimizing adaptively the parametric surface. The surface fitting is performed by random surface error correction (RSEC), which is based on the SOM neural network method.

Evolutionary algorithms have been developed for NURBS surface reconstruction in (Beielstein *et al.*, 2003) but on relatively simple objects. Krause *et al.* (Krause *et al.*, 2003) presented a surface reconstruction approach, which integrates artificial neural networks (ANNs) with subdivision techniques. The neural network is used to create a triangular mesh that approximates the shape of the object under consideration, and detects its topology, while the subdivision approach applies smooth surfaces onto this mesh. The reconstruction process is based on the following stages: (a) a manifold mesh is reconstructed using an extended neural gas neural network, where the neural units are the edges of the mesh. As the neural network is being trained, the mesh vertices move in order to minimize their difference between the cloud of points and the vertices. (b) The resulting mesh is simplified using flat area detection; triangles are grouped together into areas of flat curvature, according to the difference in the direction of their normal vectors. (c) The mesh is interpolated by NURBS, using a subdivision algorithm. Every polygon of the mesh is split into convex quadrilaterals, while the quadrilateral mesh is converted into a NURBS surface by utilizing the Patching Catmull-Clark Meshes (PCCM) algorithm (Peters, 2000). The main advantage of the proposed methodology is its applicability to objects with arbitrary topology. Additionally, the produced model can be integrated with traditional CAD systems using a NURBS representation which preserves continuity.

In the work of (Gálvez *et al.*, 2012) an iterative, GA-based methodology for the surface reconstruction from clouds of noisy 3D data points was presented; the surface is modelled as a B-spline one. The initial input to the computational procedure includes the following: the cloud of 3D data points, the degrees  $p, q$  in the two parametric directions  $u, v$  of the B-spline surface, the number of control points  $(n+1, m+1)$  in the two parametric directions, and a threshold error of the iterative procedure. The problem of computing the parameters of the B-spline surface is formulated as an iterative approximation of the surface. In each iteration, four successive steps are utilized. Firstly, the parametric values  $u, v$ , of the provided data points are computed using a real-coded GA. At the second step, the knot vectors of the B-spline

surface are computed using a second GA. At the third step of each iteration the control points of the B-spline surface are computed by solving a linear least-squares minimization problem, using either SVD or a modified LU decomposition method. At the fourth step the computed parameters of the B-spline surface (in the previous 3 steps) are used to reconstruct the surface, and the reconstructed data points are compared to the original ones, by computing the mean and the maximum error values. These four steps are repeated until the mean error of two successive iterations becomes smaller than a predefined threshold value. The proposed method requires from several minutes to few hours to converge, depending on the complexity of the surface to be reconstructed. The most important factors that affect its time efficiency are the number of data points, the required accuracy, and the population size of the GAs.

A particle swarm optimization (PSO) methodology was used in (Gálvez & Iglesias, 2012) to reconstruct the shape of objects using NURBS surface parameterization. The surface is described in the form of point clouds of (organized or scattered) noisy data points. The proposed methodology provides the ability to obtain all relevant surface data of the utilized NURBS parameterization, such as parametric values of the provided data points, knot vectors, control points and weights, without the need of pre- and post-processing. The method is based on approximating the provided 3D data points (rather than interpolating them).

In this work a surface reconstruction process is developed, to obtain the appropriate design variables for reconstructing a NURBS surface, which accurately adapts to a target input triangular mesh or point cloud. The proposed methodology enables the interaction between a geometry definition software, capable of designing three-dimensional blades (Tools for Turbomachinery - T4T (Koini *et al.*, 2009; Strofylas *et al.*, 2014), and a parallel, metamodel-assisted differential evolution (DE) algorithm (Nikolos, 2011, 2013; Strofylas *et al.*, 2018). As the evolutionary-based shape-reconstruction techniques are time consuming, a parallel strategy was incorporated in the DE algorithm, while further acceleration is achieved through the use of two artificial neural networks (ANNs) that serve as surrogate models to the DE procedure. Regarding the blade surface parameterization, the utilized geometric modelling software (T4T) employs a custom procedure for the construction of the blade surface, based on a set of parameters with physical meaning for each cross-section that forms the blade.

### 1.4.2 Mesh deformation

During the past decade, mesh deformation methods have attracted continuously increasing interest in Computational Fluid Dynamics (CFD), as its applications are extended in the fields of aerodynamic shape optimization and aeroelastic analysis. In such simulations the flow prediction over a deforming body calls for a methodology to describe the movement of the examined computational grid in a robust, accurate, and computationally-efficient way. For problems involving Fluid-Structure Interaction (FSI) numerical simulations, where the CFD mesh has to be adapted several times during the analysis, such characteristics are essential.

Various approaches have been developed in the past for mesh deformation, managing both structured and unstructured grids. The Transfinite Interpolation Technology (TFI), originally introduced by Wang and Przekwas (Wang & Przekwas, 1994), is one of the most popular techniques for structured meshes. According to this method the displacement of the internal domain points is obtained via the linear interpolation of boundaries' displacement along the mesh lines. Although the aforementioned approach seems to produce a smooth as well as an efficient interpolation/deformation, it cannot guarantee orthogonality (Cizmas & Gargoloff,

2008). (Bhardwaj *et al.*, 1998) proposed an algebraic approach instead, in which the grid deformation is achieved by reordering the nodes along grid lines in a direction perpendicular to the solid wall surfaces. Nevertheless, the method's applications are restricted to test cases involving small deformations, since large ones may lead to low-quality grids (Kouskouris & Nikolos, 2012). For multi-block structured grids, Potsdam and Guruswamy (Potsdam & Guruswamy, 2001) introduced an alternative method, in which the surface deformation is spread out with an exponential decay, while the final displacement is obtained using an improved version of TFI blending functions. However, the extensive use of three-dimensional unstructured (tetrahedral or hybrid) grids, due to their capability to describe accurately complex geometries, led the researchers to develop more generic methods and hence more suitable for this type of meshes. These techniques can be divided mainly in two categories, the connectivity-based and the point-based ones.

The method of spring analogy, initially presented by Batina (Batina, 1990), appears to be the most widely applied connectivity-based technique. According to this approach the grid edges are represented by springs, whose stiffness is inversely proportional to their length. Despite its popularity, in test cases involving large displacements the procedure might create elements with negative volumes or even fail (Farhat, Degand, *et al.*, 1998). Improved versions of the aforementioned methodology include the use of torsional (Farhat, Degand, *et al.*, 1998), semi-torsional (Blom, 2000) or ball-vertex (Bottasso *et al.*, 2005) springs, to prevent the extraction of degenerated elements. However, they result in excessively time-consuming processes, especially in cases involving large computational grids. Similarly to spring analogy, the elastic analogy model assumes the examined mesh to be a physical body with elastic properties (Lynch & O'Neill, 1980), while the stiffness of each region is defined as inversely proportional to its volume; improvements to the method include the usage of strain field as a stiffness criterion (Hsu & Chang, 2007). Another connectivity-based deformation methodology regards the solution of a Laplacian (Löhner & Yang, 1996) or Biharmonic (Helenbrook, 2003) equations' system. Unfortunately, the first approach is susceptible of failing to preserve orthogonality, while the second one entails significantly increased computational cost (Luke *et al.*, 2012). The main shortcoming of the methods of this category concerns handling of hanging nodes, for which special treatment is required. Moreover, they frequently result in a stiff matrix system, which makes large displacements unmanageable.

In the second classification of point-based methods, requiring no information about nodes' connectivity, each point can be moved independently to its adjacent ones. Thus, these techniques are applicable for any type of mesh, structured or unstructured, tetrahedral or hybrid (triangular or quadrilateral in two dimensions), etc. A well-established approach of this category, mainly oriented toward shape optimization, is the Free Form Deformation (FFD) technique (Barr, 1984), according to which a Non-Uniform Rational B-Spline (NURBS) lattice is constructed in a way to wrap the computational grid; the movement derives from the displacements of the control points included in the lattice. Another point-based methodology, introduced by Melville (Melville, 2000), considers grid deformation based on the fusion of the displacements of the closest points from a set of moving surfaces. (Liu *et al.*, 2006) proposed an alternative one, in which a Delaunay graph is used and the displacements of mapped boundaries are propagated via barycentric interpolation. Although this technique retains the main feature of point-based methods, i.e., it is eligible to be applied to any type of mesh, it is revealed inappropriate for grids over complex geometries. It may produce intersections, while it cannot account for surface rotations (Tianhang *et al.*, 2008). (Allen, 2007) presented a methodology according to which the desired deformation is achieved taking into account the displacements and rotations of moving surfaces by means of their inverse

distances from the interior points, while Witteveen (Witteveen, 2010) developed the Inverse Distance Weighted (IDW) interpolation method, where the deformation depends on the weighted-averaging of the displacements of boundary nodes. Another widely employed point-based technique is based on Radial Basis Functions (RBFs) (de Boer, van der Schoot, *et al.*, 2007) for the interpolation of the displacements; the grid points on the boundary surfaces are used to construct the interpolant that subsequently will result to the deformation of the entire grid. It has been proved to be a very effective methodology, able of handling large deformations, preserving mesh orthogonality and producing high-quality deformed grids (de Boer, van der Schoot, *et al.*, 2007; Rendall & Allen, 2009b). However, it calls for excessive computational effort and memory storage, especially in large-scale problems involving meshes with millions or tens of millions nodes. This inefficiency derives from method's requirement to solve a large system of equations with dimensions  $N_s \times N_s$ , where  $N_s$  denotes the number of surface mesh nodes, as well as from the need to implement an interpolation procedure for all the internal mesh nodes at the evaluation stage (Rendall & Allen, 2009b).

A remedy to the aforementioned non-trivial shortcoming was revealed to be the adoption of the Fast Multipole Method (FMM), initially introduced in RBF-interpolation by Beatson (Beatson & Newsam, 1992); a tree data structure is utilized to organize the points of the mesh. Subsequently, the displacement of a point is computed, taking into account the real contribution of nearby points, while faraway points' contribution is approximated based on a Taylor series expansion. FMM appears to improve the efficiency of the evaluation stage, although its implementation seems rather complex (Wendland, 2002). Another approach, also focusing on the efficiency improvement of this stage, considers the Partition of Unity (PoU) (Wendland, 2002; Rendall & Allen, 2009c), according to which the initial problem is divided in several smaller ones, by grouping base points into sets to be processed separately. However, merging of local interpolations back together can cause discontinuities or decrease the grid quality (Rendall & Allen, 2009b). An improved version of this technique, proposed in the study of Cordero-Gracia *et al.* (Cordero-Gracia *et al.*, 2012), regards the use of an octree algorithm to cope with three-dimensional meshes. The interpolation procedure is implemented to each sub-domain separately, beginning from the moving surfaces and spreading out successively to the far-field ones. The main drawback of this procedure is that the computational cost is highly affected by the mesh type (Cordero-Gracia *et al.*, 2012). In a similar way the fast multilevel approach (Livne & Wright, 2005) has been employed to improve evaluation stage's efficiency by interpolating the RBFs' coefficients into a coarser mesh, evaluating deformation at this resolution and interpolating these values back to the initial fine mesh. The aforementioned procedure is similar to the Full Approximation Scheme (FAS multigrid scheme) in CFD (Carré *et al.*, 2000; Blazek, 2001; Lygidakis & Nikolos, 2014a).

The bottleneck of the latter methodology remains the solution of the linear system, which might entail even a prohibitive computational cost. Toward this direction, Beatson *et al.* (Beatson *et al.*, 1999) proposed a method considering the iterative solution of this system via Krylov sub-space schemes and preconditioning approach with approximate cardinal functions (Blazek, 2001). Similarly to mesh deformation, RBFs may produce ill conditioned matrices resulting to relatively inefficient iterative solvers; the parallel direct ones seem to be a better choice. An alternative technique is the moving sub-mesh approach of Lefrançois (Lefrançois, 2008), extended in the context of RBFs by Liu (Liu *et al.*, 2012). According to this method deformation is initially obtained through RBFs at a coarse grid, while updating of the fine one is performed with an interpolation scheme, based on the position of the fine mesh nodes at the coarse grid graph. The main drawback of this approach is that the coarse grid has to be composed of tetrahedral only elements (Liu *et al.*, 2012).

Another way to improve the efficiency of such algorithms is the use of a reduced set of surface points, which results in a smaller system and consequently in significant computational savings. In the paper of Jakobsson and Amoignon (Jakobsson & Amoignon, 2007) a reduction algorithm was presented for the selection of a subset of points in an equidistant way, by clustering neighbouring points that lie inside a constant radius. Variations of this method include the utilization of an adaptable radius that depends on the local average deformation or interpolation error (Kroll *et al.*, 2008). To further enhance RBFs' efficiency Rendall and Allen (Rendall & Allen, 2009a, 2009b) employed a greedy reduction algorithm, according to which the point selection method depends on the error minimization on the interpolated surface. The error signal is obtained by taking into account the actual displacement of the deformed surface, hence requiring to be known prior to the deformation procedure. They extended their initial work by implementing more error functions (Rendall & Allen, 2010); the best results were obtained by the unit function imposing a unit displacement to the boundary, though it does not guarantee that the error criterion will be always met. Another version of this methodology, proposed by (Michler, 2011), considers the addition of multiple centers and selection of different point sets of them in each spatial direction. (Kowollik *et al.*, 2013) introduced a technique in which the reduction of nodes is based on a *kd*-tree; a number of generated search points for each leaf is used to identify the closest surface point that will be utilized for the subsequent RBF-deformation step.

Alternatively, a surface version of the fusion strategy of the well-established in CFD agglomeration multigrid method (Mavriplis & Venkatakrishnan, 1995, 1996; Carré *et al.*, 2000; Blazek, 2001; Zhang *et al.*, 2008; Nishikawa & Diskin, 2011; Thomas *et al.*, 2011; Nishikawa *et al.*, 2013; Lygidakis & Nikolos, 2014a, 2014b) can be implemented to derive the desired reduced surface point selection, as it is proposed in this study. Multigrid methodology is a powerful acceleration technique, based on the relaxation of governing Partial Differential Equations (PDEs) on successively coarser grids, in order the low frequency errors to be transformed in high frequency ones on the coarser resolutions, allowing for their efficient damping (Mavriplis & Venkatakrishnan, 1996; Ferziger & Peric, 2002; Lygidakis & Nikolos, 2014a). Although it was initially proposed to increase the convergence rate of the numerical solution of elliptic problems (Brandt & Livne, 2011), it has been employed since then along with compressible (Mavriplis & Venkatakrishnan, 1995, 1996; Carré *et al.*, 2000; Nishikawa & Diskin, 2011; Nishikawa *et al.*, 2013; Lygidakis *et al.*, 2016) or incompressible (Ferziger & Peric, 2002; Zhang *et al.*, 2008; Lygidakis *et al.*, 2016) flow solvers, radiative heat transfer ones (Lygidakis & Nikolos, 2014a), etc. The sequence of the coarser meshes can be obtained either by the generation from the very beginning of completely independent grids or by the construction of associated meshes (nested approach) (Blazek, 2001; Lygidakis & Nikolos, 2014a). In case of the last approach, the successive coarser resolutions can be extracted either by starting from the coarsest one enriching it with more mesh nodes via a grid adaptation technique, such as *h*-refinement (Fuchs, 1986; Jimack & Kirby, 2011; Lygidakis & Nikolos, 2013; Rossi *et al.*, 2013), or by starting from the finest one removing nodes and implementing re-triangulation (Blazek, 2001). Similarly to the last topological technique, the agglomeration multigrid type, originally introduced by (Lallemand, 1988a), considers the generation of a coarser grid via merging the neighbouring control volumes of denser resolution; in that way coarser meshes including irregular polyhedral elements are constructed (Ferziger & Peric, 2002; Lygidakis & Nikolos, 2014a). In tetrahedral grids an isotropic agglomeration (Carré *et al.*, 2000; Lygidakis & Nikolos, 2014a) is usually followed, while in hybrid ones with highly stretched elements at the boundary layer region, a directional (semi- or full-coarsening) fusion (Mavriplis & Venkatakrishnan, 1995; Nishikawa & Diskin, 2011; Nishikawa *et al.*, 2013;

Lygidakis *et al.*, 2016) is preferred, treating control volumes at prismatic or hexahedral regions separately. The directional full-coarsening procedure begins by merging the solid wall boundary control cells (implicit lines' bottoms) (Nishikawa & Diskin, 2011), while at next a line-agglomeration step is performed fusing prismatic or hexahedral control volumes along implicit lines starting directly above the boundary volumes (Nishikawa & Diskin, 2011; Lygidakis *et al.*, 2016). It is the first step of the aforementioned procedure that can be employed along with an RBF-based mesh deformation code to successively reduce the set of surface points and consequently the dimensions of equations' system  $N_s \times N_s$ ; it considers the agglomeration of boundary areas of the corresponding control volumes instead of merging all of them. Any information is transferred from the coarser to the denser resolution by means of interpolation, similarly to the prolonging operation of variables and flux balances in FAS multigrid approach (Blazek, 2001; Lygidakis & Nikolos, 2014a). In that way the advantages of the agglomeration strategy are extended beyond the efficiency improvement of numerical solvers to mesh deformation algorithms.

### 1.4.3 Fluid Structure Interaction

During the past decades the analysis of Fluid-Structure-Interaction (FSI) with numerical methods gained significant popularity, mainly due to the broad range of its applications, e.g., in aerodynamic shape optimization, aeroelastic analysis, etc. (Hansen *et al.*, 2006; Baxevanou *et al.*, 2008) Although several FSI solvers, commercial, industrial or academic, have already been developed, the need for more accurate and more efficient methodologies is still on the run, making FSI a continuously evolving scientific field. One of the most widely applied techniques to deal with this kind of problems is the partitioned coupling approach. According to this method the fluid and structural physical systems are solved independently, while interaction between them is obtained via the exchange of appropriate variables on boundary surfaces (Piperno *et al.*, 1995; Farhat, Lesoinne, *et al.*, 1998); as a result, it allows for the integration of separately developed CSD (Computational Structural Dynamics) and CFD (Computational Fluid Dynamics) applications. Nevertheless, the aforementioned solvers' combination is raising several issues in terms of information transfer, considering the main task of accurate as well as efficient FSI simulations.

For the numerical solution of FSI problems, the Dirichlet-Neumann domain decomposition coupling approach is usually followed (Smith, 1997). It begins with the flow prediction, while at next the obtained pressure field at the flow side of the CFD/CSD interface is transferred to the corresponding structural one. Subsequently, a CSD analysis is performed, deriving the deflections of the structure and a new flow grid is generated incorporating the new interface location. Information transfer across the fluid-structure interface has to be performed with respect to a kinematic condition (Dirichlet), assuring the continuity of the velocity field, and a dynamic condition (Neumann) satisfying the requirement for conservation of normal stresses (Felippa & Park, 1980). Hence, the most important criteria to be fulfilled over the interface, concern the consistency between the structural and the flow solver and the instabilities that may occur due to the partitioning process as well as the strategy followed for the data transfer. Similarly to stresses' conservation, an energy one has to be succeeded at the common interface (Piperno *et al.*, 1995) to ensure the stability of the whole numerical solution, i.e., the virtual work of the flow forces and this of the structural ones must be equal. The aforementioned conservation requirements across the fluid-structure interface depend strongly on both the employed temporal and spatial coupling schemes (Boer *et al.*, 2009).

This thesis focus is mainly towards the spatial coupling and information transfer between the CFD and CSD solvers. The computational nodes of the structural and flow grids do not coincide usually at their interfaces, since their underlying resolutions can significantly differ. This non-matching interface may lead to gaps and intersections between them, necessitating thus for an extra interpolation step to enable information transfer.

Numerous methods can be found in the literature to deal with this problem, while the simplest technique remains the nearest neighbor interpolation (Thévenaz *et al.*, 2000); according to this method information for a structural grid node is obtained from the closest point of the flow mesh and vice versa. Despite this scheme complies with force conservation requirement, it derives acceptable results only in test cases involving flow and structural meshes with similar resolutions. A more sophisticated way to encounter this problem considers the use of weighting methods (Maman & Farhat, 1995; Cebal & Löhner, 1997; Quaranta *et al.*, 2005), according to which both sides (structural and flow) are multiplied by a set of weighting functions and integrated over the interface. Nevertheless, this integration is not a straightforward procedure, since there isn't a common surface on which both structural and flow functions can be defined.

Various methods can be found in the literature to overcome this drawback, while most of them are based on projections and integrations of the elements from the one mesh to the other; such projections concern usually a vertex, the center or the quadrature Gauss points of a mesh element, or even the whole element. An assessment of the last two types has been reported in (de Boer, van Zuijlen, *et al.*, 2007), revealing that with the first one conservation of forces over the interface is succeeded, though it doesn't guarantee that all structure points will get information. In case of the latter type, the flow mesh elements might not be covered by the projections of the structural ones, resulting in incompliance with the force conservation condition. A popular method, dealing with the data transfer problem is the so-called Constant Volume Tetrahedra (CVT) (Goura *et al.*, 2001; Jiao & Heath, 2004; Sadeghi & Liu, 2005). According to this method, a tetrahedron is constructed at each flow grid node, including additionally the three nearest to this node structural mesh nodes. Since this element is defined, the displacement of each flow grid node is obtained with the movements of the structural mesh ones in a way preserving the volume of the tetrahedron; the transpose matrix can be used accordingly to transfer forces. However, an important drawback of this approach is revealed in junction regions where several structural components meet; in these areas the selection of a triad of structural nodes that do not produce discontinuities may not be an easy process. Another widely applied method, the Boundary Element Method (BEM), initially proposed in (Chen & Jadic, 1998), assumes that the examined space between the structural and flow meshes is filled with a linear elastic homogenous material. Hence, the information transfer between the two grids is based on the governing equations modeling elastic behavior of this material. Nevertheless, accuracy issues are raised in test cases involving poor mesh quality, and more specifically at regions where boundary surfaces intersect. Alternatively, in (Guruswamy & Byun, 1993) a virtual surface technique was introduced, which utilizes an intermediate surface to succeed interpolation of loads and displacements in a conservative manner, while in (Smith *et al.*, 1995, 2000) a method has been proposed involving interpolation based on splines. Moreover, in the aforementioned studies an overview of various different interpolation schemes based on RBF's have been reported, e.g., including infinite-plate spline (IPS), multiquadric biharmonic (MQ), non-uniform B-spline (NUBS), thin-plate spline (TPS), finite-plate spline (FPS) and inverse isoparametric mapping (IIM) interpolations. In 2001 Beckert and Wendland (Beckert & Wendland, 2001) studied the use of RBF's with local support, revealing their appealing properties for transferring data in FSI computations; the

main advantage of this method derives from its inherent capability to conserve total force and moment. Furthermore, RBF's methodology requires no connectivity information, as it relies only on arbitrary point data. As a result it can handle equally structured and unstructured grids.

Nevertheless, the method, in its initial formulation, calls for relatively excessive memory and computation time requirements. For the alleviation of the aforementioned shortcoming the Partition of Unity (PoU) (Ahrem *et al.*, 2006) approach is adopted in data transfer, which regards the decomposition of the interfacing boundary into several smaller regions, where local interpolations are performed independently and hence more efficiently. Although this method ensures energy conservation and offers physical distribution of the forces on the interfacing area (Ahrem *et al.*, 2006; Rendall & Allen, 2009c), it is revealed to be inappropriate for mesh deformation, as the dependence matrix takes into account all the flow mesh nodes, entailing significantly increased computational cost.

In this thesis a partitioned FSI coupling procedure was developed to facilitate interaction between an open-source CSD (Dhondt, 2014) and an in-house academic CFD (Lygidakis *et al.*, 2016) code. Attention is mainly directed toward the efficient and accurate transfer of predicted displacements (by CSD) and forces (by CFD). Therefore, spatial coupling is obtained by a RBFs interpolation scheme, employing the aforementioned PoU approach (Rendall & Allen, 2009c) to accelerate the exchange procedure. For mesh deformation a corresponding RBFs interpolation process is applied; its efficiency is significantly improved with a surface point reduction technique, based on the agglomeration of adjacent boundary nodes.

# Chapter 2: Parametric design of wind turbine blades using T4T

## 2.1 Introduction

In this chapter, a software tool named "T4T" (Tools for Turbomachinery) is presented for the parametric design of turbomachinery and especially wind turbine blades. The complete design procedure is object-oriented and parametric, providing the ability to the user to define various types of blades (Koini *et al.*, 2009). It is developed in QT C++, utilizing OpenCascade graphical and computational libraries. The software allows the user to design the outer surface either by specifying some physical parameters for each blade section, or by directly interpolating a surface through a cloud of points. The new/enhanced version of "T4T" software tool (developed during this work) introduces the definition of internal blade structure for wind turbines rotors and is fully parametric and customizable; thus, allowing the user for defining the internal blade structure, including shear webs. The computational procedure finally produces compound solids, which can be further imported to mesh generation and analysis software through standard geometry exchange protocols, for cooperation with CFD and CSD solvers.

Specifically, this enhanced version of "T4T" has the following functions of the outer surface module:

- The ability to modify trailing edge geometry and add thickness to the trailing edge.
- The methodology for the computation of a) the mean camber line (for an arbitrary blade section), b) its leading edge and c) its chord.
- The modification of the lofting procedure, for the proper adaptation of the surface at the leading and trailing edges.
- The ability to manipulate airfoils with different number of points for the blade construction.
- The ability to create custom stacking lines at custom positions.

Concerning the blade's internal structure definition, the blade is divided into regions that are used to assign different laminate thicknesses and materials. This is accomplished by defining the desirable surface patches as percentages of blade's radius and blade's section chord at selected positions. In each blade patch, the thickness of the laminated material can be parametrically defined; the location of each shear web is also prescribed, by utilizing user-defined parameters, while the profile of the web is produced as a sweep surface. "T4T" has the ability to create compound solids, which can be used for Finite Element Analyses; this feature is not common between other similar software.

"T4T" provides an intuitive user-interface for defining the outer and inner blade geometry, while highly detailed 2D and especially 3D views of the produced blades are provided to the user during the whole blade design process. The completely parametric definition of all produced geometries allows for the automatic cooperation with optimization methodologies, as the whole procedure can be performed in a batch mode without the need of the graphical interface, by importing the design parameters from appropriate text input files. The final geometry is exported in standard STEP format, which is then imported to GMSH (Geuzaine & Remacle, 2014), an open source mesh generation software, to produce the computational

mesh of the internal structure. The open source FEM software *CalculiX* (Dhondt, 2014) is used afterwards for the structural analysis of the blade in a fully automated procedure.

The rest of this chapter is structured as follows: In Sections 2.2 and 2.3 the methodology for the parametric geometry definition is presented, for blades' surface and internal structure respectively. The graphical interface, along with sample designs are presented in order to demonstrate its capabilities. In Section 2.4 the procedure of defining the NREL 5MW (Jonkman *et al.*, 2009) baseline rotor is presented as a test-case, followed by conclusions (Section 2.5).

## 2.2 Blade surface

A rotor blade consists of several airfoil profiles, which define the 3-dimensional blade surface through a "skinning" procedure (Piegl & Tiller, 1995; Koini *et al.*, 2009). In this work, using "T4T", each one of these profiles, as well as the final surface, are produced following the NURBS theory (Piegl & Tiller, 1995). NURBS (Non-Uniform Rational B-Splines) is a mathematical representation of a line, surface or volume that can accurately describe simple or complicated shapes with a small number of parameters. More specifically, "T4T" software in its current form allows the user to define the airfoil profiles through two alternative methods: either by specifying some physical parameters, such as mean camber line and thickness distribution, or by directly interpolating a curve through a set of points. The last feature of interpolation is very useful for wind turbine blade design, as it is very common to use standard NACA (Abbott & von Doenhoff, 1959) or DU (Delft University) type airfoils, or non-standard ones, to define a wind turbine blade.

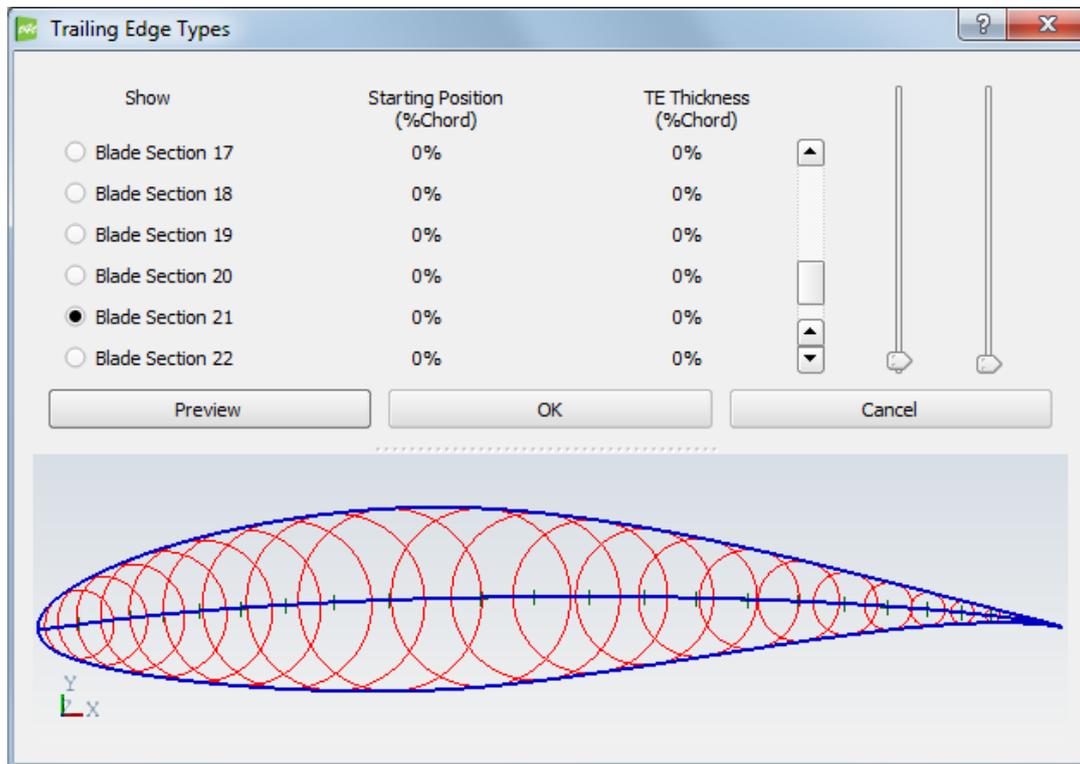
According to the first method, each airfoil is defined using two geometric entities, a "mean camber line" and a "thickness distribution" along the airfoil (the quotations are used because these entities have a more generalized definition, compared to the classic mean camber line and thickness distribution of an airfoil). The computation of the mean camber line and the airfoil is done according to the procedure described in (Koini *et al.*, 2009; İlkılıç *et al.*, 2011).

Concerning the second method, cross-sections are computed using the method of curve fitting (Piegl & Tiller, 1995). "T4T" provides the ability to import the airfoils' data through text files, as a sequence of discrete points; in its current form each airfoil can be defined through different number of points. The degree of the interpolation curves, describing the corresponding cross sections (and as a consequence the degree of blades' surface in circumferential direction), is user-defined, being always higher than 2, to allow for the continuity of the 1st derivative. This is crucial for the construction of the 3D solid blade. Airfoils' coordinates are usually provided in dimensionless form (as a function of section's chord); they are converted to a dimensional form by determining the corresponding chord's length at the specific stacking position of the airfoil, along the blade's span.

As soon as the determination of NURBS curves that describe blade cross sections is finished, the computation of the mean camber line, the leading edge, and the chord line are performed. In order to determine the mean camber line, which is the bisector curve between the pressure and suction sides of the airfoil, the unique section curve should be splitted in its two separate sides.

However, this is not a straightforward procedure, as the position of the leading edge is usually unknown. For that purpose, the airfoil section is initially splitted using an approximate procedure. Then, the bisector curve between these two sides/curves is computed for the 95% of their lengths, starting from the trailing edge. In order to find this bisector curve

the inscribed circles between the two curves are successively computed, and finally the mean camber line is produced as a sequence of their centers (FIGURE 2.1). At the end of this curve, which is close to the leading edge, its tangent vector is computed, to expand the mean camber line to its forward direction. The intersection point between the tangent vector and the airfoil curve defines the true leading edge point, used to accurately split the section and define its chord.



**FIGURE 2.1:** Mean camber line and inscribed circles between pressure and suction sides.

Additional adjustments to the generated profiles may be needed, as most of the standard airfoils have sharp trailing edges, making blade manufacturing practically infeasible. "T4T" offers a technique to modify trailing edge geometry, in order to become rounded. A circle with user-defined radius is placed at the trailing edge region. To ensure smooth thickness transition to the new trailing edge, airfoils' thickness is increased using a parabolic additional thickness distribution, starting (for each airfoil side) from a point defined by the user (as a percentage of the chord) and increasing to its maximum at the position that meets the trailing edge circle. This additional thickness is applied in a direction normal to the mean camber line. The procedure ensures that the airfoil chord length remains unchanged, as well as smooth transition to the desired thickness at the trailing edge region. As shown in FIGURE 2.2, the user may define the starting point and the desired trailing edge thickness for each airfoil.

The construction of the blade using planar airfoils is performed with respect to a stacking line; this is usually a straight line, however the software provides the ability to define a curve to be used as a stacking line, in order to produce pre-bend blades. All cross-sections are distributed in perpendicular planes along the stacking line. Sections can be stacked with respect to their centers of gravity, their leading edge points or specific positions, such as their aerodynamic centers; this last feature is commonly preferred for wind turbine blade construction. Airfoil sections are stacked by determining their position along the stacking line (either as an absolute length or as a percentage of the length of the stacking line) as shown in

FIGURE 2.3. Additionally, the twist of the blade can be defined by applying the desirable twist angle in each section. Consequently, by applying the appropriate translation and rotation matrices to each cross section, section curves are accordingly positioned in the 3D space.

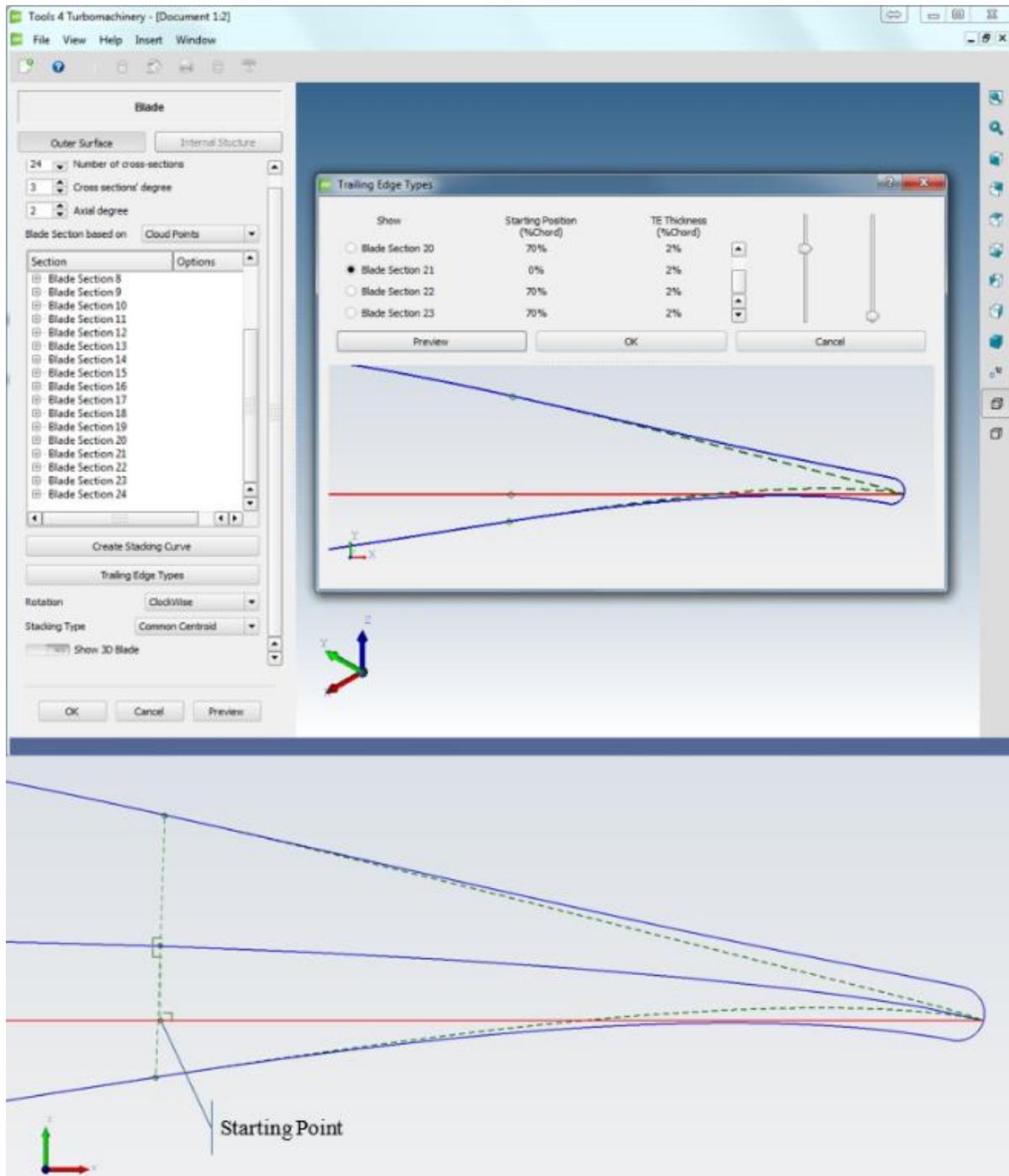


FIGURE 2.2: Modified trailing edge thickness.

The external blade surface is produced by the procedure of "skinning" through these sections. According to this procedure interpolation must be performed in both directions (circumferential and axial). The user should first specify the degrees ( $p, q$ ) of the resulting surface ( $p$  in the circumferential direction,  $q$  in the axial (span-wise) direction of the blade). Each section should have the same number of interpolation points so as to create a feasible

surface. The interpolation is initially performed in the circumferential direction. The needed points for this interpolation are computed using a full cosine distribution applied at the current cross-section curves, in order to properly adapt at the leading and trailing edges of each section; the user can specify the required number of points for this representation. The curves obtained from this procedure coincide to the original cross sections of the blade, but all have been produced with the same number of interpolation points. Subsequently, the generated NURBS curves are interpolated in the axial (span-wise) direction for the production of the control lattice that describes the final 3D surface of the blade (Piegl & Tiller, 1995; Koini *et al.*, 2009). The customized methodology for the lofting procedure ensures the continuity of the surface, as well as the accurate adaptation of the surface at the leading and trailing edge areas. It was realized that the lofting procedure of some commercial CAD software failed to accurately represent the blade geometry near leading and trailing edges.

"T4T" software provides the ability to define the number of sections which determine the blade, as well as the degree of the NURBS curves in the circumferential and axial directions. In order to construct a feasible surface, all sections should have the same degree in the circumferential direction while in axial direction the degree cannot be greater than the number of sections minus one. To construct the cross sections the user should specify one of the two geometrical definition methods provided by the software. The ability of 2-D and 3-D preview is available along the whole phase of the design procedure. All information about the created curves and surfaces is stored on a construction tree on the main window of the software, enabling the user to make modifications easily. A sample design of a 3-dimensional blade surface is presented in FIGURE 2.3 along with the blade sections, which were used to produce its outer surface.

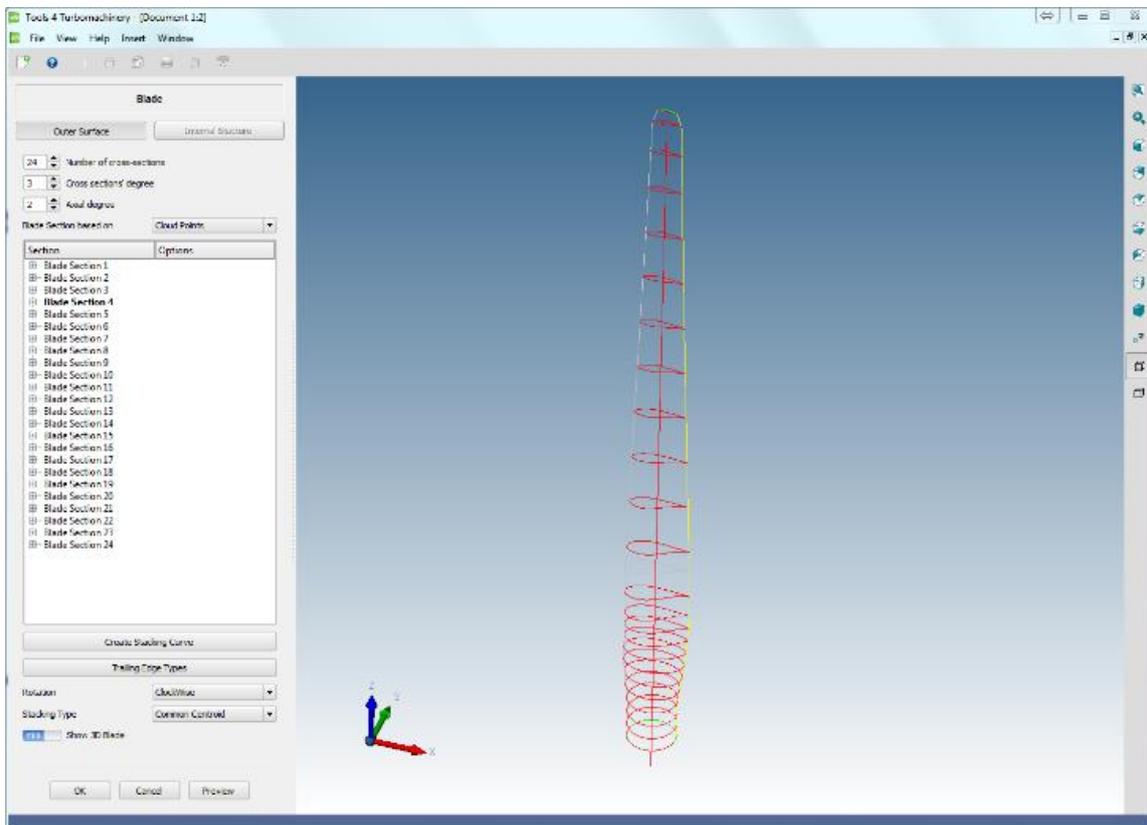


FIGURE 2.3: 3-dimensional blade along with its cross-sections.

## 2.3 Internal structure

The structural definition approach used in "T4T" is general and representative of the current commercial blade designs and design trends. The whole procedure is fully parametric and customizable, allowing the user to create thick blades with shear webs.

### 2.3.1 Thickness Assignment

In order to assign the different thicknesses (and materials) along the blade, the number of regions in which the blade is to be split should be specified. Such regions are defined by selecting the desirable cutting planes, which are perpendicular to the stacking line. Thus, for each region a lower and an upper cutting plane are defined, either by specifying its absolute span-wise position, or as a percentage of blades' span. Subsequently, the blade surface is sectioned by the defined cutting planes and each resulting zone is manipulated separately, so as to specify different laminate thicknesses and material properties. FIGURE 2.4 presents a 3-dimensional surface that is sectioned in 10 different regions. The developed graphical user interface provides several functions for the interactive selection, definition and visualization of the corresponding blade regions.

For each blade region the bounding cross-sections are computed, as well as their mean camber lines and chord lines. The computation of such blade sections and their geometrical characteristics is performed due to the fact that they do not necessarily coincide with the initial sections, used for the construction of the blade's surface. This characteristic provides the user with the ability to define unlimited number of blade regions in arbitrary positions.

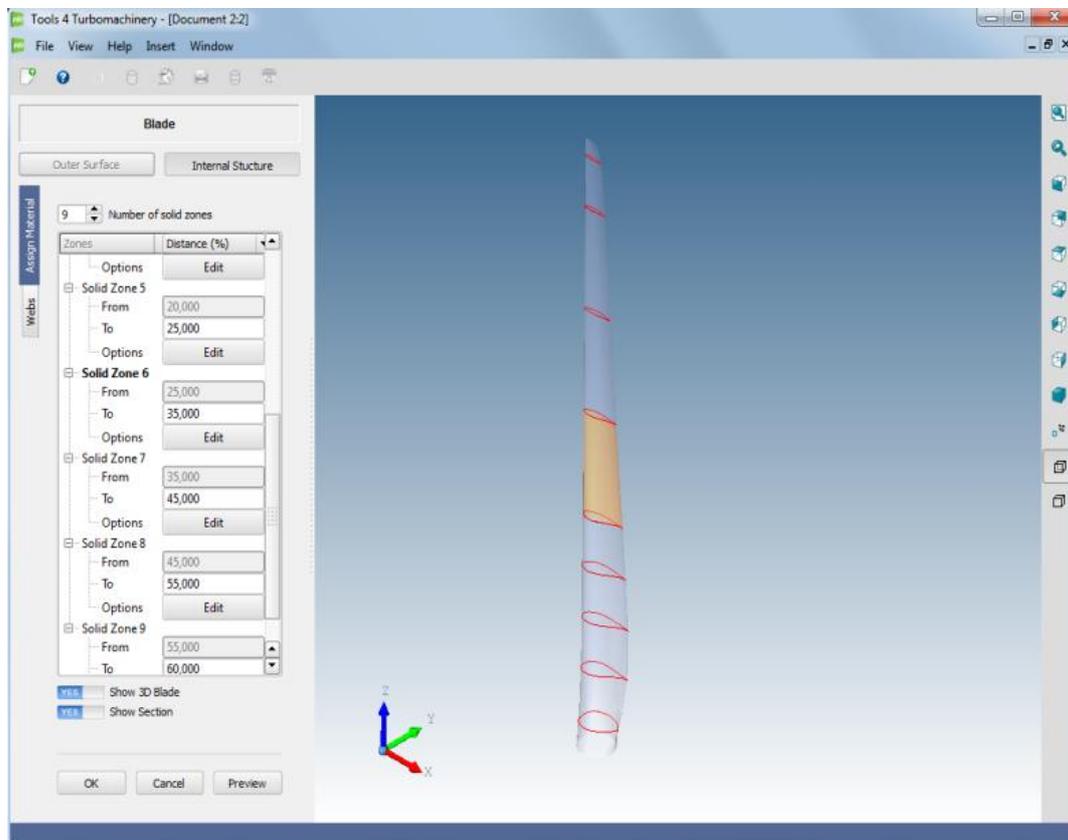


FIGURE 2.4: Blade surface sectioned in 10 regions.

Subsequently, for each blade region, using the options tree at the left side of the graphical user interface (FIGURE 2.5), the user may determine unlimited splitting points, on the suction and pressure sides for each one of the two bounding section curves of each blade region. The number of splitting points between the two bounding sections must be the same; this is ensured by the fact that such points are defined using the same spin-box for both bounding sections, as shown in FIGURE 2.5.

The splitting points for each bounding section are defined as percentages of a reference curve; this curve may be either the chord or the mean camber line of the corresponding section. In order to define these points, the total length of the reference curve should be pre-computed. Thereinafter, the perpendicular vector to the reference curve is computed at the preferred position; the intersection of that vector with the pressure or suction side provides the requested splitting point.

After the definition of all splitting points on the two bounding sections that define the region, the computation of the curves that join the corresponding points at the two sections takes place. These joining curves are mapped on the blade's surface. For each pair of corresponding points, at the two bounding sections that define the region, the parametric coordinates  $(u, v)$  of the NURBS surface are computed. In accordance with these coordinates, a 2-dimensional NURBS curve is created on the parametric space, in order to pass through these points. Finally, this curve is mapped on the 3D surface of the blade, creating the 3D NURBS curve that splits the surface. Subsequently, using the splitting points on each blade section, the blade sections are split in successive curves, which, along with the joining curves, define surface patches (FIGURE 2.5). As a result, each blade region has been split to several successive patches. Depending on the number of the splitting points (on pressure and suction sides of the bounding sections), different numbers of patches can be created in each blade region, where different laminate thicknesses can be assigned, as shown in FIGURE 2.5 and FIGURE 2.6. This procedure is fully interactive and can be performed through the GUI.

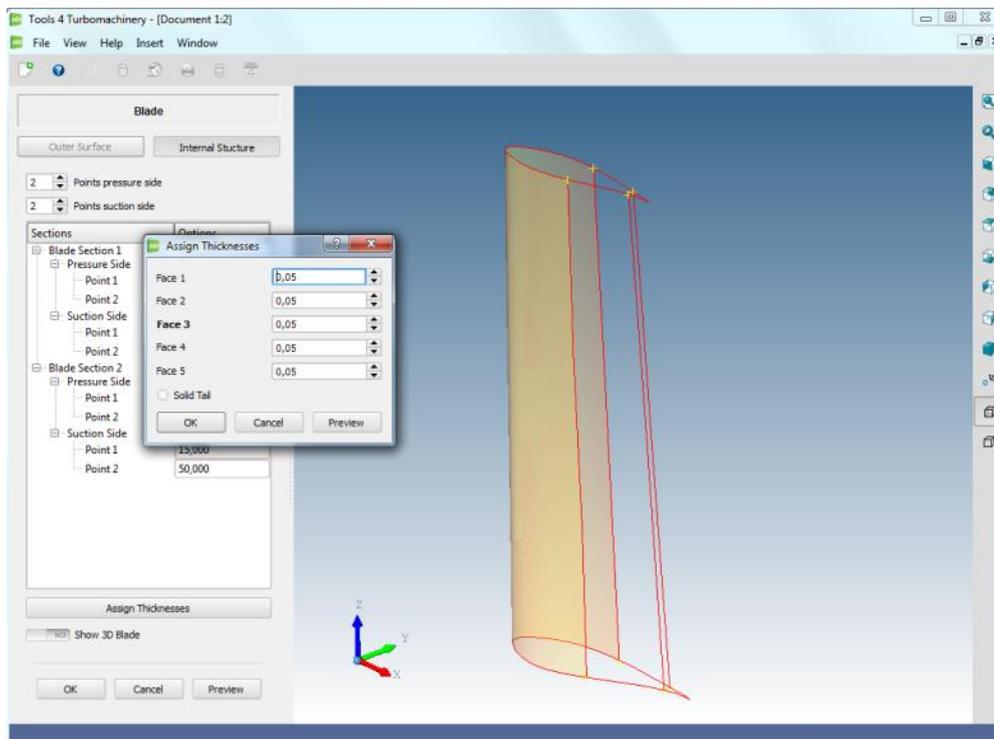
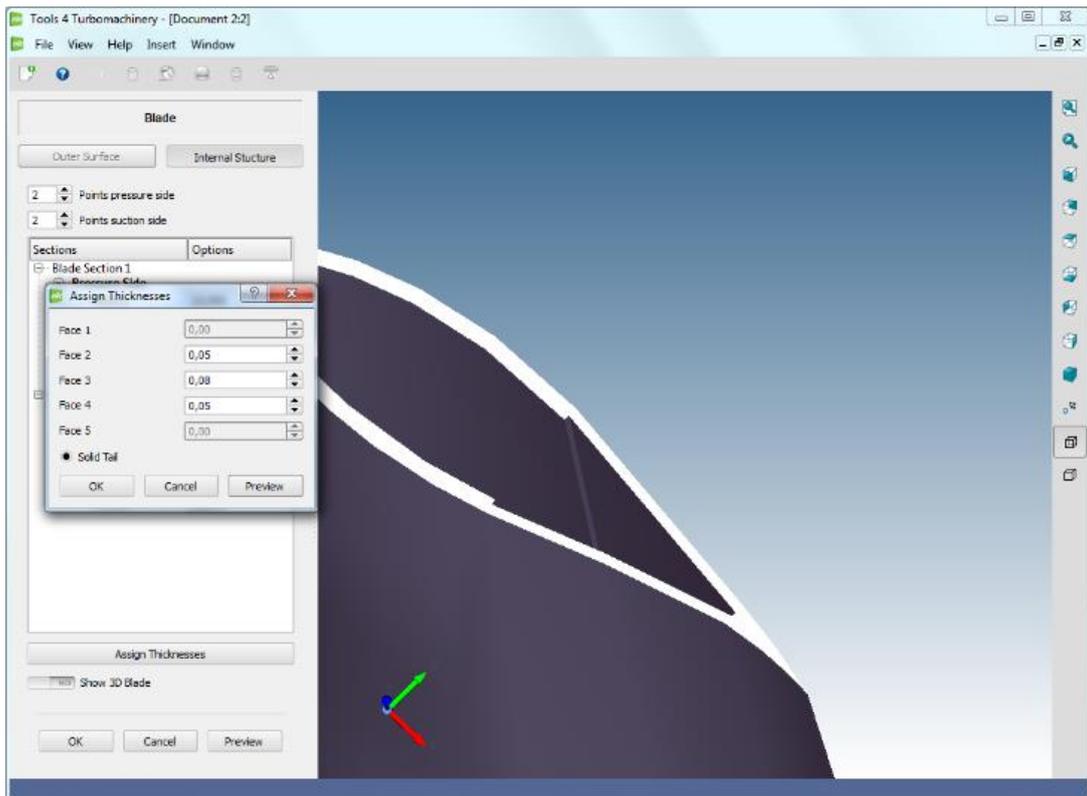


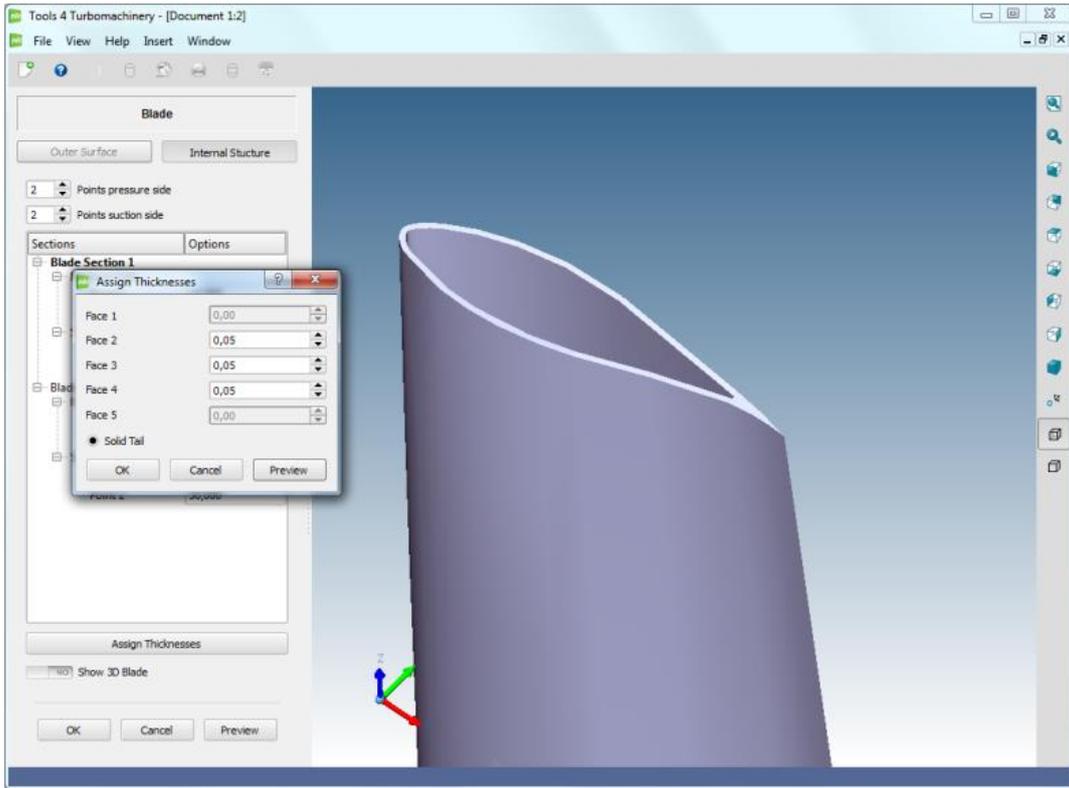
FIGURE 2.5: Blade region split in 5 faces.



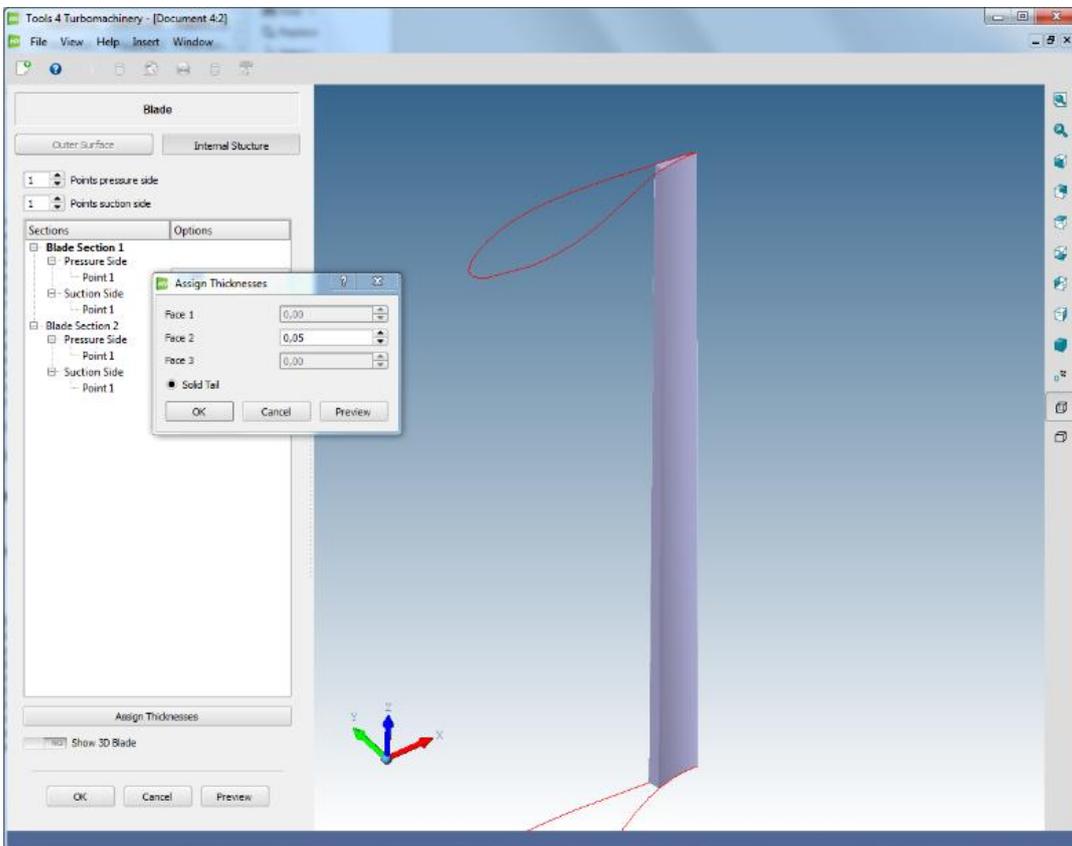
**FIGURE 2.6:** *Different laminate thicknesses.*

Each resulting solid patch is computed using an "offset" procedure, applied to the corresponding surface patch with a prescribed value of thickness. In order to realize this "offset" procedure, the normal vectors to each point of the surface are computed, and the prescribed thickness is distributed all over the selected patch surface, as shown in FIGURE 2.7.

The ability to define a solid tail at the ending of the surface region is also provided. The solid tail is constructed by "sewing" the following surfaces: the first patch on the pressure side and the first patch on the suction side of the blade region (starting from the trailing edge), the two (almost triangular) ruled (planar) faces on the bounding sections, bounded by the two previous surfaces, and, finally, the sweep surface that is produced by the corresponding 3D mapped curves on the blade surface that define the two patches (FIGURE 2.8).



*FIGURE 2.7: Solid region with solid tail.*

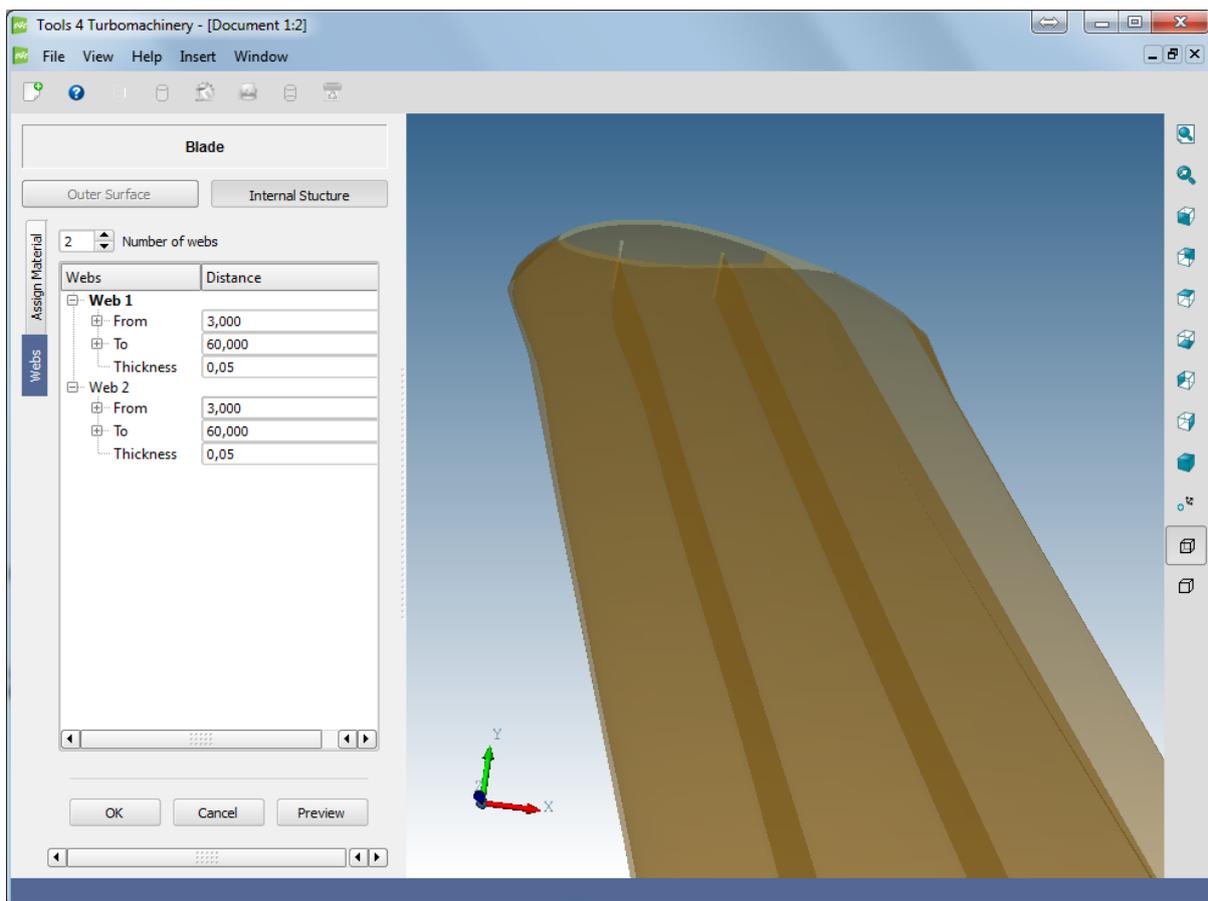


*FIGURE 2.8: Solid tail formation for a blade section.*

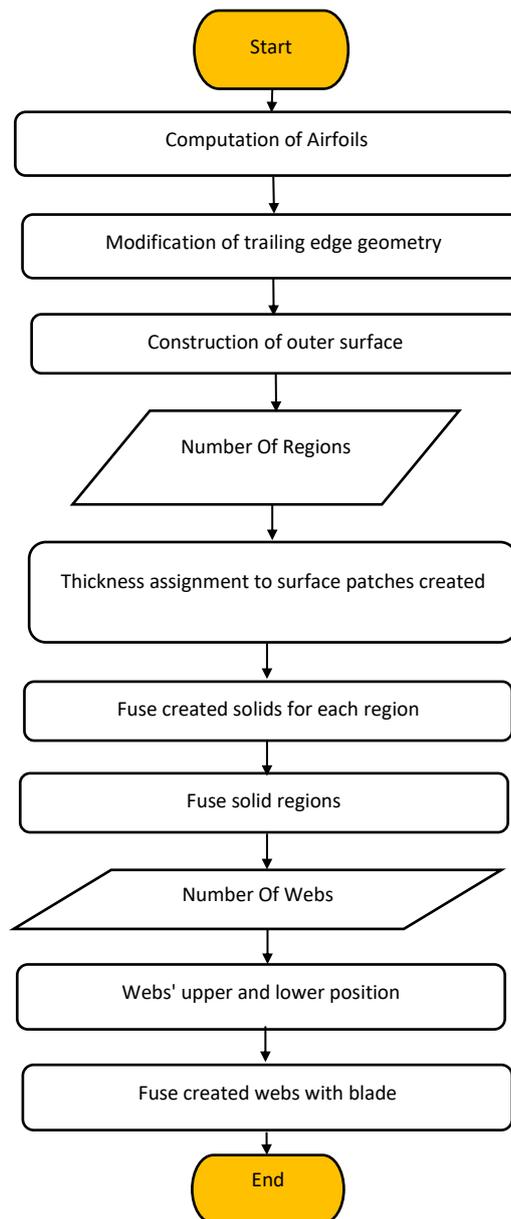
Finally, all these solid patches are fused into one solid through the use of Boolean operations (provided by OpenCascade library). However, since all these solid patches are produced by means of normal vectors on the blade surface, the solids' boundaries for each region may not coincide with the cutting planes bounding the corresponding region. To address this problem, the solid patches are extended outside the boundaries of the blade region and subsequently are cut by the bounding (planar) sections, in order to produce consistent blades without geometric discontinuities between successive regions. This procedure is repeated for every region, producing the final 3-dimensional solid blade.

### 2.3.2 Shear Webs

In "T4T" software the shear webs are computed as sweep surfaces to which a desirable thickness is symmetrically applied to both sides; the number of webs is unlimited. Each web is determined by a starting and an ending plane section, in a similar way to blade regions. For each web, the user should specify its position along sections' chord length for the two bounding planar sections (the lower and the upper one). The position and the possible twist of the web arises from the determination of the four intersection points at the pressure and suction sides of the lower and upper bounding sections.



*FIGURE 2.9: Definition of two shear webs.*



**FIGURE 2.10:** Flowchart of the blade definition procedure.

Each one's position is defined as a percentage of the corresponding section's chord. The mapping curves on blade's surface, which join the corresponding points at pressure and suction sides accordingly, are computed and used as rail curves: the profile of the web is produced as a sweep surface along these two rail curves. A blade consisting of two shear webs is presented in FIGURE 2.9: the webs were defined between a lower and an upper plane section; different characteristics of these webs may be defined in the remaining length of the blade. A flowchart of the complete blade definition procedure is presented in FIGURE 2.10.

## 2.4 Sample 3D blade design

### 2.4.1 Geometry definition

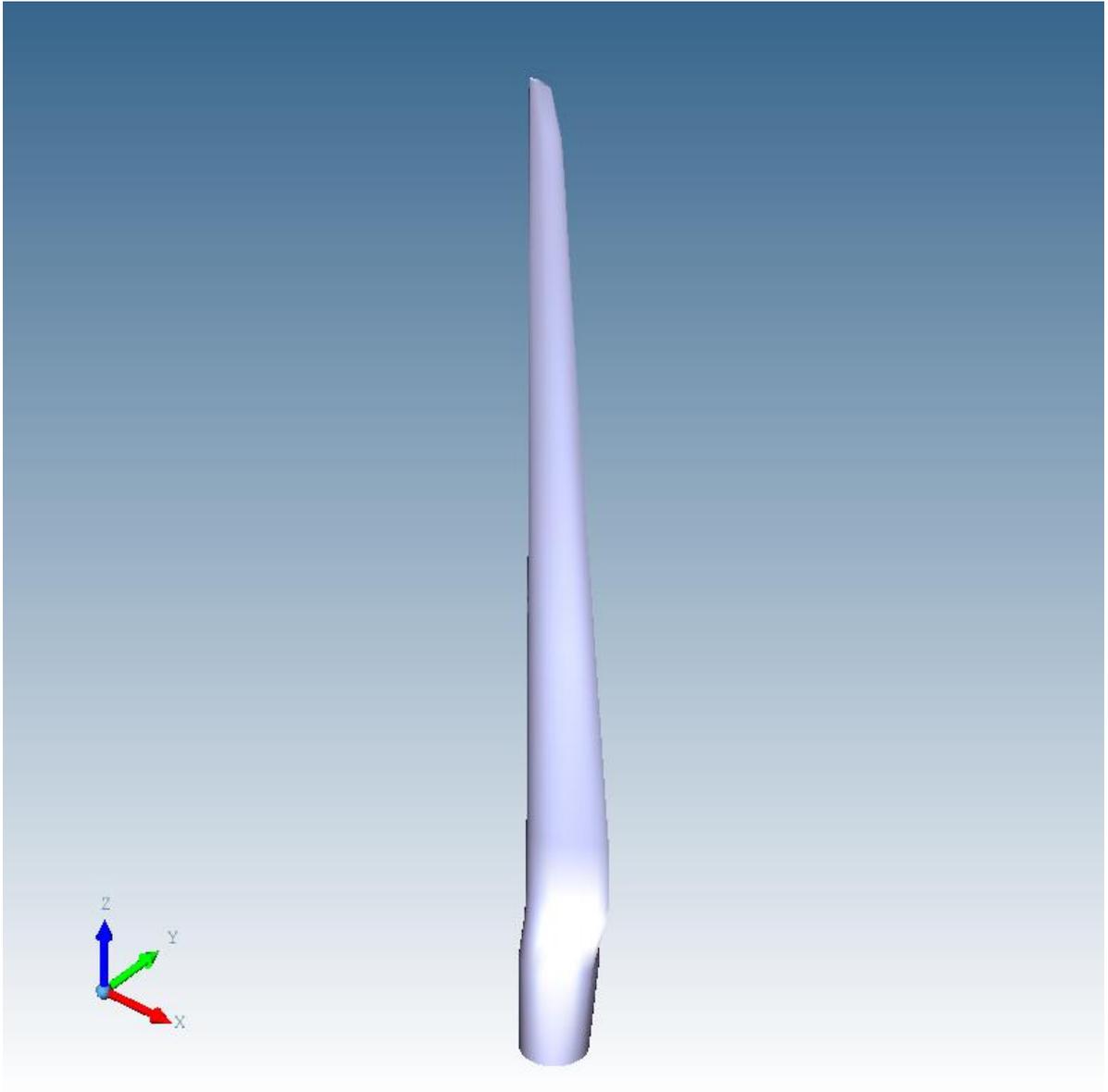
The geometrical data of the frequently used “NREL offshore 5-MW baseline wind turbine” (Jonkman *et al.*, 2009) were used as a starting point to demonstrate the corresponding capabilities introduced to the "T4T" software. The geometric details of the blades used in this work are those contained in (Martin, 2011). These data concern the cross-sections of the blade and are contained in TABLE 2.1.

**TABLE 2.1:** Distributed blade aerodynamic properties.

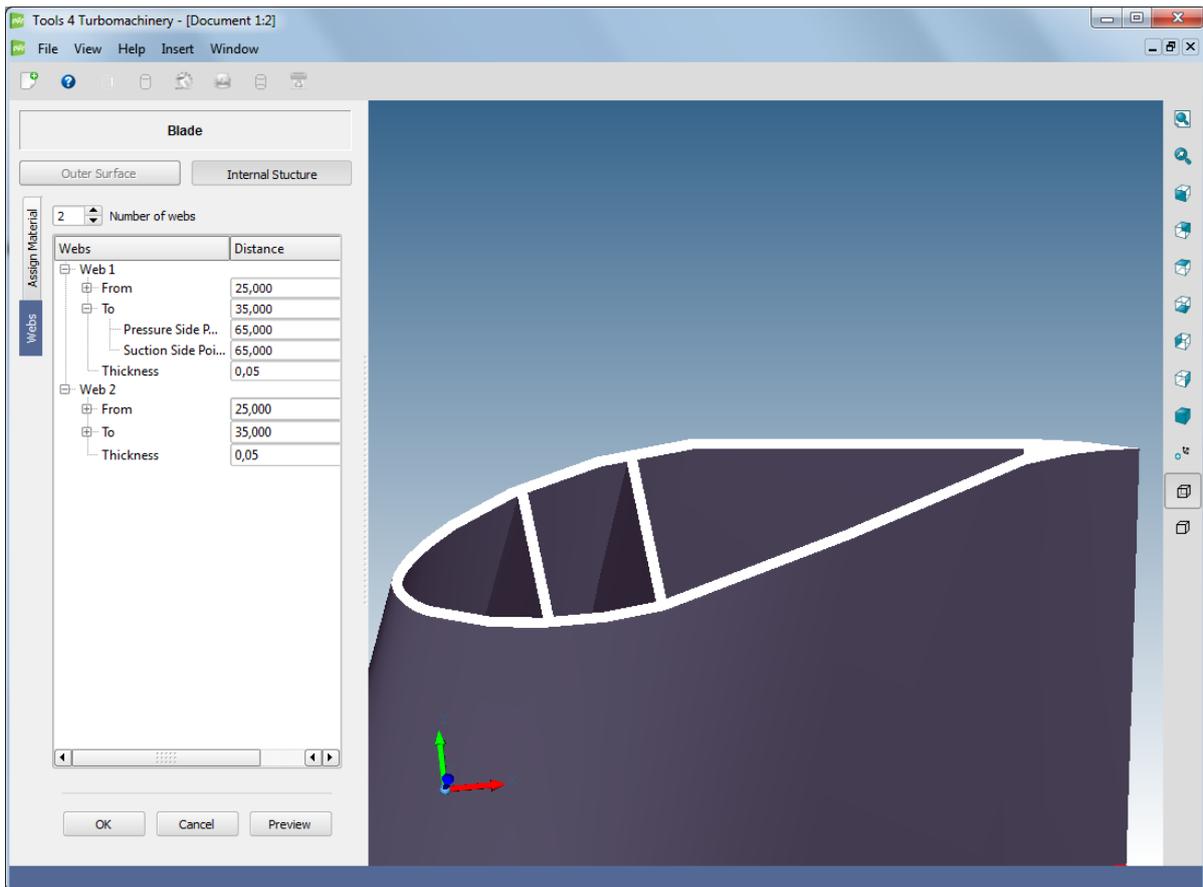
<i>Node</i>	<i>RNodes(m)</i>	<i>AeroTwst</i>	<i>Chord</i>	<i>Airfoil Type</i>
1	2.8667	13.308	3.542	Cylinder1
2	5.6000	13.308	3.854	Cylinder1
3	8.3333	13.308	4.167	Cylinder2
4	11.7500	13.308	4.557	DU40_A17
5	15.8500	11.480	4.652	DU35_A17
6	19.9500	10.162	4.458	DU35_A17
7	24.0500	9.011	4.249	DU30_A17
8	28.1500	7.795	4.007	DU25_A17
9	32.2500	6.544	3.748	DU25_A17
10	36.3500	5.361	3.502	DU21_A17
11	40.4500	4.188	3.256	DU21_A17
12	44.5500	3.125	3.010	NACA64_A17
13	48.6500	2.319	2.764	NACA64_A17
14	52.7500	1.526	2.518	NACA64_A17
15	56.1667	0.863	2.313	NACA64_A17
16	58.9000	0.370	2.086	NACA64_A17
17	61.6333	0.106	1.419	NACA64_A17

In TABLE 2.1, “RNodes” refer to the blade node locations, which are directed along the blade-pitch axis from the rotor center (apex) to the blade cross sections, while AeroTwist is the aerodynamic twist angle, in degrees. The airfoils’ type Cylinder1 represents cylinders with drag coefficients of 0.50, Cylinder2 with drag coefficients of 0.35 and no lift. For DU40\_A17 the “DU” refers to Delft University and “NACA” refers to the National Advisory Committee for Aeronautics. All these airfoil types are included in the documentation of the NREL case.

However, even though NREL blade is well documented regarding the aerodynamic characteristics, several details are missing regarding the construction of the 3D blade surface. Information gaps exist concerning the coordinates of the 2D airfoils and their stacking positions. Therefore, all non-dimensional 2D airfoil geometries and blade pitch axis locations were obtained through other sources (“NREL 5MW Rotor Geometry - NWTC,” n.d.; Martin, 2011; Resor, 2013). Combining all the available afore-mentioned resources, resulted in a blade which was constructed using "T4T", as shown in FIGURE 2.11. Reasonable custom laminate thicknesses were applied on its surface to produce a solid blade with a solid trailing edge and two shear webs, as shown in FIGURE 2.12.



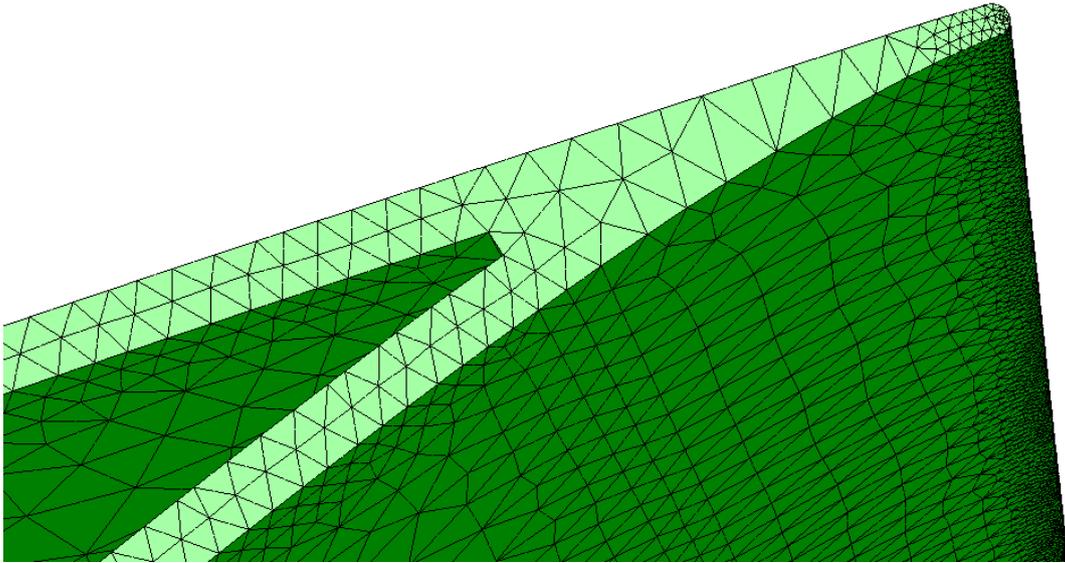
**FIGURE 2.11:** NREL 5mw blade surface.



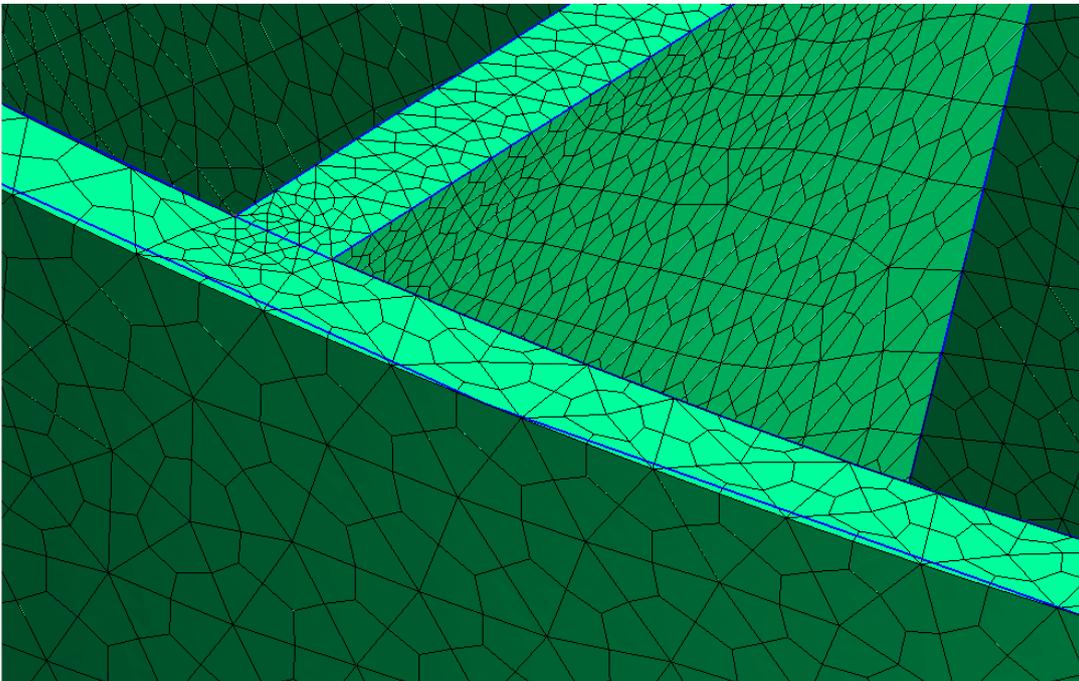
*FIGURE 2.12: Cross-section of a solid blade with solid tail and 2 shear webs.*

## 2.4.2 Mesh generation

As soon as the geometry is defined, a STEP file is extracted, which is subsequently imported to the mesh generation software, to initiate the analysis phase of the blade. The open source GMSH (Geuzaine & Remacle, 2014) mesh generation software was used for generating the mesh of the internal structure. However, there is no limitation on the mesh generation software that can be used, as other types of mesh generation software have been also tested. GMSH has the ability to create three-dimensional mesh, while it has embedded pre- and post-processors. It is a fast, light and easy to use mesh editor, which also incorporates parameterization characteristics. One of the advantages of this software is its ability to input and recognize different formats of geometric files, including STEP format, which renders the communication with "T4T" a straightforward procedure. Additionally, GMSH has the ability to create unstructured and structured mesh, using as a basis an initial surface mesh.



**FIGURE 2.13:** Blade mesh with solid tetrahedral elements, focused on trailing edge.



**FIGURE 2.14:** Blade mesh with solid hex elements.

In FIGURE 2.13 and FIGURE 2.14 two different types of meshes on the exported geometry are presented, produced using GMSH. FIGURE 2.13 focuses on the rounded trailing edge area, presenting a body mesh constructed of tetrahedral elements. FIGURE 2.14 presents a body unstructured mesh utilizing hexahedral elements, constructed through the division of the tetrahedral element of the dual mesh.

The output mesh can be automatically introduced to the *CalculiX* solver (Dhondt, 2014), a widely used open source FEM software. It is able of performing different types of analyses, such as static, modal, dynamic, buckling, heat transfer, CFD, conjugate thermomechanical etc. Furthermore, it has the ability to handle the non-linearity of material and geometry. A mesh converter was also developed in this work to allow for the automatic import of mesh files from GMSH to *CalculiX*.

TABLE 2.2: Eigenvalues of the fixed root blade.

Mode	Frequency [Hz]		Description
	"T4T"-CalculiX	(Laco, 2012)	
1	0.6613	0.7066	1st flapwise bending
2	1.0748	1.0188	1st edgewise bending
3	1.8912	1.8175	2nd flapwise bending
4	3.2338	3.3403	2nd edgewise bending
5	3.7823	3.9493	3rd flapwise bending
6	5.8526	6.4682	1st torsion

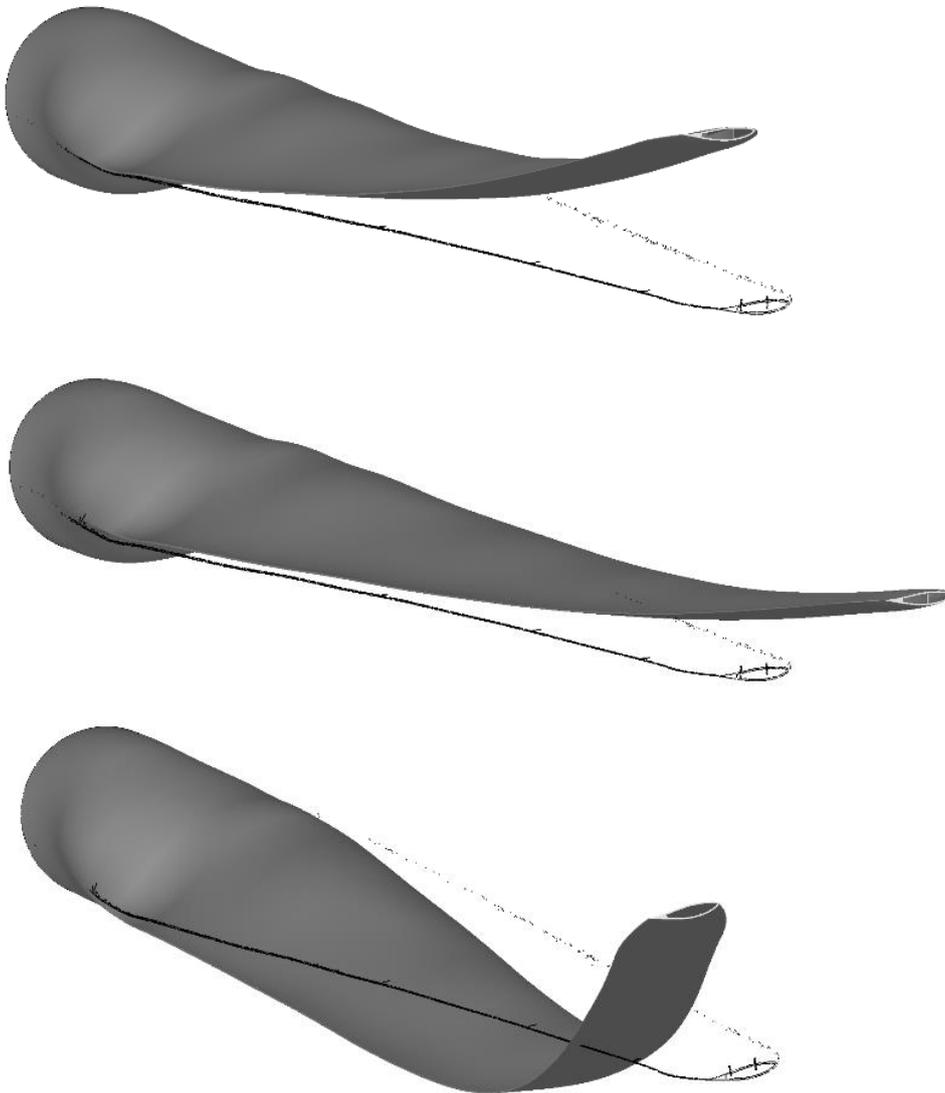


FIGURE 2.15: Three first modes of a blade model with a fixed root.

In order to demonstrate the functionality of the complete computation procedure (blade definition - mesh generation - finite element analysis) a modal analysis of the constructed blade was performed with *CalculiX*. The first six modes, compared with the corresponding

results contained in (Laco, 2012), are summarized in TABLE 2.2. FIGURE 2.15 presents the calculated first three deformation modes of the blade.

## **2.5 Conclusions**

This chapter focused on the description of the additional functions developed for the "T4T" software that provide the ability to define wind turbine blades with internal structure in a full parametric way. Variable material thickness and different material properties can be applied in different blade regions in an interactive manner, through a user friendly graphical interface. After the interactive definition of the blade geometry (both 3D surface and internal structure) the software may be used in an automated way (batch mode) to produce other candidate geometries in an optimization cycle, while retaining its topology unchanged.

# Chapter 3: The Parallel Asynchronous Differential Evolution Algorithm

## 3.1 Introduction

Evolutionary Algorithms imitate nature's selection process using a population-based searching mechanism for dealing with demanding high-dimensional real-world optimization problems. They constitute a class of search methods with remarkable balance between exploitation of the best solutions and exploration of the search space, as well as low sensitivity to local minima treatment. They combine elements of directed and stochastic search and, therefore, are more robust than directed search methods. However, despite the important contribution of EAs in solving complicated problems, they suffer from a significant drawback; a considerable number of evaluations is needed, which usually calls for significantly increased computation time. In order to overcome this barrier, the use of surrogate models (metamodels), in conjunction with parallel processing, are commonly used; see for example (Giannakoglou *et al.*, 2001, 2006; Emmerich *et al.*, 2002, 2006; Giannakoglou, 2002; Karakasis & Giannakoglou, 2006; Asouti *et al.*, 2009; Kapsoulis *et al.*, 2018), (Karakasis *et al.*, 2007; Asouti & Giannakoglou, 2009; Kampolis & Giannakoglou, 2009) and the references therein.

In this Chapter, the development of a parallel asynchronous DE algorithm is presented. It is based on the panmictic approach, by using a unique population that is distributed among the processors with a Master-Slave architecture. Furthermore, the utilization of two ANNs as surrogate models within the DE algorithm enhances its performance by substituting the computationally time-consuming exact evaluations of the fitness function with low-cost approximations.

The outline of this chapter is as follows: In Section 3.2 a detailed description of the DE algorithm is presented, while Chapters 3.3 and 3.4 present the incorporation of the surrogate models and the parallel implementation.

## 3.2 The Differential Evolution Algorithm

Within the proposed numerical optimization scheme, a Differential Evolution (DE) Algorithm is utilized, which is a versatile stochastic search method, introduced by Storn and Price (Storn & Price, 1995, 1997), capable of handling non-differentiable, nonlinear and multimodal cost functions, providing superior convergence performance than other EAs (Storn & Price, 1995, 1997; Price *et al.*, 2005). Contrary to other EAs, the DE compares each new candidate member (offspring) of the population only against a single existing one (parent), which is its counterpart in the current population. The new parameter vector (offspring) is a linear combination between a randomly selected member of the current population (chromosome) and a weighted difference between two other randomly chosen chromosomes.

Below, an analytical description of the basic elements composing a classic DE algorithm is presented. Given a cost function

$$f_{cost}(\mathbf{x}) = f_{cost}(x_1, x_2, \dots, x_n) \rightarrow \min \quad (3.1)$$

where,  $\mathbf{x}$  denotes the vector containing the  $n$  design variables (number of genes) of the problem under consideration and  $f_{cost}(\mathbf{x}): \mathbb{R}^n \rightarrow \mathbb{R}$  a real function. The optimization target is the minimization of the cost function  $f_{cost}$  by modulating the values of its design variables, while each one of the design variables is bounded between an upper  $x_i^u$  and a lower  $x_i^l$  value. Differential Evolution evolves a fixed size population of  $N_p$  individuals (chromosomes) for a finite number of generations  $G_{max}$ . The initialization of the first population is established by randomly assigning values to the design variables within their given boundaries

$$x_{k,i}^0 = r \cdot (x_i^u - x_i^l) + x_i^l, \quad i = 1, \dots, n, \quad k = 1, \dots, N_p, \quad (3.2)$$

$$x_i^l \leq x_{k,i}^0 \leq x_i^u,$$

where  $r$  denotes a random number generated with uniform probability within the range  $[0, 1]$ . After the evaluation of each individual's cost function, operators are applied to the population, simulating the according natural processes. The first operator applied is the mutation scheme, which generates a new chromosome (mutant), based on three randomly selected individuals (chromosomes) of the current generation  $G$ . The formation of the new parameter vector is realized by adding a weighted difference vector between the two members of the triad to the third one, the so-called "donor". Then, the uniform crossover scheme is applied; the mutant and the chromosome of the current population (parent) are subjected to a discrete recombination, which produces the final candidate solution

$$x_{k,i}^{G+1} = \begin{cases} x_{C_k,i}^G + F(x_{A_k,i}^G - x_{B_k,i}^G) & \text{if } r \leq C_r \text{ or } i = i^* \forall i = 1, \dots, n \\ x_{k,i}^G & \text{otherwise,} \end{cases} \quad (3.3)$$

$$k = 1, \dots, N_p, i = 1, \dots, n$$

$$A_k \in [1, \dots, N_p], \quad B_k \in [1, \dots, N_p], \quad C_k \in [1, \dots, N_p], \quad A_k \neq B_k \neq C_k$$

$$C_r \in [0,1], \quad F \in [0, 1 +], \quad r \in [0,1],$$

where  $x_{C_k,i}^G$  are the elements of the "donor" vector (chromosome),  $G$  is the current generation and  $i^*$  is a randomly selected integer within  $[1, n]$ , chosen once for all members of the population. The random number  $r$  is seeded for every gene of each chromosome whereas the parameters  $F$  and  $C_r$  consider the mutation and crossover operations, respectively. Specifically, the scale factor  $F$  controls the diversification rate of the population, while the crossover probability  $C_r$  controls the fraction of design values that are inherited from the mutant. Moreover, the design variable, which corresponds to the randomly selected index,  $i^*$ , is taken from the mutant to ensure that the trial vector (offspring) does not duplicate the initial one (parent). Scaling vector differences ensures that trial vectors do not duplicate existing members in the population. Additionally, scaling can shift the focus of the search between local and global.

Subsequently, each member of the resulting intermediate population (offspring) is evaluated and competes against its counterpart in the current population (parent); the best-fitted individuals are the ones that will form the next generation. The DE selection scheme ensures the survival of the elitists and can be described as follows (for minimization problems):

$$\mathbf{x}_k^{G+1} = \begin{cases} \mathbf{x}_k^{G+1} & \text{if } f(\mathbf{x}_k^{G+1}) \leq f(\mathbf{x}_k^G), \\ \mathbf{x}_k^G & \text{otherwise.} \end{cases} \quad (3.4)$$

The process is successively repeated, providing populations with better-fitted individuals.

The DE selection scheme has a substantial difference compared to other EAs, owing to the fact that the offspring is not compared against all the members of the current population, but only against its parent, replacing it if better-fitted. This important characteristic allows for a relatively easy implementation of an asynchronous parallelization procedure, as it will be described in a following paragraph. If each population member is assigned to a different processor (or core), this processor can proceed to the evaluation of a new individual, after completing the evaluation of its parent. Communication between the different processors is mainly required in order to perform the mutation operation. This does not necessitate the existence of a generation in the strict sense; a population comprising chromosomes belonging to different generations may be used instead. Additional communication with the master node is needed in the case of utilizing surrogate models, which are re-trained in each generation (which is the case in our implementation).

### 3.3 The combined use of surrogate models

In each DE generation, each trial vector (offspring) must be first evaluated (by computing its cost function using simulation software) and then compared with its parent, so as to select the better-fitted between them to pass to the next generation. The computation of the offspring's cost function is (in most real world applications) a time-consuming operation. The concept of utilizing surrogate models in this evaluation procedure is to replace the costly exact evaluations with fast inexact approximations, without sacrificing the robustness of the DE algorithm. These surrogate models are established using a data driven-approach, where only the input and output behavior of the simulation model of the cost function is taken into account, in order to create a mechanism that mimics that behavior. Two types of Artificial Neural Networks (ANNs) are used as surrogate models; a Multi-layer Perceptron (MLP) and a Radial Basis Function (RBF) ANN, respectively. A detailed description of their implementation within the DE optimizer can be found in (Nikolos, 2011, 2013), along with related references to available types of surrogate models and their combination with EAs. Each offspring is pre-evaluated, using the available surrogate models, in a fast screening procedure. If an offspring is pre-evaluated and found lower-fitted than its parent, then no further exact evaluation is taking place, and the current vector (parent) is transferred to the next generation, while the offspring is abandoned. In the opposite case, where the offspring is pre-evaluated as better fitted than its parent, an exact (and costly) re-evaluation is performed after the pre-evaluation, along with a second comparison between the two vectors. If the offspring is found again better-fitted than its parent, then the offspring passes to the next generation. Otherwise, its parent will pass to the next generation and the offspring will be abandoned.

An additional small percentage (5%-10%) of the candidate solutions are selected with uniform probability to be exactly evaluated, without taking into account their pre-evaluation by the utilized surrogate models, to further enhance the robustness of the procedure. Moreover, in the first two generations of the DE, all trial vectors are exactly evaluated (without using the surrogate models), as to initialize the central database (pool) required for the training of the surrogate models. As it was previously described, only exactly-evaluated

candidate solutions have the opportunity to pass to the new generation. Consequently, in each generation the current population always comprises individuals that have been selected using exact evaluation. Therefore, one part of the comparison (the parent) in the pre-evaluation phase is always an exactly-evaluated vector, and this enhances the robustness of the procedure. It should be emphasized that the surrogate model predictions replace exact and costly evaluations only for the less-promising individuals, using the pre-evaluation phase to quickly reject them without spending valuable computational resources to exactly-evaluate them.

Each evaluated chromosome, along with its resulted fitness function value, are stored in the central database. The training and testing data sets are selected in each generation from the corresponding database, to be used by all available surrogate models. If  $NR$  is the length of the training set and  $NT$  is the length of the testing set (defined by the user in the beginning of the optimization procedure), the  $NR + NT$  best members of the central database are deterministically selected and, from this set,  $NR$  members are randomly selected (with uniform probability) to be utilized as the training set, while the rest  $NT$  are being used for testing. In this way the surrogate models, which are re-trained and re-tested in each generation, evolve with the population and use only the currently most-promising individuals for approximating the cost function.

The utilized surrogate models can be used either independently or as an ensemble. In the first case, a single surrogate is used throughout the whole optimization procedure; while in the second case all surrogates are re-trained and re-tested in each generation (using the same training and testing data sets for all surrogates). Then, only the best one is used in the pre-evaluation phase of the trial vectors. The selected surrogate (different in each generation) is the one with the lower value of the testing error. The second procedure is usually preferred, since it is not known a-priori which is the best surrogate for a new cost function and for each region of the cost function, thus this automated procedure decides in each generation for the surrogate to be used. The re-training of all the available surrogates in each generation adds negligible cost, compared to the cost of the evaluations (Nikolos, 2004, 2011, 2013; Strofylas & Nikolos, 2015).

### 3.4 Parallel implementation

Despite the important contribution of Evolutionary Algorithms (EAs) in solving complicated problems, they tend to be excessively time-consuming, since they require a considerable number of evaluations. Thus, appropriate acceleration through parallel processing is commonly used (Karakasis *et al.*, 2007; Kampolis & Giannakoglou, 2009); this is supported by the fact that EAs are inherently parallel algorithms, as they deal with a population of different candidate solutions in each generation. The concept behind the developed parallelization strategy is to enable the cooperation of the DE with different simulation software in the form of executables. The required data transfer between the DE and the simulation software is succeeded with appropriate text files, while the communication among the processors and the parallel implementation is achieved using Message Passing Interface (MPI) library functions. The proposed strategy appears to be quite efficient, regardless of the use of text files, considering that the computational time of data transfer is negligible compared to that of the evaluation step (for most engineering optimization problems).

The population members are distributed a priori among the available processors; each processor is in charge of the evaluation of one individual. Next, a unique rank is assigned to

each processor, while one of them is identified as the master node that keeps track of the whole procedure. This node performs all the pre-process that is required prior to the beginning of the optimization procedure, which includes the creation of a working folder for each processor where the executables comprising the evaluation step and their corresponding text files are replicated. Furthermore, the master node distributes all the necessary information concerning the DE algorithm to all other processors, i.e., the number of the design variables, their upper and lower bounds and control parameters for the DE algorithm.

In the asynchronous implementation, the generation is not strictly defined and the current population (at each time instant) can comprise individuals belonging to different generations. More specifically, each newly generated trial vector (offspring) can replace its parent (if better fitted) and become a member of the current population, just after the completion of its evaluation process, without waiting for the completion of the evaluation phases of the rest members of the auxiliary population (FIGURE 3.1). Thus, individuals evolve independently, without strict central control or full coordination between generations. Consequently, asynchronous update has the clear advantage that the improved solutions can contribute to the evolution immediately, without time-lags, and can speed up the convergence to become faster than the synchronous update. As the cycle over all the population members is removed, the concept of generations is obsolete in the asynchronous DE.

The choice of a triplet of randomly selected individuals for each population member, used in the mutation DE operator (equation (3.3)), is an issue which emerges as soon as we switch from a synchronous to an asynchronous update population mode. As the generation concept is no longer applicable, the random selection of three members of the current population (at the corresponding time instant) means that those individuals may not belong to the same generation. However, this proved to introduce no convergence problems to the asynchronous DE version; on the contrary, as the various individuals evolve independently to each other, with a faster convergence rate, the randomly chosen triplet is likely to have better characteristics, compared to the synchronous case.

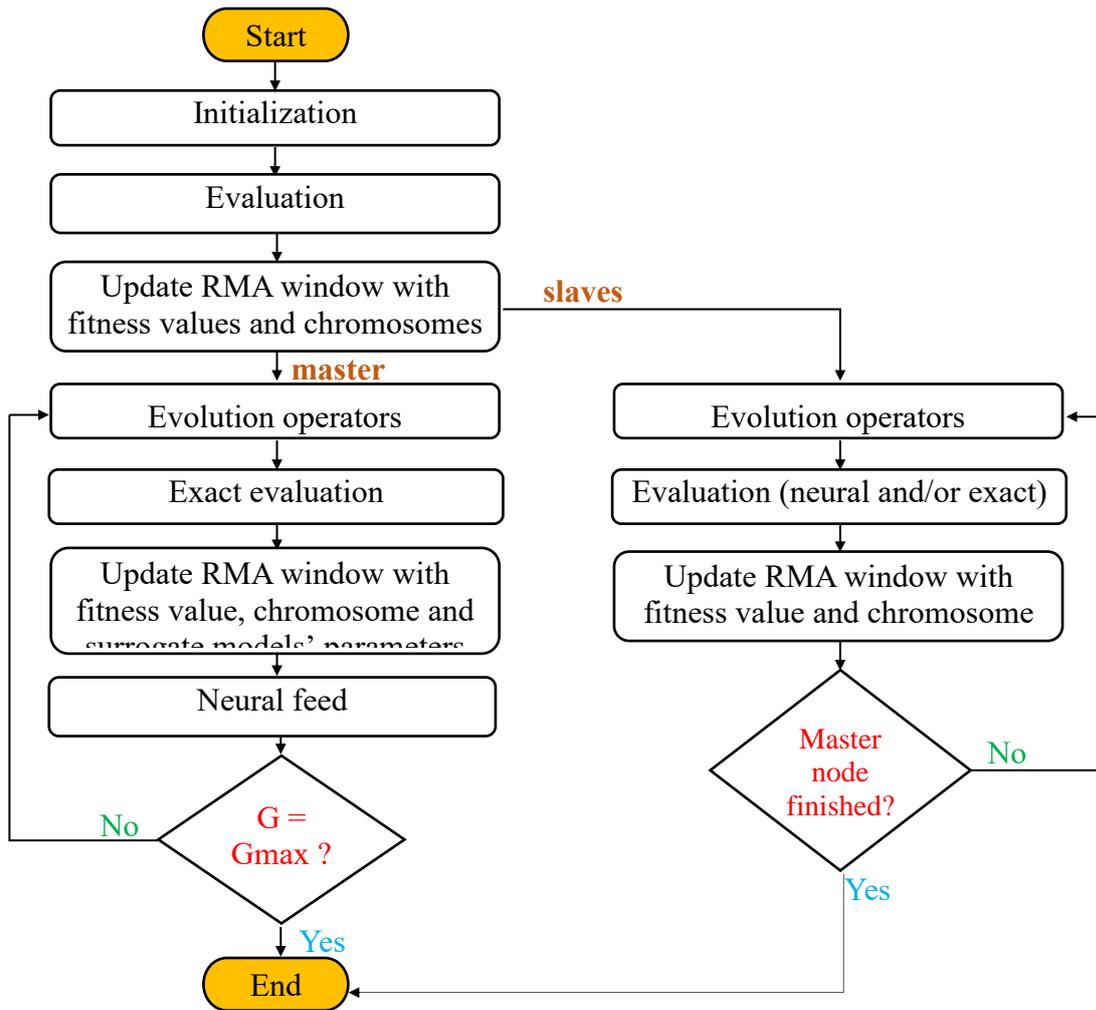
The proposed asynchronous implementation of the DE algorithm uses non-blocking MPI communication operations. An asynchronous master-slave architecture has been followed, i.e., the master process steers the evolution procedure, trains and tests the surrogate models, and collects/distributes data from all other processes. This is achieved using Remote Memory Access (RMA) windows. To allow for remote memory access, the master node exposes contiguous regions of memory to the rest of the processes, which are called windows. MPI accomplishes this by a collective function called "MPI\_Win\_create". A process can get and set data to remote memory via "MPI\_Get" and "MPI\_Put". As MPI forbids concurrent access to the same memory location in the window, it is necessary to have a mechanism which ensures that access operations are completed before using the data. Thus, updates to the RMA windows by other processes are protected by the master node, using exclusive locks in case of a conflict. Non-conflicting accesses (such as read-only accesses or accumulate accesses) are protected by shared locks, both for local accesses and for RMA accesses.

Each slave node works independently, updating periodically only the fitness function value and its corresponding chromosome to the RMA windows. Each one of the slave nodes executes the same process in an infinite loop, checking at first the receiving buffer to see if the termination message from the master node has arrived. Subsequently, it acquires all the updated data, essential for the calculations performed during a DE iteration, which include the fitness values of the candidate solutions and their corresponding chromosomes, as well as

the parameters of the surrogate models through the RMA windows, in order to perform the evolution operators.

Due to the non-blocking communication, the processes never wait for one another. That is, they run completely asynchronously, and the slowest process does not slow down the others. This characteristic is essential, especially when using surrogate models, since the pre-evaluation of the trial vector is performed very fast. This means that, if the trial vector (offspring) is lower-fitted than the corresponding vector of the current population (parent), the processor won't have to wait for all other processors to finish the computations, but instead it can proceed to the production of the next trial vector and its evaluation.

As it has been previously stated, in the asynchronous version of the parallel DE algorithm the concept of generation is no longer present. However, a definition for a "generation" number is needed for defining a frequency for the re-training and re-testing of the surrogate models, for interpreting the convergence history of the optimization run, and for comparison purposes. Thus, the generation number for the asynchronous parallel DE is defined as follows: the master node, which controls the DE procedure, is enforced to always exactly evaluate its corresponding chromosome; therefore, the change between successive individuals for the master node defines a generation for this node, which is also used to define a "generation" for the whole population. This definition is rather connected to computation time intervals, than to a real generation concept. However, as the exact evaluation time for each individual is roughly the same for all processes in a system comprising identical processors or cores (as the one used in this work), the utilized definition of "generation" in the asynchronous DE case renders the comparisons between synchronous and asynchronous runs, with respect to the generation number, valid.



**FIGURE 3.1:** Flowchart of the asynchronous implementation of the parallel Differential Evolution algorithm.

# Chapter 4: Reverse engineering of wind turbine blade surface using Differential Evolution

## 4.1 Introduction

This chapter describes the development of a dedicated shape reconstruction methodology, used for the geometry definition of wind turbine blades from a given set of data points. The resulting blade geometry definition should have a low number of design variables, which include specific geometric parameters, critical to the blade design process. To this end, the proposed shape reconstruction methodology is designed to make the resulting blade definition consistent with the specialized modelling software tool "T4T", that was developed for the parametric design of turbomachinery components and wind turbine blades (refer to Chapter 2). The shape reconstruction is formulated as an optimization procedure, based on a parallel, metamodel-assisted differential evolution (DE) algorithm (Chapter 3). The candidate blade geometries are automatically produced by altering the design variables of a pre-defined "T4T"-model of the blade, while the cost function to be minimized is the sum of the squared distance between each given data point and the resulting NURBS surface of the candidate blade. The proposed methodology is evaluated in specific test cases, and its capabilities are discussed.

The structure of the present chapter is as follows: In Section 4.2 the blade surface parametrization approach implemented by "T4T" is briefly presented, whereas Section 4.3 describes the formalization of the utilized fitness function. In Section 4.4 the performance of the proposed methodology is validated against 2 benchmark test cases, to prove its efficiency and effectiveness. Finally, Section 4.5 summarizes the results and conclusions of the surface reconstruction methodology.

## 4.2 Blade Surface parametrization

In the present dissertation the construction of the blade surface is realized utilizing the "T4T" software. More specifically, as described in Chapter 2, "T4T" is capable of constructing 3D blades, based on several airfoil profiles, which are distributed along a stacking line, according to their predefined positions and twist angles, while the final surface arises through a custom skinning procedure (Koini *et al.*, 2009; Strofylas *et al.*, 2014). The basic design process is related to some physical parameters of the blades that correspond to 2D sections, such as their inlet and outlet angles and their chord length; these are the parameters which will be used as design variables for the optimization procedure, while the proper selection of their limits is of critical importance as it determines the corresponding search space.

The whole procedure begins by formulating the 2D cross sections; each one of them is formed by a mean camber line and a "thickness distribution" imposed to it. The mean camber line is constructed by a 2<sup>nd</sup> degree NURBS curve with three control points,  $P_i$ , where the first ( $P_0$ ) and last one ( $P_2$ ) coincide with the leading and trailing edges of the airfoil, respectively. The middle control point ( $P_1$ ) is located at the intersection of two straight lines that cross the

leading and trailing edges, tangent to the mean camber line (FIGURE 4.1); since the control polygon of an open non-periodic (clamped) NURBS curve is inherently tangent to the ends of the curve, by specifying the two blade angles along with the stagger one and the chord, all the required variables have been established to calculate the position of the middle control point. Denoting as  $\beta_{1m}$ ,  $\beta_{2m}$  the inlet and outlet blade angles, respectively,  $\gamma$  the stagger angle and  $L_{ref}$  the chord, the coordinates of the second and third control points result as:

$$y_2 = y_0 + L_{ref} \tan \gamma \quad (4.1)$$

$$x_2 = x_0 + L_{ref} \quad (4.2)$$

$$y_1 = \frac{(y_2 - L_{ref} \tan \beta_{1m})}{\left(1 - \frac{\tan \beta_{2m}}{\tan \beta_{1m}}\right)} \quad (4.3)$$

$$x_1 = \frac{y_1}{\tan \beta_{1m}} \quad (4.4)$$

After the construction of the mean camber line a 2D cross section is “built” around it, according to a given “thickness distribution”; the sets of the desirable thicknesses and their corresponding positions are provided as fractions of cambers’ length. Consequently, airfoil control points can be computed by means of their normal distance from the camber line, according to the aforementioned “thickness distribution”.

Subsequently, the final 3D surface of the blade is produced according to a custom skinning process. The procedure begins with the distribution of all cross-sections in planes perpendicular to a stacking line; the stacking can be performed with respect to either their leading edges, or their centers of gravity, or custom positions (Chapter 2). Next, interpolations should be performed in both directions e.g. the circumferential and span-wise direction of the blade, to extract a single NURBS surface. The first set of interpolations are implemented on the derived cross-sections, where the required points are computed according to full cosine distribution, resulting in better adaptation of the final surface at the leading and trailing edges. The derived curves coincide with the original cross-sections of the blade, having the same number of discrete points, enabling in that way the interpolation in the span-wise direction. Finally, the corresponding curves are interpolated in the span-wise direction for the construction of the 3D blade polygon that defines the final blade surface. The proposed methodology is robust, it produces one single surface for the definition of the blade, and assures the accurate adaptation in the leading and trailing edge areas.

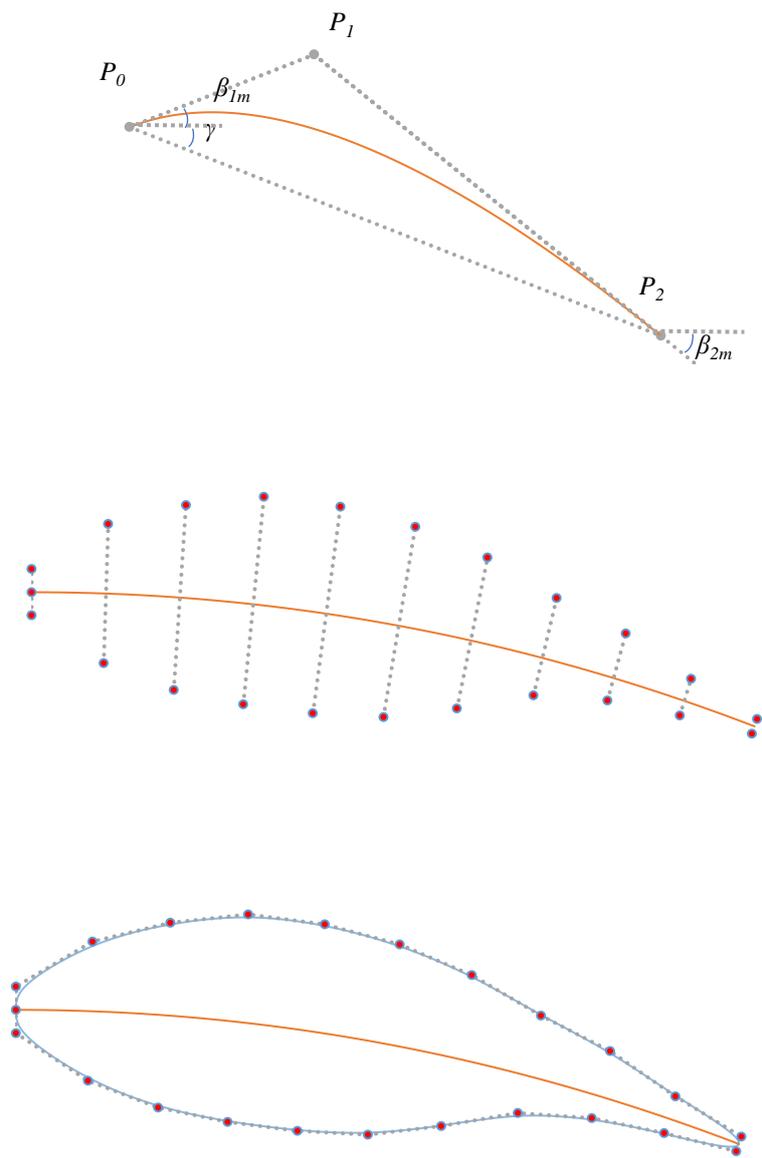


FIGURE 4.1: Airfoil construction procedure.

### 4.3 Fitness function formulation

The optimization problem addressed herein regards the minimization of the deviation between the input target mesh or point cloud and its corresponding nominal "T4T" model. Hence, the goal of the fitness function is to compute for each candidate solution the averaged sum of squared distances between the fitted surface and the original point cloud. Its formulation is given as

$$f(\mathbf{x}) = \frac{\sum_{j=1}^N \|\mathbf{x} - \mathbf{x}_p\|^2}{N} \quad (4.5)$$

where,  $\mathbf{x}$  denotes a vector of the Cartesian coordinates for each point of the given data set,  $N$  is the number of those points and  $\mathbf{x}_p$  is the nearest orthogonal projection of the point on the surface.

However, the computation of the nearest orthogonal projection is not a trivial task, due to the fact that the Cartesian coordinates of a point on a NURBS surface can't be found explicitly. The most commonly applied strategy considers an iterative method to calculate the parametric coordinates of the projected point, using a gradient descent approach. As this can be a very expensive process in terms of computational time, especially in cases involving large subsets of points, the parallel evaluation of the candidate solution is required. In the current work, the OpenCASCADE computational libraries ("OpenCASCADE," 2015) are incorporated to compute the projection points.

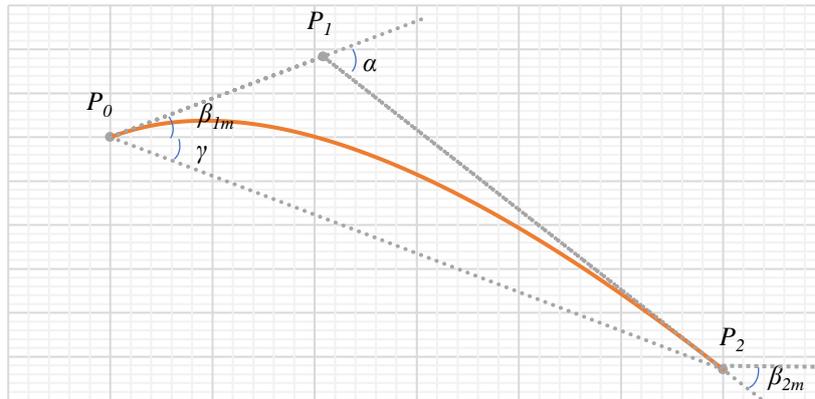
Summarizing, the evaluation stage of the evolutionary procedure is implemented in two steps. Firstly, the candidate solutions (chromosomes) produced by the DE algorithm are imported into the "T4T" software to produce the corresponding blade designs (free-form surface). Then, each individual is evaluated against the target mesh according to equation (4.5). As already stated, the optimization procedure aims to identify the optimal values of the design variables of "T4T" to obtain the better fitted surface. The free-parameters for each cross-section, which can be used as the design variables, are listed in TABLE 4.1.

**TABLE 4.1:** List of free parameters for the definition of each cross section of a blade.

$\beta_{1m}$	Inlet blade angle
$\beta_{2m}$	Outlet blade angle
$\gamma$	Stagger angle
$L_{ref}$	Chord length
$ps\_w_i$	"thickness distribution" for $N_{ps}$ points of the pressure side
$ss\_w_i$	"thickness distribution" for $N_{ss}$ points of the suction side

For all airfoil sections the number of control points distributed along the pressure and suction sides of the mean camber lines are pre-determined, depending on the given data points set; more complicated geometries require more control points to be captured accurately. In order to enhance the efficiency of the computational procedure the outlet blade angle ( $\beta_{2m}$ ) is replaced by angle  $\alpha$  (FIGURE 4.2), in the free parameters that form the chromosome; it provides the means for controlling the "curvature" of the airfoil, allowing for the rational confinement of the search space to regions close to the one of the target point cloud. The computation of the outlet blade angle is explicitly performed internally by "T4T" as:

$$\beta_{2m} = \beta_{1m} + \alpha \quad (4.6)$$



**FIGURE 4.2:** Mean camber line design parameters.

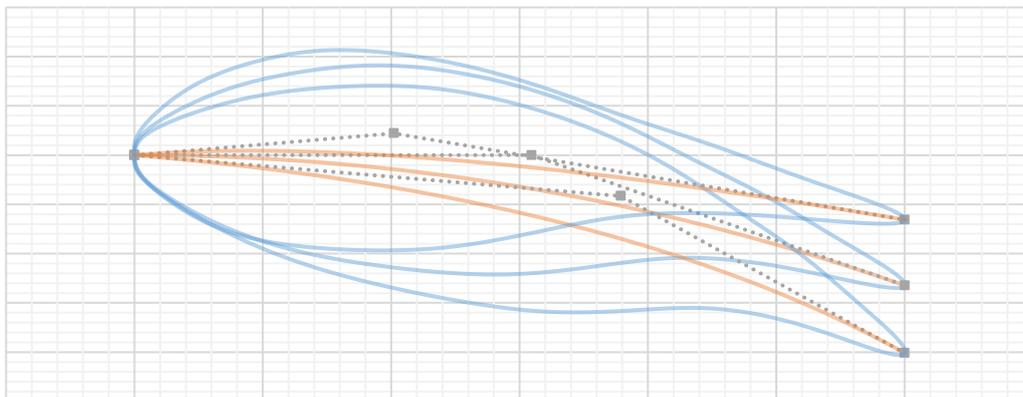
## 4.4 Validation

In this section the results of numerical tests are presented for the proposed methodology. More specifically, the optimization process is utilized to reconstruct the surface of two sample blade meshes; their corresponding designs were produced using "T4T", in order to make feasible the validation of the output parameters. All simulations were performed on a DELL™ R815 PowerEdge™ server with four AMD Opteron™ 6380 sixteen-core processors at 2.50 GHz (64 cores in total).

### 4.4.1 The 1<sup>st</sup> test case

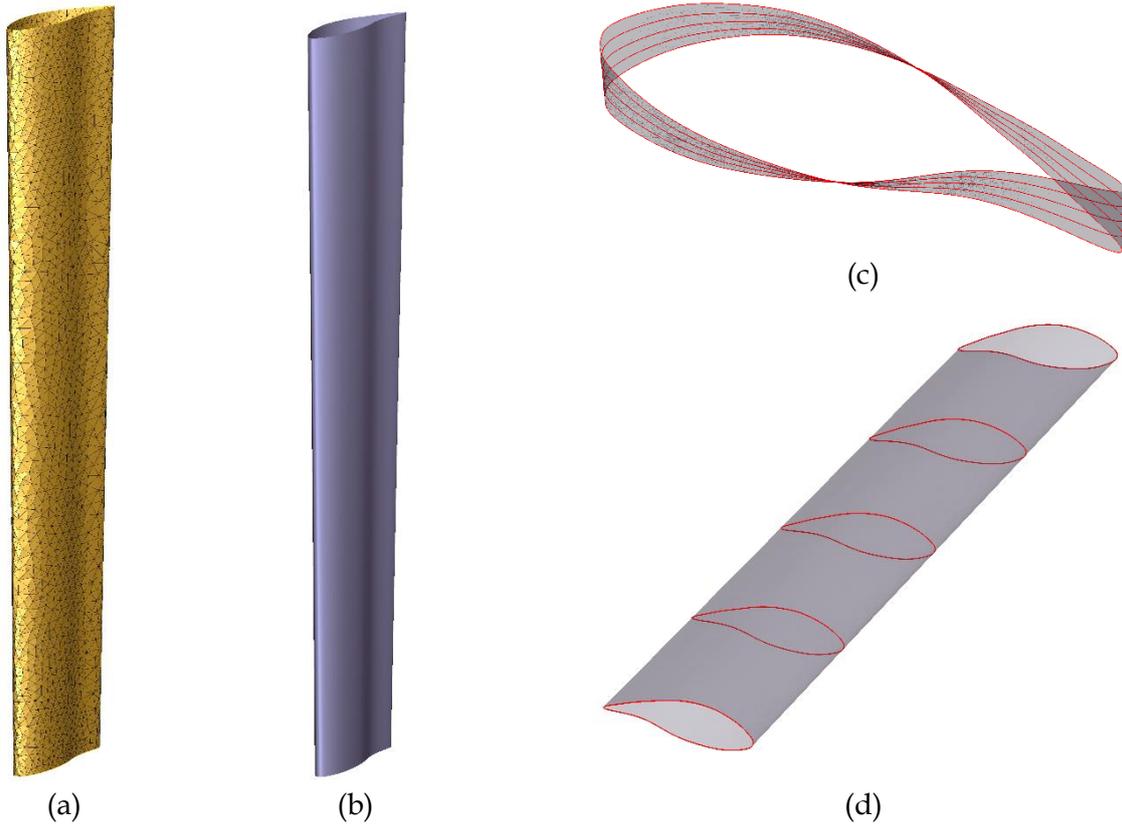
The first benchmark test case concerns the surface reconstruction of a 3D blade with known the chord of all sections and the "thickness distributions". All cross-section chords are set to 3 m, while the length of the blade is 20 m. The blade is constructed utilizing 5 cross sections and, consequently, the number of the design variables is 15 (3 angles for each section). The degrees of the surface take the values 3 and 2 for the chordwise and spanwise directions respectively, while the stacking of the sections has been performed with respect to their centres of gravity.

Regarding the parameters of the DE, the population size was set equal to 50, the values of  $F$  and  $Cr$  constants are 0.6 and 0.45 respectively, while the algorithm was executed for 1200 generations. The permitted range for each angle was set equal to  $10^\circ$  with respect to their estimated values. This range is illustrated in FIGURE 4.3 for the extreme cases of the upper and lower bounds of the angles for each cross-section (the medium blade being the target one); all intermediate positions can be produced by the DE during the evolution procedure of the population.



**FIGURE 4.3:** Illustration of the upper and lower permitted bounds for the blade angles, for the 1st test case.

FIGURE 4.4 illustrates the density of the utilized target surface mesh (its vertices being the target set of data points to be approximated), along with the final blade surface of the optimization procedure. Runs were performed with and without the use of surrogate assistance, to evaluate and compare their numerical results. The convergence history of the fitness function for the best and worst solution of each generation is depicted in FIGURE 4.5 for the two methodologies. As expected, the optimization with the use of surrogates converges faster (in a smaller number of exact evaluations), providing also a better-fitted final solution.



**FIGURE 4.4:** (a) Target surface mesh; (b)-(d) Reconstructed surface.

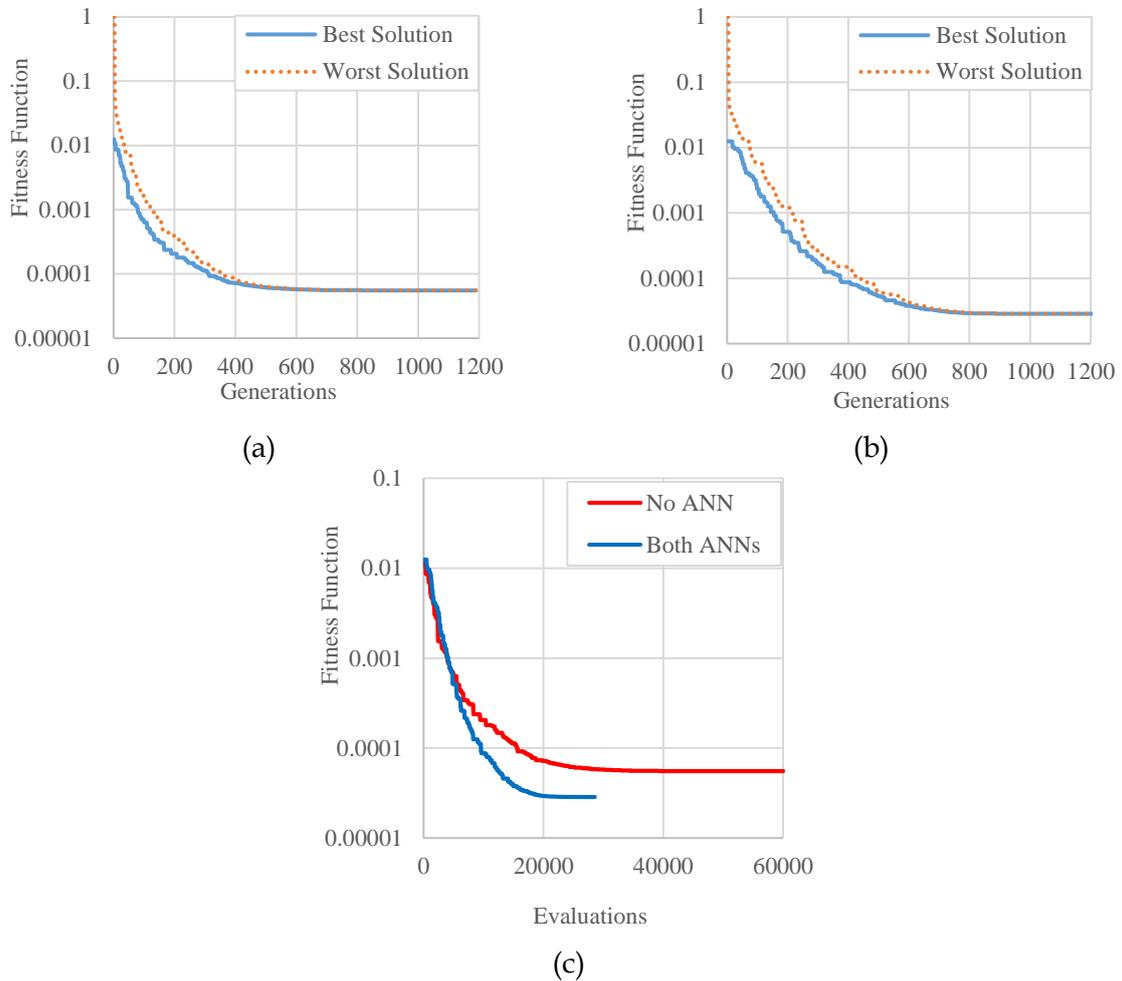
Furthermore, the computation time required by the optimization procedure along with the average and max errors of the final surface are included in TABLE 4.2; this error corresponds to the mean absolute Euclidian distance between the points of the mesh and the final surface. FIGURE 4.6 illustrates the corresponding distributions of errors on the blade surface for both encountered runs (with and without surrogate assistance).

**TABLE 4.2:** Wall-clock computation times and surface errors of the proposed surface reconstruction procedure, with and without surrogate assistance.

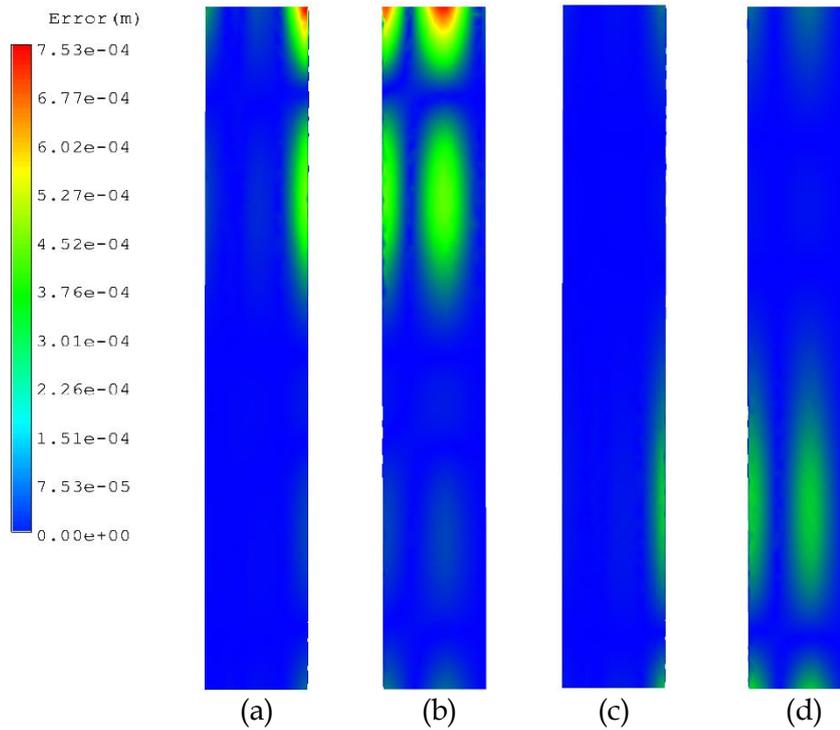
	<i>Total time (hours)</i>	<i>Average Error (m)</i>	<i>Maximum Error (m)</i>
<i>No ANN</i>	50	$5.53 \times 10^{-5}$	$7.53 \times 10^{-4}$
<i>Both ANNs</i>	18.5	$2.85 \times 10^{-5}$	$2.13 \times 10^{-4}$

The derived results reveal that significantly reduced wall-clock computation time was achieved by the use of surrogate models, due to the efficient rejection of the less promising solutions; thereby, the computationally intensive exact evaluation of their fitness value is avoided. It should be mentioned here that the computational time needed for the evaluation of each candidate solution varied a lot, since the algorithm utilized for the computation of the orthogonal projections employs an iterative process, which relies on the initial estimation of the parametric coordinates of the point. A bad estimation of the initial value leads to increased computation times in order to achieve an adequate solution. Therefore, by reducing the number of exact evaluations, the possibility of experiencing long time evaluations is decreased. Furthermore, the total computation time indicates the necessity for the parallel

implementation of the evaluations in order to obtain solutions in acceptable time periods. Regarding the accuracy of the procedure, the surface errors, which correspond to the deviation of the surface from the given data points, remain at very low levels.



**FIGURE 4.5:** The convergence history for the best and worst value of the fitness function: (a) no ANN; (b) both ANNs. (c) Comparison of the two approaches (best fitness with respect to exact evaluations).

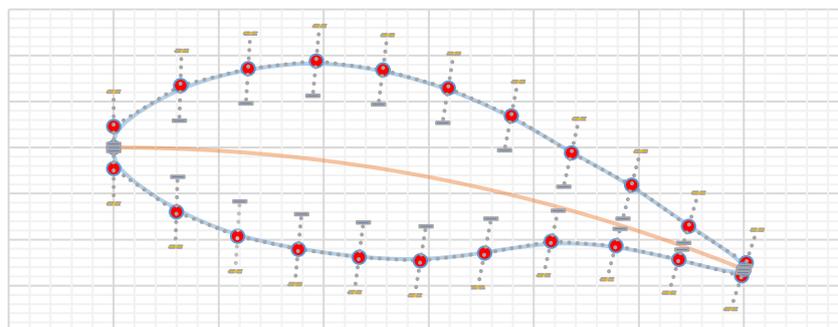


**FIGURE 4.6:** Distribution of the surface error: (a)-(b) front and back views for the case without ANN; (c)-(d) front and back views for the surrogate-assisted case.

#### 4.4.2 The 2<sup>nd</sup> test case

*The second test case presents the scenario of reconstructing the surface of the blade with all the available design parameters of "T4T" to be free to move. The 20 m testing blade comprises of three cross-sections, while, for each of them 11 stations were selected per side of the mean camber line to apply "thicknesses" (distances of the corresponding control points from the mean camber line), resulting to a total number of 78 design variables. The same parameters for the DE run, as in the 1<sup>st</sup> test case, were used. The permitted varied ranges of thicknesses as percentage of cambers' length are illustrated in FIGURE 4.7 and summarized in*

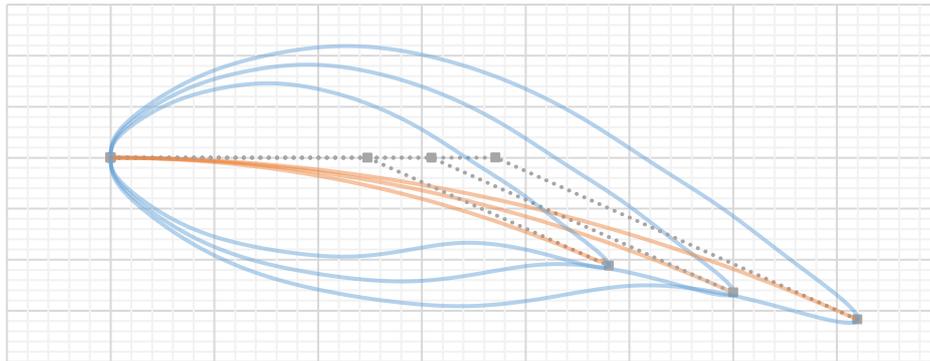
TABLE 4.3. The range of values for the chords, corresponding to a  $\pm 20\%$  fluctuation of the real chord, is depicted in FIGURE 4.8.



**FIGURE 4.7:** Upper and lower bounds for the control points of each section.

**TABLE 4.3:** Permitted range of “thickness distribution” for both sides of the mean camber line.

<i>Pressure Side</i>			<i>Suction Side</i>		
<i>Position</i>	<i>Lower Bound</i>	<i>Upper Bound</i>	<i>Position</i>	<i>Lower Bound</i>	<i>Upper Bound</i>
<b>0.0</b>	0.005	0.080	0.0	0.005	0.080
<b>0.1</b>	0.040	0.140	0.1	0.040	0.140
<b>0.2</b>	0.070	0.170	0.2	0.070	0.170
<b>0.3</b>	0.080	0.180	0.3	0.090	0.190
<b>0.4</b>	0.080	0.180	0.4	0.090	0.190
<b>0.5</b>	0.070	0.170	0.5	0.080	0.180
<b>0.6</b>	0.040	0.140	0.6	0.060	0.160
<b>0.7</b>	0.005	0.100	0.7	0.030	0.130
<b>0.8</b>	0.005	0.080	0.8	0.010	0.110
<b>0.9</b>	0.005	0.070	0.9	0.005	0.080
<b>1.0</b>	0.005	0.060	1.0	0.005	0.060



**FIGURE 4.8:** Upper and lower permitted bounds for the chord of each section (the middle one is the target blade).

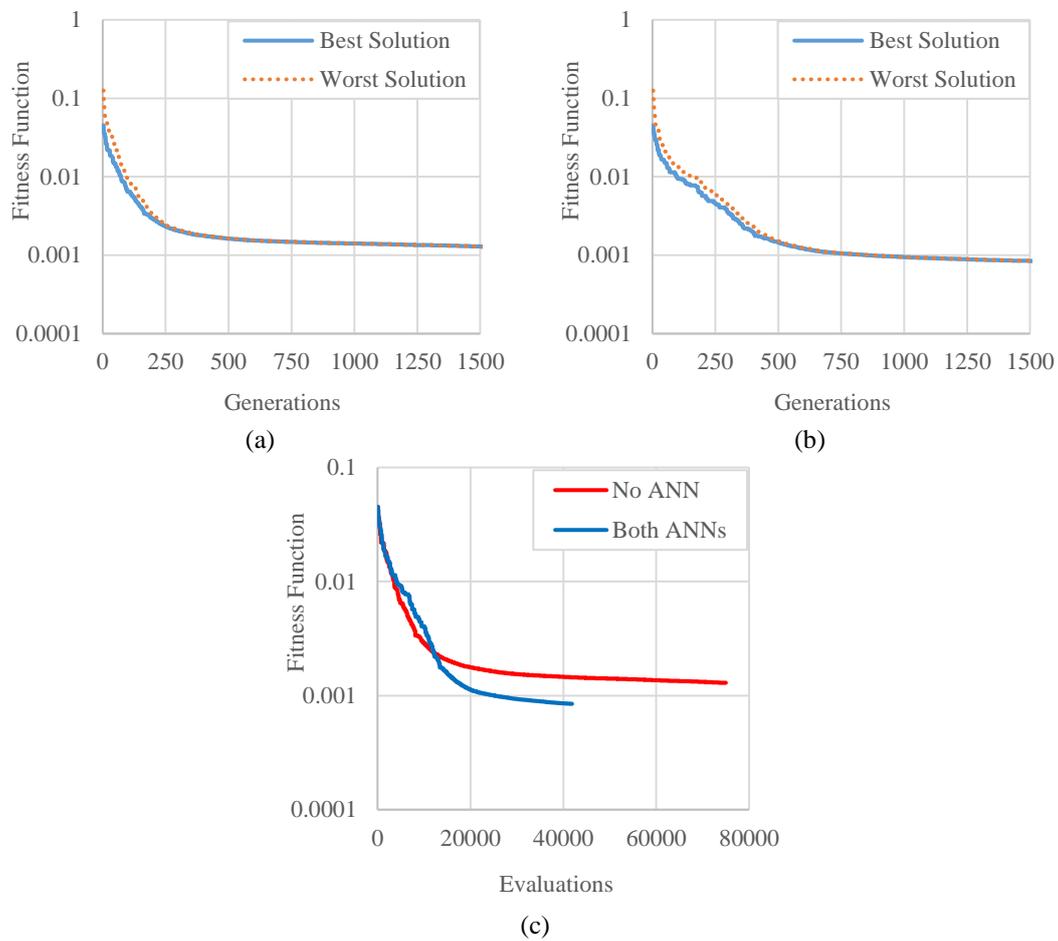
Similarly to the previous case, runs with and without the use of surrogate models were performed. In FIGURE 4.9 the corresponding convergence histories are showed for the first 1500 generations, regarding the best and worst solution for each one of them, as well as a comparison of the runs with respect to the number of exact evaluations.

*Wall-clock computation times and surface errors of the proposed surface reconstruction procedure, with and without surrogate assistance.*

**TABLE 4.4:** Wall-clock computation times and surface errors of the proposed surface reconstruction procedure, with and without surrogate assistance.

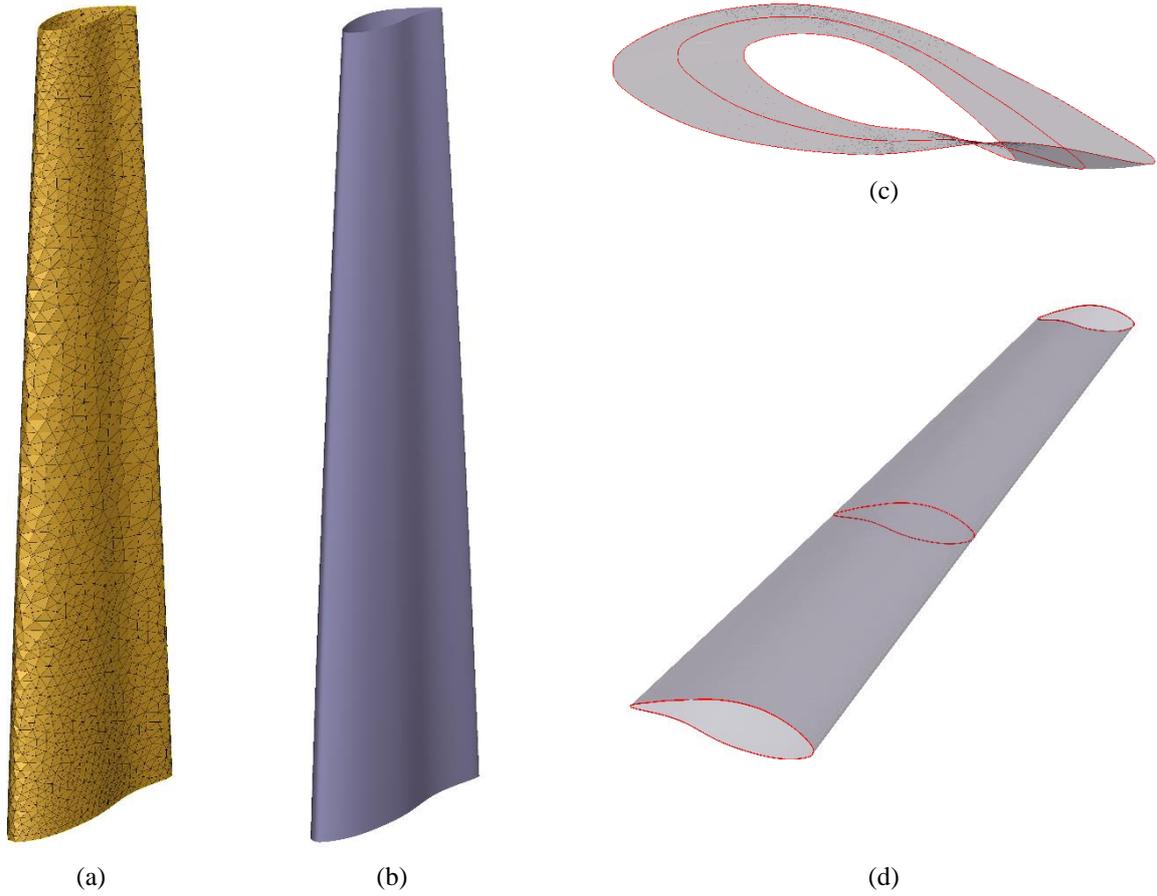
	<i>Total time (hours)</i>	<i>Average Error (m)</i>	<i>Maximum Error (m)</i>
<b>No ANN</b>	48.5	$1.28 \times 10^{-3}$	$9.48 \times 10^{-3}$
<b>Both ANNs</b>	19.5	$8.11 \times 10^{-4}$	$6.28 \times 10^{-3}$

The final blade surface along with the surface mesh of the target set of data points is illustrated in FIGURE 4.10, while **Error! Reference source not found.** and FIGURE 4.11 present the derived wall clock computation times as well as the blade surface errors.

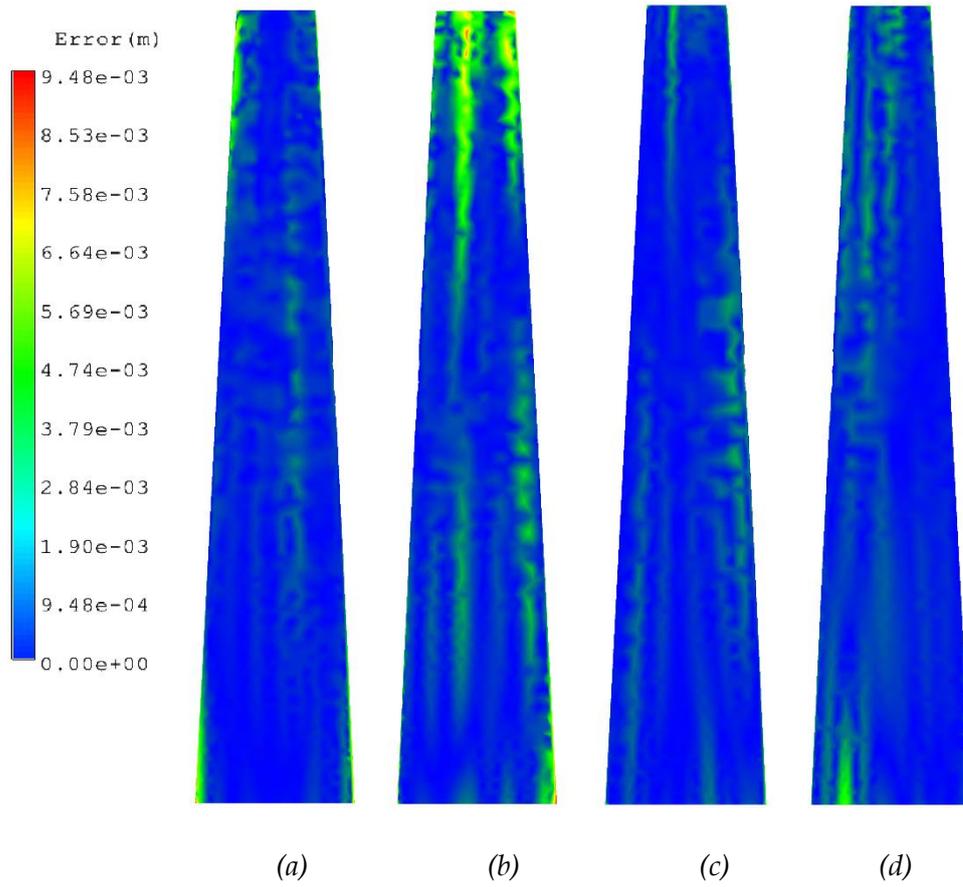


**FIGURE 4.9:** The convergence history for the best and worst value of the fitness function: (a) no ANN, (b) both ANNs. (c) Comparison of the two approaches (best fitness with respect to exact evaluations).

Comparing the results it is evident once more that the use of surrogate assistance is of great importance for the acceleration of the procedure, as well as for the enhancement of the exploitation capability of the DE algorithm.



**FIGURE 4.10:** (a) Target surface mesh; (b)-(d) Reconstructed surface.



**FIGURE 4.11:** Distribution of the surface error: (a)-(b) front and back views for the case without ANN; (c)-(d) front and back views for both ANNs.

## 4.5 Conclusions

In this chapter, the development of a specialized surface reconstruction methodology was reported, developed for the geometry definition of a wind turbine blade as a single NURBS surface, from a target set of data points provided in the form of a surface triangular mesh. For the parameterization of the blade surface the dedicated blade geometry modelling software "T4T" was used, which has been developed for the parametric design of turbomachinery components and wind turbine blades (Chapter 2).

The shape reconstruction of the blade surface was formulated as an optimization procedure, which was realized with a parallel, synchronous, metamodel-assisted differential evolution algorithm. The blade design variables, used in "T4T", are automatically modified in the optimization procedure to produce different candidate blade designs, within prescribed limits. The cost function to be minimized is the sum of the squared distance between each given data point and the resulting NURBS surface of the candidate blade. Two test cases were used to assess the proposed methodology, with a different value of complexity. For both cases, the metamodel-assisted version of the DE algorithm proved to be much faster, providing a better optimal solution than the standard algorithm, without surrogate models. The proposed methodology was found to provide reconstructed surfaces with a small error with respect to the given set of surface data points, while the resulting parameterization has a physical meaning. The major drawback of the proposed methodology is the large computational time, associated with the evaluation of each candidate solution. The computation of the normal distances of the given data points from the free-form surface is a demanding calculation, and different alternatives should be examined.

# Chapter 5: Mesh Deformation

## 5.1 Introduction

Computational mesh deformation is a technique used in both design optimization methodologies and Fluid Structure Interaction (FSI) simulations, especially in wind turbine applications. RBF-based mesh deformation methods have been recognized during the past years as an essential tool for numerical simulations involving mesh deformation. Nevertheless, they call for excessive memory and computation time requirements, especially for large-scale problems. A remedy to this shortcoming appears to be the selection of a reduced number of surface mesh nodes, to be used as RBF-centers, resulting in decreased dimensions of the system of equations. In this chapter the development of an algorithm for the reduced surface point selection via agglomeration of the corresponding boundary control areas, and consequently the acceleration of an RBF-mesh deformation technique, is reported. The proposed methodology is based on an existing multigrid scheme (Lygidakis & Nikolos, 2014b, 2014a; Lygidakis *et al.*, 2016), considering the fusion of adjacent control cells on a topology-preserving framework; it resembles the advancing front technique, as it begins from regions with surface discontinuities extending successively to the internal domain (Ito, 2013). Despite the wide implementation of the agglomeration multigrid methodology in CFD, the proposed fusion strategy has not been combined to mesh deformation algorithms. The results against complex test cases, included in this thesis, confirm its capability for significant acceleration of such algorithms, without downgrading the quality of the derived grids.

The outline of this chapter is as follows: In Section 5.2 the formulation of the RBF-based interpolation is described in detail, along with the proposed agglomeration strategy, followed by the validation results of the proposed methodology against representative test cases. Finally, Section 5.3 contains some final remarks and conclusions.

## 5.2 Application of Radial Basis Functions methodology to mesh deformation

### 5.2.1 Radial Basis Functions (RBFs) interpolation

In general RBFs provide the means for a smooth interpolation of a multivariate function  $s(\mathbf{x})$ . In this dissertation, the RBF-formulation is used in the context of mesh deformation, with the boundary nodes displacement field representing the function to be interpolated. It is described as a weighted sum of the evaluations of the basis functions as (de Boer, van der Schoot, *et al.*, 2007)

$$s(\mathbf{x}) = \sum_{i=1}^{n_c} a_i \Phi(\|\mathbf{x} - \mathbf{x}_{c_i}\|) + p(\mathbf{x}), \quad (5.1)$$

where  $a_i$  denote the scalar coefficients (weights), while  $p(\mathbf{x})$  is a polynomial term;  $\mathbf{x}_{c_i} = [x_{c_i}, y_{c_i}, z_{c_i}]$  are the centers of RBFs with known values of displacements,  $n_c$  is the number of those centers and  $\Phi$  is the basis function, which depends on the Euclidean distance of a point from each center:

$$\|\mathbf{x} - \mathbf{x}_{c_i}\| = \sqrt{(x - x_{c_i})^2 + (y - y_{c_i})^2 + (z - z_{c_i})^2}. \quad (5.2)$$

In a mesh deformation algorithm the RBF-centers correspond to the grid points of the deformed surface, while the polynomial term is omitted in order to avoid its undesirable global coverage, which can cause the whole domain to move rather than deform (Rendall & Allen, 2009b). Thus, equation (5.1) is modified as follows:

$$s(\mathbf{x}) = \sum_{i=1}^{n_c} a_i \Phi(\|\mathbf{x} - \mathbf{x}_{c_i}\|). \quad (5.3)$$

In order the previous formulation to have a unique solution,  $\Phi$  must be positively defined (Iske, 2003). The coefficients  $a_i$  are calculated fulfilling equation's requirements at the centers of basis functions

$$\mathbf{s}(\mathbf{x}_{c_i}) = \mathbf{U}_{c_i} = [u_{c_i}^x, u_{c_i}^y, u_{c_i}^z]^T, \quad (5.4)$$

where  $\mathbf{U}_{c_i}$  is the vector of the known displacement (in three dimensions) of each surface node  $c_i$ . In order to take into account the vector displacement, equation (5.4) has to be employed to each direction, deriving the following three linear systems

$$\mathbf{M}_{cc} \mathbf{A}_x = \mathbf{U}_x, \quad (5.5)$$

$$\mathbf{M}_{cc} \mathbf{A}_y = \mathbf{U}_y, \quad (5.6)$$

$$\mathbf{M}_{cc} \mathbf{A}_z = \mathbf{U}_z, \quad (5.7)$$

where

$$\mathbf{A}_x = [a_{c_1}^x, a_{c_2}^x, \dots, a_{n_c}^x]^T, \mathbf{A}_y = [a_{c_1}^y, a_{c_2}^y, \dots, a_{n_c}^y]^T, \mathbf{A}_z = [a_{c_1}^z, a_{c_2}^z, \dots, a_{n_c}^z]^T, \quad (5.8)$$

$$\mathbf{U}_x = [u_{c_1}^x, u_{c_2}^x, \dots, u_{n_c}^x]^T, \mathbf{U}_y = [u_{c_1}^y, u_{c_2}^y, \dots, u_{n_c}^y]^T, \mathbf{U}_z = [u_{c_1}^z, u_{c_2}^z, \dots, u_{n_c}^z]^T, \quad (5.9)$$

while  $\mathbf{M}_{cc}$  is a  $n_c \times n_c$  matrix, not depending on direction, defined as follows

$$\mathbf{M}_{cc} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \dots & \Phi_{1n_c} \\ \Phi_{21} & \Phi_{22} & \dots & \Phi_{2n_c} \\ \vdots & \vdots & \vdots & \vdots \\ \Phi_{n_c 1} & \Phi_{n_c 2} & \dots & \Phi_{n_c n_c} \end{bmatrix} \quad (5.10)$$

with  $\Phi_{ij} = \Phi(\|x_i - x_j\|/R)$ ,  $R$  being the support radius defined in the following paragraph,  $\mathbf{A}_x, \mathbf{A}_y, \mathbf{A}_z$  the vectors of the scalar coefficients in each spatial direction, and  $\mathbf{U}_x, \mathbf{U}_y, \mathbf{U}_z$  the displacement vectors of the surface nodes in  $x, y$ , and  $z$  directions respectively. The coefficients  $a_{c_i}^{x,y,z}$  of the linear systems are obtained by employing direct or iterative solvers.

Subsequently, the required displacements of the volume mesh nodes are calculated according to those of the boundary points, by applying the following equations

$$[u_1^x, u_2^x, \dots, u_{n_v}^x]^T = \mathbf{M}_{cv} \mathbf{A}_x, \quad (5.11)$$

$$[u_1^y, u_2^y, \dots, u_{n_v}^y]^T = \mathbf{M}_{cv} \mathbf{A}_y, \quad (5.12)$$

$$[u_1^z, u_2^z, \dots, u_{n_v}^z]^T = \mathbf{M}_{cv} \mathbf{A}_z, \quad (5.13)$$

where  $n_v$  denotes the number of the volume mesh nodes and  $\mathbf{M}_{cv}$  is a  $n_v \times n_c$  matrix, described as:

$$\mathbf{M}_{cv} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \cdots & \Phi_{1n_c} \\ \Phi_{21} & \Phi_{22} & \cdots & \Phi_{2n_c} \\ \vdots & \vdots & \vdots & \vdots \\ \Phi_{n_v1} & \Phi_{n_v2} & \cdots & \Phi_{n_vn_c} \end{bmatrix}. \quad (5.14)$$

Several approaches have been reported in the literature about the type of the aforementioned basis function  $\Phi$ ; they can be divided in two main categories: functions with compact and functions with global support. For the first ones the variable to be interpolated is scaled with a support radius  $R$  ( $\xi = x/R$ ), which represents the region of influence of the function from each center. In that way, only points that lie inside that radius (around a center  $x_c$ ) are affected by its displacement. Mathematically it is expressed as:

$$\Phi(\xi) = \begin{cases} \Phi(\xi) & 0 \leq \xi \leq 1 \\ 0 & \xi > 1 \end{cases}. \quad (5.15)$$

An increase of the support radius results in a more smooth and accurate deformation, in expense of efficiency, due to the fact that a large support radius leads to dense matrix systems, prone to ill-conditioning. Thus, an acceptable trade-off between accuracy and efficiency has to be fulfilled, as a large support radius may produce good approximations but a small one results in sparse systems more stable and more efficiently solved. A remedy to this compromising shortcoming is the use of a local support radius, different at each center; nevertheless, the solvability of the system is guaranteed only by a global radius (Beckert & Wendland, 2001). A detailed examination of the influence of the support radius  $R$  on accuracy and condition number of RBFs' methods has been presented in the study of (Lombardi *et al.*, 2013). Unlike the techniques of the previous category, functions with global support influence the whole domain. Therefore, the far field boundary points have to be also included in the interpolant, to prevent their displacements. An extended comparison of various basis functions, focusing on the quality of the deformed mesh and the efficiency of the method, has been reported in (de Boer, van der Schoot, *et al.*, 2007), as well as in (Rendall & Allen, 2008). Based on the results of such analyses the Wendland's  $C2$  function was adopted in the proposed methodology due to its robustness and accuracy; it is defined as

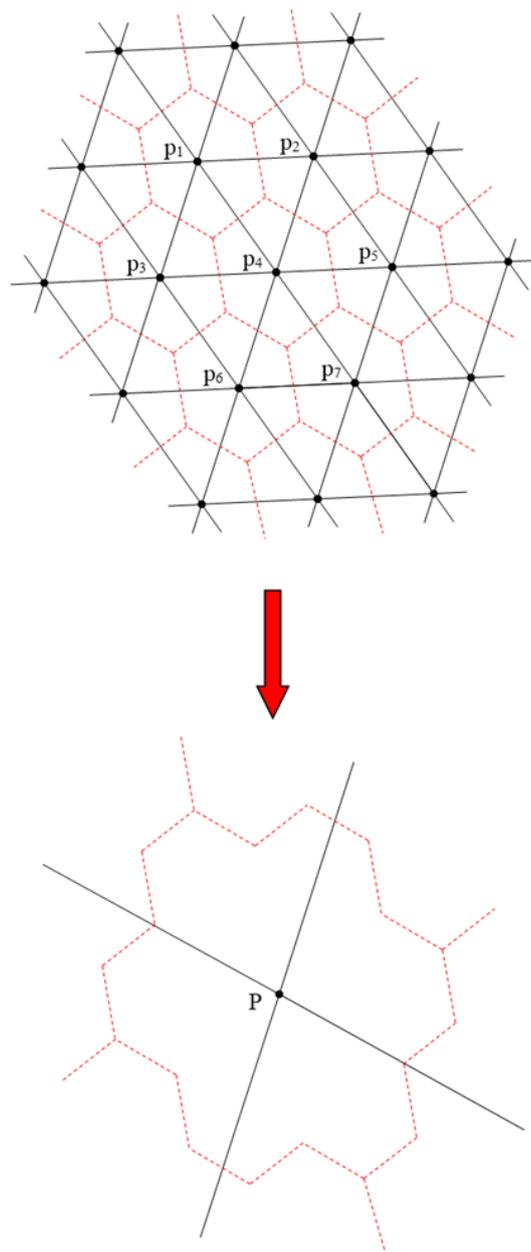
$$\Phi(\xi) = (1 - \xi)^4 (4\xi + 1) \quad (5.16)$$

while with the appropriate selection of support radius the possibility of generating an ill-conditioned matrix is minimized, even for large-scale problems.

Summarizing, the deformation process is performed in two discrete stages, namely the solution and the evaluation one. The first one is accomplished by calculating  $\mathbf{M}_{cc}^{-1}$  only once prior to deformation, since this matrix can be reused without requiring to update the coefficient vectors; it remains constant for any displacement field. Nevertheless, it is this stage that suffers from excessive computational and memory requirements, especially when large-scale problems are encountered, and calls for a method deriving a reduced set of surface nodes (Rendall & Allen, 2009b), as the one analyzed in the next Section. Once this stage is complete the evaluation stage is performed, which derives the deformation of each internal mesh node by employing equations (5.11)-(5.13).

## 5.2.2 The agglomeration strategy

As mentioned in the Introduction, an agglomeration procedure is iteratively implemented in order a reduced set of surface nodes to be obtained and consequently the computational performance of the proposed RBF-based mesh deformation algorithm to be improved, without compromising the quality and accuracy of the deformation procedure. A node-centered formulation is used for the construction of the control areas around each node. For the boundary nodes, used in this study, the control area of each node is constructed by connecting the barycenters of the surrounding triangles with the mid-points of triangle edges (FIGURE 5.1).



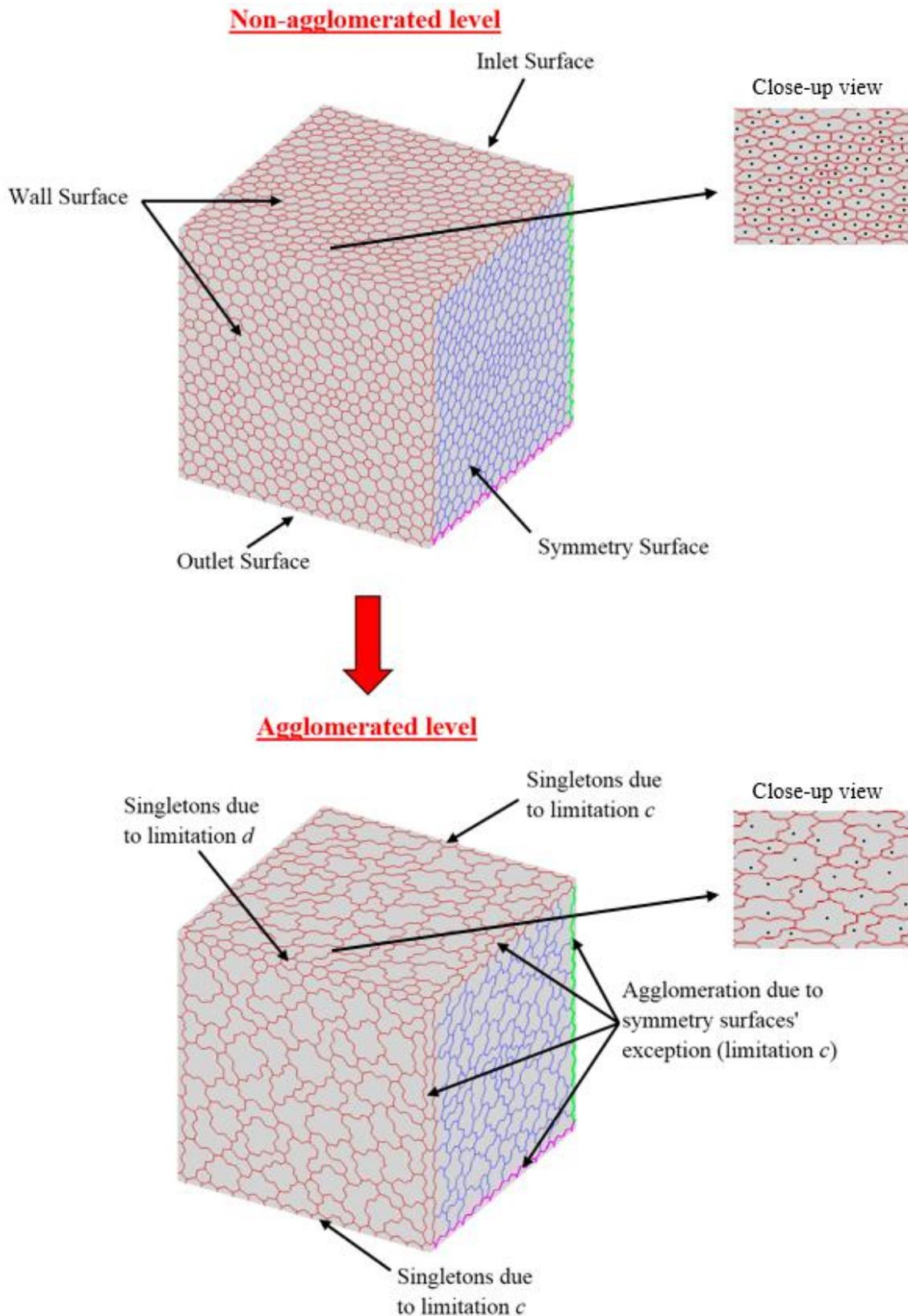
**FIGURE 5.1:** Generation of a supernode  $P$  via agglomeration.

The fusion of those control areas is performed on a topology-preserving framework, resembling the advancing front technique (Ito, 2013). It is limited by pre-defined rules, which

concern mainly nodes with different boundary-condition-type closures or more than one boundary slope discontinuities (Nishikawa & Diskin, 2011; Lygidakis & Nikolos, 2014a), preserving in that way consistency between the initial finest grid and the coarser generated ones. More specifically, the following constraints are employed:

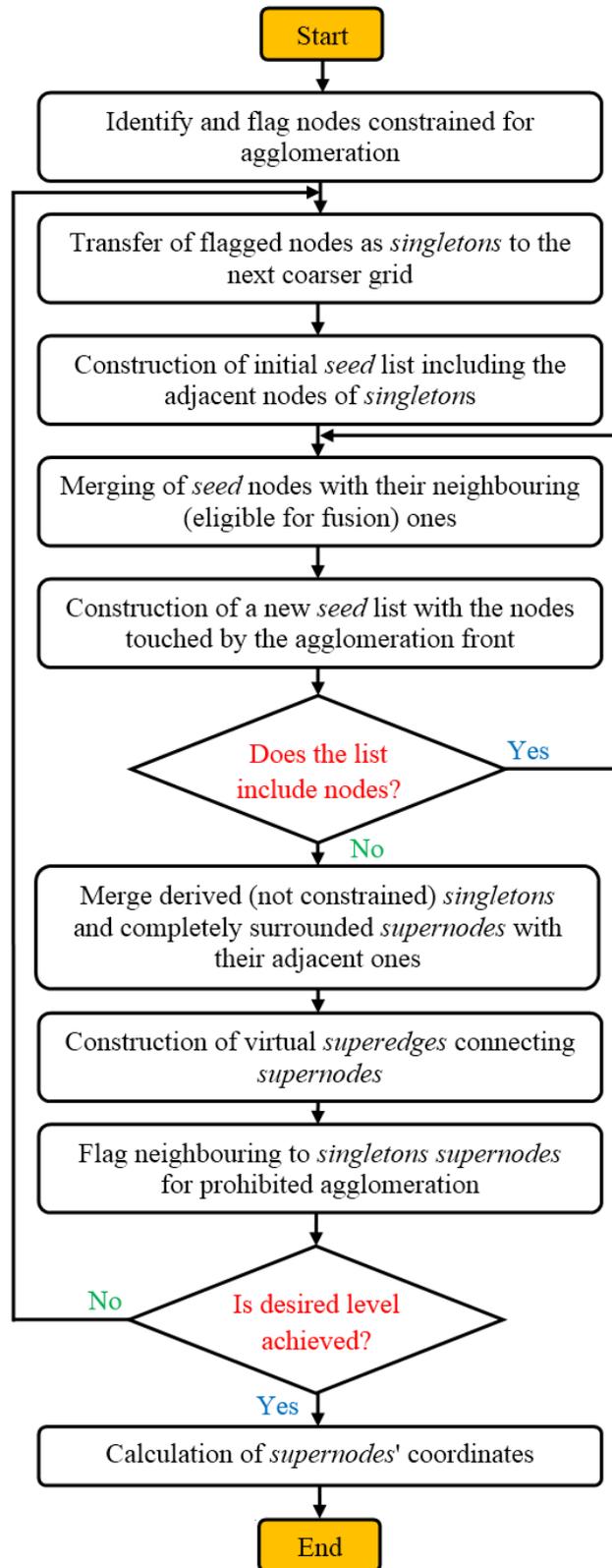
- a) The control area of a node can be merged only with its adjacent ones belonging to the same surface, e.g., a node positioned at a solid wall boundary can be associated with another one belonging to the same surface but not to an inlet, outlet or symmetry one (Lygidakis & Nikolos, 2014a).
- b) Two neighbouring nodes can be fused to construct a new virtual *supernode*, only if their control areas' unit normal vectors form an angle less than  $10^\circ$ . An adaptation of this limit may be applied, depending on the encountered test case, in order to allow for further reduction of the surface points.
- c) A node belonging to two or more boundary-condition-type closures is not agglomerated and remains as *singleton* at the next coarser level (Nishikawa & Diskin, 2011; Lygidakis & Nikolos, 2014a; Lygidakis *et al.*, 2016). For example, the control area of a node imposed simultaneously to inlet and solid wall boundary surfaces cannot be merged with its adjacent ones. However, the nodes belonging to two boundary-condition-type closures, with the one being a symmetry surface, are excluded from this limitation; their fusion with nodes belonging to the same two boundary-condition types is allowed (Lygidakis & Nikolos, 2014a).
- d) Similarly to the previous limitation, a node with one or more boundary slope discontinuities cannot be agglomerated and it will be transferred as a *singleton* at the next coarser grid (Nishikawa & Diskin, 2011; Lygidakis & Nikolos, 2014a; Lygidakis *et al.*, 2016), e.g., a node placed on an edge formed by the intersection of two adjacent surfaces. In this algorithm the limit, set to restrict the fusion procedure, is equal to  $30^\circ$ , but it may be adapted, depending on the encountered test case.

An example of the aforementioned constraints is depicted in *FIGURE 5.2*, including the surface control areas of the initial and agglomerated grids representing a cubic domain. Wall, inlet, outlet and symmetry surfaces have been defined with red, green, magenta, and blue coloured control areas respectively; the constraints (c) and (d) have been also applied. More specifically, *singletons* have been produced, due to coupled wall-inlet and wall-outlet boundary conditions (limitation (c)), as well as due to wall boundary slope discontinuities (limitation (d)). In addition, the exclusion of constraint (c) in case of two boundary-condition-type closures, with the one being a symmetry surface, can be also identified. Finally, close-up views of a part of the initial and agglomerated upper surfaces are included, illustrating the corresponding control areas as well as their centers.



**FIGURE 5.2:** Control areas on the surface of the initial and agglomerated grids, representing a cubic domain.

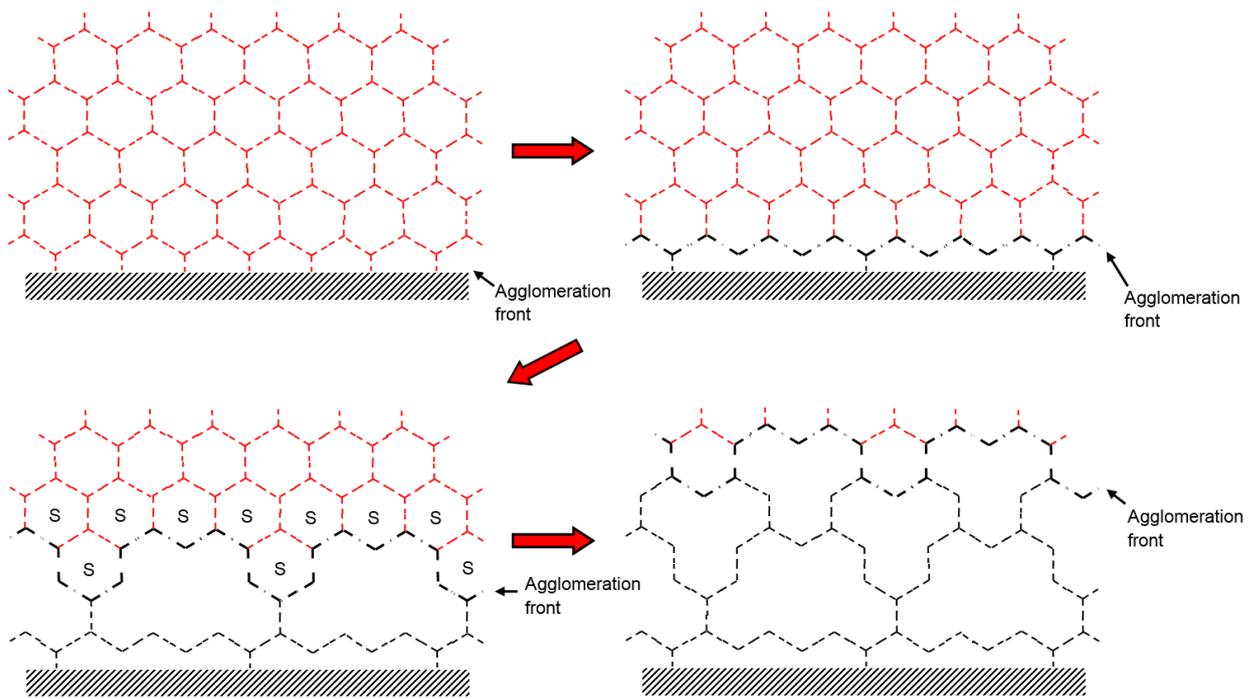
Considering the aforementioned limitations, the surface agglomeration procedure is performed in the following sequence, described also in the flowchart of *FIGURE 5.3*:



*FIGURE 5.3: Flowchart of the proposed agglomeration procedure.*

- 1) The computational nodes of the initial finest grid, which cannot be merged due to the previous limitations (c) and (d), are identified and marked.
- 2) The procedure begins at each level with the simple transfer of marked (confined for fusion) nodes as *singletons* to the next coarser grid. Similarly to step 1, these *singleton supernodes* are marked for prohibited agglomeration in case an even coarser resolution is desired (Lygidakis & Nikolos, 2014a; Lygidakis *et al.*, 2016).
- 3) The initial list of the so-called *seed* nodes is constructed (nodes under examination for possible agglomeration), including actually the neighbouring nodes of the pre-described *singleton* ones (i.e., of those simply transferred to the next level), in order the main fusion process to begin. Thus, the starting point of the proposed procedure regards regions with surface discontinuities, such as the trailing edge and the tip of a wing. It is a common tactic in agglomeration multigrid methodologies to start the procedure from internal boundaries, i.e., from probably problematic nodes due to the limited number of neighbouring points eligible for merging. In that way the produced control volumes or control areas are getting larger gradually, as agglomeration extends from the internal to the external boundary surfaces. As a result, more smooth polyhedral grids are produced. In case that a reverse procedure was selected, the quality of the produced mesh would be jeopardized.
- 4) At this step the main agglomeration procedure begins by looping over the nodes of the aforementioned list and examining whether their adjacent, non-merged yet, are eligible for fusion due to the imposed constraints. If no limitation is identified, their control areas are merged with this of a *seed* node, deriving the corresponding control area of a *supernode*. In FIGURE 5.1 the schematic representation of the generation of a *supernode P* via the agglomeration of the control areas of seven nodes  $p_i$  ( $i = 1, \dots, 7$ ) is presented. In case no neighbouring node is appropriate for fusion, the adjacent *supernodes* of the *seed* node are examined in order to include it; if more than one *supernodes* are identified as eligible for this merging, the one with the fewer number of agglomerated nodes will be selected. If no agglomeration is achieved again, the examined *seed* node remains as *singleton* to the next coarser level. The procedure is assumed to be accomplished when all *seed* nodes are agglomerated or transferred as *singletons* (Lygidakis & Nikolos, 2014a).
- 5) A new list of *seed* nodes is constructed, including the non-agglomerated nodes, whose control areas have been touched by the agglomeration front (FIGURE 5.4). They are actually the neighbouring nodes of the already fused ones, confirming in that way the similarity to advancing front technique of this method (Lygidakis & Nikolos, 2014a; Lygidakis *et al.*, 2016). In addition, a priority hierarchy is imposed for this list, based on the number of times a node is touched by the agglomeration front, as well as the number of its adjacent nodes. According to this hierarchy the set of nodes having the maximum touch number will be initially examined for agglomeration, while for the same set the procedure begins from the node with the minimum number of neighbouring ones (Lygidakis & Nikolos, 2014a).
- 6) Steps 4 and 5 are repeated until all the surface nodes are agglomerated or assigned as *singletons* to the next level (Lygidakis & Nikolos, 2014a). In FIGURE 5.4 the progress of the agglomeration front as well as the merging procedure are presented schematically, beginning from a two-boundary-condition-type closure (solid wall and symmetry boundary conditions) and continuing to the internal area of the domain.

- 7) *Singleton supernodes* are examined again for possible fusion with their adjacent ones, according to the procedure described in step 4. Additionally, the *supernodes* completely surrounded by other ones are identified (each one is connected with only one *supernode*), in order to be merged with the surrounding *supernodes* (Lygidakis & Nikolos, 2014a).
- 8) The new virtual *superedges*, connecting adjacent *supernodes* obtained by the previous steps, are constructed. Although this data structure is a prerequisite for multigrid methodology, as it is utilized in edge-wise evaluation of numerical fluxes (Lygidakis & Nikolos, 2014a; Lygidakis *et al.*, 2016), in this algorithm it is employed only to enhance the definition procedure of nodes' connectivity.
- 9) At the final step of the main agglomeration procedure the *supernodes*, which are neighbours to *singletons*, are marked to become also *singletons* at the next coarser mesh, avoiding in that way the generation of adjacent control areas with large differences in size, and consequently surface grid harsh irregularities (Lygidakis & Nikolos, 2014a; Lygidakis *et al.*, 2016).
- 10) Steps 2 to 9 are repeated in case an even coarser resolution is desired. Depending on the test cases encountered in this study, a number of five agglomeration levels has been identified by the authors to be adequate.



**FIGURE 5.4:** Progress of the agglomeration front and fusion procedure. In the bottom-left illustration the seed nodes for the next agglomeration step are marked with an "S".

Once the agglomeration procedure is complete, the pseudo-coordinates of the *supernodes* at each coarser resolution have to be computed. Although for such calculations the common practice in multigrid methodology is to employ volume- or area-weighting (Nishikawa & Diskin, 2011; Lygidakis & Nikolos, 2014a; Lygidakis *et al.*, 2016), in the proposed algorithm the coordinates of each *supernode* are set equal to those of an initial node, included in this super-region, having the minimum distance to the area-weighted center of the super-area (FIGURE 5.2). In that way it is assured that the nodes of the reduced surface set (which are

going to be used in the  $N_s \times N_s$  equation system) lie indeed on the examined boundary. The pseudo-coordinates of the area-weighted center are evaluated, based on the coordinates and the control area of each included node, as follows:

$$\mathbf{x}_{P_{weighted}} = \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{P_{weighted}} = \frac{\sum_{i=1}^p \mathbf{x}_i A_i}{\sum_{i=1}^p A_i}, \quad (5.17)$$

where  $P$  denotes the *supernode*,  $p$  the included nodes, while  $A_i$  stands for their control areas' size.

In FIGURE 5.5 the defining procedure of the pseudo-coordinates of a *supernode*  $P$ , including five nodes  $p_i$  ( $i = 1, \dots, 5$ ), is schematically presented; the blue-coloured lines  $L_1$  to  $L_5$ , represent distances, between the orange-coloured area-weighted center  $P_{weighted}$  and the included in the *super-area* black-coloured points  $p_1$ ,  $p_2$ ,  $p_3$ ,  $p_4$  and  $p_5$ . According to the methodology developed in this study, the *supernode*  $P$  acquires the coordinates of node  $p_4$ , as the latter is located closer to  $P_{weighted}$  comparing to the remaining nodes ( $p_1$ ,  $p_2$ ,  $p_3$  and  $p_5$ ).

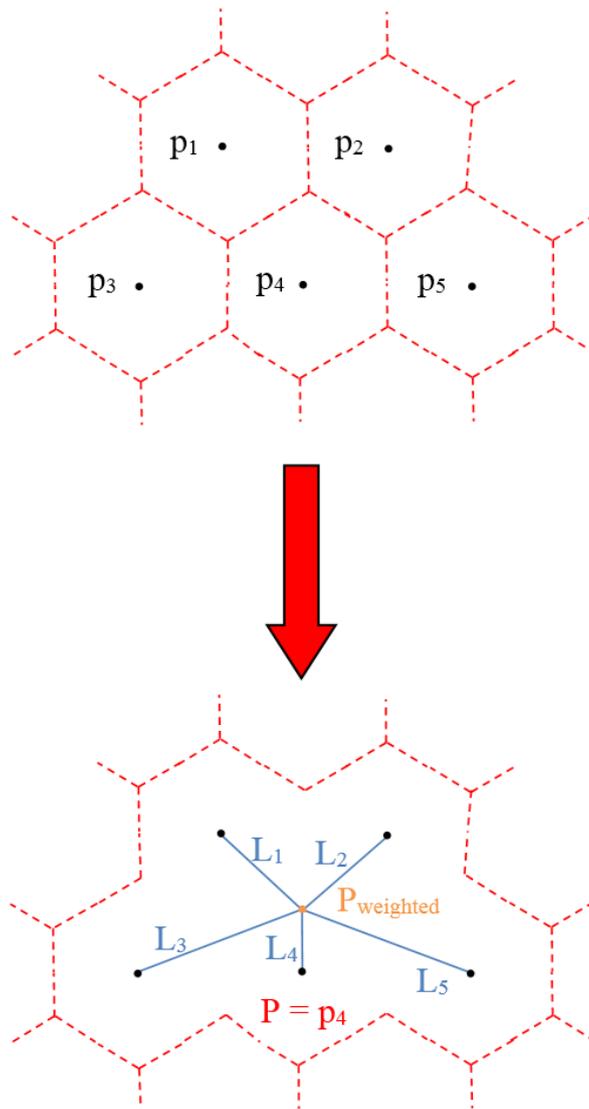


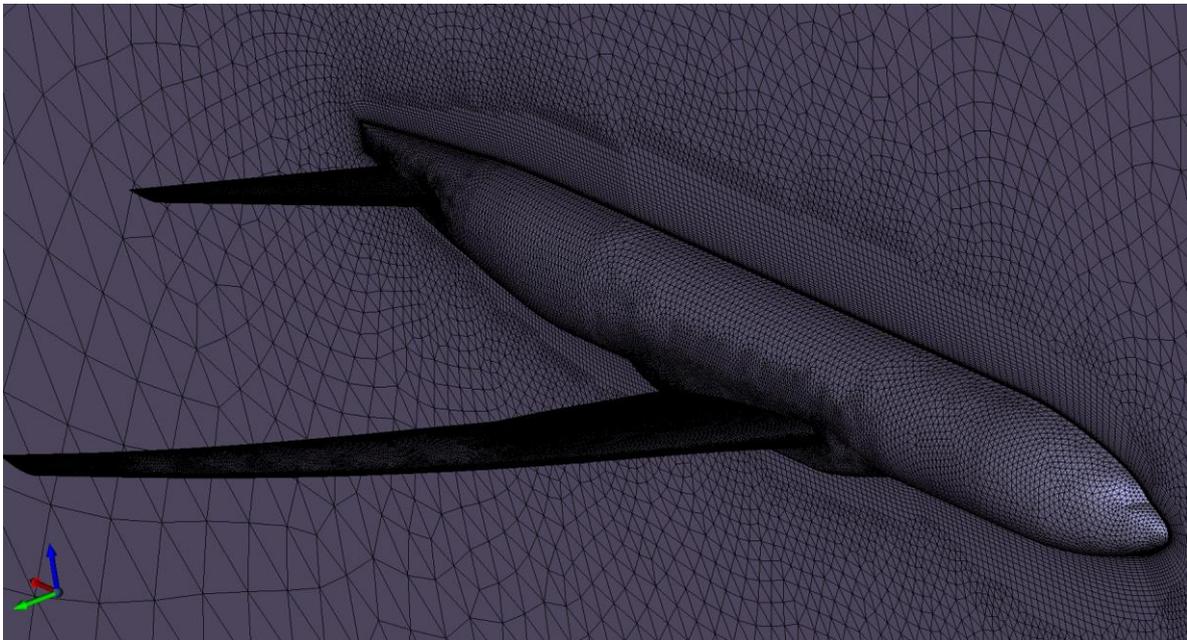
FIGURE 5.5: Schematic representation of supernodes coordinates' computation process.

### 5.2.3 Validation

Three benchmark test cases have been encountered in this Section to assess the performance of the proposed mesh deformation methodology. The open-source library Eigen (Guennebaud & Jacob, 2010) has been employed for the factorization and inversion of the coefficient matrices via LU decomposition (partial pivoting, vectorized), while the matrix-vector multiplications at the evaluation stage have been accelerated with OpenMP (Dagum & Menon, 1998a). Further acceleration has been achieved through parallel processing on a DELL™ R815 PowerEdge™ server with four AMD Opteron™ 6380 sixteen-core processors at 2.50 GHz (64 cores in total). Finally, for visualization of the results a custom GUI was developed based on OpenCascade graphical libraries (“OpenCASCADE,” 2015).

#### 5.2.3.1 NASA Common Research Model (CRM)

The first test case concerns the deformation of the wing of NASA CRM aircraft in a Wing-Body-Horizontal Tail (WBHT) configuration, which was examined in the fourth and fifth AIAA Drag Prediction Workshops (DPWs) (Lee-Rausch *et al.*, 2010; Ceze & Fidkowski, 2013; Levy *et al.*, 2013; Murayama *et al.*, 2013; Park *et al.*, 2013; Scalabrin & Felix, 2013; Sclafani *et al.*, 2013; Lygidakis & Nikolos, 2015). It is a 2.7% scaled geometry, representative of a modern commercial transonic flow aircraft (Lee-Rausch *et al.*, 2010), specifically designed for the validation of CFD codes (Sclafani *et al.*, 2013; Lygidakis & Nikolos, 2015). The mesh utilized in this work was provided from the site of the NASA Langley Research Center (LARC), where most of the grids used by the participants in the aforementioned DPWs are available. It consists of 4,653,134 nodes, 2,685,704 tetrahedra, 8,226,704 prisms and 53,630 pyramids, while it includes only the half configuration for computational savings (Lygidakis & Nikolos, 2015); in FIGURE 5.6 its density on aircraft and symmetry surfaces is illustrated.



**FIGURE 5.6:** Mesh density on aircraft and symmetry surfaces for the NASA CRM case.

The deformation examined herein corresponds to a parabolic deformation of the 26.5 m aircraft main wing beginning from its root and increasing successively to its maximum value at the wing tip. The coordinates of the deformed surface of the wing can be obtained employing the parabolic expression  $\Delta y = ax^2$  with maximum deflection 5 m at wing tip. Although being an unphysical deformation, due to the resulting extension of the wing, it remains suitable for the evaluation of the proposed methodology. The final mesh deformation was succeeded via the Wendland's  $C2$  function with a support radius of 18 m; consequently, the practical restriction that the support radius should be larger than the largest displacement of any surface point was taken into account (de Boer, van der Schoot, *et al.*, 2007). In FIGURE 5.7 the initial (un-deformed) and final (deformed) wing is presented.

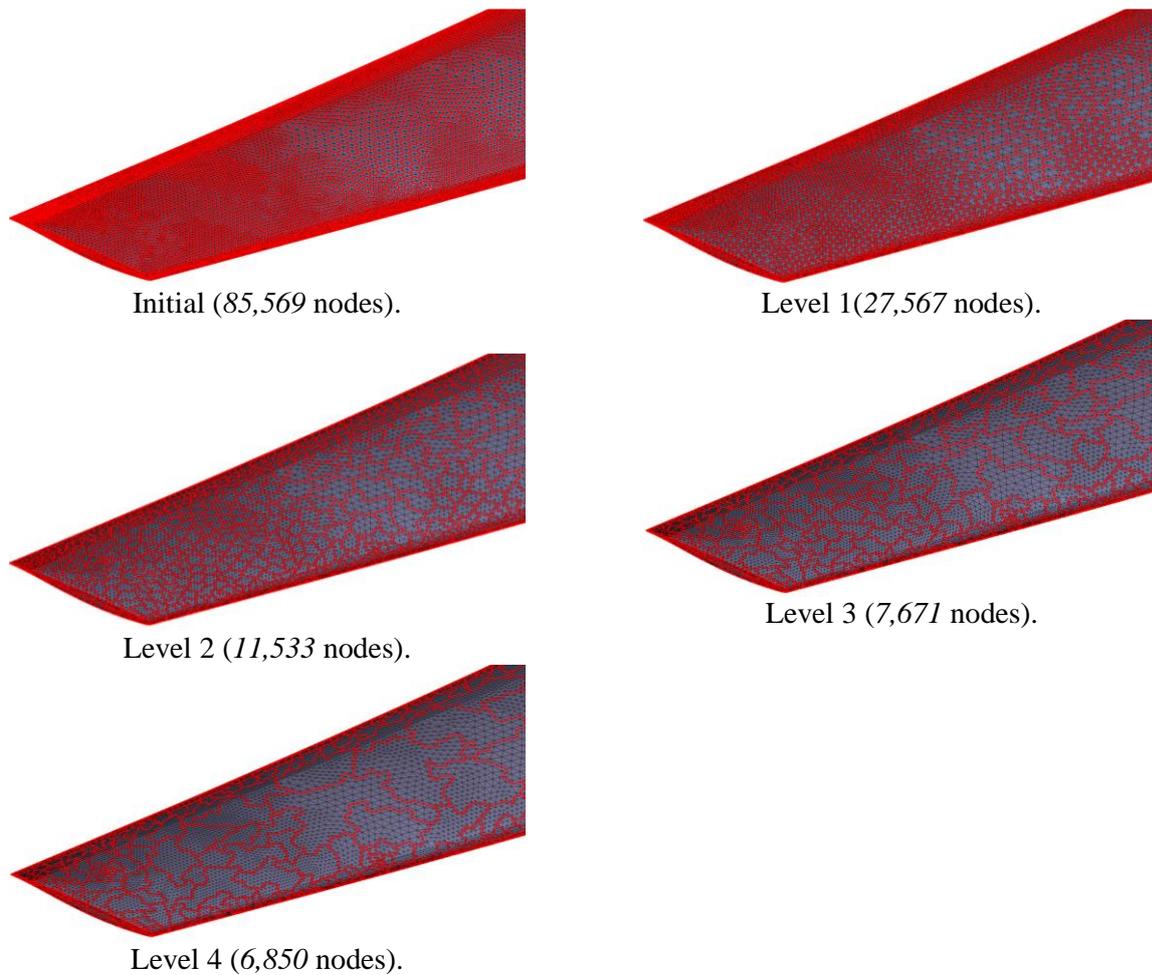


**FIGURE 5.7:** Parabolic deformation of the NASA CRM aircraft wing (max deflection 5 m at wing tip).

For the acceleration of the computational process four coarser resolutions were generated via agglomeration. TABLE 5.1 includes for each fusion level the number of wing surface nodes used as centers for RBFs and consequently for the implementation of the proposed deformation algorithm; FIGURE 5.8 shows their boundary control areas at the initial and the four successive coarser grids.

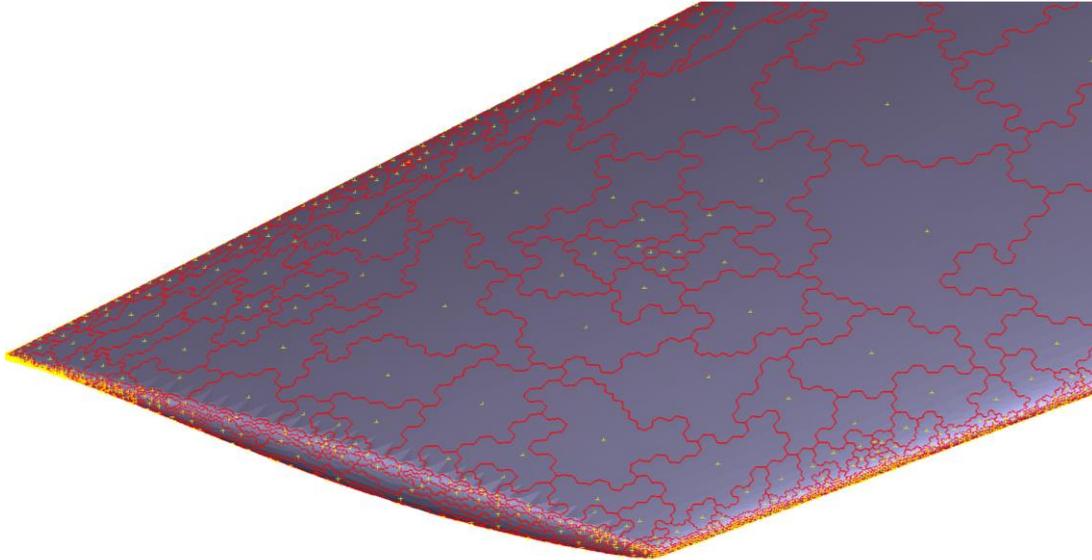
**TABLE 5.1:** Number of wing surface nodes at each agglomeration level (NASA CRM case).

<i>Agglomeration level</i>	<i>Number of wing surface nodes</i>
<i>Initial</i>	85,569
<i>Level 1</i>	27,567
<i>Level 2</i>	11,533
<i>Level 3</i>	7,671
<i>Level 4</i>	6,850

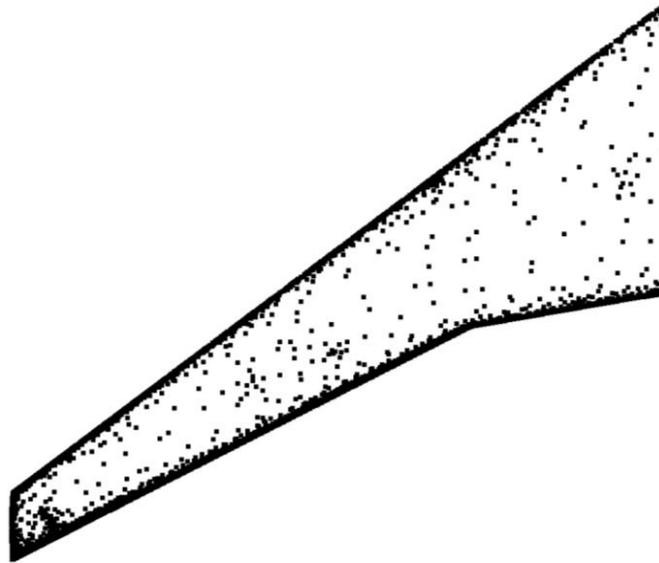


**FIGURE 5.8:** Mesh density on the wing surface of the initial and agglomerated control area grids (NASA CRM case).

FIGURE 5.9(a) presents the extracted supernodes (yellow-coloured), as well as their control areas (red-coloured) at the coarsest resolution; the first ones correspond to the points of the initial finest grid with the minimum distance from the area-weighted centers of their control areas. In FIGURE 5.9(b) the distribution of the final base points over the wing is illustrated. As expected, most of the points were selected in the regions near the leading and trailing edges of the wing, due to the limited or even prohibited agglomeration in areas with high curvatures and slope discontinuities (constraints (b) and (d)). In that way the initial shape of the geometry was preserved, regardless the significant reduction of the RBF-centers.



(a)



(b)

**FIGURE 5.9:** *Super-areas' centers (yellow-coloured) at the coarsest grid (a) and their distribution over the wing surface (b) (NASA CRM case).*

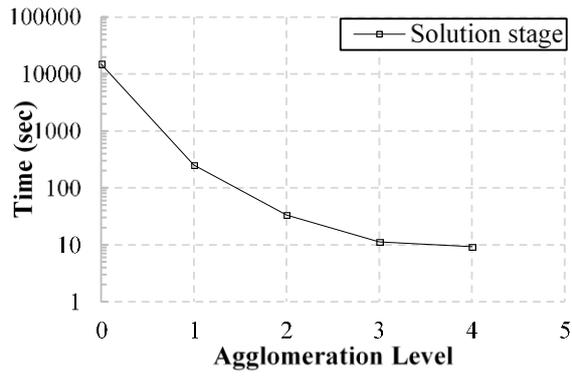
Furthermore, runs with less (one, two and three) agglomeration levels were performed, as well as for the initial grid, in order to assess the contribution of each additional coarser resolution to the acceleration of both stages of the procedure (solution and evaluation ones). As the computation time required for the generation of the coarser grids is relatively negligible, it is included to the computation times presented at next. Besides computational cost, the average and maximum error of the derived deformed surfaces were computed; this error corresponds to the Euclidean distance between the final location of each point on the deformed surface and the one calculated with the aforementioned parabolic expression. TABLE 5.2 and FIGURE 5.10 include the numerical results for all the runs (with different numbers of employed agglomeration levels), while FIGURE 5.11 illustrates the

corresponding error distributions on the wing surface. For reference, the height of the first grid layer on the wing is 0.02 m.

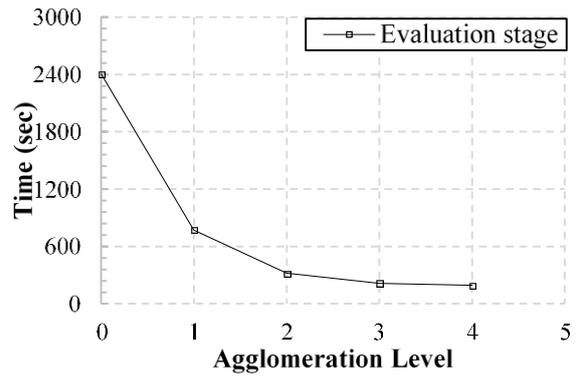
**TABLE 5.2:** *Wall-clock computation time and surface errors of the RBF-mesh deformation procedure for different agglomeration levels (NASA CRM case).*

<i>Agglomeration level</i>	<i>Solution stage (sec)</i>	<i>Solution stage (% of Initial)</i>	<i>Evaluation stage (sec)</i>	<i>Evaluation stage (% of Initial)</i>	<i>Average Error (m)</i>	<i>Maximum Error (m)</i>
<i>Initial</i>	15088.4		2404		-	-
<i>Level 1</i>	253.3	1.679	773	32.155	$2.07 \times 10^{-8}$	$6.21 \times 10^{-6}$
<i>Level 2</i>	33.4	0.221	319	13.269	$8.97 \times 10^{-8}$	$2.39 \times 10^{-5}$
<i>Level 3</i>	11.3	0.075	214	8.902	$3.45 \times 10^{-7}$	$2.97 \times 10^{-5}$
<i>Level 4</i>	9.3	0.062	193	8.028	$1.38 \times 10^{-6}$	$5.54 \times 10^{-5}$

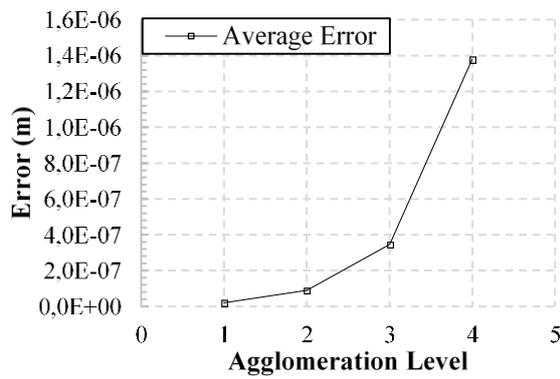
Comparing the derived (wall-clock) computation times it is quite clear that a large reduction of the surface nodes (succeeded in this study via agglomeration) has a significant impact on the efficiency of the mesh-deformation algorithm, even for highly demanding test cases as the one considered here. The boundary points of the initial finest grid were decreased approximately 12.5 times at the coarsest resolution. In that way a theoretically unmanageable deformation problem was solved in just a few seconds, since the proposed fusion methodology influences both the solution and evaluation stages of the procedure. It should be highlighted that the difference in computation time of the solution stage of the RBF-procedure between the final agglomeration level (4) and the initial grid is more than three orders of magnitude. Despite the considerable reduction of centers, and consequently of memory and computational requirements, the accuracy of the deformed surface was maintained at a competitive level. Although the imposed 5 m max deflection is in general larger than a real one encountered by a wing in an aeroelastic simulation, both average and max surface errors remain relatively low.



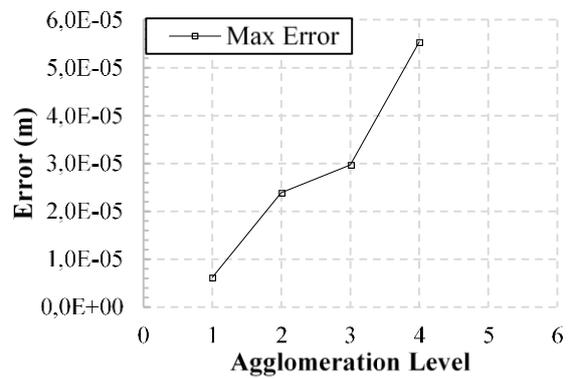
(a)



(b)



(c)



(d)

**FIGURE 5.10:** Distributions of computation time of the solution (a) and evaluation (b) stages, as well as of average (c) and maximum (d) wing surface errors for different agglomeration levels (NASA CRM case).

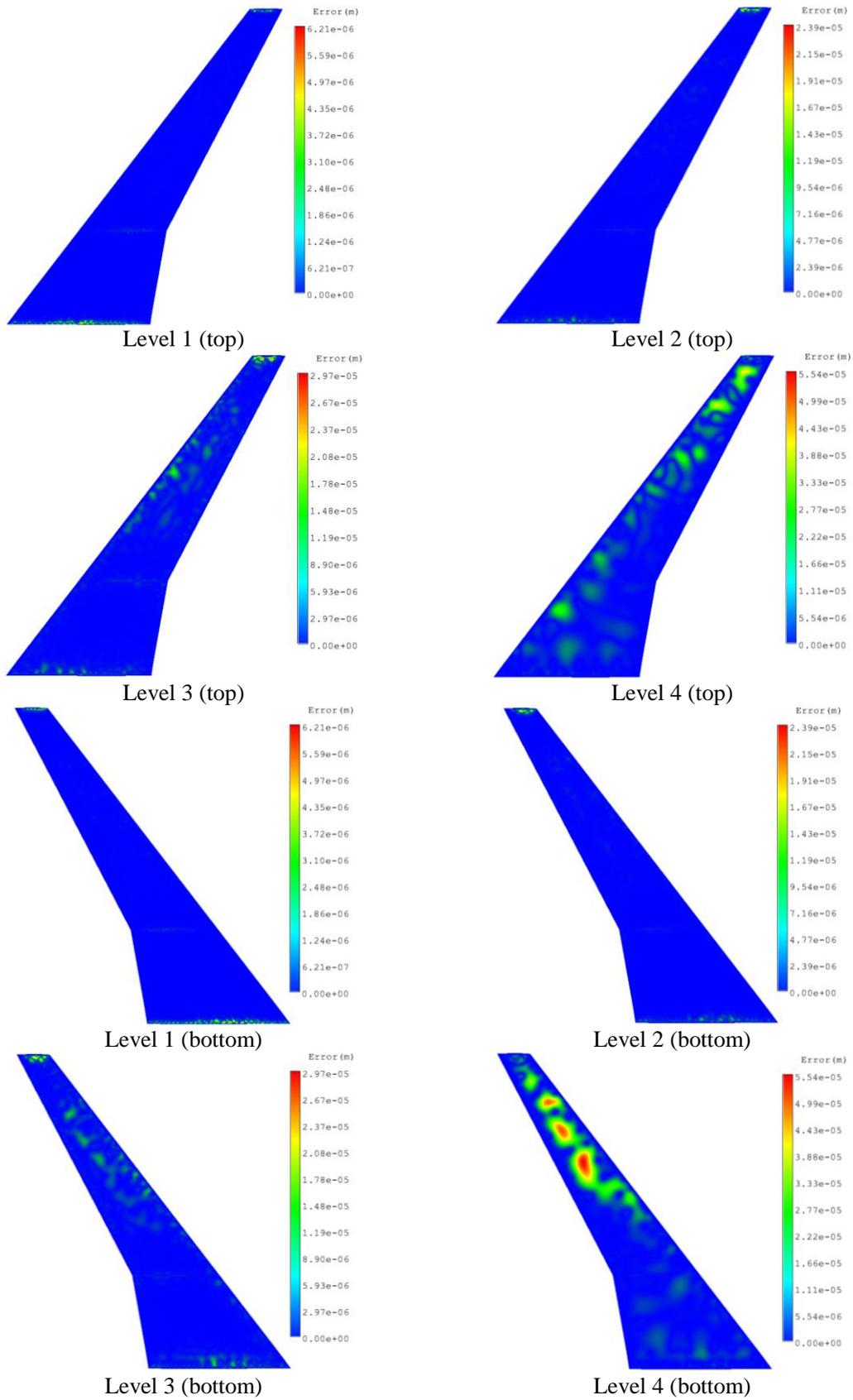
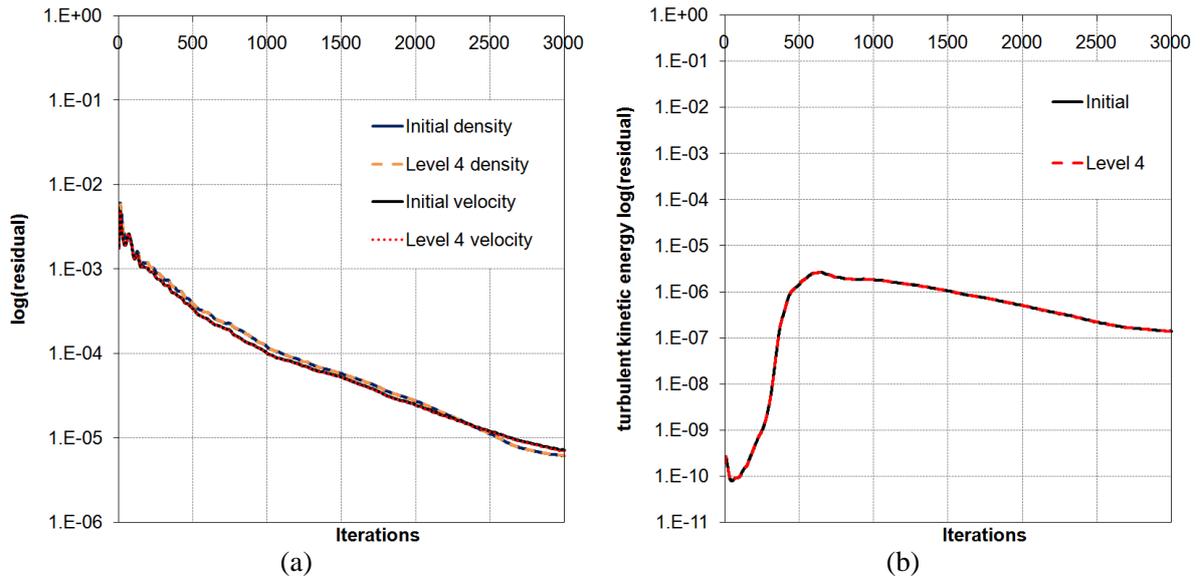


FIGURE 5.11: Distribution of wing surface error for different numbers of agglomeration levels (CRM case).

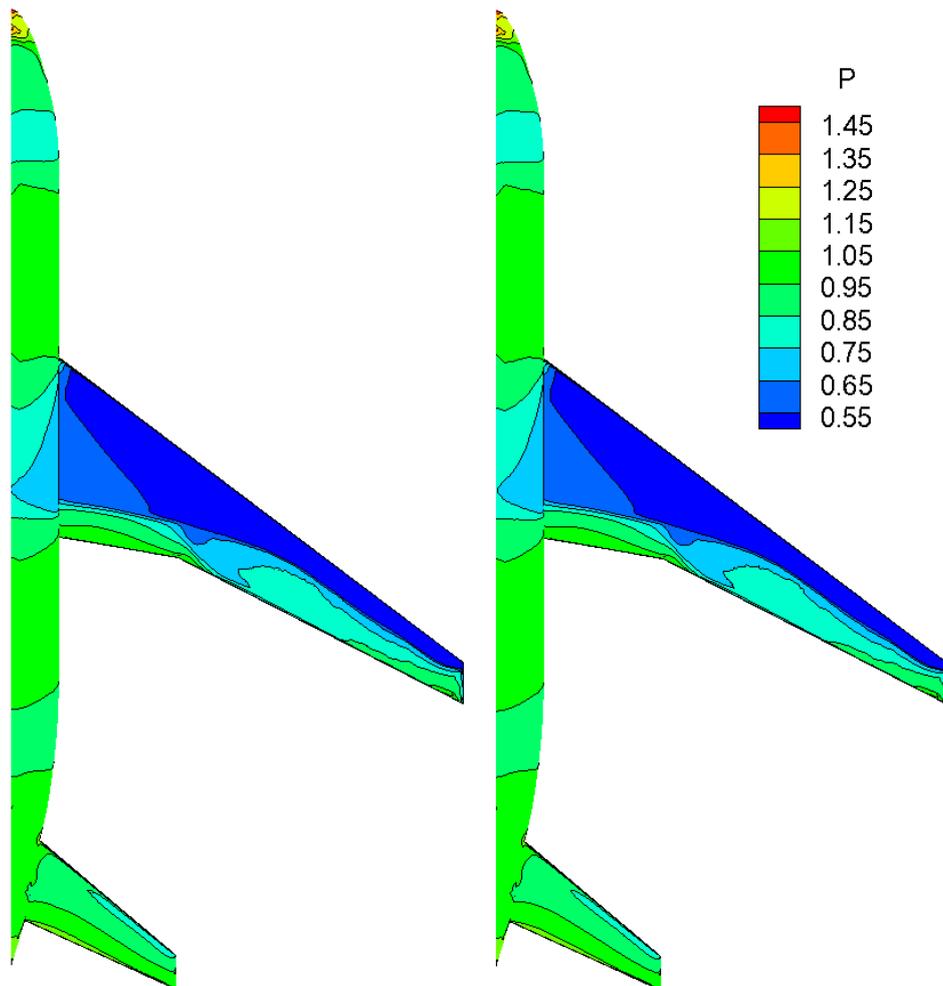
To assess the quality of the obtained deformed grids, different mesh-quality metrics have been calculated, with the corresponding results contained in

In order to evaluate the differences between the extracted grids and consequently validate their quality, fully turbulent flow over the deformed CRM WBHT case was examined with the academic CFD solver *Galatea* (Lygidakis & Nikolos, 2015). Due to its dimensionless formulation, both meshes were re-dimensionalized in order the mean aerodynamic chord to become equal to unity (Lygidakis & Nikolos, 2015). The Mach and Reynolds numbers (based on the unit mean aerodynamic chord) characterizing the free-stream flow, were assumed equal to 0.85 and  $5.0 \times 10^6$  respectively. The angle of attack and reference temperature were set equal to  $5^\circ$  and  $310.927 \text{ K}$  ( $100^\circ\text{F}$ ), respectively. The inviscid fluxes were computed with a second-order accurate spatial scheme, based on the Monotone Upwind Scheme for Conservation Laws (MUSCL) along with Van Albada-Van Leer limiter (Blazek, 2001; Lygidakis & Nikolos, 2015), while for the computation of the viscous ones the required gradients at control volumes' interfaces were evaluated with the nodal-averaging scheme (Blazek, 2001; Lygidakis & Nikolos, 2015). Turbulence prediction was succeeded via the Shear Stress Transport (SST) model, while no transition area was assumed (Lygidakis & Nikolos, 2015). For the approximation of the final steady-state solution the four-stage Runge-Kutta method (RK(4)) was implemented with a CFL number equal to 0.5. The solution procedure was accelerated with the incorporated agglomeration multigrid scheme (Nishikawa & Diskin, 2011; Nishikawa *et al.*, 2013; Lygidakis & Nikolos, 2014b) and parallelization strategy (Lygidakis & Nikolos, 2013, 2015); for the implementation of the first methodology two coarser grids were generated via full-coarsening directional agglomeration, while for parallel processing the initial mesh, as well as the coarser ones (for multigrid technique), were divided in sixteen sub-domains. The numerical results were obtained after density residual was decreased approximately three orders of magnitude. Although the succeeded residual reduction is rather small, it is considered sufficient enough for the purpose of the present study. Moreover, such aircraft test cases often exhibit unsteadiness after similar residual decreases.

In FIGURE 5.12 the convergence history per number of iterations (multigrid V-cycles) for density, velocity in x-direction (left) and turbulent kinetic energy (right), are presented for both the deformed grids, extracted with (Level 4) and without (Initial) agglomeration of centers. Almost no difference can be identified between them, indicating the potential of the proposed algorithm for the generation of equally accurate deformed meshes in a relatively negligible computation time (comparing to those derived without agglomeration). In addition, FIGURE 5.13 includes the predicted contours of dimensionless pressure; a perfect agreement is obvious between the utilized deformed meshes.

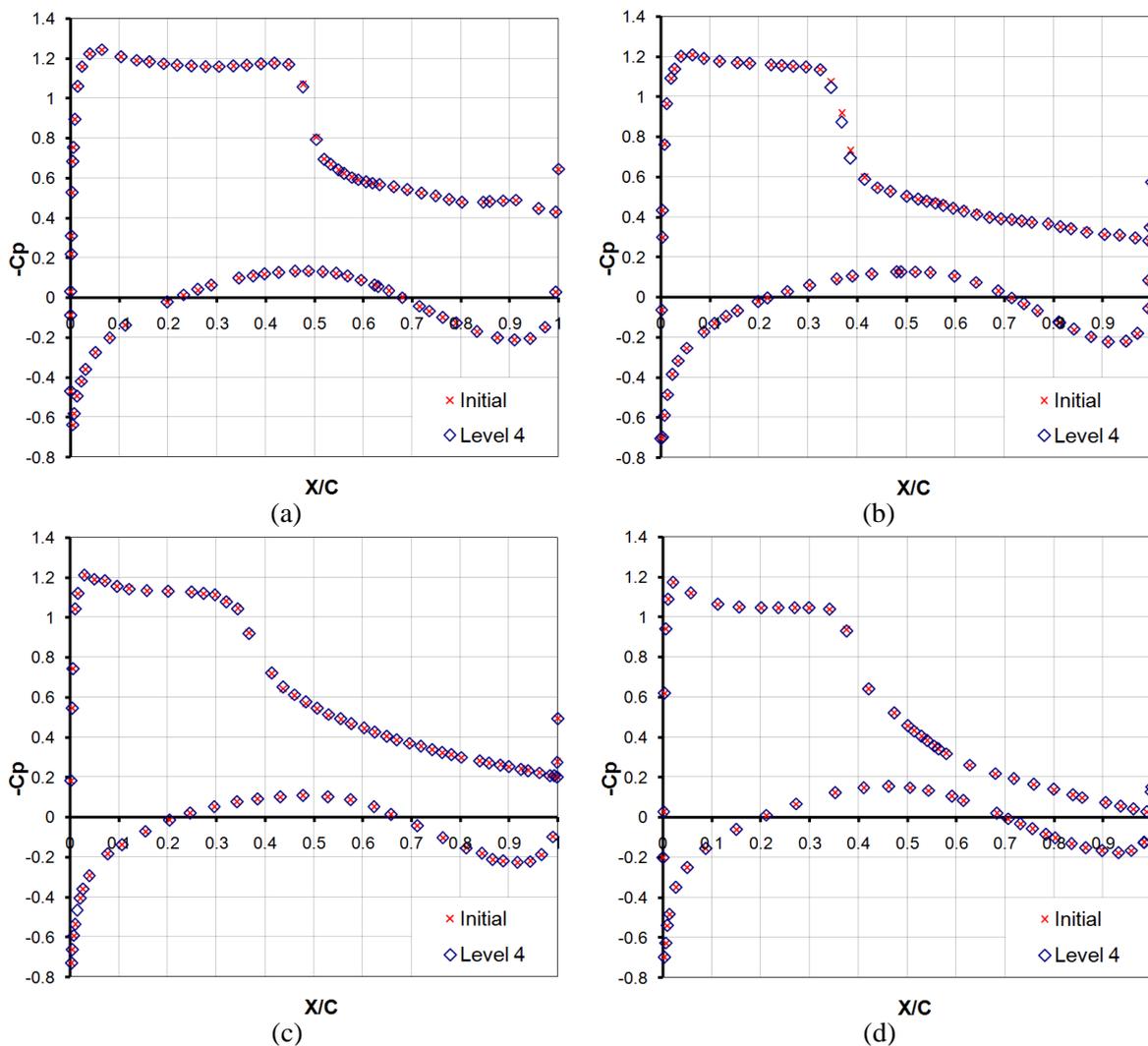


**FIGURE 5.12:** Convergence histories for density and velocity (left), and turbulent kinetic energy (right) for the deformed grids, extracted with (Level 4) and without (Initial) agglomeration of RBF centers (NASA CRM case).



**FIGURE 5.13:** Top view of aircraft surface dimensionless pressure contours of the deformed grids, derived with (left) and without (right) agglomeration of centers (NASA CRM case).

In FIGURE 5.14 the aforementioned agreement is confirmed quantitatively with the extracted distributions of pressure coefficient at different wing span-wise sections (a: 50.24%, b: 72.68%, c: 84.56% and d: 95.00%). The selection of these sections was based on the maximum values of error distribution, shown in FIGURE 5.11; errors seem to be larger in the region close to the wing tip. Nevertheless, even at the lower-outer wing surface where the largest errors appear, a perfect match can be identified between the pressure coefficient distributions for the two examined deformed grids. The aforementioned qualitative and quantitative comparisons confirm the potential of the proposed deformation algorithm for effective generation of deformed meshes in terms of both efficiency and accuracy.

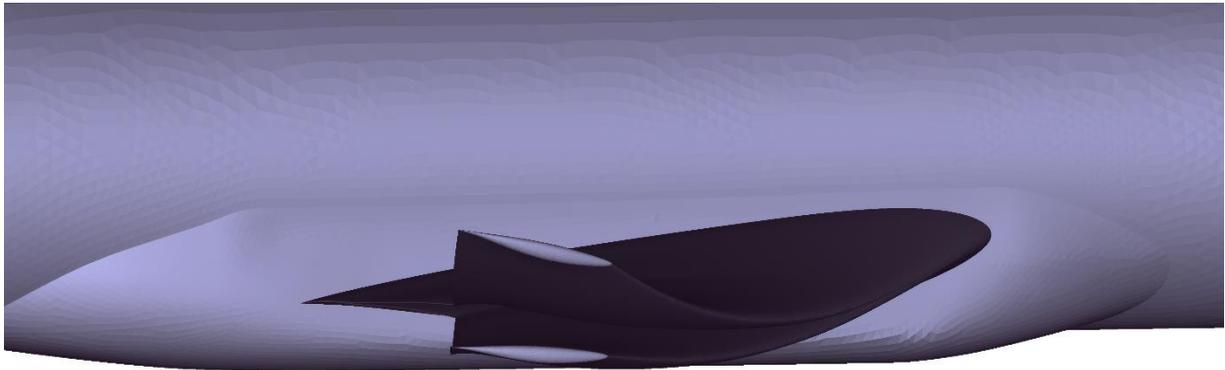


**FIGURE 5.14:** Distribution of pressure coefficient at different wing span-wise sections (a: 50.24%, b: 72.68%, c: 84.56% and d: 95.00%) of the deformed grids, obtained with (Level 4) and without (Initial) agglomeration of RBF centers (NASA CRM case).

Finally, the ratio of the wall clock computational time required for the mesh deformation procedure and this for the solution of the flow PDEs per each real time step was computed approximately equal to 2.0 (which goes down to 0.83 if loose constraints are used in the agglomeration procedure, as it will be presented later -TABLE 5.4). However, it should be highlighted here that such unsteady compressible solvers apply the relaxation of the

governing equations only once per each real time step (e.g., the Runge-Kutta scheme in *Galatea* solver is performed only once at each real time step), resulting thus in a relatively increased temporal ratio. On the contrary, the unsteady incompressible solvers (usually employed in FSI problems) entail a considerable number of iterative solutions per each real time step, resulting in much smaller or even negligible such ratios (Lygidakis *et al.*, 2016).

In order to evaluate the potential of the proposed methodology against more complex problems, a modified test case was encountered, considering an additional torsional deformation (over-imposed to a bending one) of the CRM aircraft main wing. Moreover, the influence of the pre-defined limitations (b) and (d) of the agglomeration procedure was also assessed. The additional deformation examined herein corresponds to a parabolic torsional deformation of the aircraft main wing, beginning from its root and increasing successively to its maximum value of 10 degrees at the wing tip; the imposed maximum (bending) displacement at the flapwise direction was 1.5 m. The torsional deformation was obtained with the rotation of the wing around an axis connecting the gravity centers of the cross-sections at the wing tip and wing root. FIGURE 5.15 illustrates the initial and final (deformed) geometry of the wing.

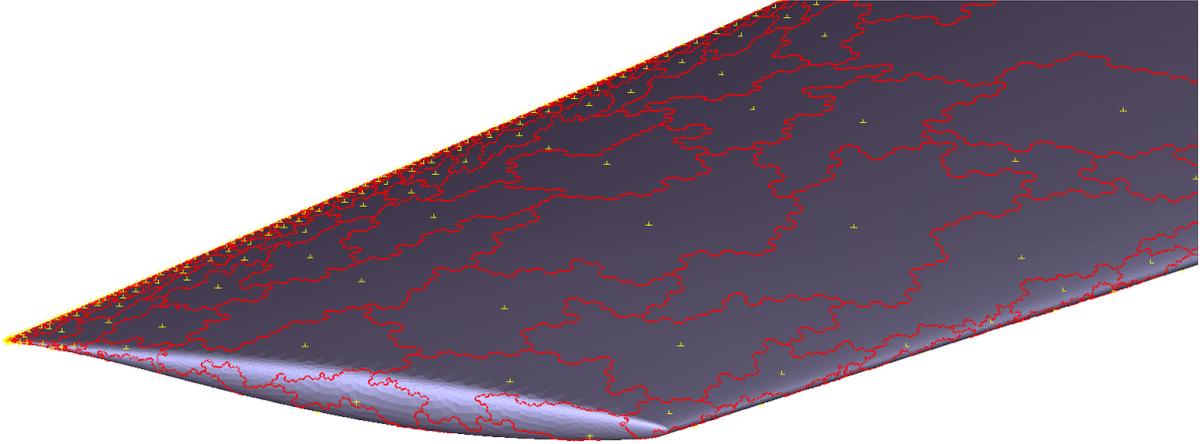


**FIGURE 5.15:** Deformation of the NASA CRM aircraft wing (max bending deflection 1.5 m at wing tip, max rotation of 10° at wing tip).

Considering additionally the evaluation of the constraints (b) and (d) of the agglomeration procedure, two different point-reduction approaches were examined. For the first one with the *strict constraints*, the restrictions (b) and (d) were imposed with the angular limits of 10° and 30°, respectively, while for the second one with the *loose constraints* the limits were set equal to 75° and 80°, respectively. Similarly to the previous tests, four coarsest resolutions were generated via the proposed agglomeration strategy. TABLE 5.3 reports the number of the wing surface nodes at each fusion level for both the aforementioned point-reduction approaches. FIGURE 5.16 depicts the distribution of the selected *supernodes* (yellow-colored) on the same surface as well as their control areas (red-coloured) at the coarsest resolution, extracted with the approach with the *loose constraints*.

**TABLE 5.3:** Number of wing surface nodes at each agglomeration level (NASA CRM case with additional torsional deformation).

<i>Agglomeration level</i>	<i>Number of wing surface nodes (Strict constraints)</i>	<i>Number of wing surface nodes (Loose constraints)</i>
<i>Initial</i>	85,569	85,569
<i>Level 1</i>	27,567	24,493
<i>Level 2</i>	11,533	8,119
<i>Level 3</i>	7,671	3,781
<i>Level 4</i>	6,850	2,742



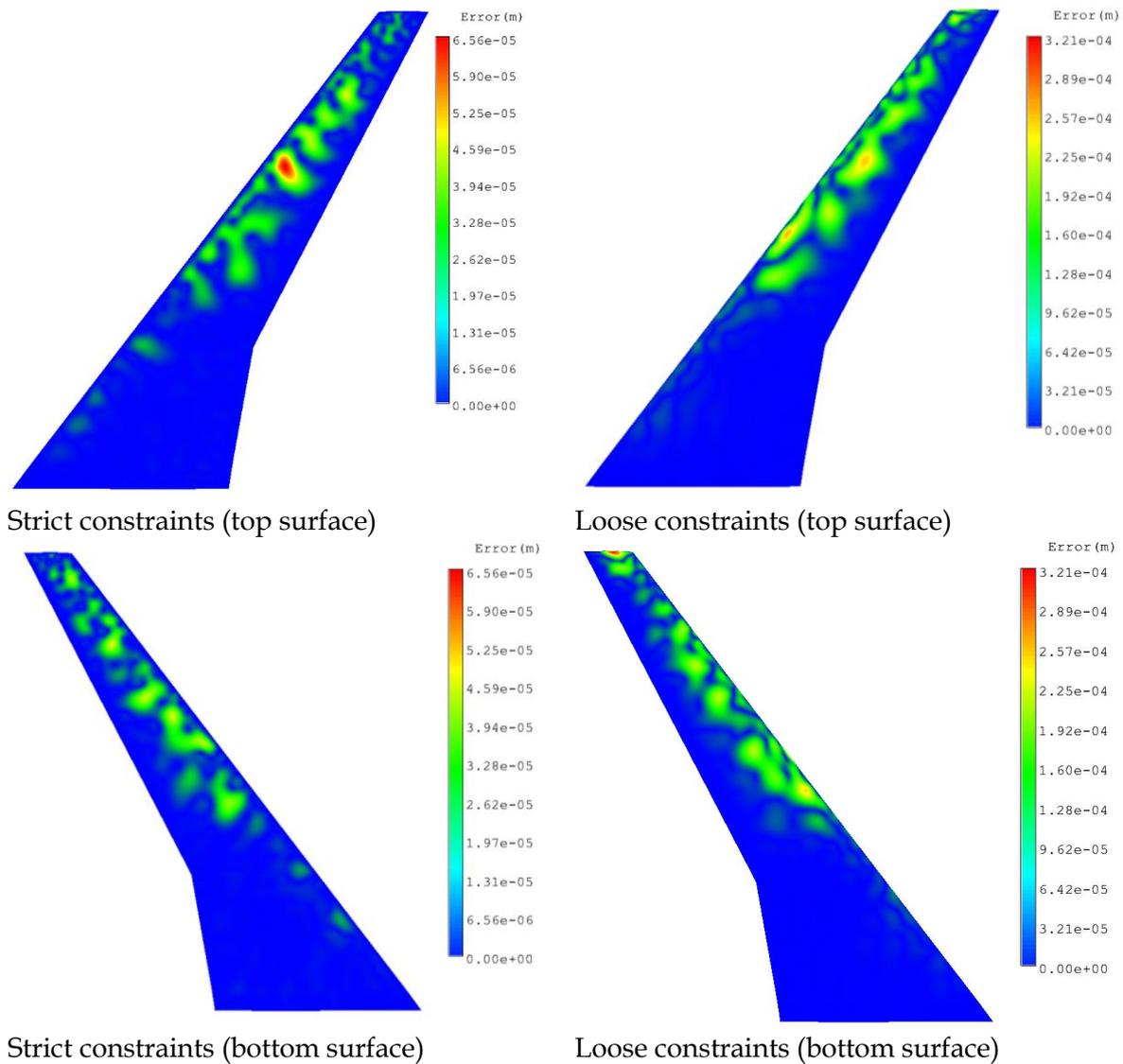
**FIGURE 5.16:** Super-areas' centers (yellow-coloured) at the coarsest grid, extracted via the approach with the loose constraints (NASA CRM case with additional torsional deformation).

Comparing the distribution of the RBF-centers on the wing surface, extracted with the *loose constraints* to this derived by the *strict* ones, it is obvious that less control points are located in the regions with high curvatures or slope discontinuities, e.g., leading edge, trailing edge, wing tip, etc. As a result, a deeper agglomeration and point reduction is succeeded. Nevertheless, the initial shape of the geometry is adequately maintained.

**TABLE 5.4:** Wall-clock computation time and surface errors of the RBF-mesh deformation procedure employing strict and loose constraints (NASA CRM case with additional torsional deformation).

<i>Agglomeration level</i>	<i>Solution stage (sec)</i>	<i>Solution stage (% of Initial)</i>	<i>Evaluation stage (sec)</i>	<i>Evaluation stage (% of Initial)</i>	<i>Average Error (m)</i>	<i>Maximum Error (m)</i>	<i>DTS ratio*</i>
<i>Strict con.</i>	7.6	0.051	178	7.404	$1.66 \times 10^{-6}$	$6.56 \times 10^{-5}$	1.99
<i>Loose con.</i>	1.4	0.009	76	3.161	$1.80 \times 10^{-5}$	$3.21 \times 10^{-4}$	0.83

\*Ratio of the wall clock computational time required for the mesh deformation procedure and this for the solution of the flow PDEs per each time step.



**FIGURE 5.17:** Distribution of wing surface error for different constraint angles (CRM case with additional torsional deformation).

TABLE 5.4 and FIGURE 5.17 contain the derived computation times and the distribution of the error over the deformed surface, respectively. Although the accuracy of the deformed surface was maintained at a competitive level for both the applied approaches (with strict and loose constraints), the obtained results reveal that a compromise between accuracy and efficiency has to be fulfilled. As expected, strict constraints produce better approximations with higher accuracy (smaller error) while loose ones result in smaller matrix systems and consequently in more efficient solutions.

### 5.2.3.2 Wind turbine blade

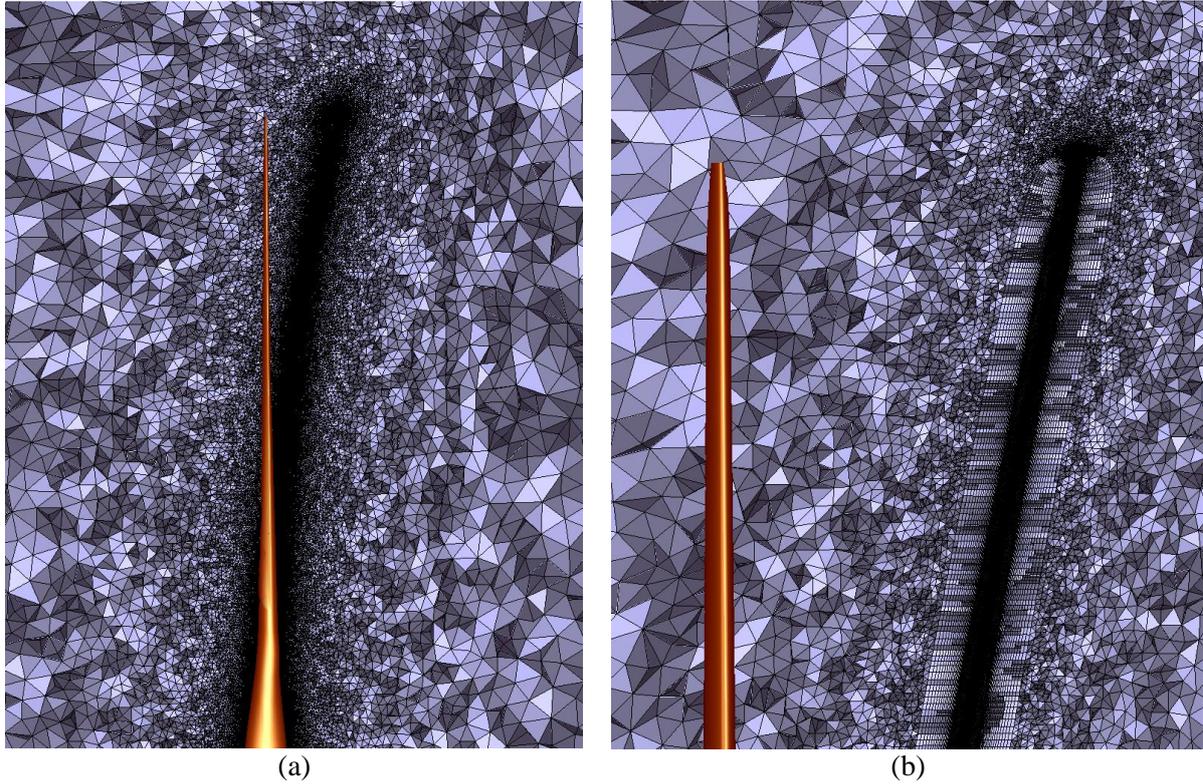
The second test case considers the deformation of a blade of a 5-MW baseline wind turbine, which was developed at the National Renewable Energy Laboratory (NREL) as a Horizontal Axis Wind Turbine (HAWT) for offshore applications (Jonkman et al., 2009). Its geometric and aerodynamic properties are available in (Resor, 2013). The utilized hybrid grid, representing the aforementioned blade, is composed of 5,648,073 nodes, 13,145,351 tetrahedra, 6,722,596

prisms and 2,678 pyramids, while for the acceleration of the computational procedure four coarser resolutions were generated via agglomeration. TABLE 5.5 includes the number of blade surface nodes for each fusion level.

**TABLE 5.5:** *Number of surface nodes at each agglomeration level (NREL case).*

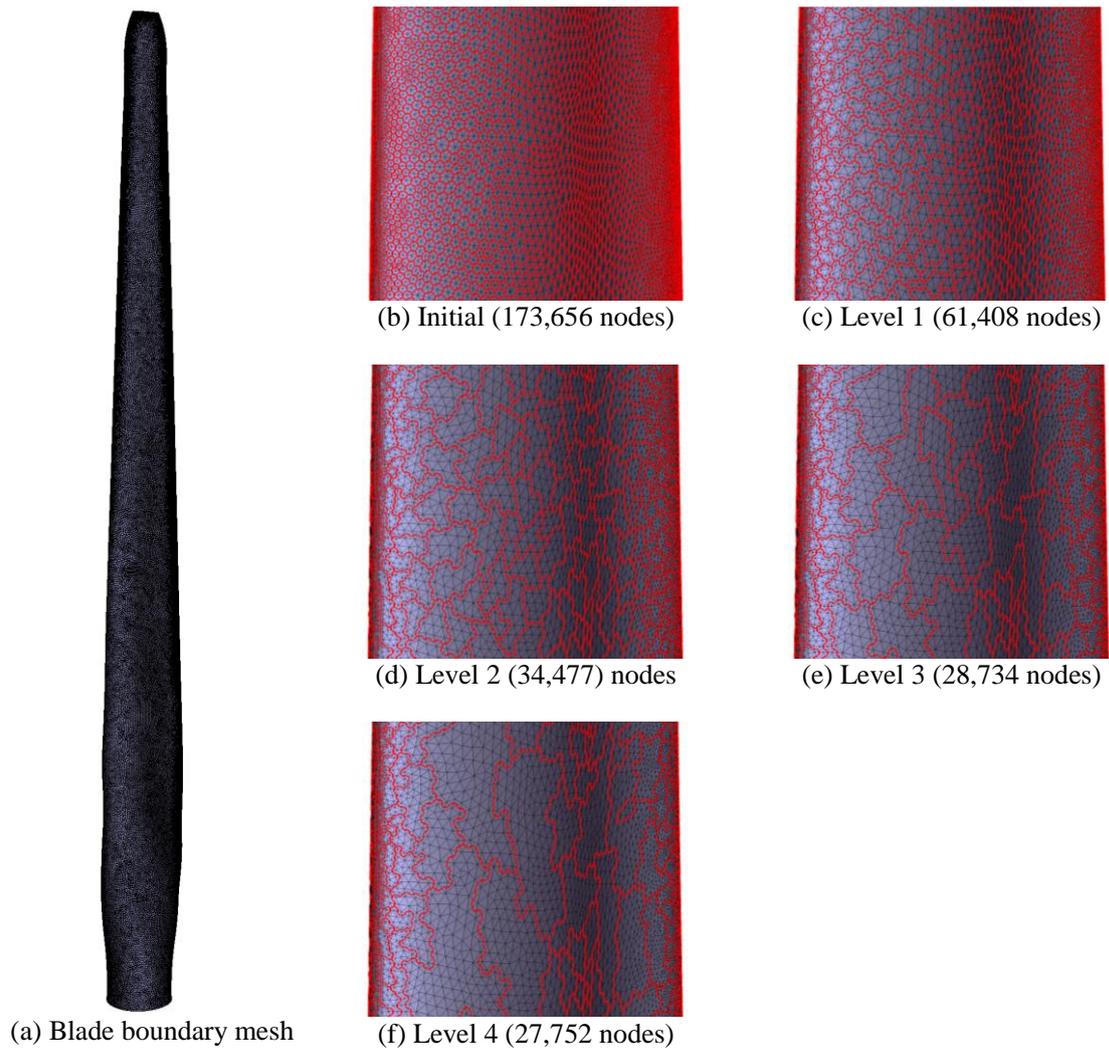
<i>Agglomeration level</i>	<i>Number of blade surface nodes</i>
<i>Initial</i>	173,656
<i>Level 1</i>	61,408
<i>Level 2</i>	34,477
<i>Level 3</i>	28,734
<i>Level 4</i>	27,752

Similarly to the previous case, a parabolic deformation was applied on the blade, assuming a maximum value of 6 *m* at the tip. The support radius of the RBF-method was set equal to 25 *m*. In FIGURE 5.18 the deformed volume mesh is presented, while FIGURE 5.19 illustrates the boundary control areas at the initial and four coarser grids. Furthermore, FIGURE 5.19 presents the extracted *supernodes* (yellow-coloured) as well as their control areas (red-coloured) at the coarsest resolution.



**FIGURE 5.18:** Deformed mesh of NREL 5-MW turbine blade (a) and a close-up view at the tip region (b).

Most of the centers again were automatically selected at the regions near the edges of the blade, outlining its shape. In that way it is confirmed once again that the proposed algorithm preserves the initial shape of the geometry, despite the significant reduction of its surface nodes used as RBF-centers. Additional runs with less fusion levels were performed, deriving the (wall-clock) computation times for both solution and evaluation stages, as well as the surface errors, presented in TABLE 5.6 and FIGURE 5.21; FIGURE 5.22 illustrates the corresponding distributions of errors on the blade surface. For reference, the height of the first grid layer on the blade is  $0.01\text{ m}$ . The initial grid is not included in TABLE 5.6 and FIGURE 5.21, as it was not possible for the available computer system to handle the resulting system of equations, due to memory limitations (128 Gb).



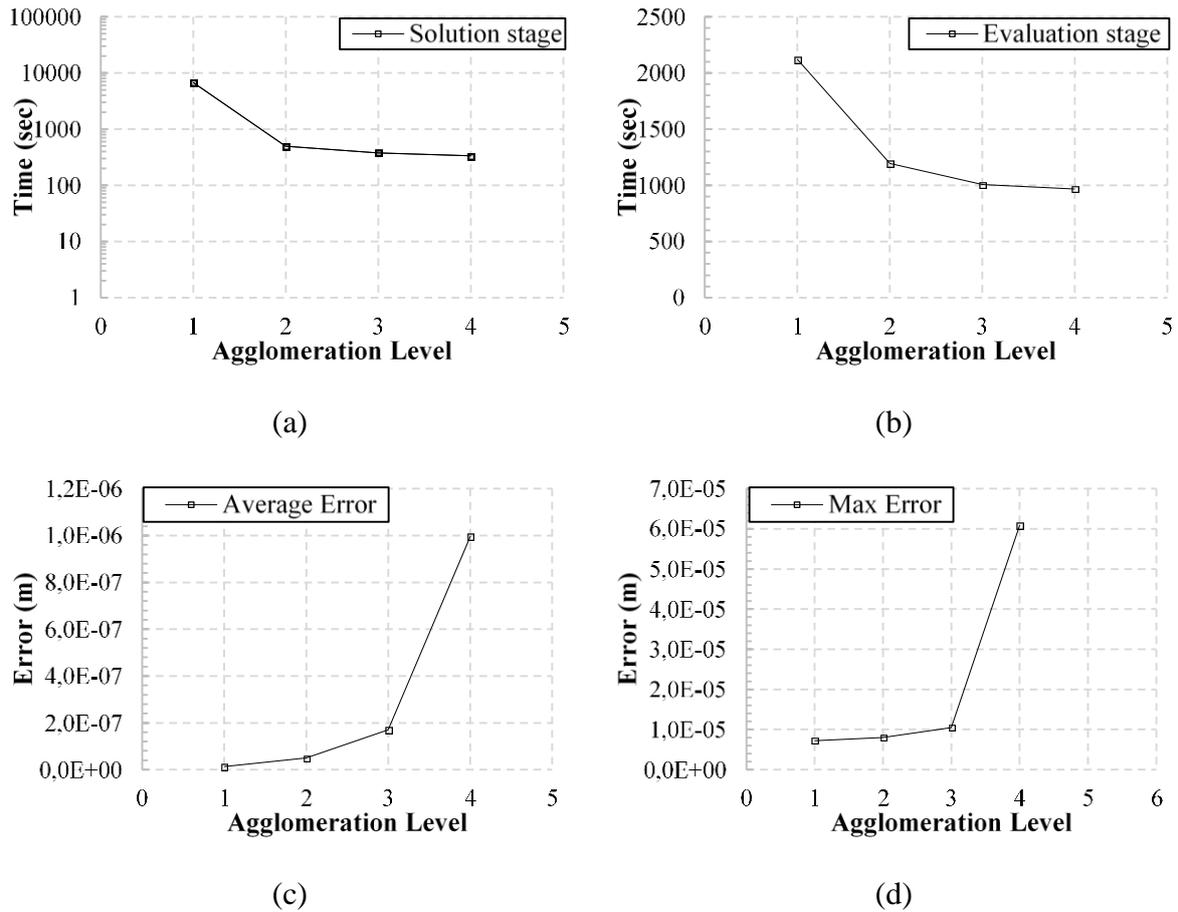
**FIGURE 5.19:** Mesh density on the blade surface of the initial (a-b) and agglomerated (c-f) surface grids (NREL case).



FIGURE 5.20: Super-areas' centers (yellow-coloured) at the coarsest resolution (NREL case).

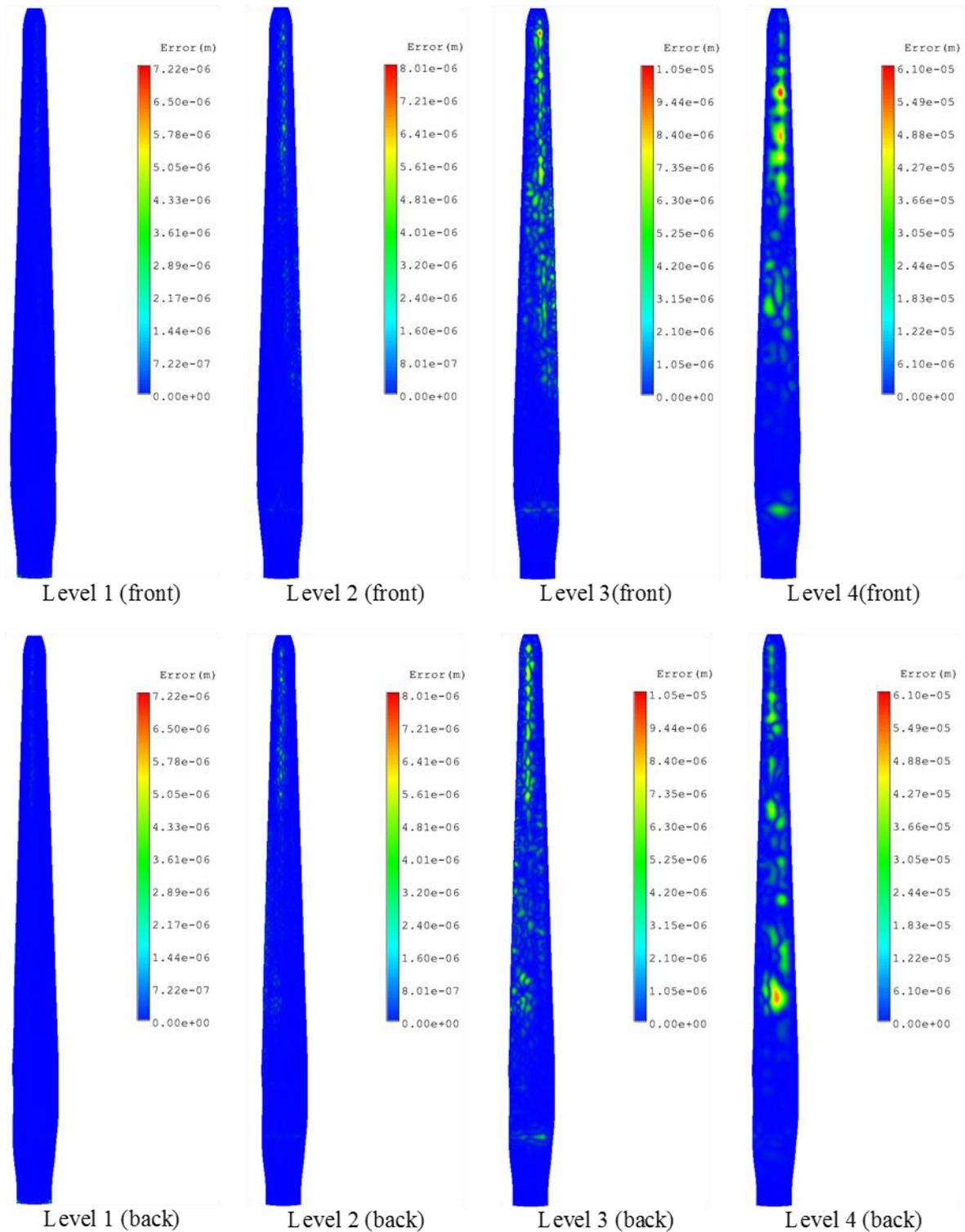
TABLE 5.6: Computation time and surface errors for different agglomeration levels (NREL case).

Agglomeration level	Solution stage (sec)	Evaluation stage (sec)	Average Error (m)	Maximum Error (m)
Level 1	6751.3	2120	$1.30 \times 10^{-8}$	$7.22 \times 10^{-6}$
Level 2	498.6	1196	$4.98 \times 10^{-8}$	$8.01 \times 10^{-6}$
Level 3	379.8	1006	$1.70 \times 10^{-7}$	$1.05 \times 10^{-5}$
Level 4	334.4	968	$9.96 \times 10^{-7}$	$6.10 \times 10^{-5}$



**FIGURE 5.21:** Distributions of computation time of the solution (a) and evaluation (b) stages, as well as of average (c) and maximum (d) blade surface errors for different agglomeration levels (NREL case).

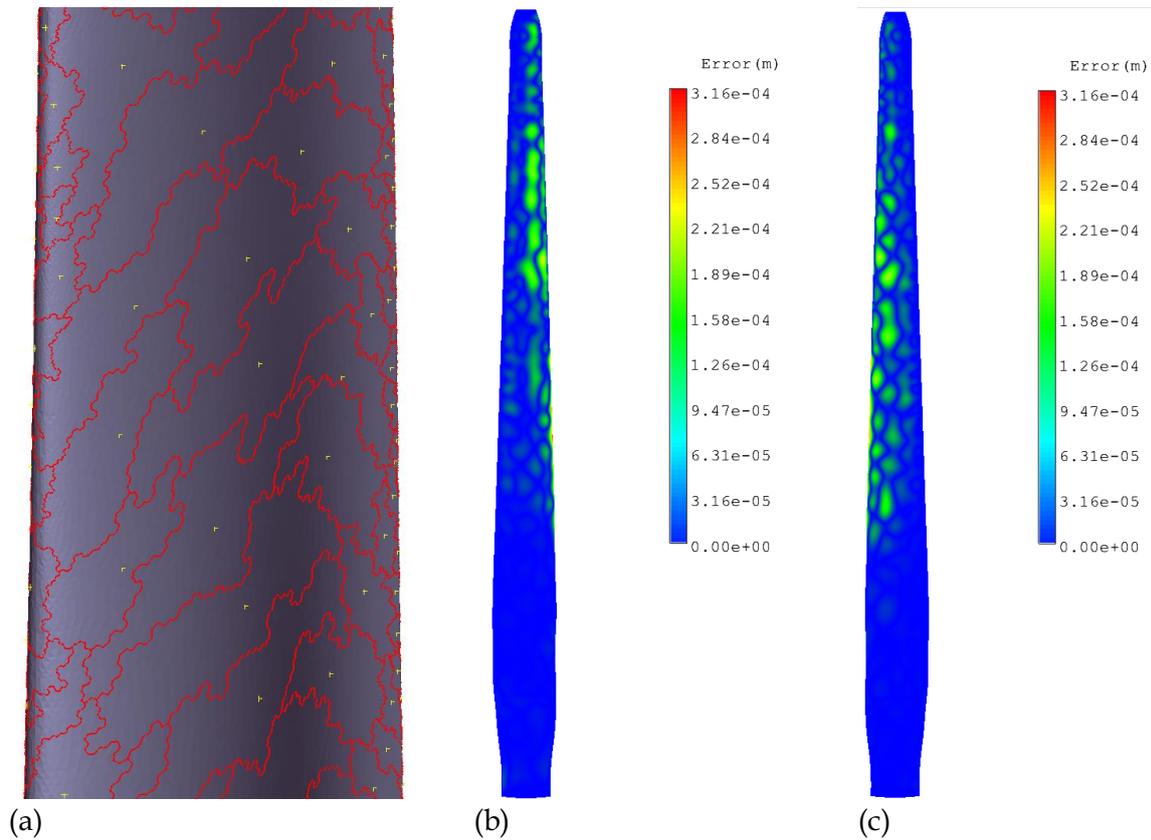
From the distributions, presented in *FIGURE 5.22*, the maximum errors are identified at the regions where the incorporated agglomeration strategy generated larger super control areas, i.e., at the regions where the curvature does not vary considerably. Nevertheless, the produced errors remain at an acceptable level, even for the higher levels of agglomeration, while once again no degenerated elements were produced. Moreover, in those areas of small curvature the produced errors have the smaller effect on the flow computation. Mesh quality checks have been performed as well (for the un-deformed and deformed grids), confirming that the proposed methodology does not reduce the quality of the derived meshes. The corresponding results are included in Appendix B.



**FIGURE 5.22:** Distribution of blade surface error for different agglomeration levels (NREL case).

Similarly to the previous test case, the agglomeration strategy with the *loose constraints* was assessed, e.g., the restrictions (b) and (d) were imposed with the angular limits of  $75^\circ$  and  $80^\circ$ , respectively. FIGURE 5.23 illustrates the extracted control areas of the *supernodes* on the blade surface, demonstrating qualitatively the deeper point-reduction succeeded with this

agglomeration approach. In the same figure the error on the deformed surface (front and back) is depicted.



**FIGURE 5.23:** Control areas of supernodes (a) and error distribution on front (b) and back (c) blade surface.

In TABLE 5.7 the number of the extracted blade surface nodes at each fusion level is presented, while TABLE 5.8 includes the obtained surface errors (average and maximum) on the deformed blade along with the corresponding computational times for the solution and evaluation stages, considering the coarsest agglomeration level only. Once more, a trade-off relation between accuracy and efficiency is demonstrated, as the approach with the loose constraints derived a more efficient solution but with larger errors. Nevertheless, the accuracy is assumed to be at an acceptable level for such a test case.

**TABLE 5.7:** Number of surface nodes at each agglomeration level employing the approach with the loose constraints (NREL case).

<i>Agglomeration level</i>	<i>Number of blade surface nodes</i>
<i>Initial</i>	173,656
<i>Level 1</i>	48,132
<i>Level 2</i>	14,078
<i>Level 3</i>	5,083
<i>Level 4</i>	2,751

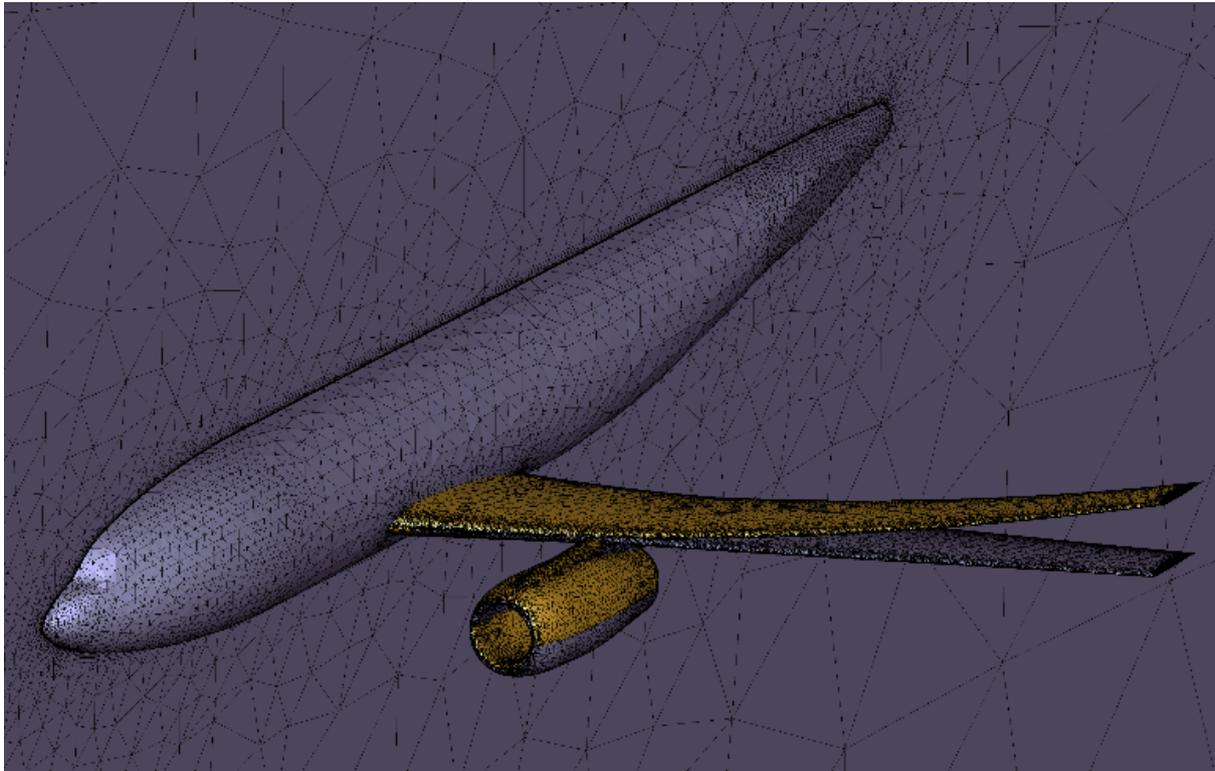
**TABLE 5.8:** Computation time and surface errors employing the coarsest agglomeration level, extracted via the approach with the strict and loose constraints (NREL case).

Agglomeration Level	Solution stage (sec)	Evaluation stage (sec)	Average Error (m)	Maximum Error (m)
4				
Strict Constraints	334.4	968	$9.96 \times 10^{-7}$	$6.10 \times 10^{-5}$
Loose Constraints	1.47	91.9	$1.05 \times 10^{-5}$	$3.16 \times 10^{-4}$

### 5.2.3.3 DLR-F6

The third problem encountered, concerns the deformation of the wing of the DLR-F6 aircraft model in a Wing-Body-Nacelles-Pylons (WBNP) configuration (Luo *et al.*, 2005; Lygidakis & Nikolos, 2014c). DLR-F6 represents a twin engine wide body aircraft of Airbus type, specifically designed for the validation of CFD codes; it was examined in the second AIAA Drag Prediction Workshop (DPW) (Luo *et al.*, 2005; Lygidakis & Nikolos, 2014c). The utilized in this study grid was provided from the site of the NASA Langley Research Center (LARC), at which most of the meshes used by the participants of the aforementioned DPW are available (Luo *et al.*, 2005). It is composed of 2,287,980 nodes, 786,755 tetrahedrons, 4,242,674 prisms and 6,642 pyramids, while it includes only the half configuration for computational savings (Luo *et al.*, 2005).

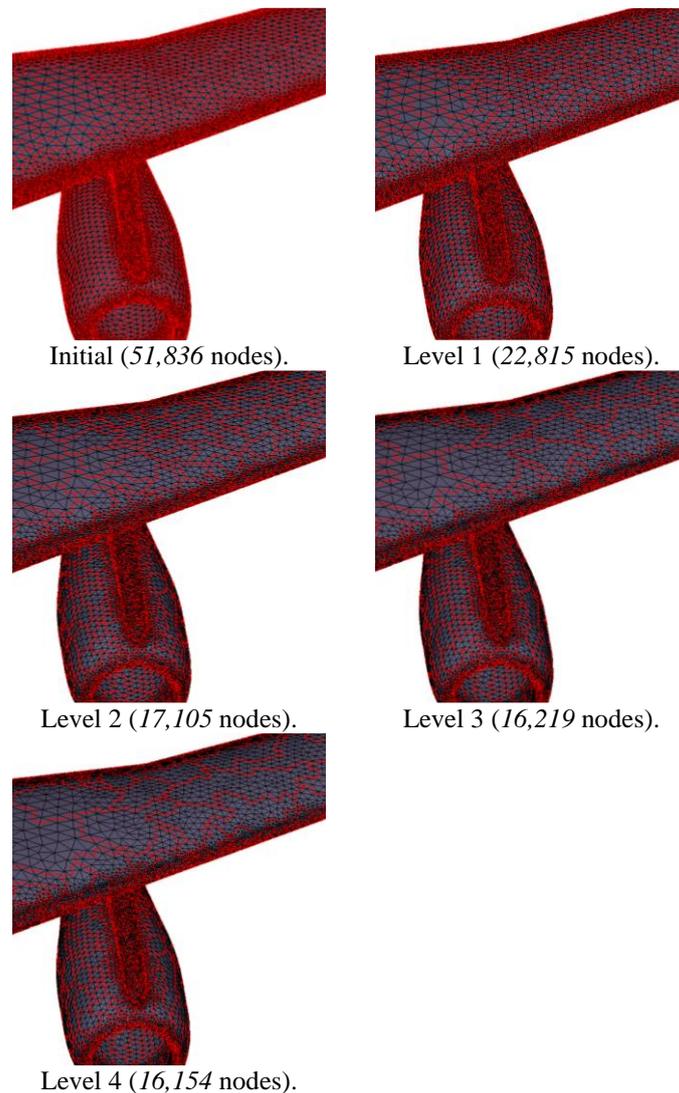
The deformation is implemented by performing parabolic deflection of aircraft wing with maximum value 6 m at the wing tip (FIGURE 5.24).



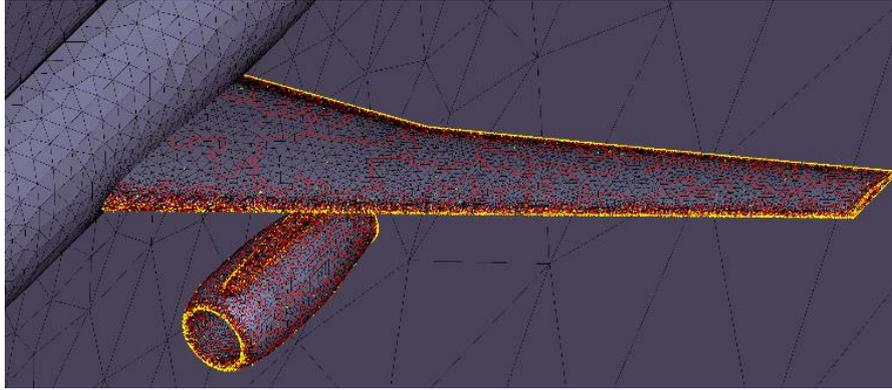
**FIGURE 5.24:** Parabolic deformation of aircraft wing (max deflection 6 m at wing tip) (DLR-F6 case).

For the assessment of the proposed methodology four coarser resolutions were generated via agglomeration of their surface nodes. In FIGURE 5.25 the boundary control areas of the corresponding nodes and supernodes are illustrated for the initial and four coarser grids respectively; those are the nodes that will be used as centers of RBF's for the implementation of the developed deformation procedure.

FIGURE 5.26 depicts the distribution of the selected base points (yellow-colored) on the wing surface at the coarsest resolution. Most of them are located in regions with high curvatures or slope discontinuities, due to the constraints imposed for the agglomeration strategy; in that way the initial shape of the geometry is preserved, regardless the significant reduction of the number of surface nodes.



**FIGURE 5.25:** Mesh density on the wing surface of the initial and agglomerated control area grids (DLR-F6 case).



**FIGURE 5.26:** Super-areas' centers (yellow-colored) at the coarsest generated grid and their distribution on the wing surface (DLR-F6 case).

In order to validate the accuracy of the proposed methodology, the average and maximum errors were computed on the deformed surfaces; they are defined as the Euclidean distance of each point on these deformed surfaces from their ideal positions calculated using the parabolic expression  $\Delta y = \alpha x^2$ . TABLE 5.9 contains the corresponding results for all the encountered runs (with different number of employed agglomeration levels), while FIGURE 5.27 illustrates the corresponding distributions of errors on the wing surface. The efficiency improvement, gained with the agglomerated grids, is presented in TABLE 5.10 and FIGURE 5.28, along with the obtained computational times for the solution and evaluation stages, compared with those for the initial level. It should be noted that the wall-clock time, required to construct the four coarsest resolutions, is relatively negligible compared to this of the interpolation process.

**TABLE 5.9:** Surface errors for different agglomeration levels (DLR-F6 case).

<i>Agglomeration Level</i>	<i>Number of Nodes</i>	<i>Average Error (m)</i>	<i>Maximum Error (m)</i>
<i>Initial</i>	51,836	-	-
<i>Level 1</i>	22,815	$3.69 \times 10^{-7}$	$1.08 \times 10^{-4}$
<i>Level 2</i>	17,105	$1.85 \times 10^{-6}$	$3.38 \times 10^{-4}$
<i>Level 3</i>	16,219	$7.58 \times 10^{-6}$	$7.36 \times 10^{-4}$
<i>Level 4</i>	16,154	$1.12 \times 10^{-5}$	$1.15 \times 10^{-3}$

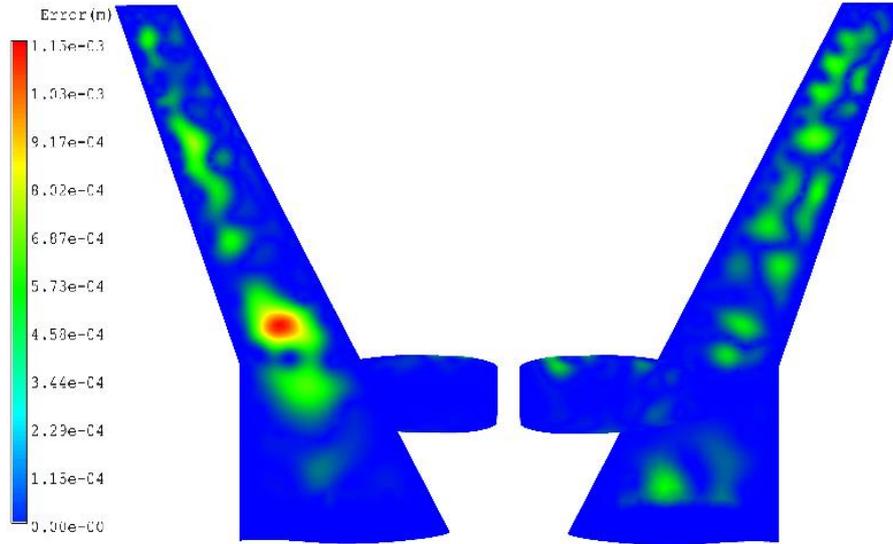
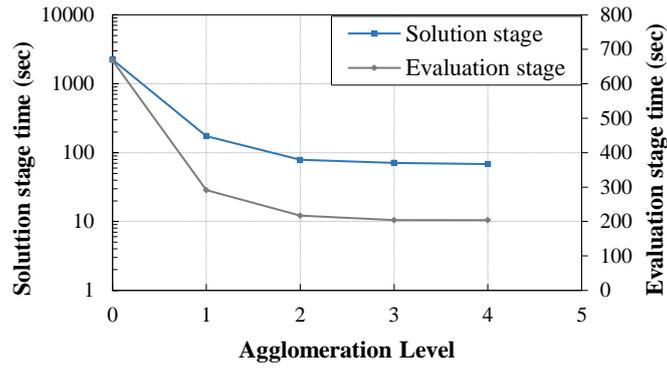


FIGURE 5.27: Distribution of (deformed wing) surface error at the fourth agglomeration level (top and bottom views) (DLR-F6 case).

TABLE 5.10: Wall-clock computation time for different agglomeration levels (DLR-F6 case).

Agglomeration Level	Solution stage		Evaluation stage	
	(sec)	(%) of initial	(sec)	(%) of initial
<i>Initial</i>	2234.8	-	667.2	-
<i>Level 1</i>	174.4	7.8	291.6	43.7
<i>Level 2</i>	78.8	3.5	216.8	32.5
<i>Level 3</i>	70.9	3.1	204.4	30.6
<i>Level 4</i>	68.2	3.0	203.9	30.6

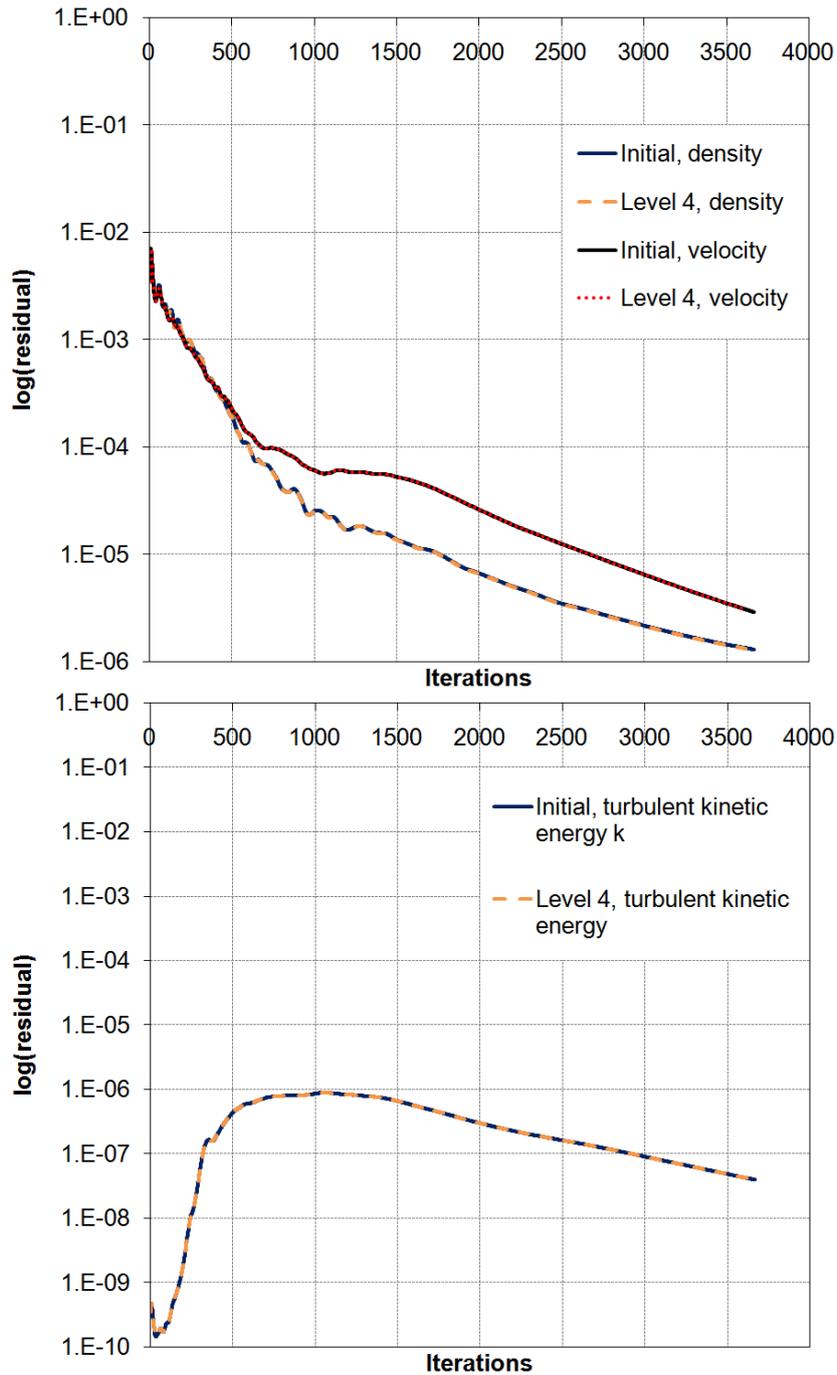
The obtained results reveal the significant acceleration, achieved by the proposed deformation procedure, without downgrading though the accuracy of the derived meshes and despite the large imposed value of deflection (6 m). Moreover, no degenerate elements were produced, as the deformation procedure remains independent of the connectivities between the control cells; connectivity-based approaches are prohibited frequently by the aforementioned problem, especially in areas with high aspect ratio cells. The maximum errors are observed in regions with large control areas, i.e., in regions with small curvature, having small impact on flow computation.



**FIGURE 5.28:** Computation time of the solution and evaluation stages for different agglomeration levels (DLR-F6 case).

To assess the differences between the obtained deformed grids (initial one and the fourth agglomeration level), as well as their quality, fully turbulent flow over the deformed DLR-F6 WBNP aircraft was examined with the academic CFD solver *Galatea* (Lygidakis & Nikolos, 2014c, 2015). Considering the dimensionless formulation of the aforementioned code, the utilized grids were re-dimensionalized in order the mean aerodynamic chord to become equal to unity (Lygidakis & Nikolos, 2014c). The Mach and Reynolds numbers of the free-stream flow, were set equal to 0.75 and  $3.0 E + 6$  respectively, while its angle of attack equal to  $0.8^\circ$  (Lygidakis & Nikolos, 2014c). Turbulence modelling was obtained with the Shear Stress Transport (SST) model, while no transition area was assumed (Lygidakis & Nikolos, 2015). For the computation of the inviscid fluxes a second-order spatial accurate scheme, coupled with the Min-mod limiter, was employed, while for this of the viscous ones a nodal-averaging scheme was applied (Lygidakis & Nikolos, 2015). For the relaxation of the governing equations, and consequently the approximation of the final steady-state solution, a second-order accurate in time four-stage Runge-Kutta method (RK(4)) was implemented with a CFL number equal to unity (Lygidakis & Nikolos, 2015). For parallel processing on the aforementioned computational system the utilized grids were decomposed in eight partitions, while a three-level agglomeration multigrid approach was implemented for additional acceleration of the solution procedure (Lygidakis & Nikolos, 2014c, 2015).

FIGURE 5.29 includes the convergence history per number of iterations (multigrid V-cycles) for density, velocity in  $x$ -direction (top) and turbulent kinetic energy (bottom), for both the deformed meshes, extracted with (Level 4) and without (Initial) agglomeration of surface nodes. The absence of any difference between the corresponding curves indicates the capability of the proposed methodology for the generation of equally accurate deformed grids but in highly reduced time, comparing to those obtained without agglomeration.

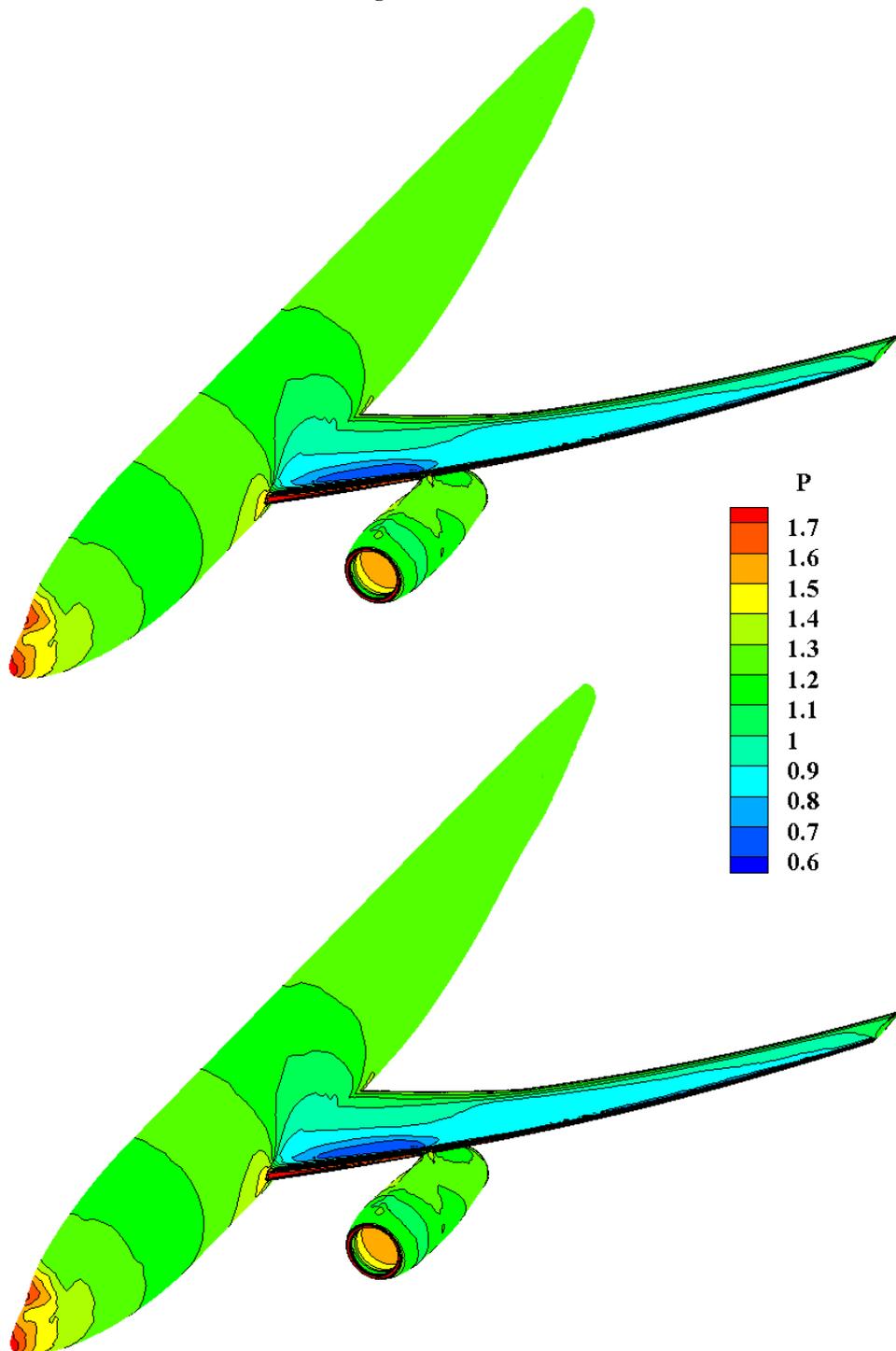


**FIGURE 5.29:** Convergence history curves for density, velocity in  $x$ -direction and turbulent kinetic energy of deformed grids (initial and 4th agglomeration level) (DLR-F6 case).

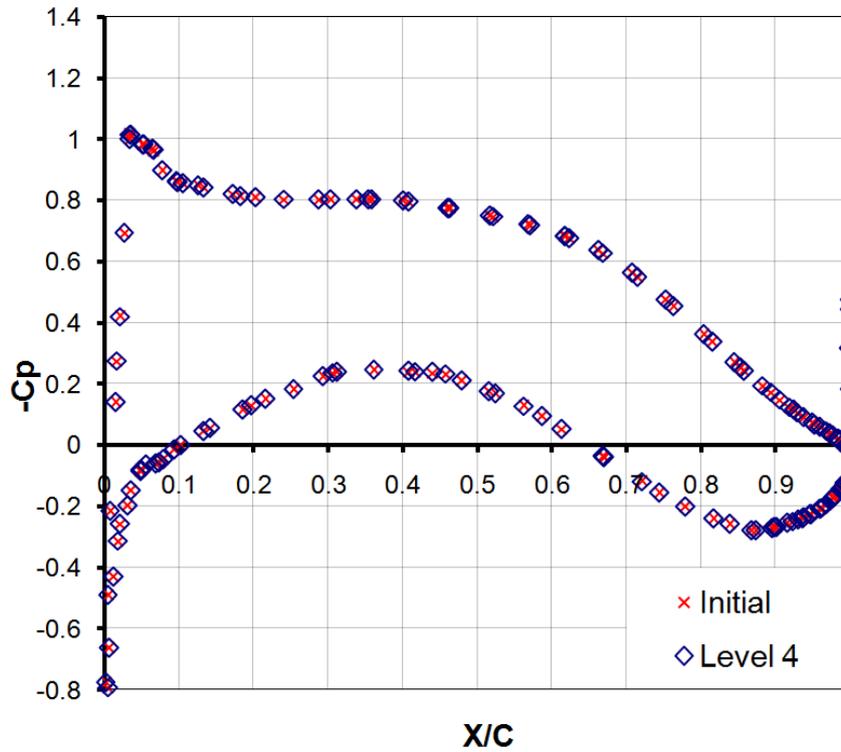
FIGURE 5.30 illustrates the corresponding dimensionless pressure contours on the surface of the aircraft; once more no difference can be identified between the utilized grids. Finally, in FIGURE 5.31 and FIGURE 5.32 the quantitative confirmation of the previous qualitative results is contained via the corresponding distributions of pressure coefficient at two different wing span-wise sections (63.8% and 84.7%). As expected, no difference was observed again between the employed grids, confirming the potential of the proposed method for simultaneously accurate and efficient generation of deformed grids.

Furthermore, in order to study the influence of the grid size on the proposed methodology, a finer grid was utilized, constituting of 3,376,236 nodes, 867,432 tetrahedrons, 6,358,494 prisms and 6,642 pyramids. It was generated via the implementation of an h-refinement approach (Lygidakis & Nikolos, 2015) to the elements at wing sections greater equal to  $\sim 70\%$ .

Similarly to the previous case four coarsest resolutions were generated by the proposed agglomeration strategy. In TABLE 5.11 the surface errors on the deformed wing are presented, while in FIGURE 5.33 the distribution of the error over the deformed surface is illustrated. The derived computation times are presented in TABLE 5.12 and FIGURE 5.34 for both the solution and evaluation stages.



**FIGURE 5.30:** Contours of dimensionless pressure on the aircraft surface of deformed grids (initial and 4th agglomeration level) (DLR-F6 case).



**FIGURE 5.31:** Distribution of pressure coefficient at wing span-wise section 63.8% of deformed grids (initial and 4th level) (DLR-F6 case).

Once more the larger errors are identified in regions with small variations in curvature, while their values remain at the same acceptable levels. Considering the required wall-clock computation times makes quite obvious that greater acceleration has been obtained with the refined mesh. Similarly to multigrid schemes in CFD, the acceleration increases with the corresponding increase to the grid size; the greater the number of surface nodes, the greater is the acceleration gained.

A similar to the initial non-refined grid evaluation was performed using *Galatea* solver (Lygidakis & Nikolos, 2014c, 2015). FIGURE 5.35 illustrates the obtained dimensionless pressure contours on the surface of the aircraft, while FIGURE 5.36 includes the corresponding distributions of pressure coefficient at wing span-wise section 84.7%. No difference can be identified between the results of the deformed refined grids, extracted with (Level 4) and without (Initial) agglomeration of surface points. Thus, the potential of the proposed method for accurate and efficient generation of deformed meshes is demonstrated once more; nevertheless, its efficiency was shown to get higher with increasing the grid size.

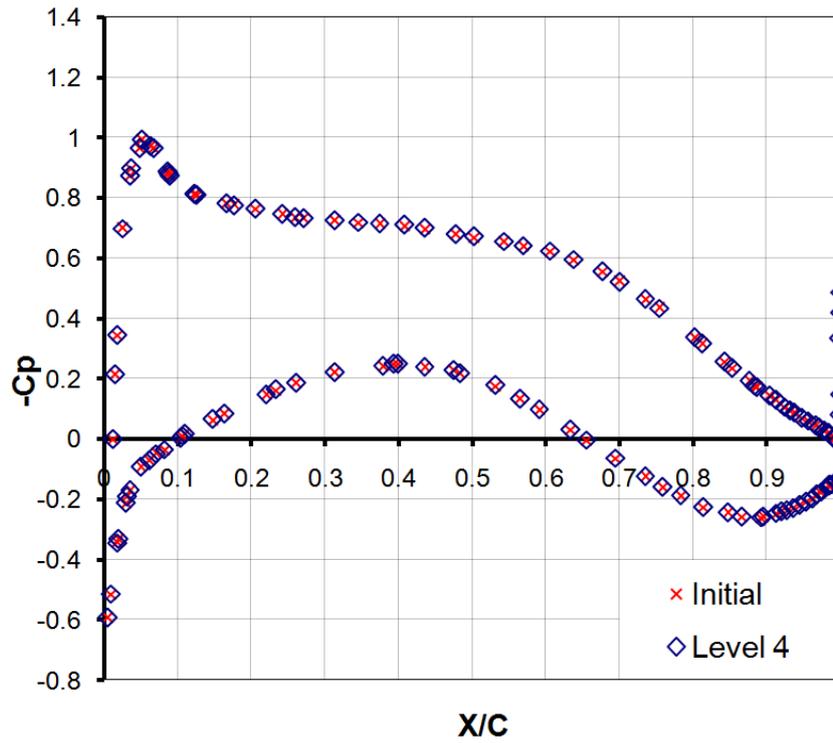


FIGURE 5.32: Distribution of pressure coefficient at wing span-wise section 84.7% of deformed grids (initial and 4th level) (DLR-F6 case).

TABLE 5.11: Surface errors of the refined grid for different agglomeration levels (DLR-F6 case).

Agglomeration Level	Number of Nodes	Average Error (m)	Maximum Error (m)
<i>Initial</i>	82,062	-	-
<i>Level 1</i>	32,762	$4.90 \times 10^{-7}$	$1.08 \times 10^{-4}$
<i>Level 2</i>	22,431	$1.17 \times 10^{-6}$	$1.77 \times 10^{-4}$
<i>Level 3</i>	20,356	$3.63 \times 10^{-6}$	$4.24 \times 10^{-4}$
<i>Level 4</i>	20,352	$8.41 \times 10^{-6}$	$6.14 \times 10^{-4}$

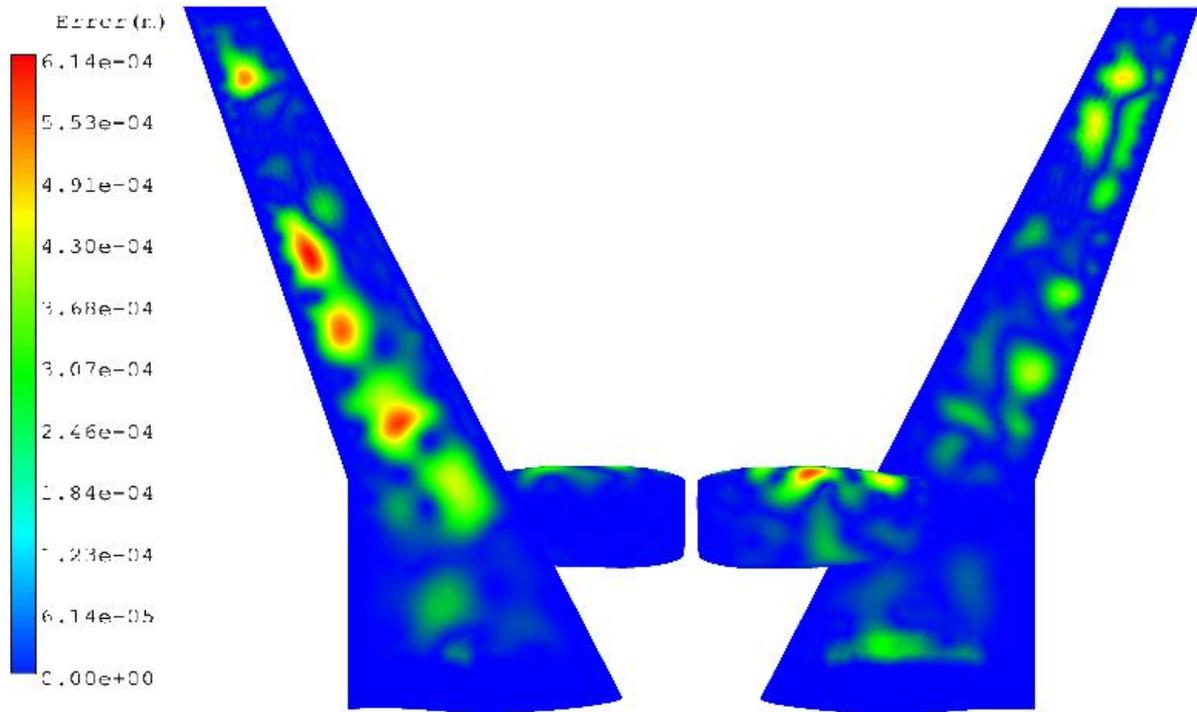


FIGURE 5.33: Distribution of wing surface error (top and bottom views) at the fourth agglomeration level (refined grid) (DLR-F6 case).

TABLE 5.12: Wall-clock computation time for different agglomeration levels (refined grid) (DLR-F6 case).

Agglomeration Level	Solution stage		Evaluation stage	
	(sec)	(%) of initial	(sec)	(%) of initial
Initial	10315.6	-	1345.2	-
Level 1	474.7	4.6	577.3	42.9
Level 2	159.9	1.6	398.2	29.6
Level 3	139.5	1.4	370.6	27.5
Level 4	122.8	1.2	362.9	27.0

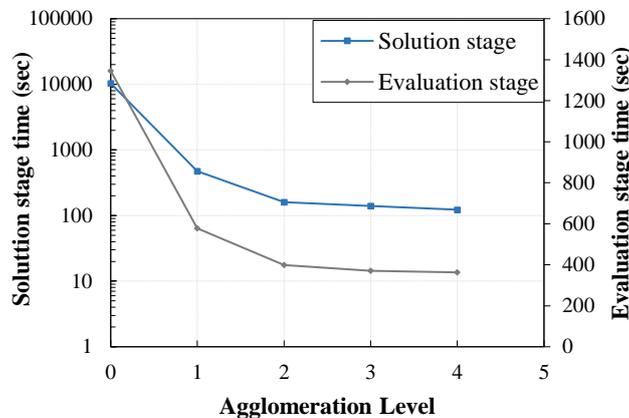
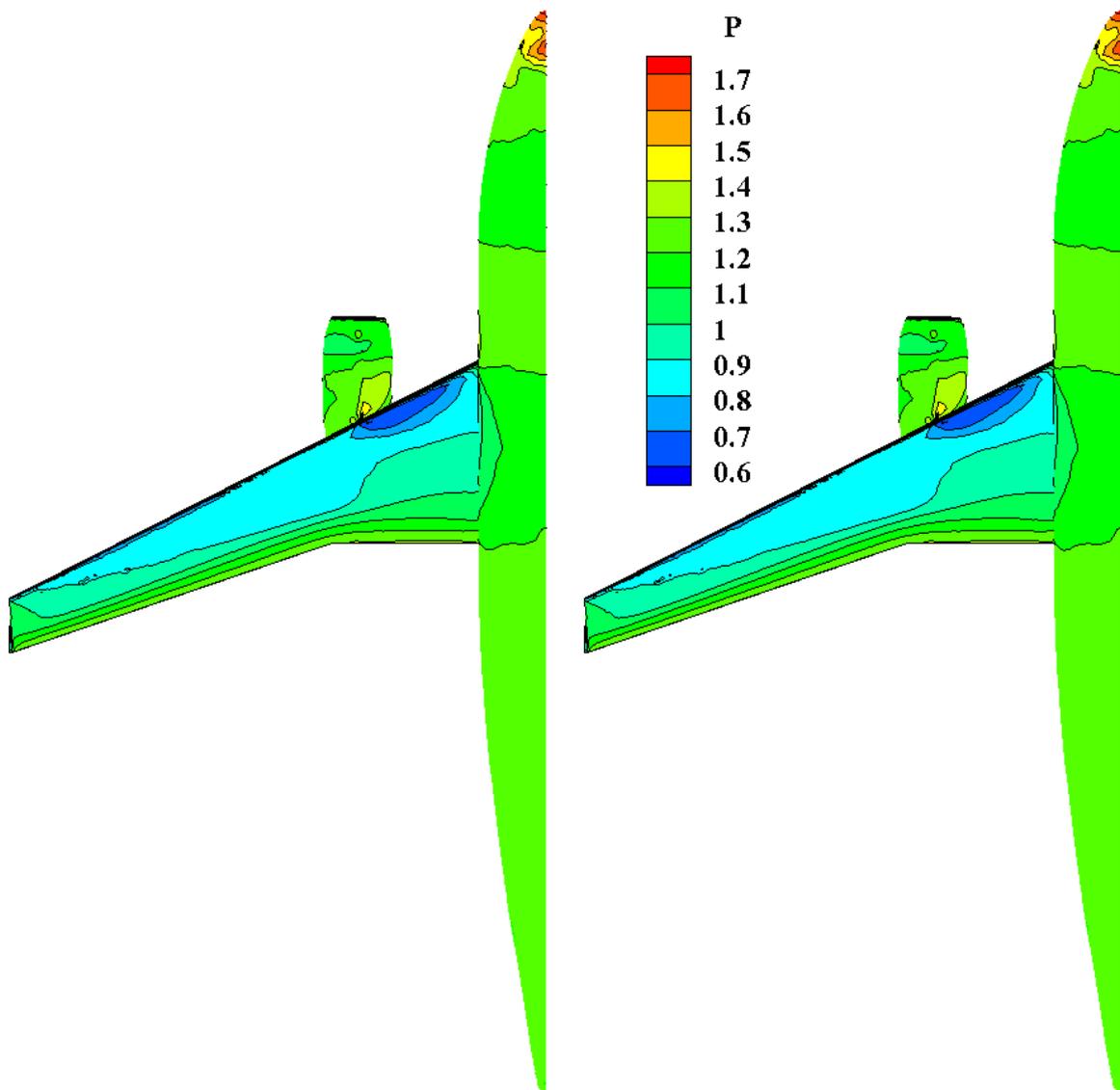
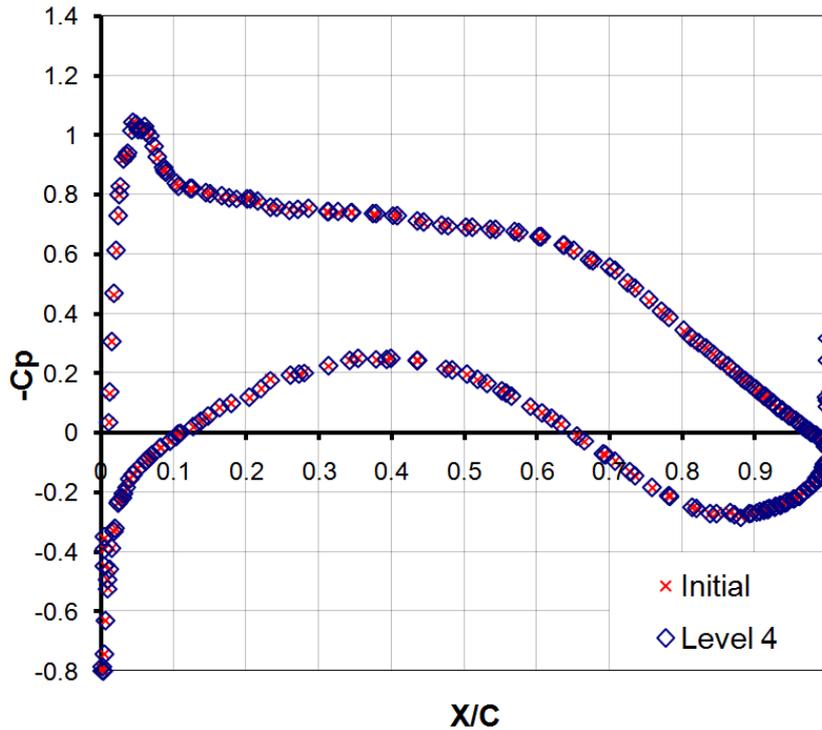


FIGURE 5.34: Computation time of the solution and evaluation stages for the refined grid (at each agglomeration level) (DLR-F6 case).



**FIGURE 5.35:** *Contours of dimensionless pressure on the aircraft surface of deformed refined grids (initial and 4th level) (DLR-F6 case).*



**FIGURE 5.36:** Distribution of pressure coefficient at wing span-wise section 84.7% of deformed refined grids (initial and 4th level) (DLR-F6 case).

### 5.3 Conclusions

In this chapter, a methodology for the acceleration of an RBF-based mesh deformation procedure via the reduction of the surface points was proposed. An agglomeration strategy was used to this end, modified from an already developed one for a multigrid methodology applied in a CFD code (Lygidakis & Nikolos, 2014b; Lygidakis et al., 2016). As to the authors' best knowledge, although agglomeration multigrid techniques are well-established in numerical simulations, the corresponding fusion strategy has not been used so far with mesh-deformation algorithms.

Agglomeration is performed on a topology-preserving framework, resembling the advancing front technique, as it begins from regions including surface discontinuities (two or more boundary-condition-type closures, one or more boundary slope discontinuities, etc.) and extends to more smooth ones. It was demonstrated to be a highly efficient scheme; it requires no more than two seconds to generate four coarser resolutions of a surface including approximately 200,000 nodes. Moreover, significant improvements to the computational performance of the mesh-deformation algorithm can be succeeded, especially for large grid sizes. Besides improving efficiency of the RBF-based mesh deformation technique, the aforementioned scheme was revealed to preserve the grid quality, despite the significant decrease of the number of surface nodes, used as RBF-centers. Due to the incorporated constraints, the nodes near the edges of the deforming surface are preserved during the fusion process, outlining the studied geometry and maintaining its topology unaffected. However, depending on the examined test case, some of the aforementioned limitations can be applied in a looser way, increasing further the efficiency of the proposed approach without significantly reducing its accuracy. Based on the presented validation results, the proposed methodology appears to significantly enhance the mesh deformation algorithm in terms of

computational efficiency, while maintaining simultaneously its accuracy at a competitive level. Indicative CFD results on the deformed grids produced with and without the proposed agglomeration methodology revealed no difference at all.

The development of an agglomeration methodology for hybrid unstructured grids is not a trivial task. However, if such a methodology has been already developed for a multigrid procedure, its modification for mesh deformation acceleration, as it is proposed in this thesis, can be accomplished in a straightforward manner. The gains of such an implementation proved to be impressive in terms of reduction of the computational cost of the mesh deformation methodology based on RBFs.

# Chapter 6: Fluid Structure Interaction methodology

## 6.1 Introduction

In this chapter, the development of a partitioned FSI coupling procedure is reported, aiming to facilitate interaction between an open-source CSD (Computational Structural Dynamics) and an in-house academic CFD (Computational Fluid Dynamics) code. Attention is mainly directed towards the efficient and accurate transfer of predicted displacements, velocities (by CSD) and loads (by CFD). More precisely, spatial coupling is achieved using Radial Basis Functions (RBFs) interpolation, which enables point-based interaction, needing therefore no information for connectivities and, consequently, allowing for the utilization of different type or even intersecting structural and flow grids. Although RBFs method seems to be particularly attractive for both data transfer and mesh deformation, it suffers from a significant drawback, as it has been pointed out in the previous Chapter; it calls for relatively excessive memory and computation time requirements (in its original formulation). In case of data transfer the Partition of Unity (PoU) approach is adopted as a remedy of the aforementioned deficiency, which regards the decomposition of the examined problem into several smaller ones, to be solved independently and hence more efficiently. In mesh deformation though, improvement of computational performance is succeeded with the surface point reduction technique, already described in Section 5.2.2. The obtained numerical results confirm its potential for such simulations, highlighting additionally the radically improved computational performance of data transfer and grid deformation procedures.

The structure of the present chapter is as follows: In Section 6.2 the coupling procedure, concerning exchange of displacements and forces, is described in detail, while in Section 6.3 the incorporated mesh deformation process is outlined. Section 6.4 includes a brief description of the flow and structural models as well as the coupling software. In Section 6.5 the validation results of the proposed methodology over 3 benchmark test cases are presented, while Section 6.6 contains some conclusions, based on the aforementioned results.

## 6.2 Coupling procedure

The spatial coupling procedure can be mainly divided in two stages, namely, (a) the transfer of the displacement vectors from the structural to the flow grid and (b) the transfer of nodal forces from the flow grid back to the structural one. The coupling scheme, for both aforementioned steps, can be expressed in a unified fashion by defining a coupling matrix  $\mathbf{H}$ , which associates the interface boundary nodes of the two grids (flow and structural). Then, interpolation of the displacements can be performed according to:

$$d_s = \mathbf{H}d_f \quad (6.1)$$

where,  $d_s$  and  $d_f$  denote the displacements at the structural and flow interface side, respectively. For forces' association, the transpose matrix  $\mathbf{H}^T$  is used as

$$f_s = \mathbf{H}^T f_f, \quad (6.2)$$

where  $f_s$  and  $f_f$  are the forces acting on the structural and flow sides of the interface, respectively. With the adoption of this scheme the virtual work can be effectively conserved over the interface (Hounjet & Meijer, 1995; Beckert, 2000).

However, the coupling procedure has to satisfy also conservation of total force and moment, which depends strongly on the choice of the coupling matrix  $\mathbf{H}$ , and hence on the corresponding method applied to obtain it.

### 6.2.1 Data transfer scheme

In this thesis, the RBFs technique is implemented to construct the interpolation matrix, which is used for exchanging information between the structural and the flow solver. As already mentioned in Section 5.3, the interpolation function is defined as:

$$s(\mathbf{x}) = \sum_{i=1}^{n_c} a_i \Phi(\|\mathbf{x} - \mathbf{x}_{c_i}\|) + p(\mathbf{x}) \quad (6.3)$$

where  $a_i$  are the weight coefficients,  $p(\mathbf{x})$  a polynomial, and  $\mathbf{x}_{c_i} = [x_{c_i}, y_{c_i}, z_i]$  the RBFs centers, e.g., the coordinates of the boundary mesh nodes;  $n_c$  is the number of those centers and  $\Phi$  is the basis function. The use here of the linear polynomial term ensures the exact recovery of translations and rotations (Beckert & Wendland, 2001). The polynomial term in  $x$  direction can be expressed as follows

$$p^x(\mathbf{x}_{c_i}) = \gamma_0^x + \gamma_1^x x + \gamma_2^x y + \gamma_3^x z \quad (6.4)$$

while the coefficients  $a_i$  and  $\gamma_i$  are computed by interpolation conditions, fulfilling equation (6.3) requirements at the centers of the basis functions, along with an additional condition, described as

$$s(\mathbf{x}_{c_i}) = \mathbf{U}_{c_i} = [u_{c_i}^x, u_{c_i}^y, u_{c_i}^z]^T \quad (6.5)$$

$$\sum_{i=1}^{n_c} a_i q(\mathbf{x}_{c_i}) = 0 \quad (6.6)$$

where  $\mathbf{U}_{c_i}$  are the known boundary displacements of the structural side of the interface.

To construct the coupling matrix  $\mathbf{H}$ , the structural mesh displacements of the boundary surfaces are used; combining equations (6.5) and (6.6), the coefficient vectors  $\mathbf{a}^{x,y,z}$  and  $\mathbf{b}^{x,y,z}$  can be evaluated by solving the following linear system in each spatial direction (expressed in matrix notation in the  $x$  direction)

$$\begin{bmatrix} \mathbf{U}_x \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{M}_{cc} & \mathbf{P}_c \\ \mathbf{P}_c^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a}^x \\ \mathbf{b}^x \end{bmatrix} = \mathbf{C}_{cc} \begin{bmatrix} \mathbf{a}^x \\ \mathbf{b}^x \end{bmatrix}, \quad (6.7)$$

where,

$$\mathbf{a}^x = \begin{bmatrix} a_{c_1}^x \\ a_{c_2}^x \\ \vdots \\ a_{c_n}^x \end{bmatrix}, \quad \mathbf{b}^x = \begin{bmatrix} \gamma_0^x \\ \gamma_1^x \\ \gamma_2^x \\ \gamma_3^x \end{bmatrix} \quad (6.8)$$

and

$$\mathbf{P}_c = \begin{bmatrix} 1 & x_1 & y_1 & z_1 \\ 1 & x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & x_{c_n} & y_{c_n} & z_{c_n} \end{bmatrix}, \quad \mathbf{M}_{cc} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \cdots & \Phi_{1c_n} \\ \Phi_{21} & \Phi_{22} & \cdots & \Phi_{2c_n} \\ \vdots & \vdots & \vdots & \vdots \\ \Phi_{c_n1} & \Phi_{c_n2} & \cdots & \Phi_{c_nc_n} \end{bmatrix} \quad (6.9)$$

with  $\Phi_{ij} = \Phi(\|x_i - x_j\|)$ .

The required approximation of displacements at the flow side of the interface are calculated by applying the following equation for each spatial direction:

$$s(x_a) = \mathbf{A}_{ac} \mathbf{C}_{cc}^{-1} x_c = \mathbf{H} x_c, \quad (6.10)$$

where,

$$\mathbf{A}_{ac} = \begin{bmatrix} 1 & x_{\alpha_1} & y_{\alpha_1} & z_{\alpha_1} & \Phi_{\alpha_1 c_1} & \Phi_{\alpha_1 c_2} & \cdots & \Phi_{\alpha_1 c_n} \\ 1 & x_{\alpha_2} & y_{\alpha_2} & z_{\alpha_2} & \Phi_{\alpha_2 c_1} & \Phi_{\alpha_2 c_2} & \cdots & \Phi_{\alpha_2 c_n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_{\alpha_{N_\alpha}} & y_{\alpha_{N_\alpha}} & z_{\alpha_{N_\alpha}} & \Phi_{\alpha_{N_\alpha} c_1} & \Phi_{\alpha_{N_\alpha} c_2} & \cdots & \Phi_{\alpha_{N_\alpha} c_n} \end{bmatrix} \quad (6.11)$$

with  $a_i$  being the  $i$ -th flow mesh node of the boundary and  $N_\alpha$  the total number of boundary nodes of the same grid. If the following term is additionally set

$$\tilde{\mathbf{M}}_{cc} = (\mathbf{P}_c \mathbf{M}_{cc}^{-1} \mathbf{P}_c^T)^{-1} \quad (6.12)$$

then the coupling matrix  $\mathbf{H}$  can be expressed as:

$$\mathbf{H} = \mathbf{A}_{ac} \begin{bmatrix} \tilde{\mathbf{M}}_{cc} \mathbf{P}_c \mathbf{M}_{cc}^{-1} \\ \mathbf{M}_{cc}^{-1} - \mathbf{M}_{cc}^{-1} \mathbf{P}_c^T \tilde{\mathbf{M}}_{cc} \mathbf{P}_c \mathbf{M}_{cc}^{-1} \end{bmatrix} \quad (6.13)$$

Following the coupling process described above, it is evident that the computation of the coupling matrix is obtained by solving the linear systems described in equation (6.7), while the interpolation of displacements and forces from the one mesh to the other involve only matrix-vector multiplications. However, the method suffers from excessive computational and memory requirements, especially when large-scale problems are encountered, since the coupling matrix  $\mathbf{H}$  is of size  $N_s \times N_\alpha$ , where  $N_s$  denotes the number of the structural mesh surface nodes and  $N_\alpha$  the number of the flow mesh surface nodes. Computational efficiency is further degraded due to the necessity to invert the  $(n_c + 4) \times (n_c + 4)$  matrix  $\mathbf{C}_{cc}$ . Even though the boundary node coarsening technique presented in Section 5.3.2 seems to be particularly attractive for the mesh deformation process, it is considered inappropriate for the data transfer one, as it might result in loss of information and, as such, jeopardize the requirement of force conservation over the interface.

Another approach, also focusing on the efficiency improvement of the method, is the Partition of Unity (PoU) scheme, according to which the initial problem is divided into several smaller ones by grouping base points into sets to be processed separately; it resembles the domain decomposition approach, employed for the parallelization of complex numerical methods. It has been successfully implemented against FSI problems in (Ahrem *et al.*, 2006; Rendall & Allen, 2009c) with very promising results. Besides reducing computational cost, it avoids the undesirable influence of support radius into the interpolation process. Considering these tempting features, this technique (analyzed in the next Section 6.2.2) was selected to be employed for data transfer in this dissertation.

## 6.2.2 Efficiency improvement - Partition of Unity (PoU)

The basic concept of the PoU method regards the decomposition of the initial problem into several smaller ones to be solved independently, while the global solution arises by a weighted combination of the local ones. Given this condition, the term “unity” refers to the summation of the weighting coefficients of the entire domain that must be equal to one. The process begins by dividing the set  $\Omega$  of the boundary nodes, including the nodes of the interface of both computational meshes (structural and flow) into a  $K$  number of mutually overlapping regions, in a way to ensure that each  $\Omega_j$  patch includes only a small number of points. For each of these patches a local displacement is computed by solving a local interpolation problem with the RBFs method. Subsequently, the global interpolation is obtained by summing all local interpolations, weighted by their respective coefficients, as

$$s(\mathbf{x}) = \sum_{j=1}^K w_j(\mathbf{x}) s_j(\mathbf{x}), \quad \sum_{j=1}^K w_j(\mathbf{x}) = 1 \quad (6.14)$$

where  $w_j(\mathbf{x})$  represents a non-negative weighting function, which comes into effect only for its associated patches  $\Omega_j$ , while it is zeroed for the rest. Using the transpose of the global matrix, formed by the local interpolants, assures that the conservation criteria are also met in the PoU approach (Ahrem *et al.*, 2006).

A variation of the pre-described method, proposed in (Rendall & Allen, 2009c), concerns the application of the PoU method on each boundary flow node separately, instead of decomposing the set of nodes into overlapping patches. Following this strategy, the complex step of decomposing the structural grid into overlapping regions is evaded, and thus the possibility of facing discontinuities between the patches due to the partitioning process is diminished. In order to calculate the local interpolants, an association of each flow mesh node with a pre-defined number of its closest structural boundary ones has to be defined. Therefore, the use of sophisticated data structures appears to be mandatory in order to link the boundary nodes of the two grids (flow and structural) in an efficient manner. Octree data structures are a common tactic for addressing such kind of problems. After the definition of the local interpolants, their merging back to a global one is essential so as to avoid distortion and non-realistic deformations, which may occur in applications involving computational meshes with different resolutions. In addition, the implementation of such a local strategy ameliorates the force distribution over the interface; the area of influence of each flow node is limited only to its nearby ones of the structural grid, unlike the corresponding area used with the complete method.

## 6.3 Mesh deformation

The formulation of RBFs in case of mesh motion is different from this of data transfer, since the two problems have different requirements. In case of data transfer the inclusion of polynomial terms of at least first order is mandatory to ensure conservation of total force and moment, while for mesh deformation these functions are omitted to avoid its undesirable global coverage, which can cause the whole domain to move rather than deform. Hence, the dependence matrix  $\mathbf{C}_{cc}$  is reduced to  $\mathbf{M}_{cc}$ , and the weighting coefficients are obtained via the following expression

$$\mathbf{a}^x = \mathbf{M}_{cc}^{-1} \mathbf{x}_{c_i} \quad (6.15)$$

with  $\mathbf{a}^{x,y,z}$  denoting the vector containing the scalar coefficients in each direction and  $\mathbf{M}_{cc}$  the matrix defined in equation (6.9).

The required displacements of the volume mesh nodes are calculated according to those of the boundary points. As such, the deformation of the volume mesh can be computed as follows

$$\mathbf{s}(x_a) = \mathbf{A}_{cv} \mathbf{M}_{cc}^{-1} \mathbf{x}_{c_i} = \mathbf{H} \mathbf{x}_{c_i} \quad (6.16)$$

where index  $v$  represents the volume nodes of the flow grid, and  $\mathbf{A}_{cv}$  a  $n_v \times n_c$  matrix, described as:

$$\mathbf{A}_{cv} = \begin{bmatrix} \Phi_{11} & \Phi_{12} & \cdots & \Phi_{1n_c} \\ \Phi_{21} & \Phi_{22} & \cdots & \Phi_{2n_c} \\ \vdots & \vdots & \vdots & \vdots \\ \Phi_{n_v 1} & \Phi_{n_v 2} & \cdots & \Phi_{n_v n_c} \end{bmatrix} \quad (6.17)$$

## 6.4 Computational software

### 6.4.1 Flow Solver

#### 6.4.1.1 *Galatea-I*

The simulation of unsteady incompressible flow phenomena is performed with the use of a recently developed academic solver, named *Galatea-I* (Lygidakis *et al.*, 2014, 2016; Sarakinos *et al.*, 2014), which employs the Navier-Stokes equations, coupled with the artificial compressibility method (Chorin, 1997; Kim & Menon, 1999; Kallinderis & Ahn, 2005). The discretization of the governing equations is succeeded with a node-centered finite-volume scheme on unstructured hybrid computational grids, comprised of tetrahedra, prisms and pyramids (Lygidakis *et al.*, 2014, 2016). The calculation of the inviscid fluxes is performed with the Roe's approximate Riemann solver (Roe, 1981), along with a second-order spatial accurate scheme based on the MUSCL approach (Blazek, 2001). For the calculation of the viscous fluxes the gradients of velocity components have to be evaluated at the interfaces of the control cells. Two methodologies have been incorporated in *Galatea-I*; an element-based approach, utilizing the edge-dual control volume scheme and a simpler nodal-averaging one (Lygidakis *et al.*, 2014, 2016). For turbulence prediction the RANS (Reynolds-Averaged Navier-Stokes) approach is implemented, along with the SST (Shear Stress Transport) turbulence model (Menter, 1994). At the inlet and outlet regions characteristics-based boundary conditions are applied (Sarakinos *et al.*, 2014), while at the solid walls free-slip or no-slip conditions can be applied depending on the type of the encountered flow (inviscid or viscous). For the time accurate computation a dual-time stepping scheme has been included (Kallinderis & Ahn, 2005). According to this scheme the relaxation of the governing equations is achieved with a four-stage Runge-Kutta (RK(4)) method over pseudo-time (Lallemand, 1988b), while the time accurate solution is obtained with the use of a second-order finite-difference scheme over real-time (Kallinderis & Ahn, 2005). Besides using edge-based data structures and a local time-stepping technique (Blazek, 2001), *Galatea-I* solver has incorporated the capability of parallel processing, based on the domain decomposition approach (Lygidakis *et al.*, 2016), and the implementation of an agglomeration multigrid scheme, based on the FAS (Full Approximation Scheme) approach (Lygidakis *et al.*, 2016), for the acceleration of the

solution procedure. With the first method the initial computational grid is divided into smaller sub-domains, each of which is attributed on a single computer core (Lygidakis *et al.*, 2016). Interaction between neighboring partitions is achieved with the MPI (Message Passing Interface) library functions. With the latter methodology successively coarser grids are generated, derived by the fusion of adjacent control volumes. The relaxation of the governing equations is performed on all meshes in a V-cycle type that accelerates solution convergence (Lygidakis *et al.*, 2016).

#### 6.4.1.2 *GAST\_aero*

As an alternative, *GAST\_aero* (Riziotis & Voutsinas, 1997), was utilized for the flow simulations. The aerodynamic loads are estimated either using the standard BEM method including corrections and add-on's or the 3D free wake vortex particle method GenUVP. Both methods apply the ONERA dynamic stall model.

### 6.4.2 Structural Solver

As far as the structural computations are concerned, the open-source software package *CalculiX* (Dhondt, 2014) is utilized, which is a finite-element (FEM) analysis suite with its own pre/post processor for linear or non-linear calculations on structured or unstructured three-dimensional computational grids. Furthermore, it is able of performing different types of analyses, such as static, dynamic, modal, buckling, heat transfer, etc., in a serial or parallel computational environment; for the latter either MPI or *OpenMP* library functions can be used.

### 6.4.3 Coupling software

#### 6.4.3.1 *Implementation*

This section exposes the most important details related to the coupling process of the two solvers. It is based on the loosely-coupled partitioned approach, according to which both part solvers are handled as black boxes. Nevertheless, the modification of some core routines is unavoidable, in order to enable the information transfer between them; in general, the more efficient a program is intended, the deeper the modifications have to be. In FIGURE 6.1 the implementation of the coupling procedure, in *FSImulator* is depicted. The CFD solver, (*Galatea-I/GAST\_aero*), runs a single time along the whole simulation, avoiding recurrent reads of input files and memory allocations. This spawning scheme is not fully consistent with the black-box paradigm and a deeper intervention in the CFD codes were needed. On the contrary, *CalculiX* is spawned in each real time step; fortunately, since structural solvers use coarser grids, this feature does not burdens the process.

At the beginning of the simulation both solvers, and the coupling manager, *FSImulator*, read their corresponding input files (grids and boundary conditions). The interaction among the three software is succeeded through data exchange via appropriate text files, while the realization of the communication is based on the use of MPI (Message Passing Interface) communication protocol. Although this strategy might seem relative slow, it appears to be quite efficient, as the computational time for exchanging information is negligible compared to the one of the simulation steps.

In order to facilitate the cooperation among the 3 software, an explicit interface has been developed since: on the one hand, *Galatea-I* and *GAST\_aero* were programmed in FORTRAN

whereas *FSImulator* and *CalculiX* has been developed in C++ and C, respectively. Moreover, *Galatea-I* has a pure MPI parallelization while *CalculiX* takes advantage of the multithreaded version of the SPOOLES solver.

The whole procedure begins by running the *FSImulator*; initially, for each node of the flow grid, the closest nodes of the structural one are identified, using the Octree algorithm. For each set of structural nodes, an RBF problem is being set; this means that the corresponding coefficients of the RBF centers are calculated, for the RBF PoU, coupling procedure. This is performed only once at the beginning of the simulation.

Subsequently, the CFD solver is spawned with the desired number of processes, returning a handle to the new communicator between *FSImulator* and the CFD solver. The communicator is an abstract construct of MPI to group up processes for the communication. Within *Galatea-I*, a master-worker MPI parallelization approach had been followed, throughout the years, i.e., the master process is in charge of all input and output and therefore distributes and collects data from all other processes. Thus, only the master process communicates with *FSImulator*. In the *Galatea-I* code, right after the MPI\_INIT line MPI\_COMM\_GET\_PARENT is executed to retrieve the communicator between *Galatea-I* and *FSImulator*.

Then the FSI iterations in time are starting; at the end of each real time step, a blocking synchronous command is called from the master process of the CFD solver; this command activates a flag informing the parent software (*FSImulator*), to proceed to calculations. In the meanwhile, the processes of the CFD solver, are waiting until the computations are finished, in order to proceed to the next time step. The blocking synchronous send is adopted for all communication between the CFD solver and *FSImulator*, for safety and simplicity reasons.

The calculations performed by the *FSImulator* software, involve the transfer of the force field from the flow to the structural grid, utilizing the RBF PoU method. At this step, only multiplication of matrices are required, which is a very efficient procedure. The resulting loads for the CSD computation are exported to a text file in the required by the *CalculiX* software format.

Then, the *CalculiX* solver is spawned by the *FSImulator* software to compute the structural response with the aforementioned aerodynamic forces. The results of the CSD analysis are the displacements of the structural grids' surface nodes. These are exported at the end of the time step in text files to be imported to the *FSImulator*. Subsequently, the deformations of the CSD surface nodes are interpolated within the *FSImulator* to the CFD grid, using the inverse procedure. Finally, the CFD grid is deformed according to the deflections of the boundary, and a new flow grid is generated to be used by CFD solver, in order to perform the next step of the flow analysis.

After the completion of the calculations of the *FSImulator* software for the current iteration, a blocking synchronous command is called, letting the CFD solver to proceed to the next time step. The procedure is repeated until the end of simulation time.

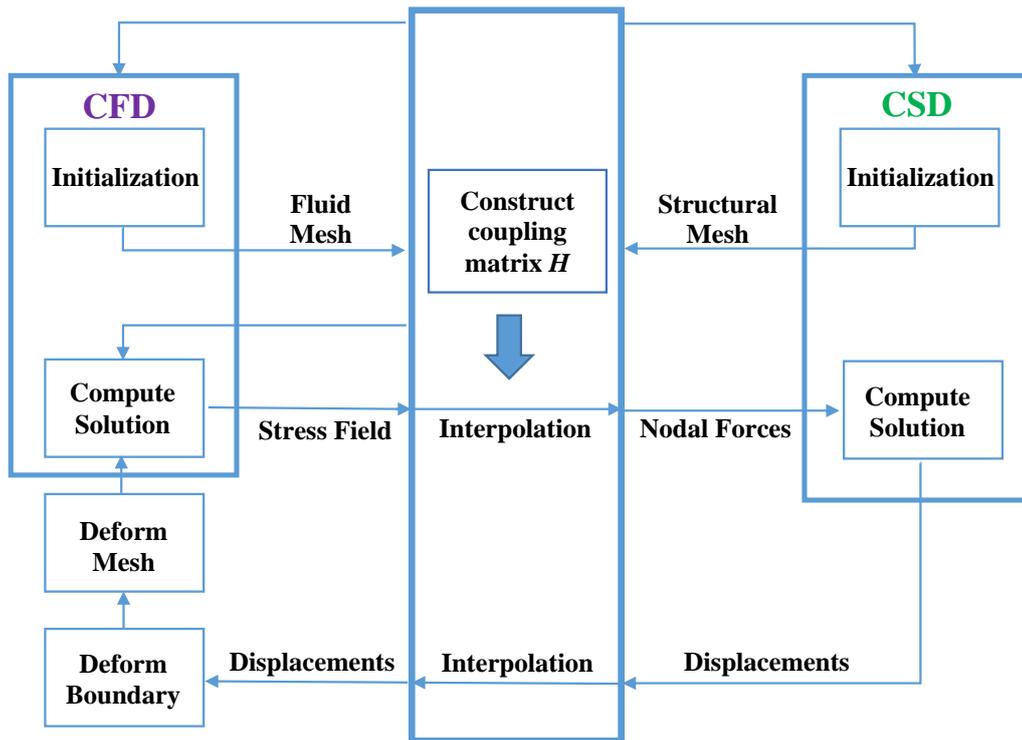


FIGURE 6.1: Flow chart of the FSI coupling scheme.

#### 6.4.3.2 GUI presentation

An in-house interfacing software (*FSImulator*) has been developed in this Thesis to couple the flow and structural solvers; it is written in C++, while it utilizes *Qt* application framework for the Graphical User Interface (GUI) and *OpenCASCADE* (“OpenCASCADE,” 2015) graphical libraries for the visualization process. Its main features are outlined below:

- A user friendly interface to set up the whole study along with a 3D viewer enabling user’s interaction.
- An integrated environment, allowing for real-time observation of the numerical simulation under examination.
- A data transfer method based on the RBF's PoU approach.
- A mesh deformation method based on the developed RBF's agglomerated interpolation approach.
- Custom Octree data structures based on the *OpenCASCADE* library functions.
- Integration of Eigen library and use of *OpenMP* for parallel processing.
- Various visualization properties
  - ✓ Display modes (shading, wireframe, etc.)
  - ✓ Initial/ Deformed grids
  - ✓ Custom clipping planes
  - ✓ Mesh data visualization (Pseudocolor, Vector, etc.)
  - ✓ Agglomeration edges
- Ability to import/export mesh data with results
  - ✓ Displacements of the structure under study
  - ✓ Pressure fields
- Ability to import/export mesh data in various formats
  - ✓ Ansys CFX format (.cfx5)
  - ✓ I-DEAS universal file format (.unv)

- ✓ STereoLithography file format (.stl)

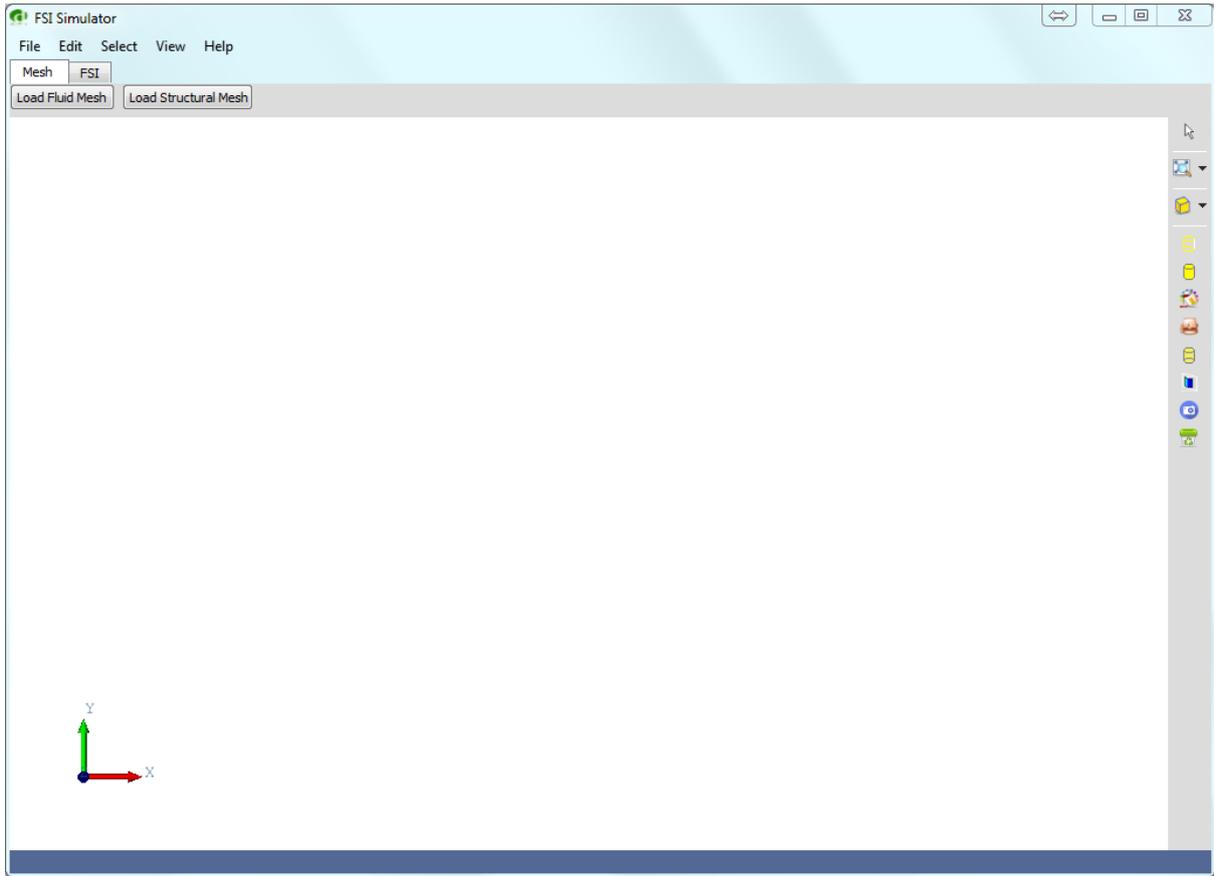
Special attention was given to the efficiency improvement of the viewer, in order to handle large-scale test cases, involving grids with millions or tens of millions nodes. Therefore, custom data structures, based on grid mapping, were constructed, while only the boundary faces of the elements are visualized by the GUI application.

The current version of this package is capable only for spatial coupling of the solvers (structural and flow) and mesh deformation; however, its potential to encounter a whole FSI simulation, including pre- and post-processing, is indicated. Considering this final goal, various tools have been developed for visualization enhancement of the code, while the GUI is formulated in such a way to enable further capabilities to define the initial and boundary conditions for the solvers.

For the time being, the coupling process relies on .txt files including data to be exchanged between the solvers. Thus, the predicted by the flow solver force field is imported to the interfacing software and interpolated to the structural grid according to the RBF's PoU method, while a similar process is followed for the interpolation of the predicted by the structural solver displacements to the flow grid. The deformation of the mesh is obtained with the RBF's agglomerated interpolation methodology and a new CFD simulation begins.

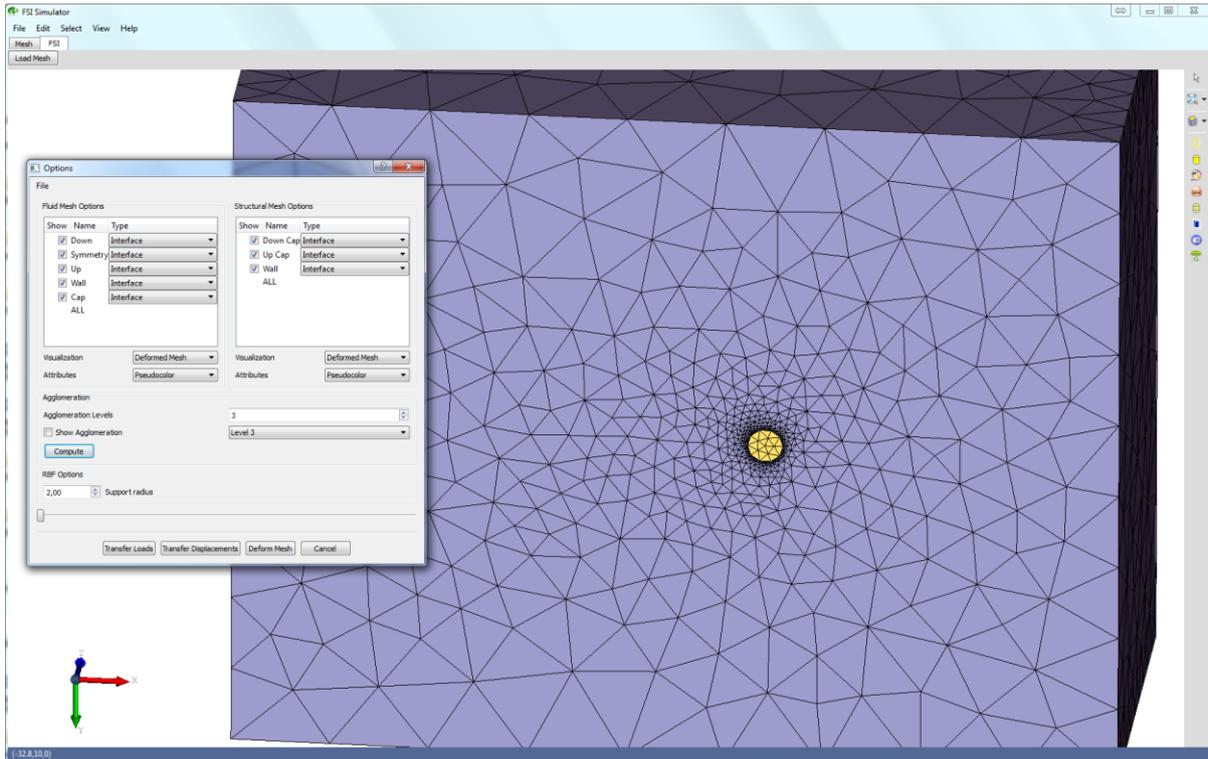
#### 6.4.3.2.1 *Coupling interface*

The main window of the software is presented in FIGURE 6.2; from the “Mesh” tab the user can import the flow and structural grids that had been generated to be used in the study. It provides the ability to import meshes in various formats (.cfx5, .unv, etc.), facilitating in that way the data exchange with other mesh generation software. Despite this potential, ongoing work concerns the development or incorporation of a mesh generation algorithm in order to enable grid generation within this application; therefore, the well-established mesh module of the SALOME platform has been incorporated.



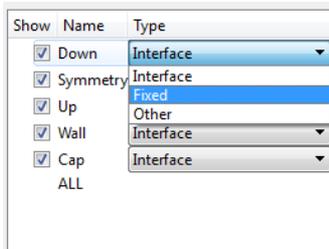
**FIGURE 6.2:** Main window – “Mesh” tab.

The next step considers the setup of the case study; the imported meshes are stored in custom data structures and imported automatically to the “FSI” tab, while an options-window appears, enabling the user to define the corresponding simulation parameters. As shown in FIGURE 6.3, the flow mesh is colored in grey, while the structural one in yellow (fluid flow over a cylindrical geometry).

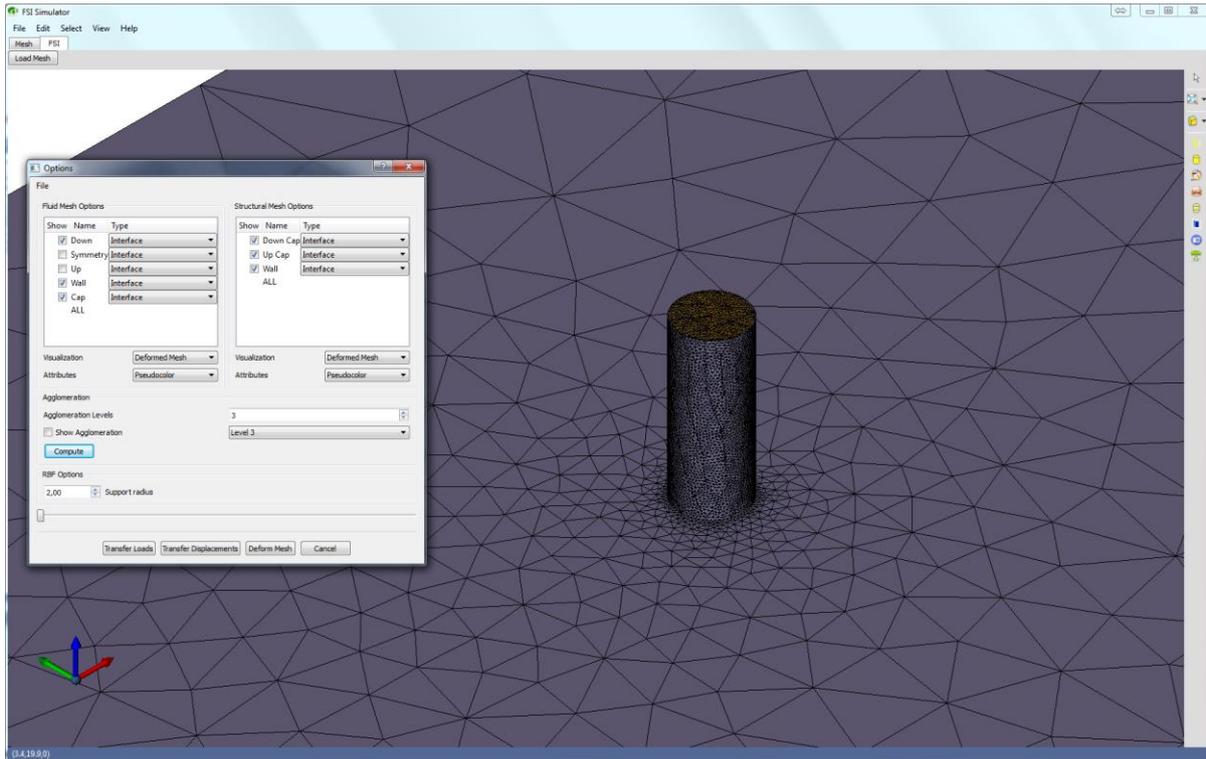


**FIGURE 6.3:** Main window – “FSI” tab.

The options-dialog, besides helping the user to define simulation parameters, aims to enhance visualization with additional tools, as well as to provide the means for the analysis of the predicted results.



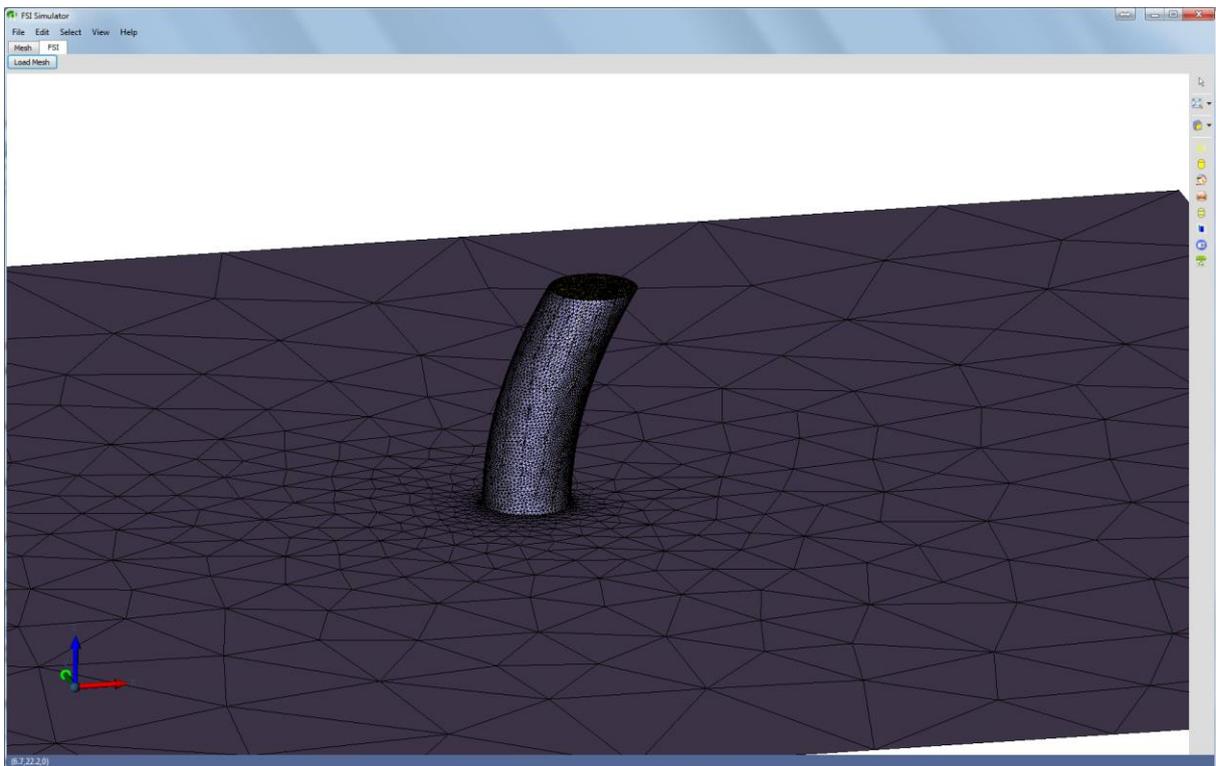
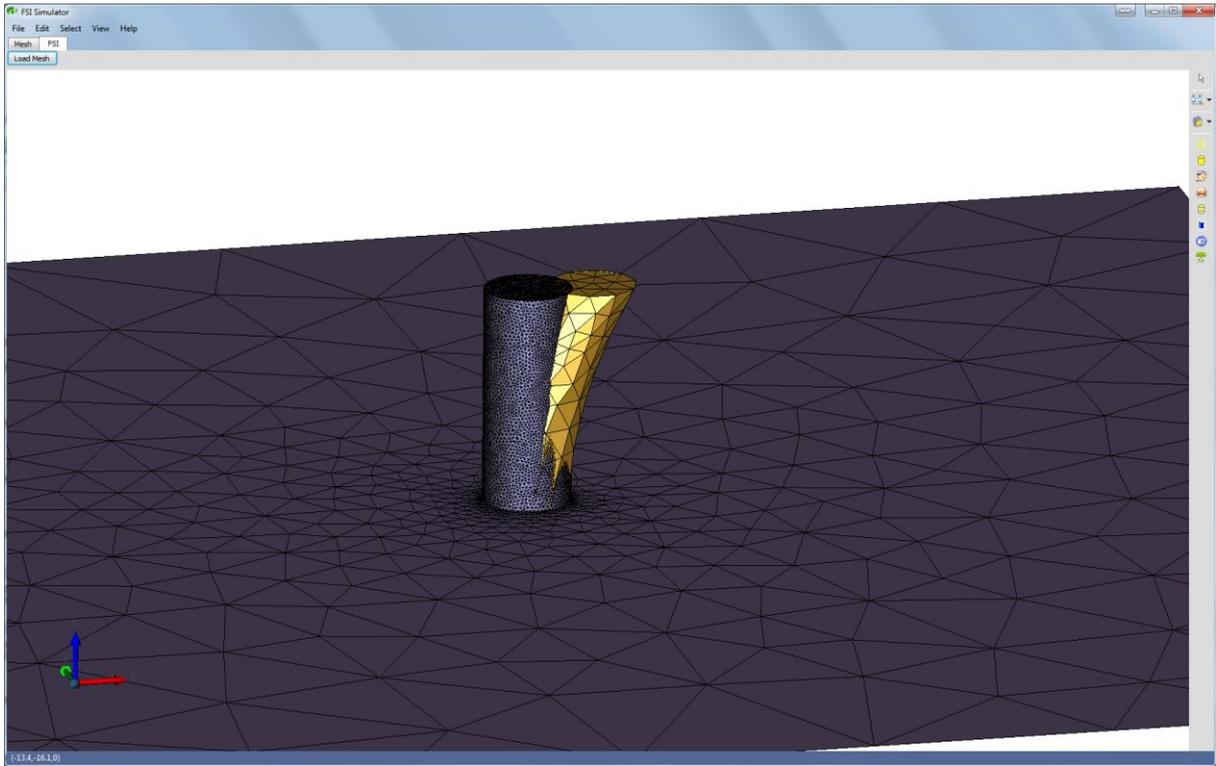
The user can define for both grids the type of their boundary closures regarding their association with the interfacing surfaces; as such, the interfacing boundaries, the fixed surfaces that can't be moved, etc. For each closure its visibility may be enabled or disabled to assist the definition process (FIGURE 6.4).



**FIGURE 6.4:** Hidden Symmetry and Up closures of the flow mesh.

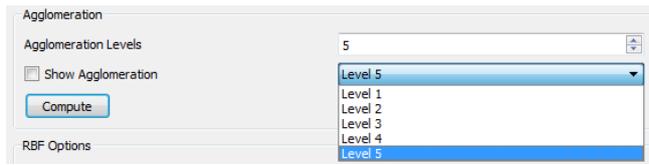
#### 6.4.3.2.2 Data transfer

As already mentioned, the predicted forces by the CFD solver acting on the interface and the computed displacements of the structure by the CSD solver are imported via appropriate .txt files. In FIGURE 6.5 the interpolation process of the displacements from the structural to the flow mesh is illustrated; the deformation examined herein corresponds to a parabolic deflection of a cylinder. The open-source library *Eigen* (Guennebaud *et al.*, 2010) was employed for the factorization and inversion of the coefficient matrices via LU decomposition (full pivoting), while the matrix-vector multiplications at the evaluation stage were accelerated with *OpenMP* (Dagum & Menon, 1998b). Moreover, octree data structures were developed in order to locate in an efficient manner the closest structural grid nodes for each flow one.



**FIGURE 6.5:** (Top) Deformed structure as calculated by the structural solver. (Bottom) Deformed flow mesh boundary surface computed by RBF's PoU interpolation.

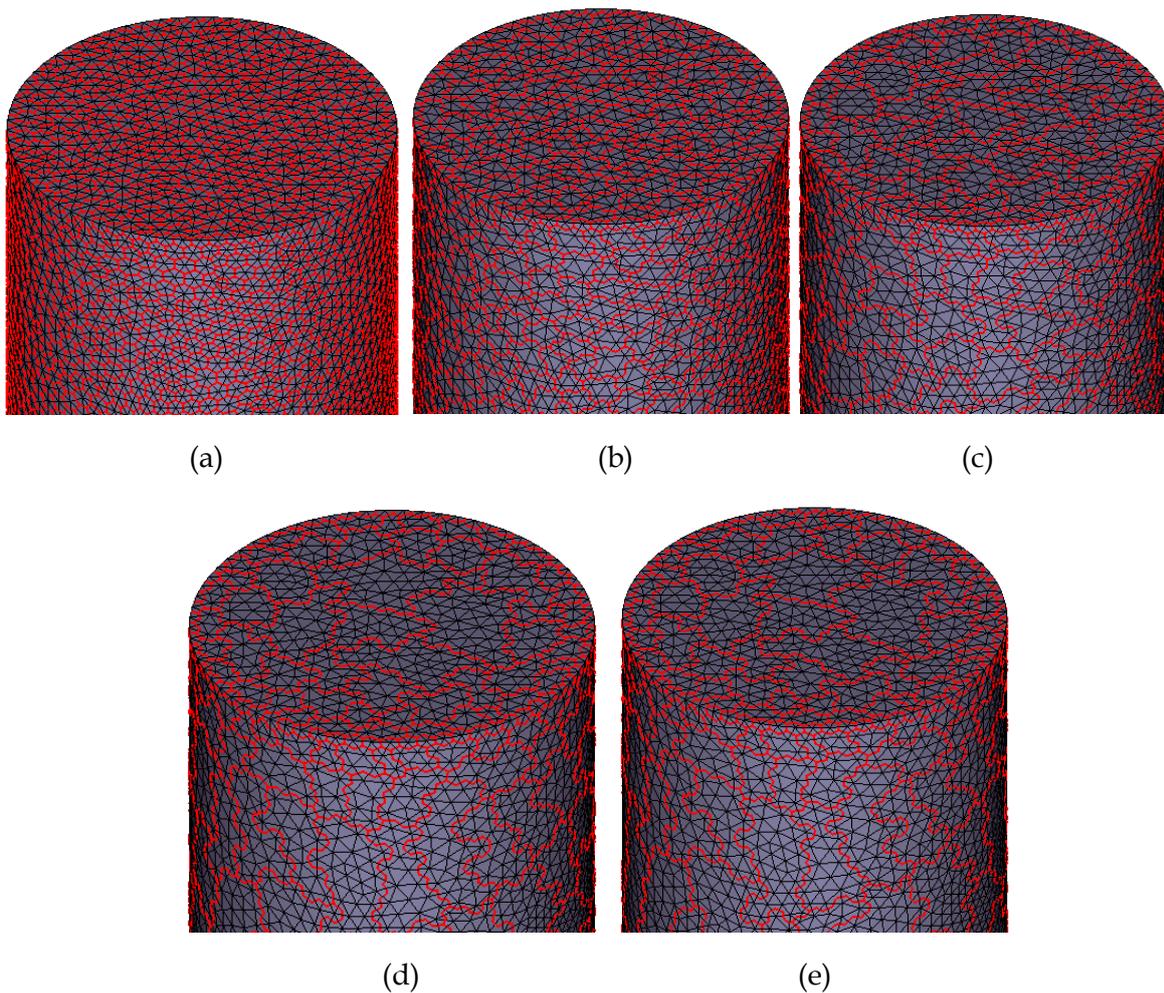
### 6.4.3.2.3 Mesh Deformation



From the options-dialog the user may select the number of agglomeration levels for the surface mesh coarsening. (Error! Reference source not found.).

**FIGURE 6.6:** Defining agglomeration levels for surface mesh coarsening.

In FIGURE 6.7 five discrete agglomerated levels are illustrated. Finally, more options can be defined in the options-dialog, e.g., the parameters of the utilized RBF's methodology.



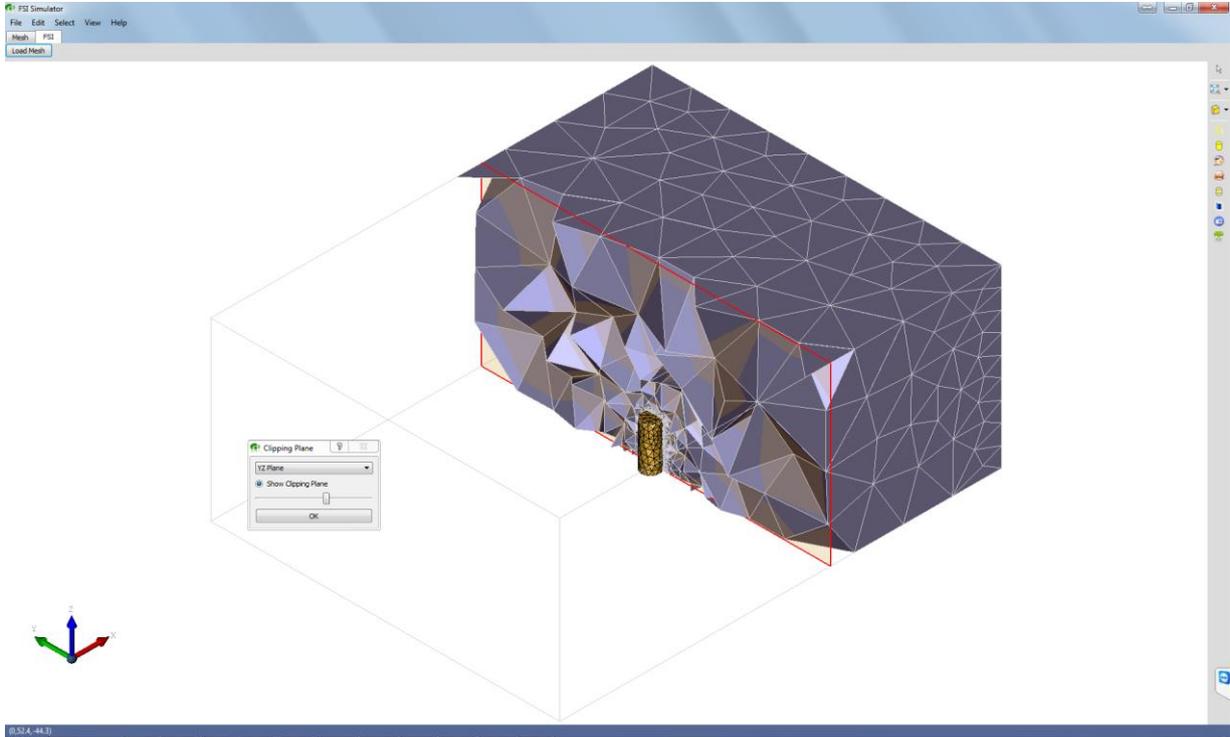
**FIGURE 6.7:** Mesh density on the cylindrical surface of the five agglomerated (a-e) control volume grids

For the analysis of the computed results, the user may select among various three-dimensional representations involving:

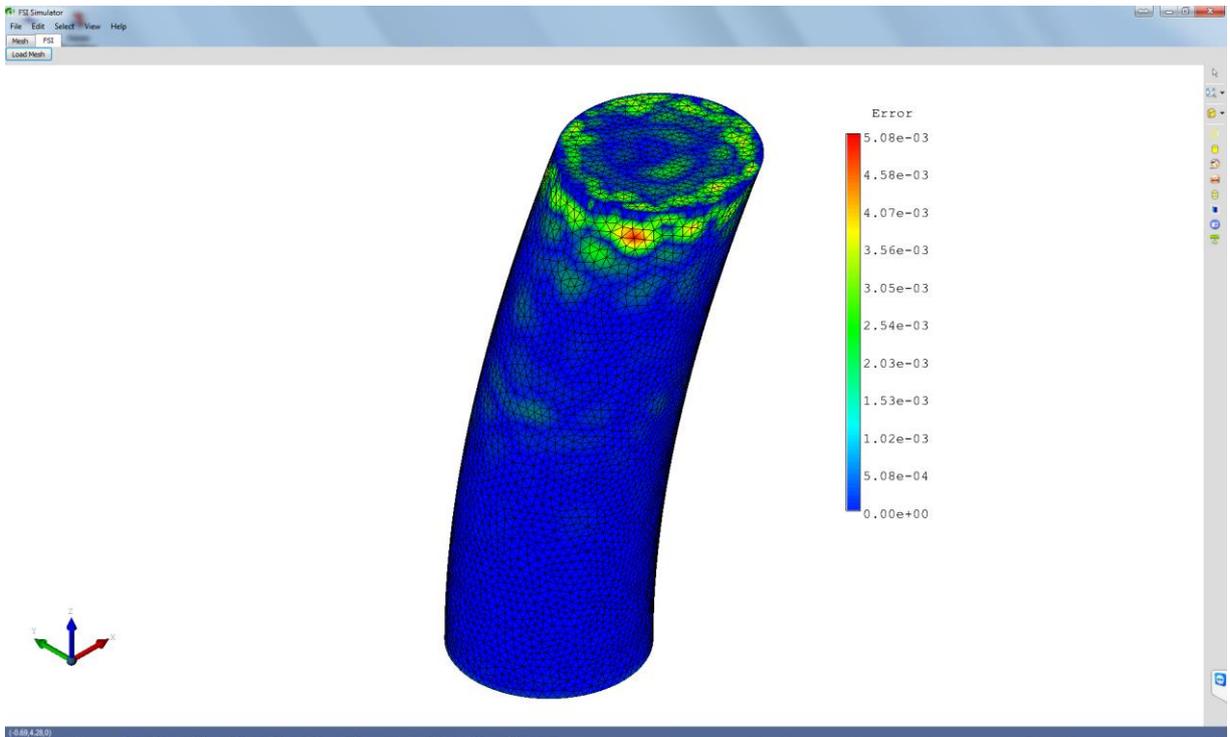
- Visualization of the internal mesh through sections at arbitrary positions defined by the user. For this reason, a custom clipping plane algorithm was developed to enable

whole element clipping; only the boundary elements are taken into account in visualization process to allow for its efficient performance (FIGURE 6.8).

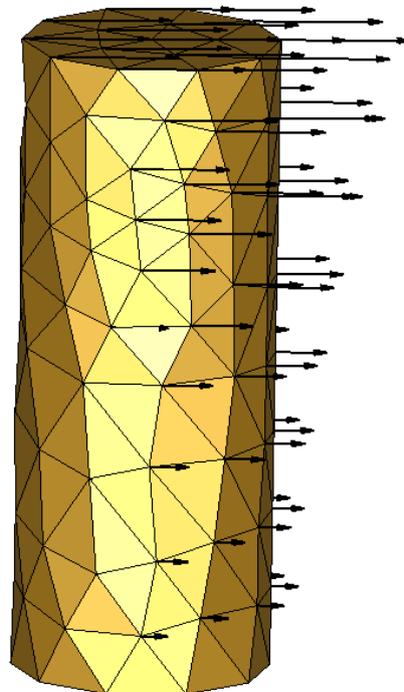
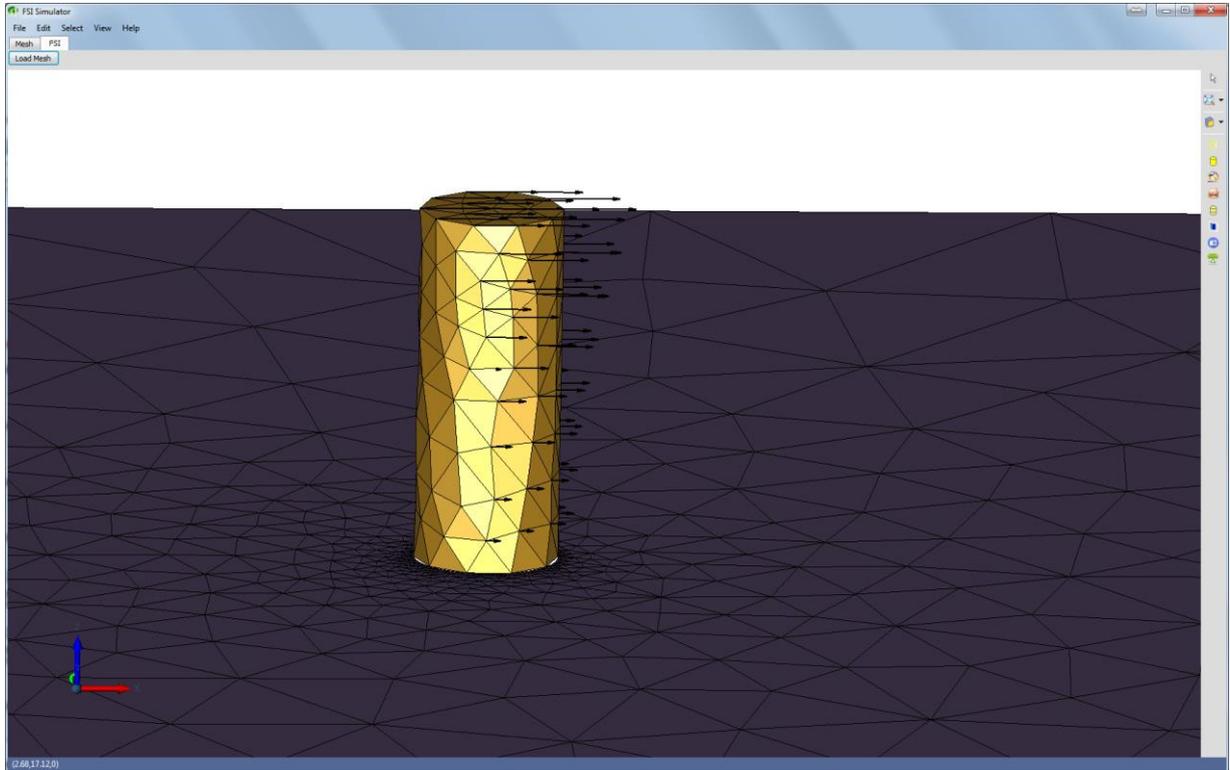
- Initial and deformed meshes.
- Pseudocolor representation with scalar maps for data visualization (FIGURE 6.9).
- Vector representation (FIGURE 6.10).



*FIGURE 6.8: Clipping plane -section of a flow mesh.*



*FIGURE 6.9: Pseudocolor representation of flow mesh boundary surface with colormap.*



**FIGURE 6.10:** Vector presentation on the structural grid, illustrating the displacements of the corresponding nodes.

## 6.5 Numerical results

### 6.5.1 Commonwealth Advisory Aeronautical Research Council (CAARC)

The first test-case in order to assess the pre-described methodologies was performed with the Commonwealth Advisory Aeronautical Research Council (CAARC) Standard Tall Building Model, which is in fact a specification for wind tunnel experiments on tall building geometries (Wardlaw & Moss, 1970; Melbourne, 1980). The full-scale geometry of the building is specified as a rectangular shaped prism with base size equal to  $100\text{ft}$  ( $30.8\text{m}$ ) by  $150\text{ft}$  ( $45.72\text{m}$ ) and height  $600\text{ft}$  ( $183.88\text{m}$ ). It is described as a flat-topped building without parapets; its external walls are flat without any geometric disturbances (Melbourne, 1980). The case has been encountered by several researchers, who have performed numerical tests of wind flow around the building, as well as FSI simulations (Huang *et al.*, 2007; Braun & Awruch, 2009; Dagnew *et al.*, 2009; Dagnewa & Bitsuamlakb, 2010); the proposed geometry has been tested in wind tunnel experiments too (Melbourne, 1980; Obasaju, 1992). A sketch of the geometric model that was prepared for use with the *Galatea-I* solver is shown in FIGURE 6.11.

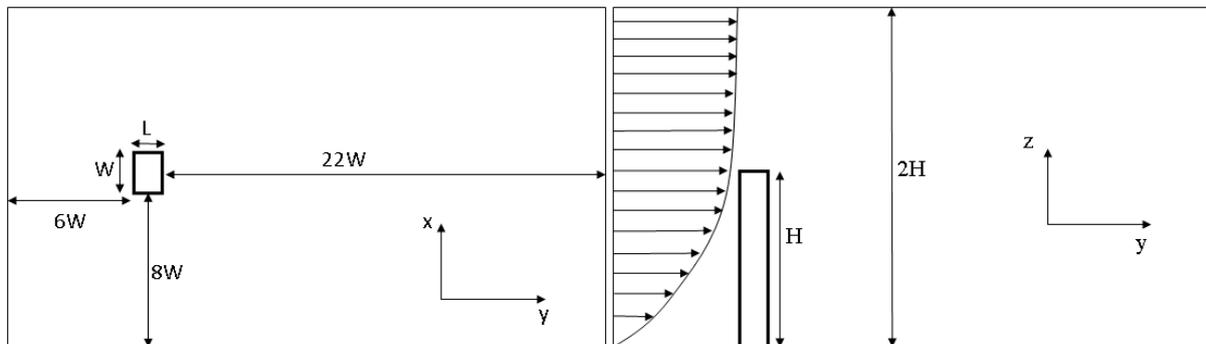


FIGURE 6.11: Sketch of CAARC test case setup.

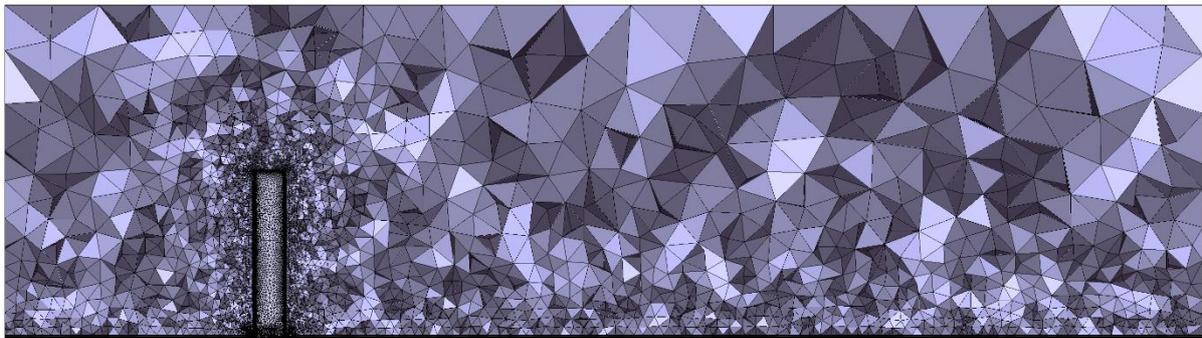


FIGURE 6.12: Computational mesh around the CAARC standard tall building model.

The computational mesh, generated for the simulation of wind flow around the CAARC building, is presented in FIGURE 6.12. As the flow around the building develops in three-dimensional fashion and is unsteady in nature, the whole building geometry had to be meshed. In order to reduce computation time, a relatively coarse mesh was generated that consisted of 434,082 nodes, 623,801 tetrahedra and 640,685 prisms. Prismatic inflation was applied on the building walls, as well as at the base of the geometry that represents the ground around the building, in order no slip boundary conditions to be applied at these areas and the corresponding boundary layer to be effectively predicted. The prismatic inflation consisted of

20 layers with the first layer height equal to  $W/2500$ . At the top and side walls of the geometry free slip conditions were imposed.

The Reynolds number was set equal to 380,000, based on the building height  $H$  and the velocity  $c_H$  at height  $z = H$  (Dagnev *et al.*, 2009). In order to simulate wind flow at such a low height, the velocity profile at the inlet was defined by the following relation

$$\frac{|\vec{c}|}{c_H} = \left(\frac{z}{H}\right)^a \quad (6.18)$$

where  $a$  is the power law exponent, which in this case was set equal to 0.3. Moreover, the turbulence boundary conditions had to be calculated, based on the turbulence intensity in the approaching flow, as it has a significant effect on the accuracy of the numerical results (Huang *et al.*, 2007; Braun & Awruch, 2009; Dagnev *et al.*, 2009). In FIGURE 6.13 the velocity profile at the inlet, as well as the turbulence intensity profile are presented. The values of turbulence intensity at the inlet of the computational domain were interpolated from those presented in FIGURE 6.13 (Huang *et al.*, 2007). Finally, all values were normalized with  $H$  and  $c_H$  for use with the dimensionless solver.

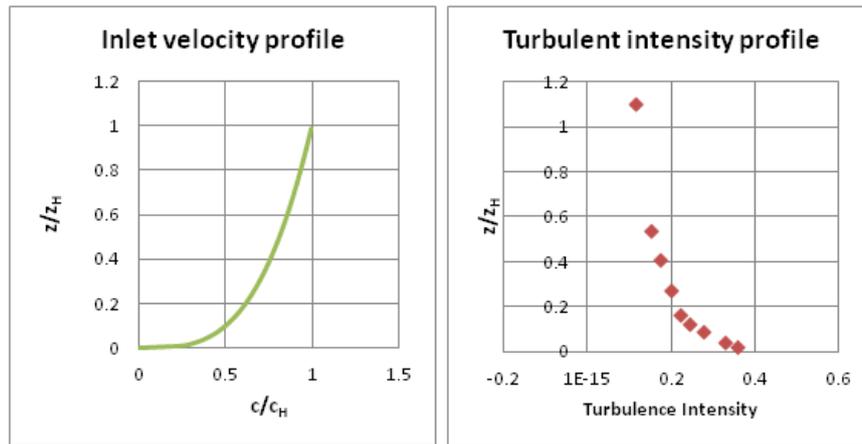


FIGURE 6.13: Inlet velocity profile (left) and turbulence intensity profile (right) (CAARC case).

The simulation was performed on a workstation with an AMD FX™ 8150 Eight-Core processor at 3.62 Ghz. Acceleration via parallel processing was applied by partitioning the initial mesh into six sub-domains, while due to the small size of the computational mesh the multigrid method was not used. The time-accurate equations were solved over a total of 1300 real time iterations, with a dimensionless real time step equal to 0.05. In each real time step 50 pseudo-time iterations were performed, with a CFL number equal to 0.5. The artificial compressibility parameter was set equal to 10.0. For the quantitative evaluation of the extracted results the mean-pressure coefficient on the wall of the building was calculated from the obtained time-accurate results. In FIGURE 6.14 the distribution of the mean-pressure coefficient along the building wall at height  $z = 2/3H$  is compared with the corresponding ones found in the literature (Huang *et al.*, 2007; Dagnev *et al.*, 2009). While a very good agreement is achieved between the current and reference data at the windward and leeward sides of the building, there seems to be some disagreement between numerical and experimental results at the side of the building. This is attributed to the method of turbulence evaluation; a more accurate method (e.g., LES-Large Eddy Simulation) could provide more accurate results at these regions.

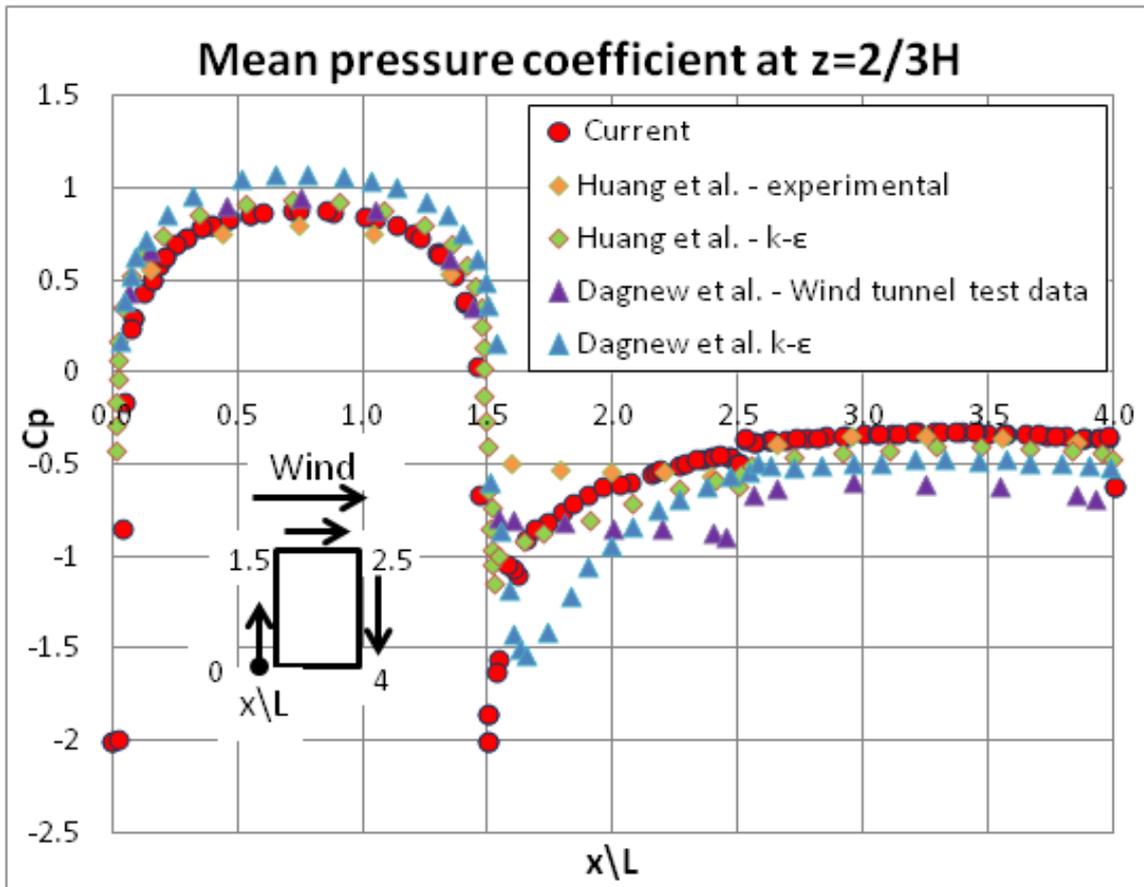


FIGURE 6.14: Mean pressure coefficient on the CAARC building wall at  $z=3/2H$ .

In FIGURE 6.15 the mean pressure coefficient distribution on the building walls and the “ground” around is presented. The maximum pressure value is located near the building top at the windward side, while under-pressure is developed at the leeward side of the building.

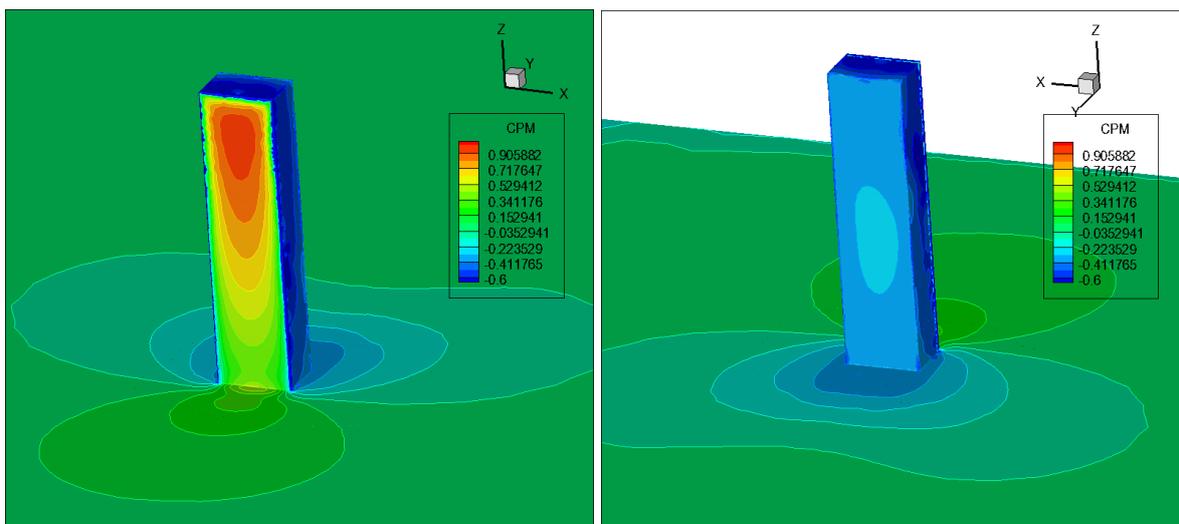
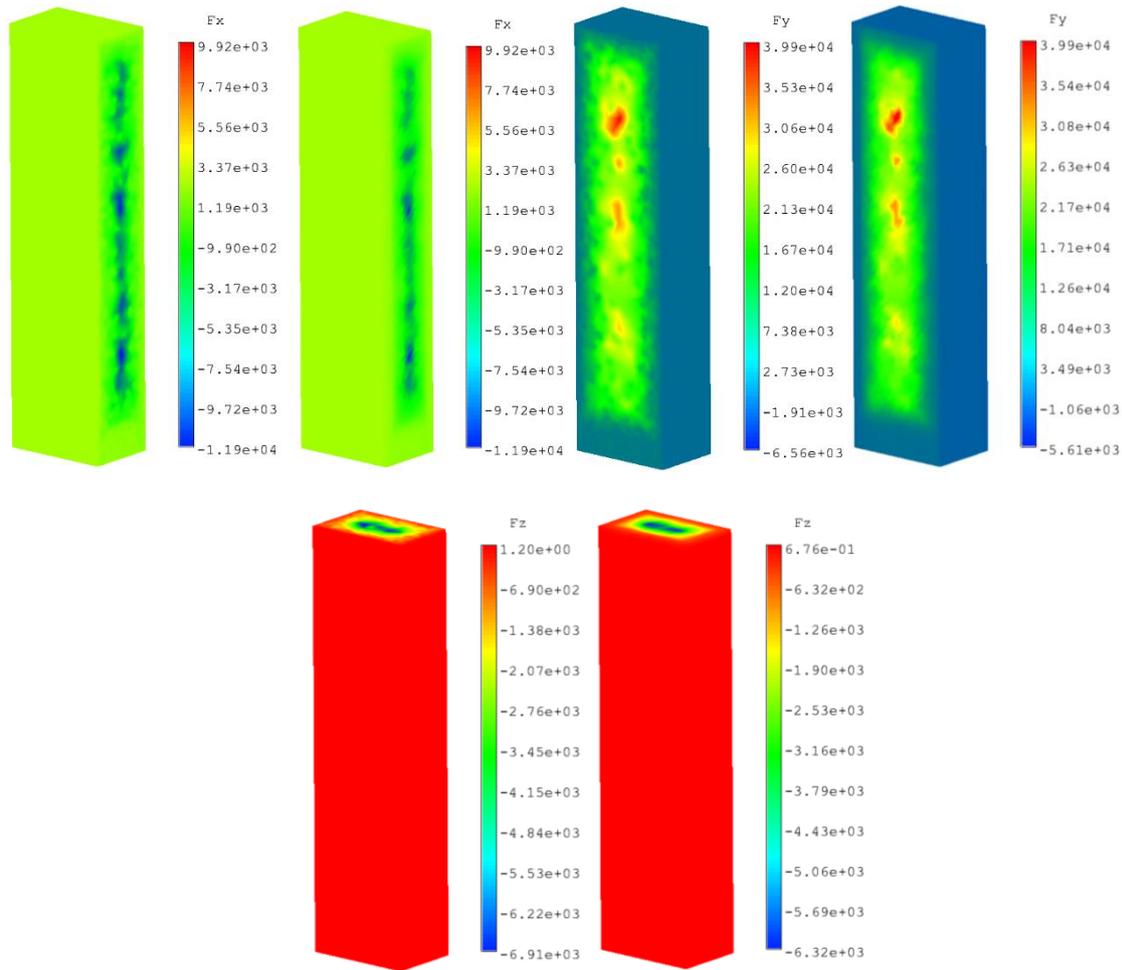


FIGURE 6.15: Mean pressure coefficient distribution on the CAARC building walls and surrounding “ground”.

The force field around the CAARC standard tall building model was then calculated from the dimensionless results provided by the *Galatea-I* solver with regard to the flow conditions, reported by (Braun & Awruch, 2009). These properties of the flow can be found in TABLE 6.1. In order to use them with the CSD code, the exported force field had to be interpolated between the two different grids, i.e., the unstructured and the structural one. For the coupling of the two grids with the RBFs PoU method, the nearest nodes of the structural mesh have to be identified for each aerodynamic grid point at the building wall surface. Nevertheless, such an identification is not a straightforward procedure; a brutal linear search method would require an enormous amount of time to perform, especially in the case of very dense meshes. Therefore, custom Octree data structures were developed to find the nearest structural grid point to each aerodynamic one, while at next the remaining nodes required by the proposed methodology were gathered with the aid of the structural mesh topology. Applying this approach, whereby the number of structural points is given a priori, ensures the sparseness of the coupling matrix, as the user can specify the number of these points according to the problem needs. Due to this PoU method, a support radius needs not be defined; it is set equal to the distance of each aerodynamic grid point from the uttermost in the specific patch.

**TABLE 6.1:** *Flow field properties (CAARC case).*

Specific mass - $\rho$ [Kgr/m <sup>3</sup> ]	1.25
Dynamic viscosity - $\mu$ [N·s/m <sup>2</sup> ]	$7.03 \cdot 10^{-2}$
Reference velocity - $c_H$ [m/s]	100
Reference length - $H$	180



**FIGURE 6.16:** Comparison of force component distribution between CFD (left) and CSD (right) meshes (CAARC case).

The utilized structural grid is composed of 38,675 nodes and 43,048 hexahedral eight-node brick elements, fully integrated ( $2 \times 2 \times 2$  integration points). For each boundary node of the CFD mesh the 20 closest points of the CSD mesh were used, while for each set of CSD nodes a RBFs problem was set deriving the corresponding required coefficients. While the coupling of the CFD with the CSD mesh may be a rather tedious procedure, it only has to be completed once (at the beginning of the FSI simulation) even for a dynamic problem. The computational cost of the coupling procedure is slightly affected by the size of the structural mesh, as the number of RBFs problems to be solved remains the same; only the nearest nodes identification process will differ. Comparisons between the force components distributions over the boundary surfaces, for the flow and structural computational grids, are illustrated in FIGURE 6.16.

These qualitative results reveal the ability of the proposed methodology to provide accurate transfer of the force field in an efficient manner though. As it can be observed, the PoU approach offers the advantage of physical distribution of forces over the interface, as opposed to the full method, where the transpose of the coupling matrix would destroy the local character of the force distribution (Ahrem *et al.*, 2006).

With the loading (i.e., the force distribution) provided by the CFD code, a modal analysis was performed with the *CalculiX* open source software. The structural properties of the model

were based on the CAARC standard tall building specifications (Melbourne, 1980), while the Young's modulus was set equal to that reported in (Braun & Awruch, 2009). All the employed structural properties are given in TABLE 6.2.

TABLE 6.2: Mechanical properties of the structure (CAARC case).

Specific mass - $\rho$	160 Kg/m <sup>3</sup>
Natural frequency - $n$	0.2 Hz
Young's modulus - $E$	$2.3 \cdot 10^8$ N/m <sup>2</sup>
Poisson's ratio - $m$	0.25
Structural damping - $\xi$	1%

In FIGURE 6.17 the von Mises stresses and the stresses at the three main axes are presented, while in FIGURE 6.18(a) the deformation of the structural mesh according to the results of the aforementioned analysis is illustrated (magnified x 10) along with the un-deformed boundary surface of the fluid mesh. The maximum displacement was 46.9 cm on the  $y$  axis (as expected) because of the influence of the first bending mode. Moreover, in FIGURE 6.18(b) the deformed boundary surface of the CFD mesh obtained by transferring the displacement field from the CSD one, is depicted; no distortion was observed on the boundary surface of the flow grid, resulting in a smooth deformed surface. The normalized along-wind displacement ( $\sigma_y/L$ ) as a function of the reduced velocity ( $c_H/nW$ ), obtained with the proposed methodology, is compared to the corresponding results of other reference works in TABLE 6.3; a satisfactory agreement is observed.

TABLE 6.3: Comparison of normalized maximum displacement (CAARC case).

Current (Braun & Awruch, 2009)	0.015633
(Thepmongkorn & Kwok, 2002)	0.011595
(Melbourne, 1980)	0.048892

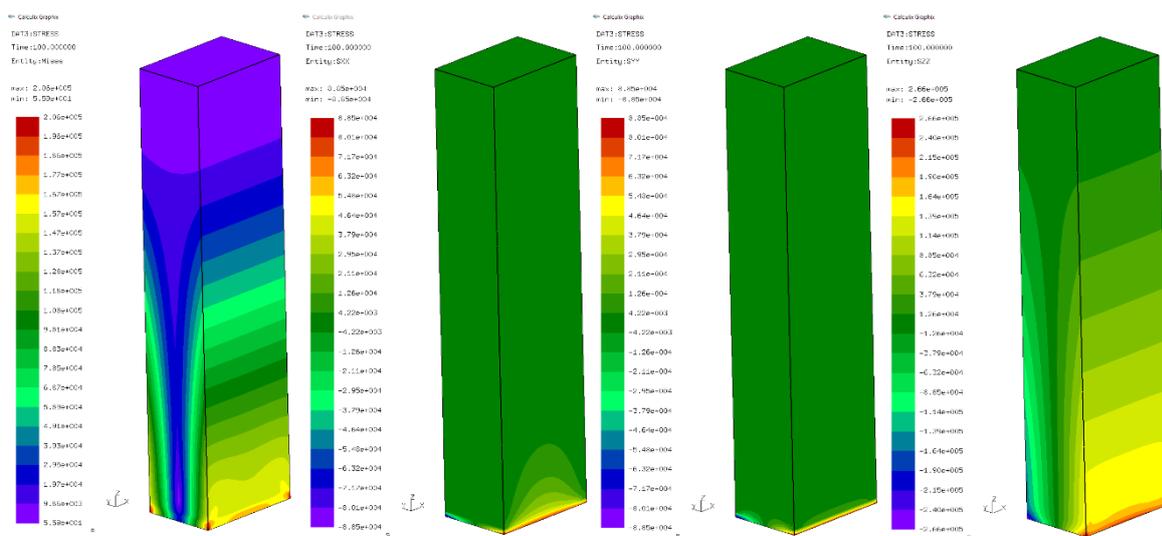
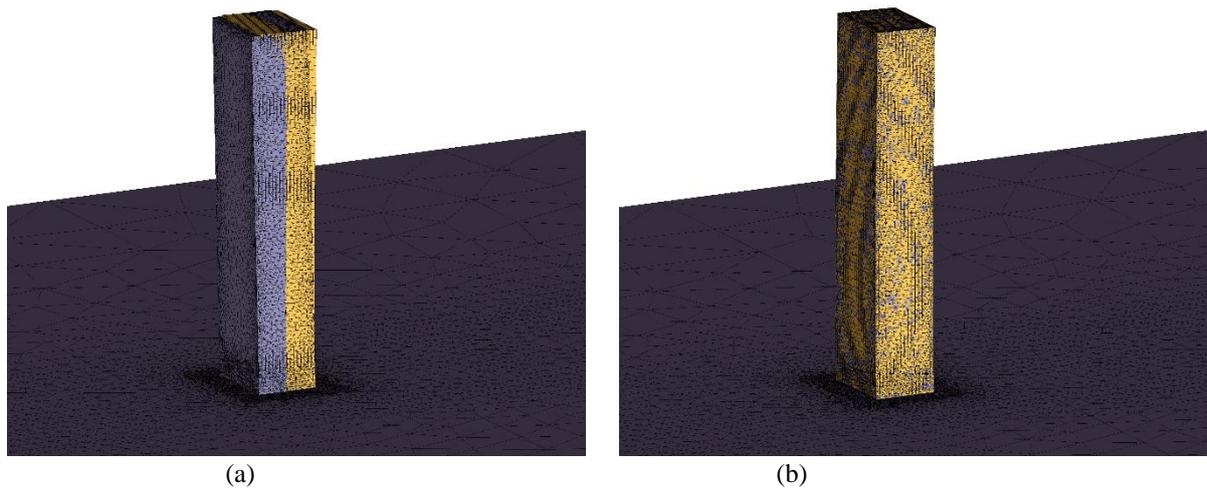
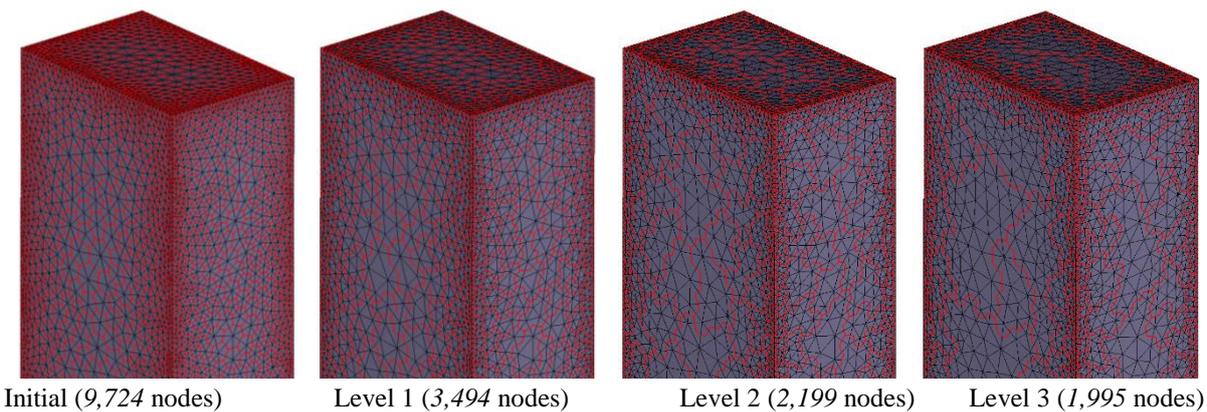


FIGURE 6.17: Von Mises,  $S_{XX}$ ,  $S_{YY}$ ,  $S_{ZZ}$  stresses of the CAARC standard tall building model.

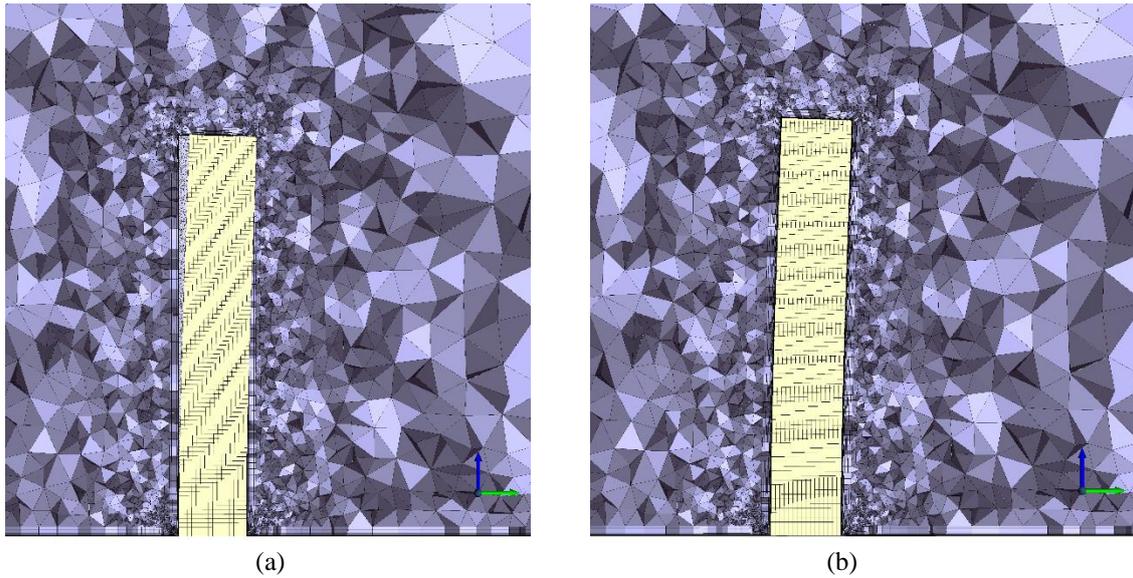


**FIGURE 6.18:** Deformation of the structural grid (a) and corresponding deformation of the CFD boundary grid (b) (CAARC case).



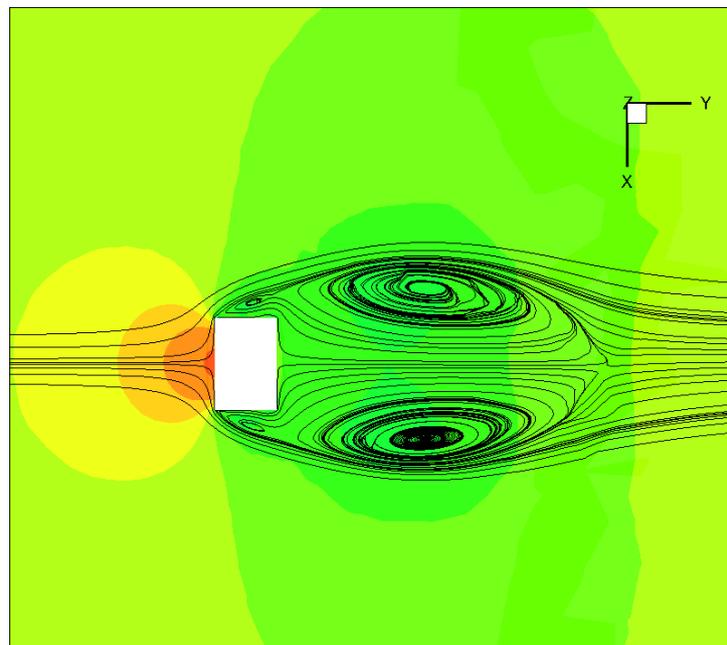
**FIGURE 6.19:** Fine and coarser grids generated with the agglomeration procedure (CAARC case).

Since this stage was accomplished, the deformation of the flow grid had to be performed, according to the deflections of the boundary. For the acceleration of the procedure four coarser grids were obtained, implementing the proposed (Chapter 5) agglomeration process on the surface nodes (Strofyilas *et al.*, 2015). In FIGURE 6.19 the boundary control areas of each level are depicted. For all of them the node with the minimum distance from their area-weighted centers is selected to be used as the base point for the RBFs method (Strofyilas *et al.*, 2015). These basis points form the interpolant, required for the final deformation of the volume grid. In FIGURE 6.20, the flow grid prior and after the implementation of the deformation procedure is illustrated.



**FIGURE 6.20:** Before (a) and after (b) deformation states of the CFD grid (CAARC case).

The deformed CFD grid was subsequently used with the *Galatea-I* solver for a new flow simulation with the same flow conditions as with the original one. The derived mean pressure coefficient distribution around the CAARC standard tall building, along with the velocity streamlines are illustrated in FIGURE 6.21. The successfully performed simulation demonstrates the deformation methodology capabilities in providing quality grids.



**FIGURE 6.21:** Mean pressure coefficient and velocity streamlines around the deformed CFD grid (CAARC case).

The aforementioned analysis was based on a specification for a tall building model which however differs a lot from a realistic one. For the completion of the current work, the results from a dynamic analysis of a similar 45-storey building are presented. The building has the same external geometry as the CAARC standard tall building model, but the interior is not solid. It is consisted of beams, slabs, columns, walls, and a lift core. Moreover, two different

materials, steel for the structural members (green color), and glass for all the outer walls (red color) were used (FIGURE 6.22). TABLE 6.4 presents the properties of the steel and glass material.

TABLE 6.4: Mechanical properties of the structure (realistic CAARC building model).

	Glass	Steel
Specific mass - $\rho$	2500 Kg/m <sup>3</sup>	7850 Kg/m <sup>3</sup>
Natural frequency - $n$	0.2 Hz	0.2 Hz
Young's modulus - $E$	$7 \cdot 10^{10}$ N/m <sup>2</sup>	$2.1 \cdot 10^{11}$ N/m <sup>2</sup>
Poisson's ratio - $m$	0.25	0.25
Structural damping - $\xi$	1%	1%

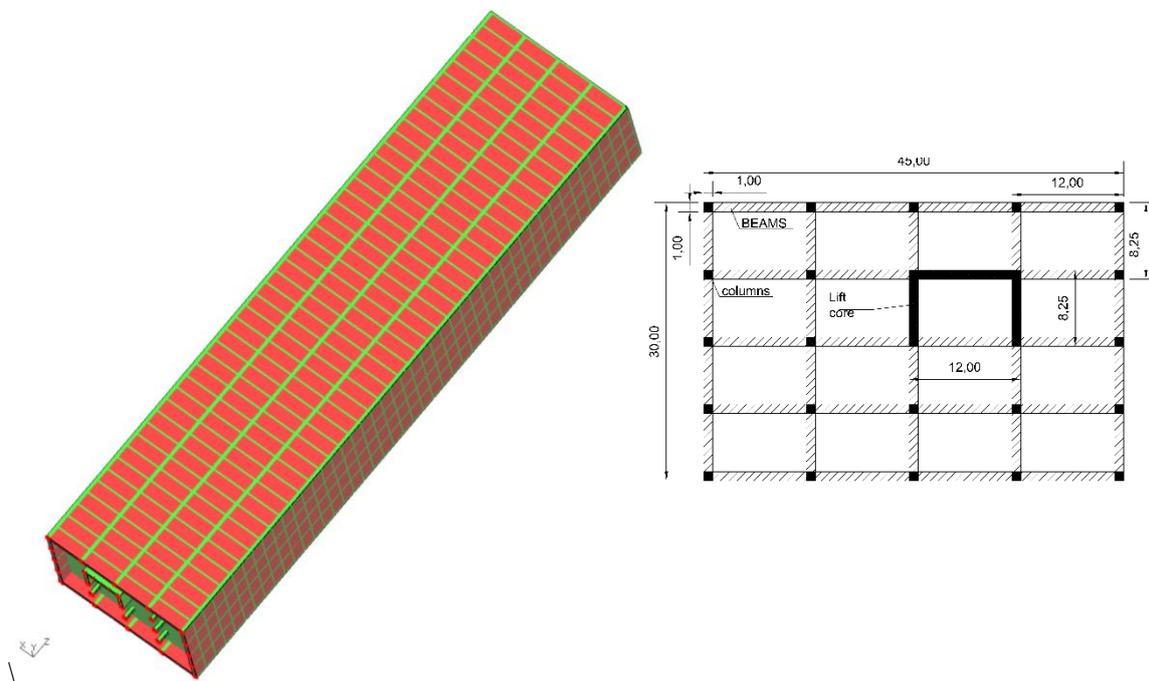


FIGURE 6.22: Realistic CAARC building model (structural members: green color, walls: red color).

In this case a dynamic analysis takes place, in which the wind loading is considered the same to the previous one (FIGURE 6.16). The deformation is also similar to the previous case but the displacements are much smaller, as expected, taking into account that the elements properties are stiffer. FIGURE 6.23 presents the extracted von Mises stresses as well as the three main stresses. Additionally, comparing to the previous problem, the stresses appear to be more realistic. The columns produce larger stresses than the walls, because of their larger stiffness, while as the height increases the stresses get smaller. In general, larger stresses were extracted (comparing to the previous case) as the same force was imposed at a smaller surface though.

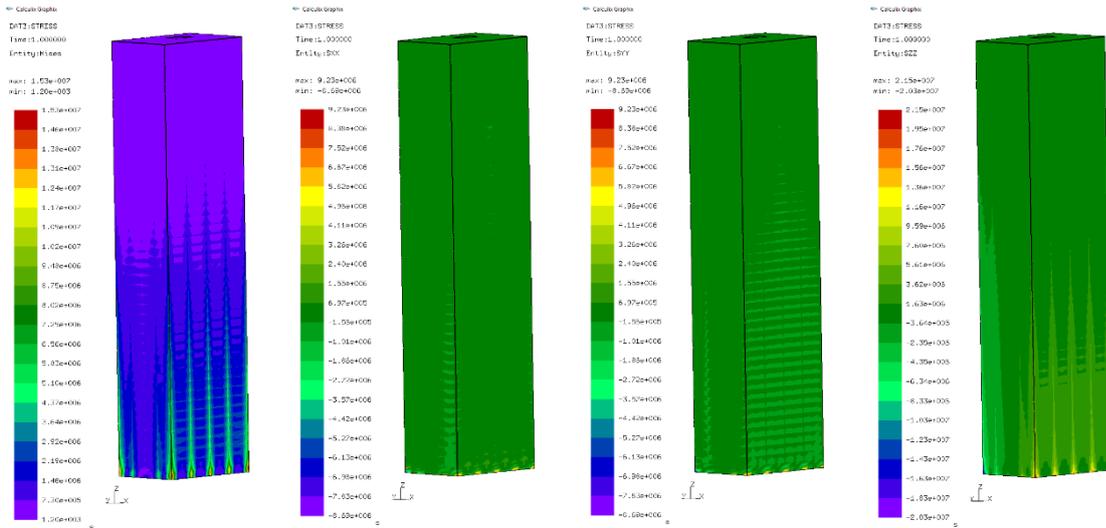


FIGURE 6.23: Von Mises,  $S_{xx}$ ,  $S_{yy}$ ,  $S_{zz}$  stresses for the CAARC realistic building model.

## 6.5.2 Bridge case

The next benchmark problem encountered in this study considers the static aeroelastic simulation of a real bridge model. The geometry was based on a real constructed bridge at Egnatia motorway in Greece. In particular, the simulated model was the G2 part from Panagia to Grevena (4.1.3s), a pre-stressed bridge during its construction phase (M3 pylons). The bridge model has been built according to the cantilever method and consists of two parts, right and left, as shown in FIGURE 6.24. Both of them have height  $69.14\text{ m}$ . In this study the left one consists of 18 sections with span  $72.3\text{ m}$ , while the right one consists of 10 sections with span  $36.7\text{ m}$ . The pylon is composed of a hollow section with external dimensions  $7.0 \times 3.5\text{ m}$ , while the deck consists of a box section with width  $14.2\text{ m}$  and height ranging from  $2.9\text{ m}$  up to  $6.10\text{ m}$ . The real model contains up to 36 tendons, which were ignored in the current case in order to achieve a less stiff structure.

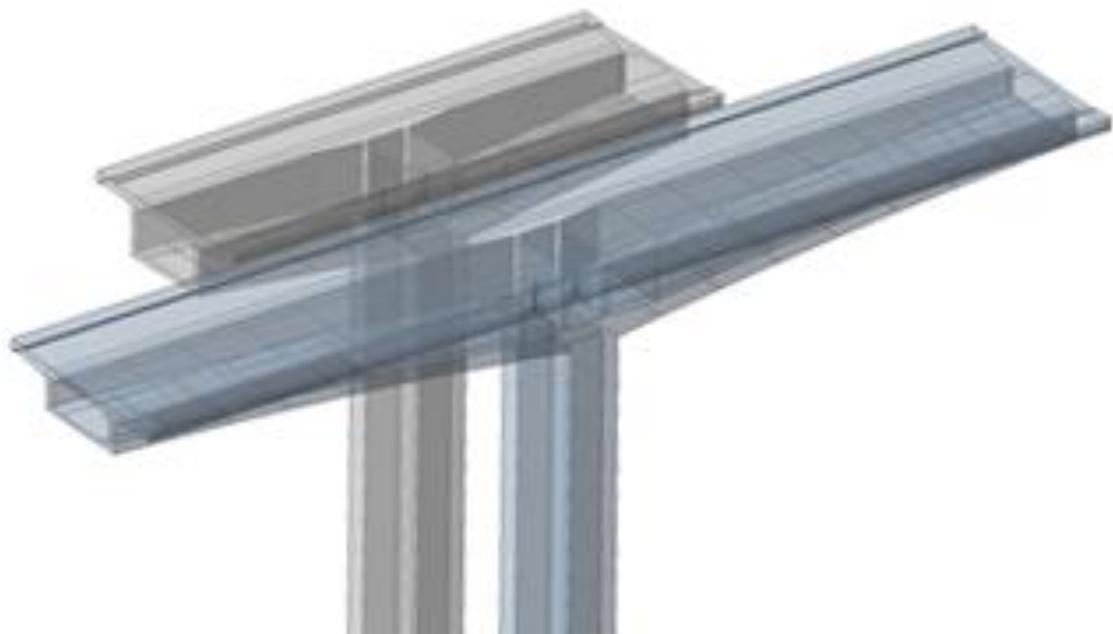
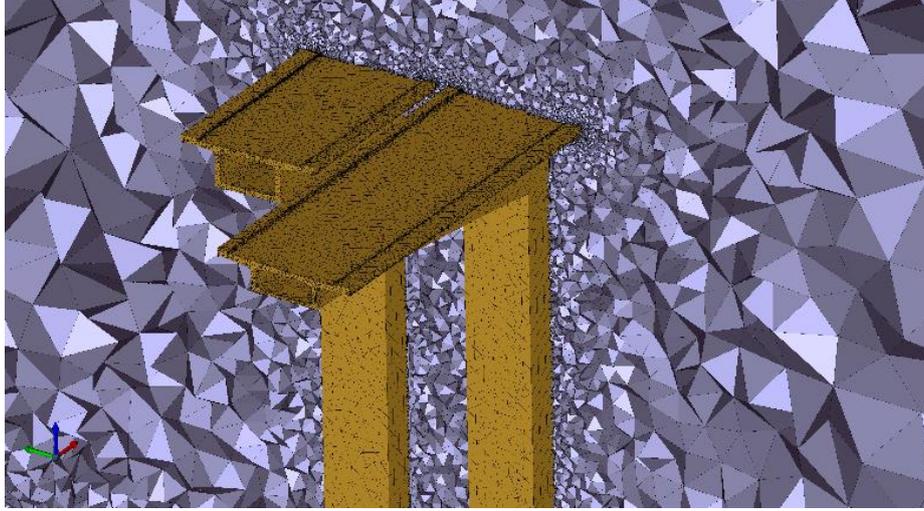


FIGURE 6.24: General view of the geometry (bridge case).



**FIGURE 6.25:** Flow (gray) and structural (yellow) computational grids (bridge case).

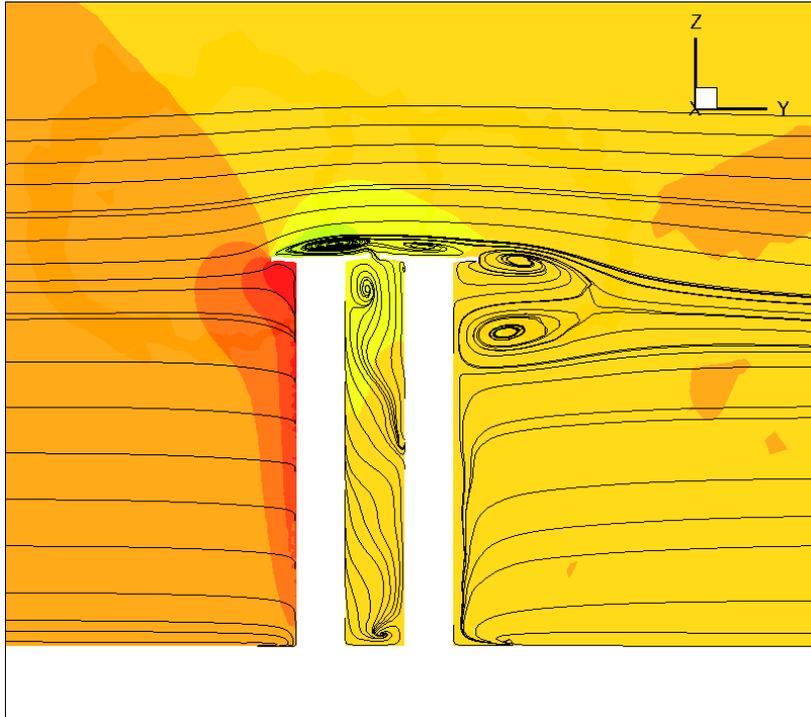
For the flow simulation over the two bridges a three-dimensional hybrid unstructured grid was constructed, composed of 2,9201,26 nodes, 5,386,114 tetrahedra and 3,844,887 prisms. The prismatic elements were distributed with a ten-layer inflation on the bridges' walls, as well as on the ground, to capture the boundary layer phenomena at the corresponding areas (FIGURE 6.25). At top and sidewalls of the computational domain free slip conditions were applied. The generated structural grid was composed of 179,761 nodes and 777,425 four-node, linear, tetrahedral elements with one integration point (FIGURE 6.25). Despite the fact that this type of element is stiff enough, there is no divergence when many of them are used.

In order to simulate wind flow at such a low height, a power-law velocity profile was defined as

$$\frac{|\vec{c}|}{c_0} = \left(\frac{z}{H_0}\right)^a \quad (6.19)$$

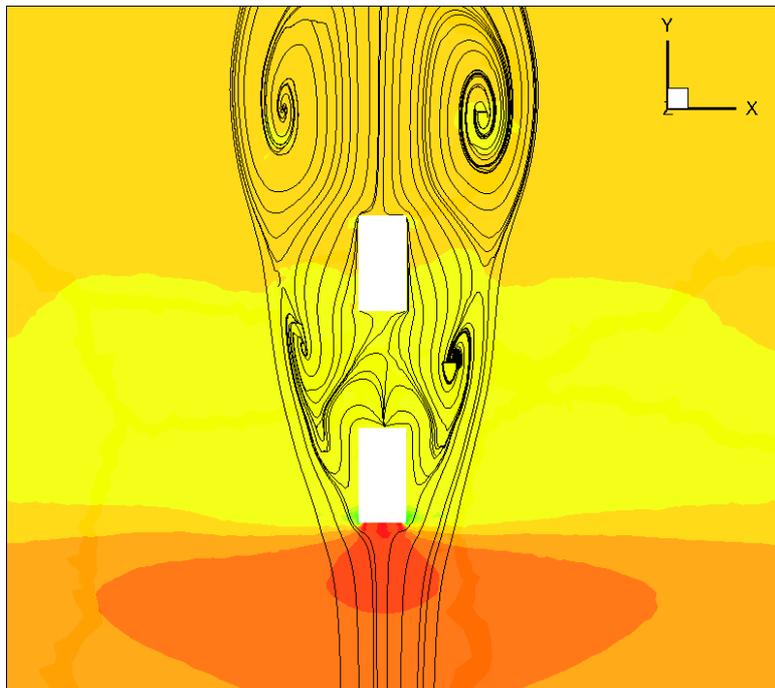
where  $a$  was set equal to 0.2 and  $|\vec{c}|$  is the velocity vector magnitude, while  $c_0$  and  $H_0$  are the reference velocity and reference height, respectively, defined in this case as follows:  $c_0 = 80 \text{ m/sec}$  and  $H_0 = 115 \text{ m}$ . Air density was assumed equal to  $\rho = 1.2 \text{ kg/m}^3$ , with reference dynamic viscosity set to  $\mu = 1.846 \times 10^{-5} \text{ kg/(m s)}$ . The Reynolds number was computed equal to  $72.0 E + 6$ , based on the reference velocity  $c_0$  and the reference length  $L$ ; the latter was set equal to the bridge width  $L = 14.20 \text{ m}$ . Finally, the computational grid, as well as all the flow variables were normalized with respect to the reference velocity and reference length to be used with the dimensionless solver *Galatea-1*. The artificial compressibility parameter was set equal to 10.0, and the CFL number to 0.3.

The simulation was performed on a computer with an AMD FX™ 8150 eight-core processor at 3.62 GHz. For parallel processing the utilized grid was divided into eight sub-domains, while a two-level agglomeration multigrid approach was implemented for additional acceleration of the solution procedure. One hundred real time iterations were performed totally with a dimensionless time step equal to 0.05, while in each of them 150 pseudo-time iterations were accomplished. The mean results of all the real time iterations were finally used for the CSD analysis.

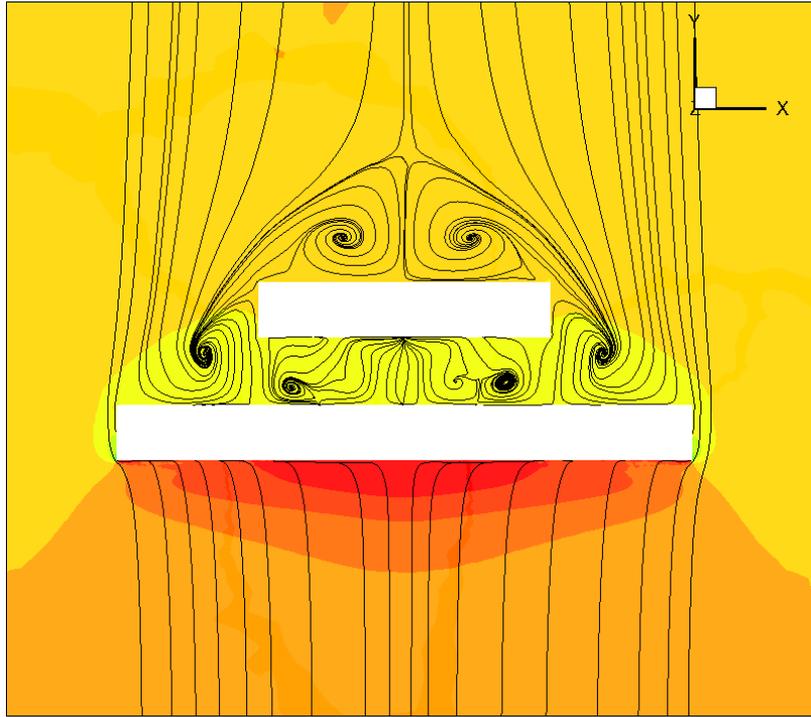


**FIGURE 6.26:** Pressure contours along with velocity streamlines at the YZ plane (bridge case).

The flow phenomena predicted around the two bridge parts proved to be of high complexity. FIGURE 6.26 to FIGURE 6.28 present the extracted dimensionless pressure contours along with velocity streamlines at different views. In FIGURE 6.26 the velocity streamlines are plotted at the mid-plane of the two bridges, demonstrating a number of recirculation phenomena, caused by their bluff body shape.



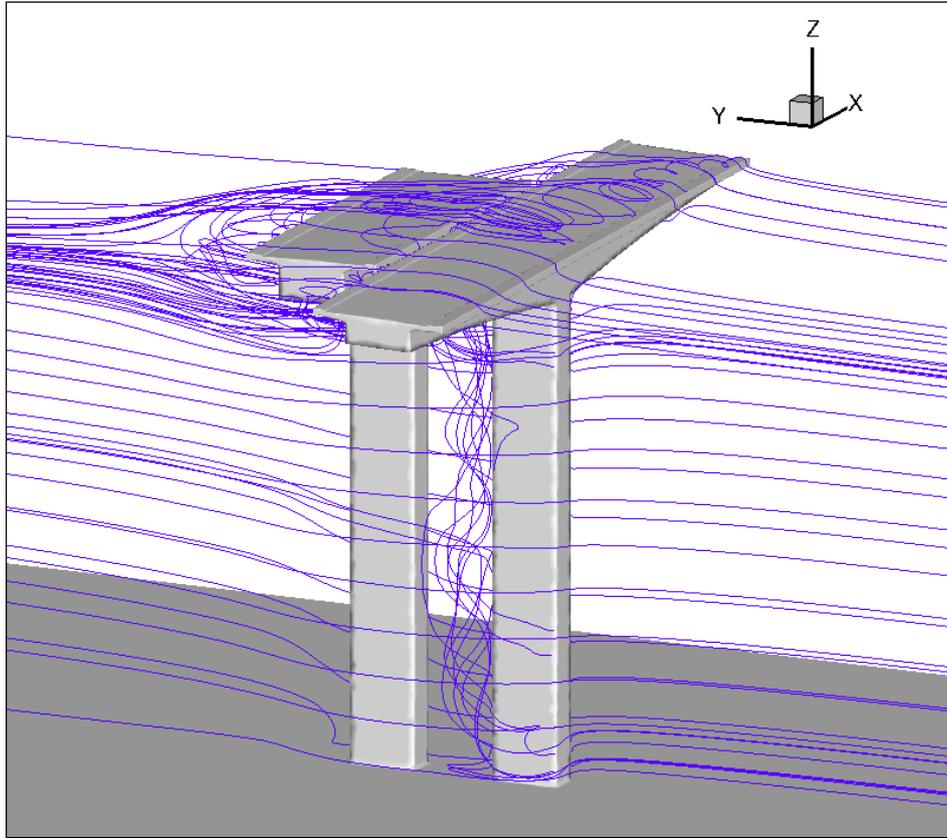
**FIGURE 6.27:** Pressure contours along with velocity streamlines at the XY plane, under the bridge deck (bridge case).



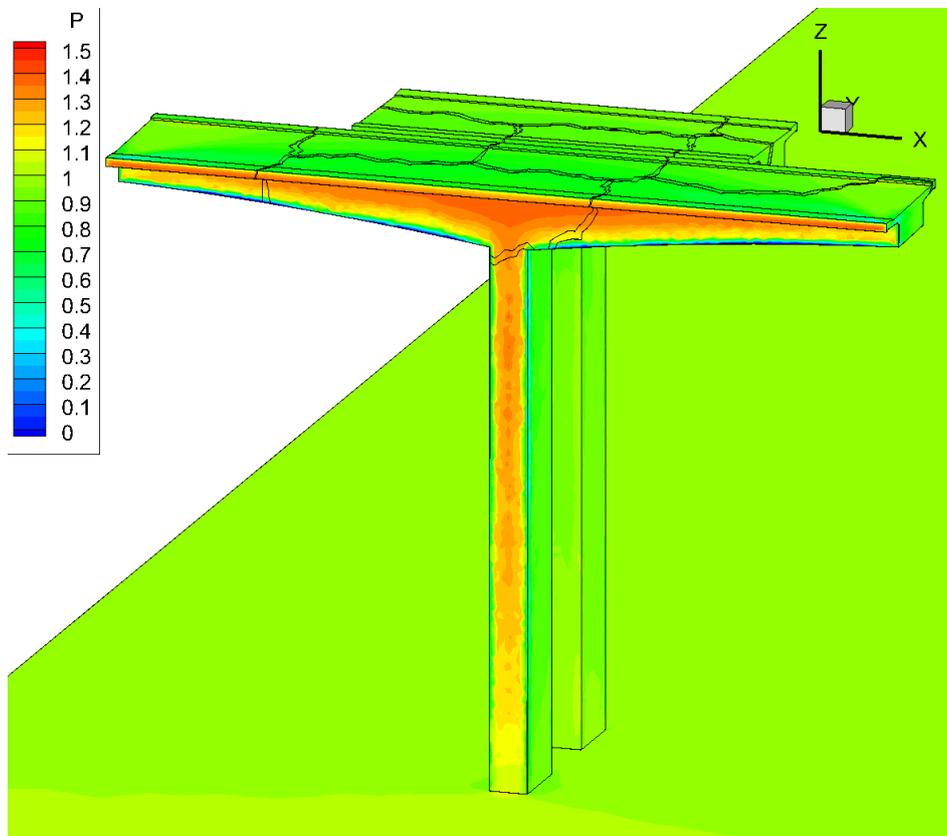
**FIGURE 6.28:** Pressure contours along with velocity streamlines at the  $XY$  plane at the bridge deck (bridge case).

The velocity streamlines in FIGURE 6.27 are plotted on the  $XY$  plane directly under the bridge deck, while in FIGURE 6.28 the corresponding streamlines at the height of the deck are illustrated. FIGURE 6.29 includes a three-dimensional aspect of the flow around the bridge, where the aforementioned flow complexity is obvious. Finally, in FIGURE 6.30 the mean (non-dimensional) pressure is illustrated on the solid surfaces, revealing the areas of higher pressure near the bridge deck, as it was initially expected.

Once the flow prediction has been completed, the obtained force field at the flow side of the interface was transferred to the corresponding structural one. For the fulfillment of the PoU coupling method, the nearest structural nodes for each aerodynamic one were identified on the boundary surfaces of the bridge. The aforementioned procedure can be performed either by selecting all nodes that lie inside a certain radius, or by using a predefined number of nodes. The first approach is susceptible to lead to unbalanced interpolations in case of grids with highly different resolutions, since the number of structural nodes linked with each aerodynamic one depends on the density of the grids.

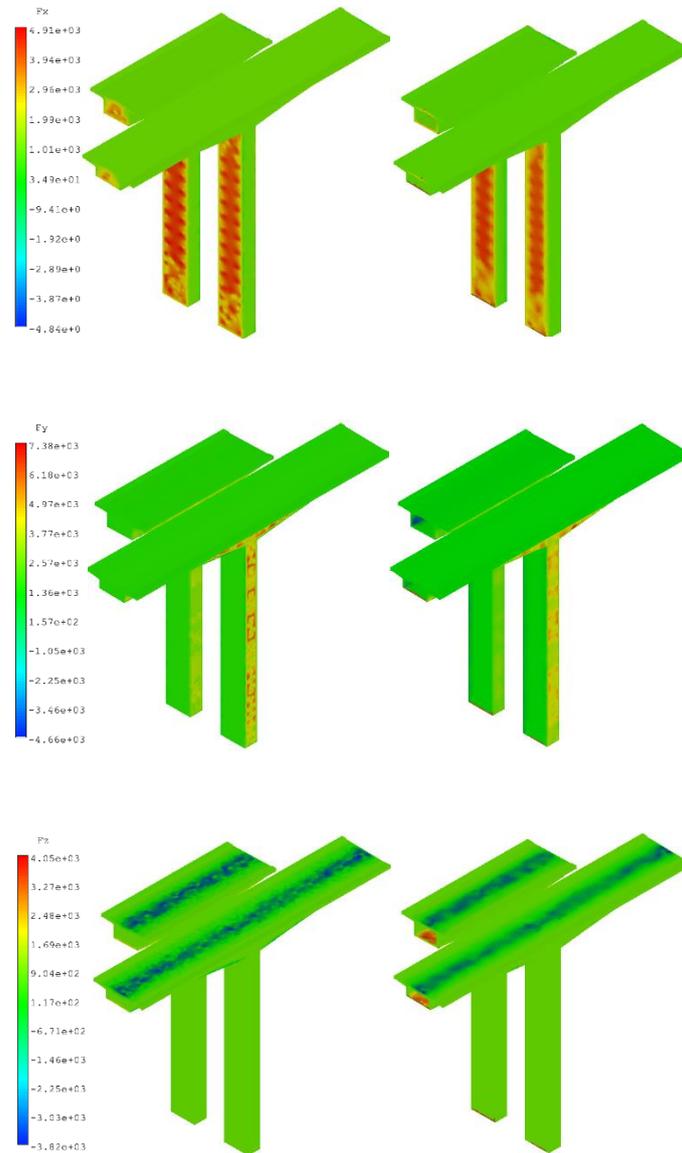


*FIGURE 6.29: Velocity streamlines over the bridges geometry.*



*FIGURE 6.30: Pressure contours on the solid surfaces (bridge case).*

Unlike it, the second approach, according to which the number of structural points is limited beforehand, results in sparse matrix systems. Therefore, the latter strategy was followed for this simulation with the use of octree data structures. At first, the nearest structural grid node was identified for each aerodynamic one, while at next the rest required nodes were selected with the aid of the structural mesh topology. An additional advantage of this approach is the no need for defining a support radius, since it is automatically computed as the distance of each aerodynamic grid node from the uttermost one in the specific patch. The number of structural nodes, used for the construction of the local interpolants in the current test case, was set equal to 12.



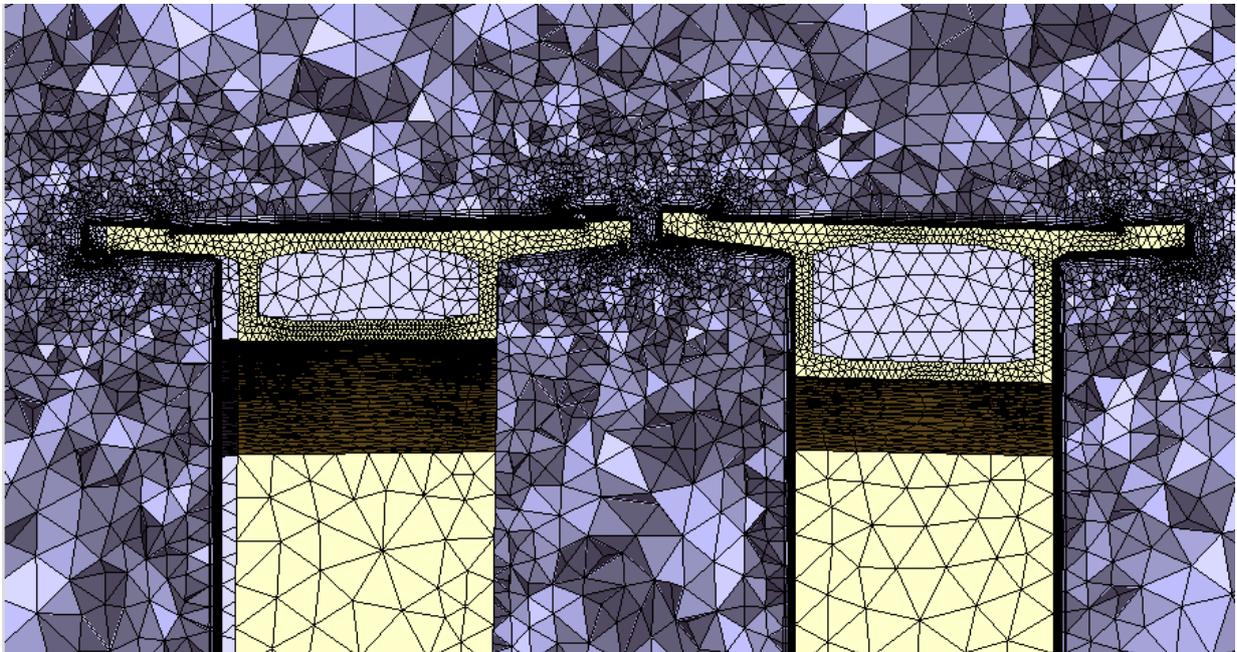
**FIGURE 6.31:** Force components' distribution at CFD (left) and CSD (right) meshes (bridge case).

The wall-clock time, required to link the boundary nodes of the two computational grids and construct the coupling matrix  $\mathbf{H}$ , was 4,814 sec. Unfortunately, it was impossible to perform a simulation with the naïve RBF-based data transfer method (to compare with the utilized PoU approach) due to memory limitations of the available computer systems; such a run would require to invert a matrix of size  $179,069 \times 179,069$ , while it would be excessively time-consuming and consequently worthless. Although the proposed scheme might seem a

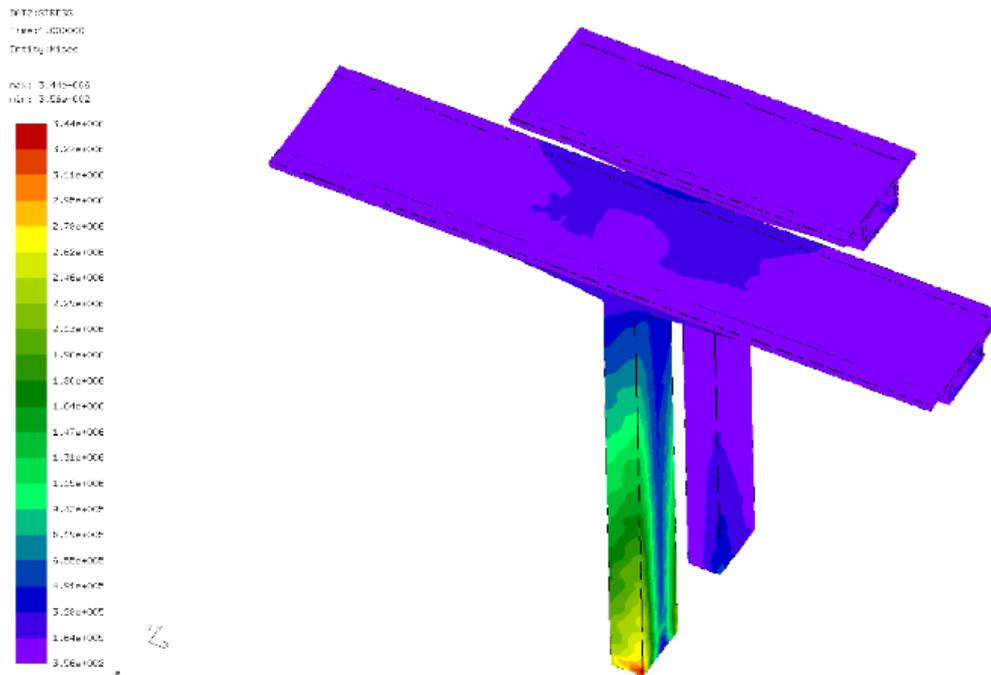
rather tedious procedure, it was revealed to be quite efficient, since the construction of the global coupling matrix is performed only once at the beginning of the simulation; the same stands even for dynamic problems. The size of the structural mesh slightly influences the computational performance of the procedure, since the size of the coupling matrix depends mainly on the number of the aerodynamic boundary nodes; only the nearest nodes identification process will differ. The (wall-clock) computational time for the interpolation of the displacement field was 8.3 sec. FIGURE 6.31 illustrates the distribution of force components over the boundary surfaces of both the CFD and CSD computational meshes. The obtained qualitative results demonstrate a good agreement of force distributions on the CFD and CSD meshes, highlighting the potential of the PoU method to provide accurately physical force transfer over flow/structural interfaces. Unlike it, the full method destroys the local character of the force distribution, due to the transpose of the global coupling matrix (Ahrem *et al.*, 2006).

**TABLE 6.5:** Mechanical properties of the structure (bridge case).

<i>Specific mass – <math>\rho</math> (Kg/m<sup>3</sup>)</i>	3080.25
<i>Young's modulus – E 1 (N/m<sup>2</sup>)</i>	50,838,148,150
<i>Young's modulus – E 2 (N/m<sup>2</sup>)</i>	32,838,148,150
<i>Young's modulus – E 3 (N/m<sup>2</sup>)</i>	328,381,48,150
<i>Poisson's ratio – <math>\nu</math>1</i>	0.300
<i>Poisson's ratio – <math>\nu</math>2</i>	0.193
<i>Poisson's ratio – <math>\nu</math>3</i>	0.193
<i>Shear module G1 (N/m<sup>2</sup>)</i>	20,335,259,260
<i>Shear module G2 (N/m<sup>2</sup>)</i>	13,135,259,260
<i>Shear module G2 (N/m<sup>2</sup>)</i>	13,135,259,260
<i>Structural damping – <math>\xi</math> (%)</i>	5

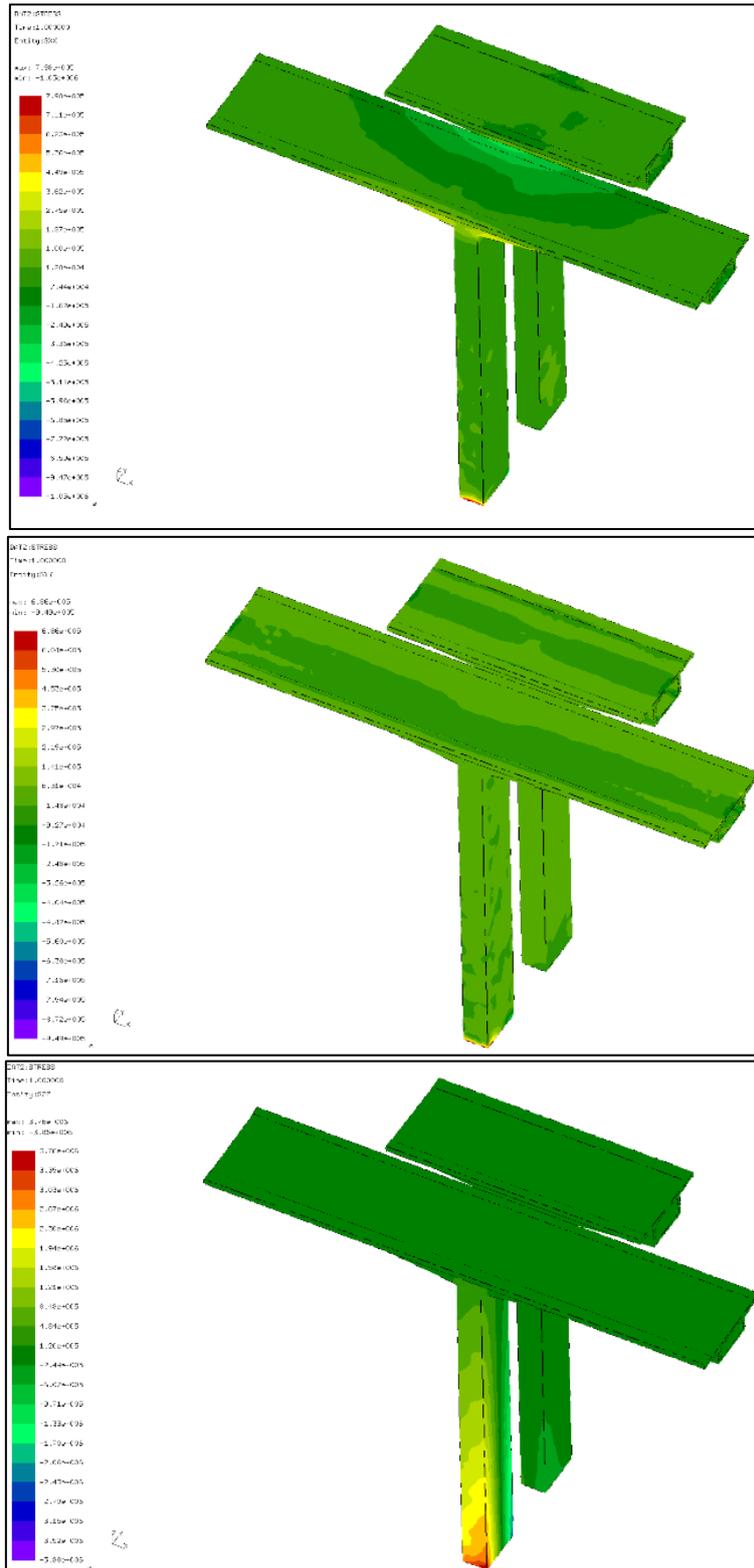


**FIGURE 6.32:** Magnified deformation of the bridge under the wind load (bridge case).



**FIGURE 6.33:** *Computed von Mises stresses of the bridge (bridge case).*

Subsequently, using the aforementioned force loading, a dynamic analysis was performed with the *CalculiX* open source software. The structural properties of reinforced concrete were calculated with the composite mixture theory (Soboyejo, 2002). In particular, they are based on a B45 quality of concrete and a B500s quality of steel; details of them are presented in TABLE 6.5.



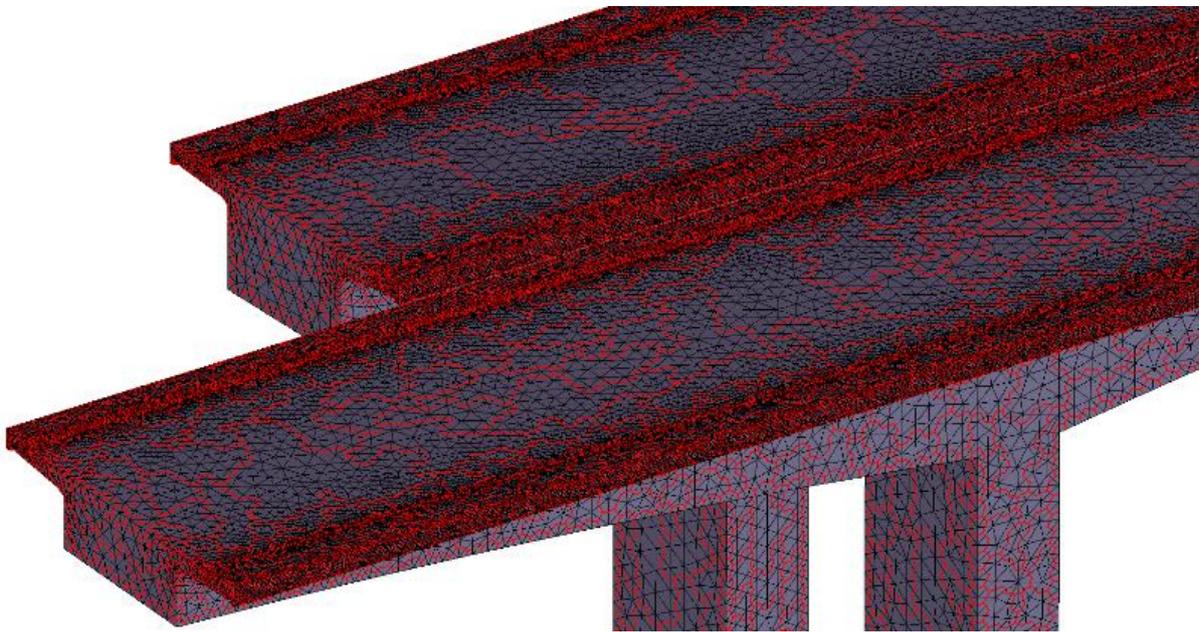
**FIGURE 6.34:** Computed  $S_{xx}$ ,  $S_{yy}$  and  $S_{zz}$  stresses on the bridge (bridge case).

FIGURE 6.32 illustrates the extracted deformation of the bridges' model. The two parts bend in opposite directions, indicating a risk of impact under the wind loading. The bridge parts mainly deform at  $y$ -direction, with a smaller bending at their weak axis, due to lateral buckling of the pylons. The maximum computed displacements were  $2.8\text{ cm}$  in  $y$ -axis and  $0.4\text{ cm}$  in  $x$ -axis. As expected, the left bridge part was deformed more than the right one, as it

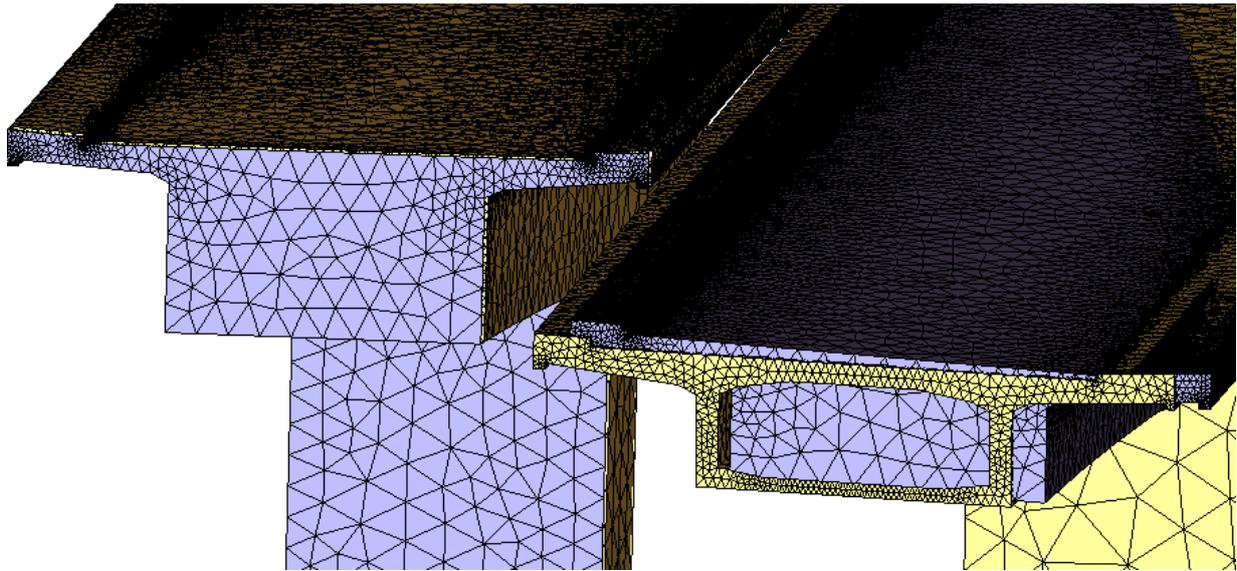
receives the main wind load. The opposite deformation caused at the two bridge parts in  $y$ -axis is justified by the recirculation phenomena revealed between them.

In FIGURE 6.33 the von Mises stresses, derived by the aforementioned analysis, are presented. As expected, the maximum stresses appear at the fixed root of the left bridge pylon. In addition, high stresses are revealed at the windward connection of the left bridge's pylon with the deck. Unlike it, the right bridge part is less affected, because it is inside the wake of the left one. Similar attitude have the main stresses, having their maximum values at the root area. In FIGURE 6.34 the stresses at the three main axes are presented; in  $x$ -axis the maximum value was  $-1.03 \cdot 10^6 \text{ Nt/m}^2$  while in  $y$ - and  $z$ -axes  $-9.49 \cdot 10^5 \text{ Nt/m}^2$  and  $-3.88 \cdot 10^6 \text{ Nt/m}^2$ , respectively.

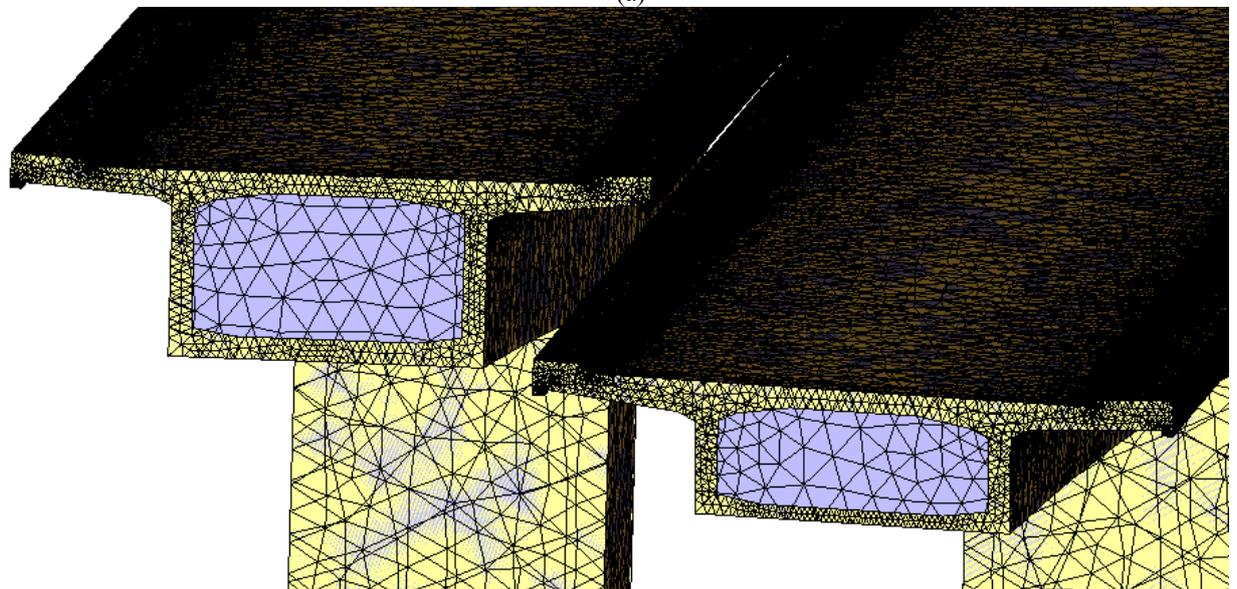
Since the structural analysis has been completed the deflections of the boundaries were transferred to the flow grid, while at next a new flow volume grid was generated, incorporating the new interface location. Four coarser grids were generated, employing the pre-described agglomeration procedure, to enhance the computational performance of the mesh deformation step; the number of boundary nodes was reduced from 179,069 to 53,090, while the (wall-clock) computation time for the calculation of the RBF coefficient vectors was 1,596 sec. In FIGURE 6.35 the control areas, obtained by the aforementioned agglomeration procedure, are depicted. The nodes, used by the RBFs mesh deformation method to construct the required interpolant, were the ones with the minimum distance from their corresponding area weighted centers (Strofylyas *et al.*, 2015). The wall-clock time, required for the interpolation of the displacements to the volume flow grid, was 916 sec. In FIGURE 6.36 the flow computational mesh prior and after the implementation of the deformation step is illustrated.



**FIGURE 6.35:** Control areas of the coarser grid generated with the agglomeration procedure (bridge case).



(a)



(b)

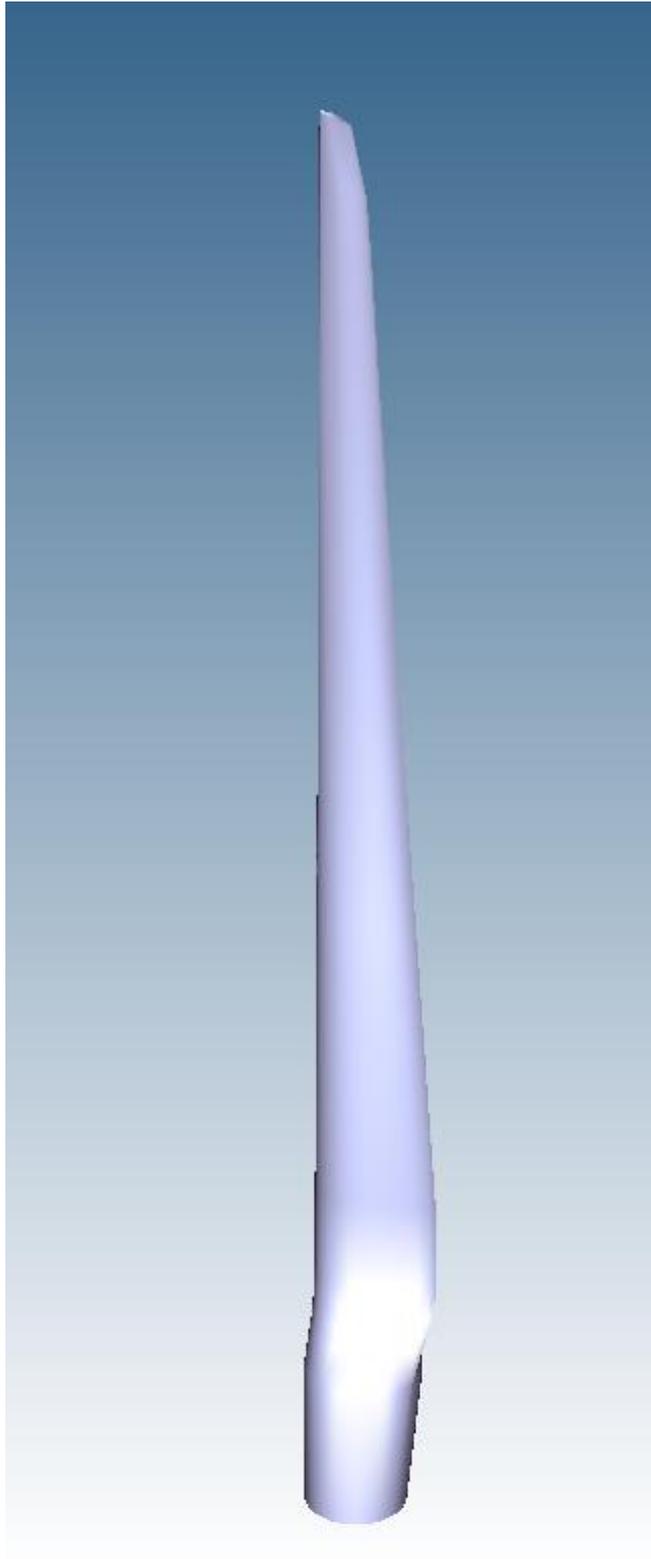
**FIGURE 6.36:** Fluid (gray) and structural (yellow) computational grids prior (a) and after (b) the fluid mesh deformation step (bridge case).

### 6.5.3 NREL offshore wind turbine

In this section the results of numerical tests are presented for the NREL offshore 5-MW baseline wind turbine (FIGURE 6.37) (Jonkman *et al.*, 2009). This wind turbine is a conventional three-bladed upwind variable-speed variable blade-pitch-to-tip-controlled turbine. The main characteristics of this wind turbine are included in TABLE 6.6. A detailed description of its geometrical characteristics can be found in (Jonkman *et al.*, 2009).

**TABLE 6.6:** Gross properties chosen for the NREL-5MW Baseline Wind Turbine (Jonkman et al., 2009).

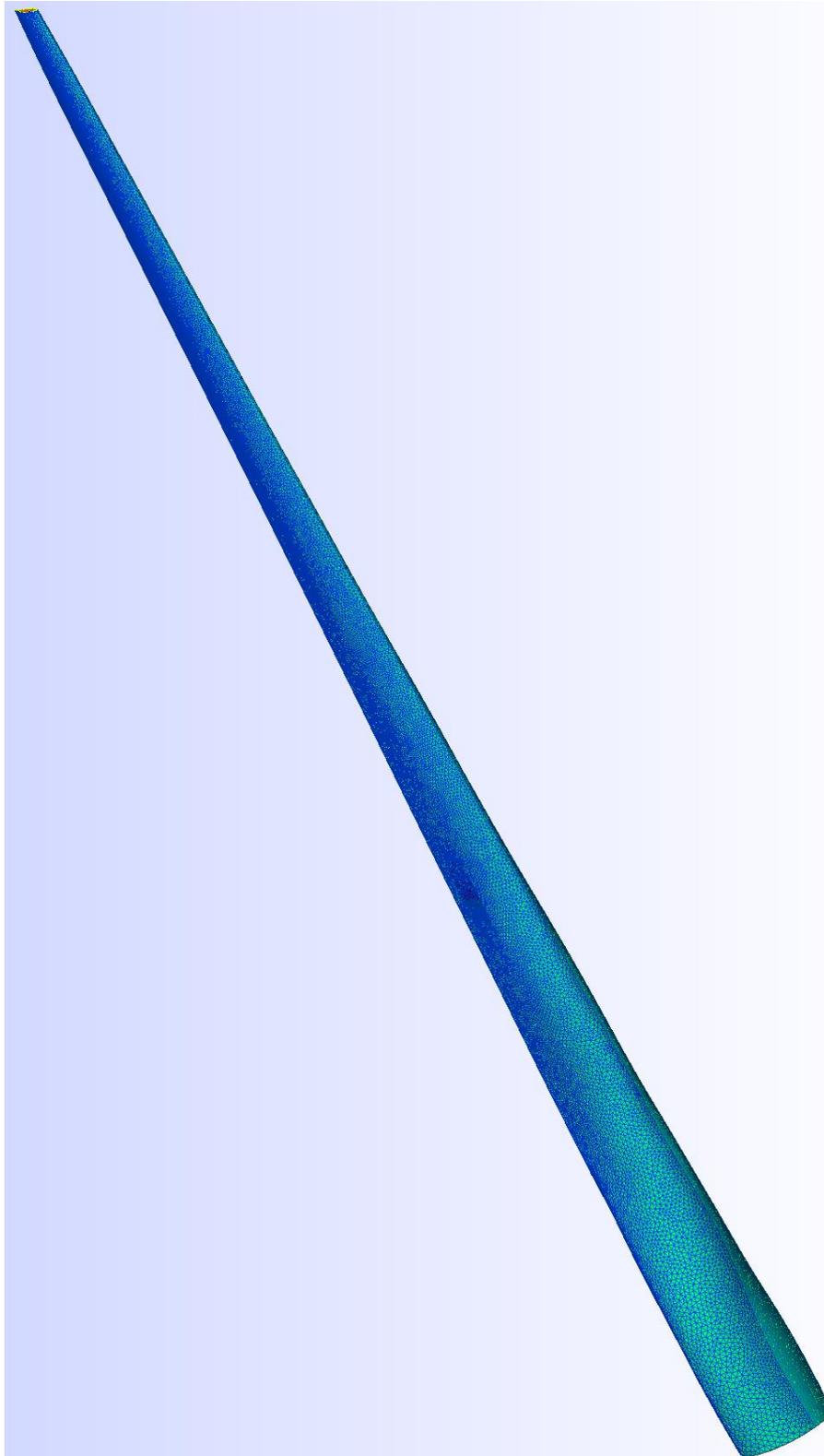
<i>Rating</i>	5 MW
<i>Rotor Orientation, Configuration</i>	Upwind, 3 Blades
<i>Control</i>	Variable Speed, Collective Pitch
<i>Drivetrain</i>	High Speed, Multiple- Stage Gearbox
<i>Rotor, Hub Diameter</i>	126 m, 3 m
<i>Hub Height</i>	90 m
<i>Cut-In, Rated, Cut-Out Wind Speed</i>	3 m/s, 11.4 m/s, 25 m/s
<i>Cut-In, Rated Rotor Speed</i>	6.9 rpm, 12.1 rpm
<i>Rated Tip Speed</i>	80 m/s
<i>Overhang, Shaft Tilt, Precone</i>	5 m, 5°, 2.5°
<i>Rotor Mass</i>	110,000 kg
<i>Nacelle Mass</i>	240,000 kg
<i>Tower Mass</i>	347,460 kg
<i>Coordinate Location of Overall CM</i>	(-0.2 m, 0.0 m, 64.0 m)



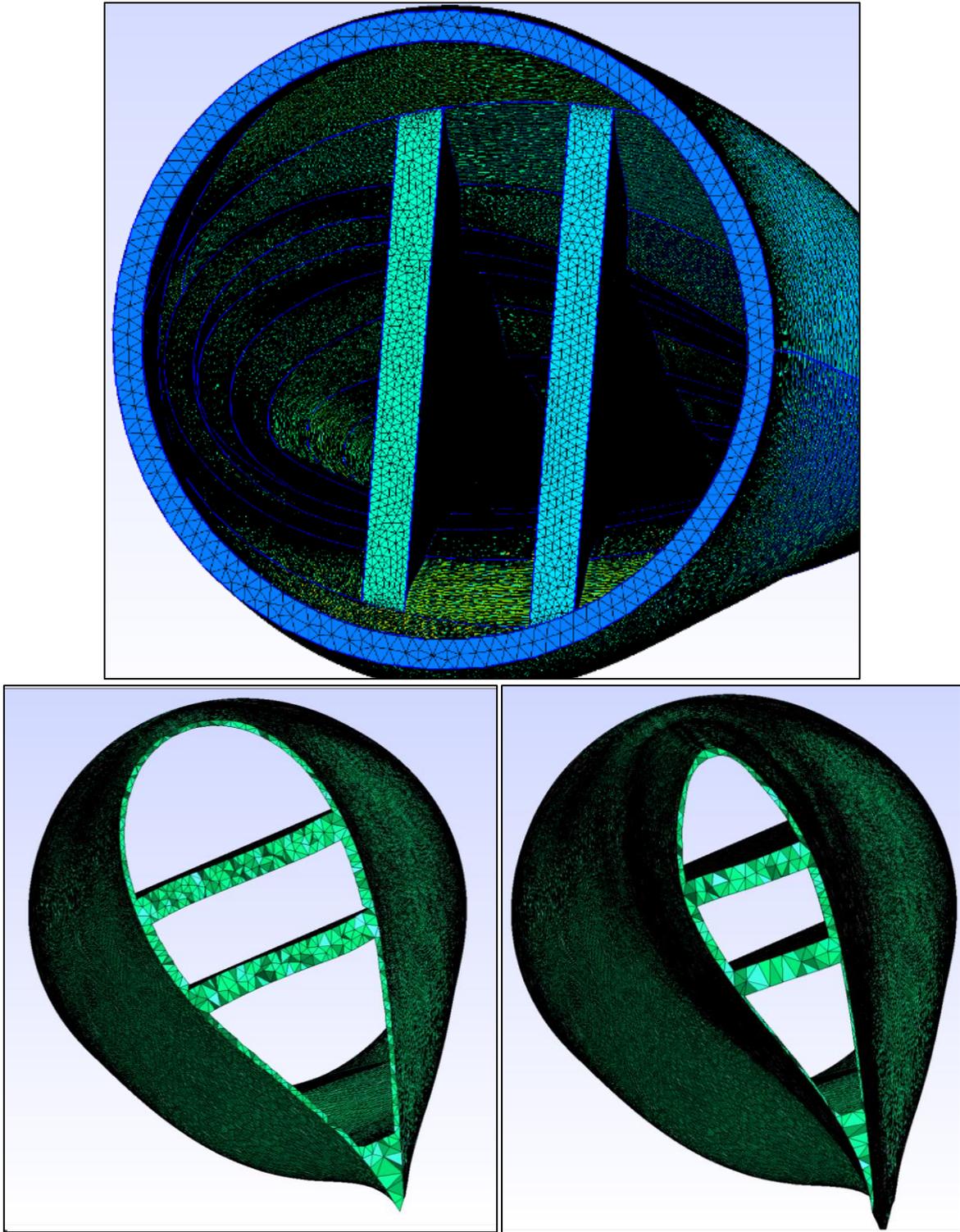
**FIGURE 6.37:** *The NREL-5MW Baseline Wind Turbine blade geometry.*

For the specific model, the creation of an unstructured CSD mesh was performed, consisting of tetrahedral elements (FIGURE 6.38), with higher discretization at areas of high curvature for the better simulation of the geometric model (FIGURE 6.39). In total 167,527 tetrahedral elements (of type C3D4), were used with 52,688 nodes, from which 11,211 nodes are located at the outer surface of the blade. The blade's thickness varies along the chord's

length. In particular, (Charalampous *et al.*, 2015) typical sections of its geometry are described in TABLE 6.7. Along the blade there are 18 sections, each one of which are divided in 5 different parts. Among the aforementioned sections a linear interpolation has been applied.



**FIGURE 6.38:** *General mesh view of the NREL blade.*

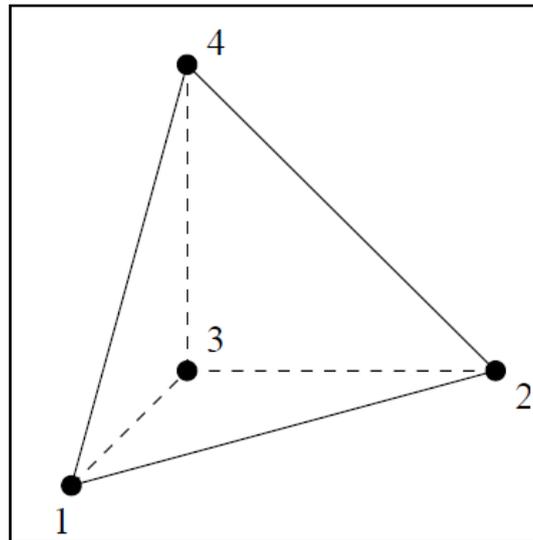


**FIGURE 6.39:** Details of the mesh of the sections 1, 4, 8 (from left to right) (NREL case).

C3D4 elements (FIGURE 6.40) are the most popular elements, which can be generated from the open source software mesh generator *GMSH* (Geuzaine & Remacle, 2014), selected for the creation of a 3D structural mesh. Despite the fact that the elements are quite rigid, their management by *CalculiX* is highly satisfactory for the requirements of the current work. These elements are equipped with one integration point.

**TABLE 6.7:** Distributed Blade Aerodynamic and Geometrical Properties for the NREL blade (Charalampous et al., 2015).

Node	RNodes (m)	Percentage along the stacking line	Aero Twist (Degrees)	Chord	Airfoil Type	1° & 5° Part Width (m)	2° & 3° Part Width (m)	Width of the 3° Part (m)
1	0	0	10.480	3.542	Cylinder1	0.0435	0.0352	0.0435
2	5.115	0.083	10.480	3.542	Cylinder2	0.0435	0.0352	0.0435
3	8.505	0.138	14.80	4.185	Cylinder3	0.0541	0.0635	0.0541
4	11.833	0.192	14.80	4.550	Cylinder4	0.0466	0.0677	0.0466
5	13.990	0.227	13.308	4.557	DU40_A17	0.0456	0.10199	0.0456
6	15.839	0.257	11.480	4.652	DU35_A17	0.0456	0.10199	0.0456
7	19.969	0.324	10.162	4.458	DU35_A17	0.0456	0.0931	0.0456
8	24.036	0.390	9.011	4.249	DU30_A17	0.00564	0.0837	0.00564
9	28.166	0.457	7.795	4.007	DU25_A17	0.0456	0.0639	0.0456
10	32.234	0.523	6.544	3.748	DU25_A17	0.0456	0.0522	0.0456
11	36.363	0.590	5.361	3.502	DU21_A17	0.0276	0.0291	0.0276
12	40.431	0.656	4.188	3.256	DU21_A177	0.0276	0.0291	0.0276
13	44.560	0.723	3.125	3.010	NACA64_A17	0.0276	0.0291	0.0276
14	48.690	0.790	2.319	2.764	NACA64_A17	0.0276	0.0291	0.0276
15	52.758	0.856	1.526	2.518	NACA64_A17	0.0276	0.0291	0.0276
16	56.147	0.911	0.863	2.313	NACA64_A17	0.0276	0.0291	0.0276
17	58.859	0.955	0.37	2.086	NACA64_A17	0.0276	0.0276	0.0276
18	61.6333	1	0.106	1.419	NACA64_A17	0.0276	0.0276	0.0276



**FIGURE 6.40:** Tetrahedral C3D4 elements (Dhondt, 2014).

Herein, the cases that will be examined are dynamic ones with linear geometry (small deformations). The composite material's properties used are presented at the following TABLE 6.7.

The test case concerns the aeroelastic analysis of a blade for wind velocity 11 m/s while the

rotor speed is 10 *rpm*. The aerodynamic mesh utilized for the CFD analysis was composed of 1007 vertices (structured mesh of 19 sections with 53 vertices each). However, the Data Transfer procedure, described above, is formed in a way to handle aerodynamic grids with bigger resolution than the structural ones or at least of similar size, since this is the case in most simulations with RANS CFD solvers. To alleviate this problem a refined, intermediate, grid was constructed, using the initial grid, by interpolating additional sections among the initial ones. For each zone between two cross sections of the initial grid additional 20 cross-sections have been interpolated to create the refined grid. This refined grid is used in every cycle of the FSI simulation. The forces computed by the *GAST\_aero* software, at the nodes of the initial coarse one are evenly distributed to the cross-sections of the refined one, while at next the forces are interpolated from the refined grid to the structural one using the RBF PoU approach. Subsequently, the velocity and displacement fields computed by the CSD solver are interpolated back from the structural to the refined aerodynamic grid, while only the values corresponding to the sections of the initial grid are taken into account, to be used by *GAST\_aero* for the next time step of the FSI procedure.

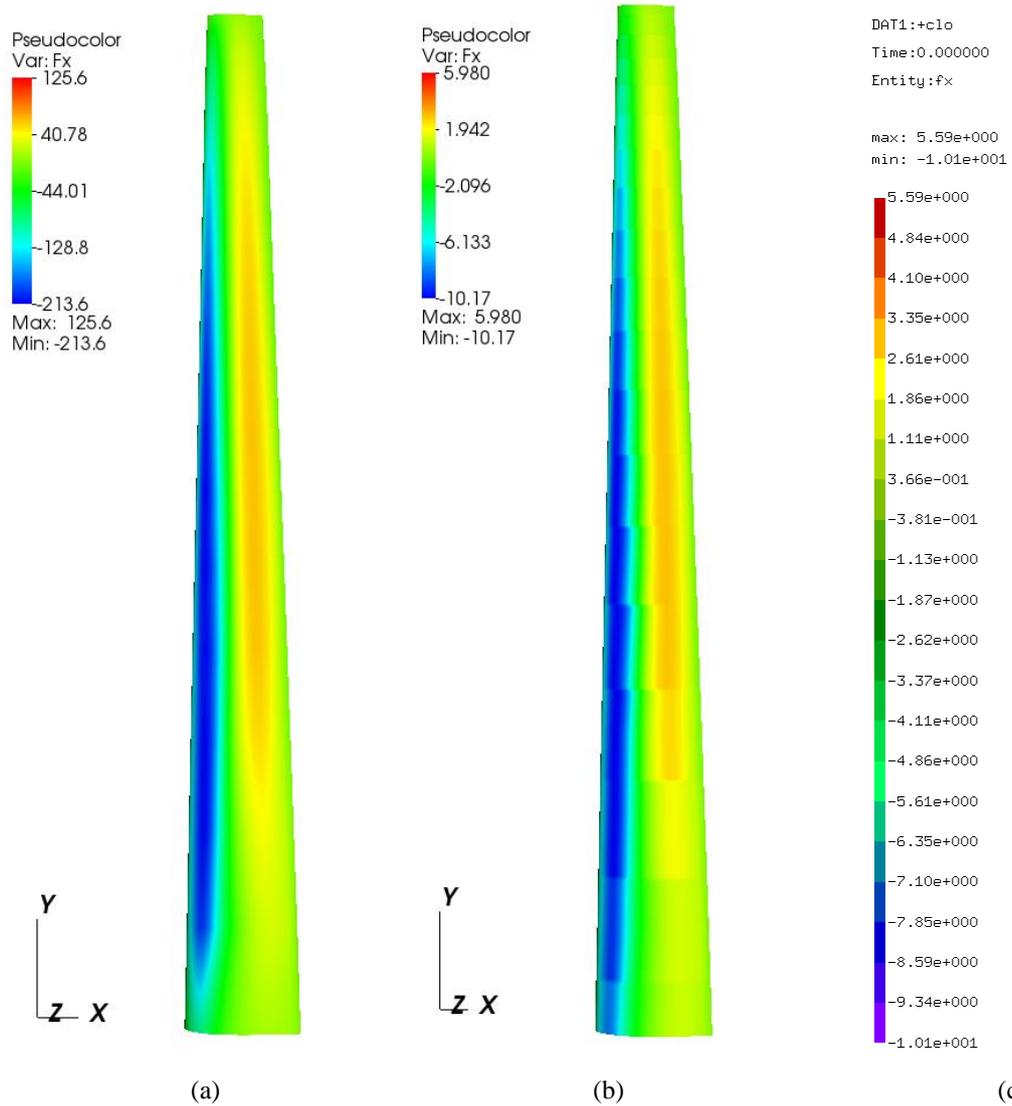
FIGURE 6.41 to FIGURE 6.46 illustrate the distribution of the forces over the blade surface for the three grids (coarse aerodynamic, refined aerodynamic and structural grid). The forces have been uniformly distributed from the coarse aerodynamic grid to the refined one, while at next (using the RBF PoU approach) have been interpolated to the structural grid, in order a CSD analysis to be performed.

From FIGURE 6.47 up to FIGURE 6.51, the deformation of the blade, during the first FSI cycle, is presented. As expected, the shape of the first mode dominates under the concentrated, on every external node, wind load and gravity and centrifugal load.

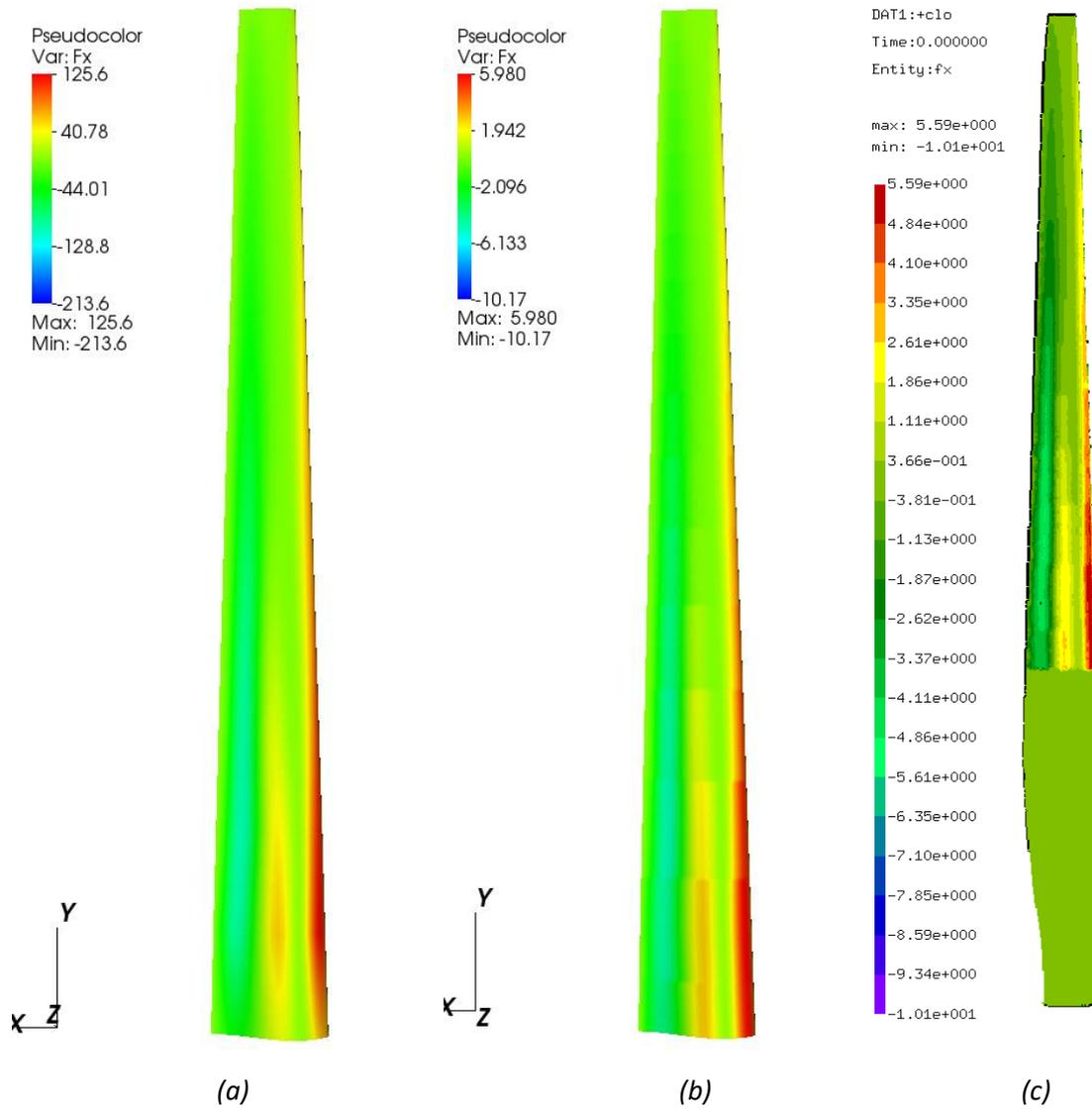
FIGURE 6.52, illustrates the blade's deformation at  $xx'$  axis during the first FSI Cycle. In this orientation, the maximum displacement was 0.06 *m*. Moreover, longitudinal deformation of the blade is illustrated in FIGURE 6.53. In addition, FIGURE 6.54 presents the deformation at  $zz'$  axis, which is the crucial one, with the maximum value being equal to 1.07 *m*. Respectively, at FIGURE 6.55, FIGURE 6.56 and FIGURE 6.57, the velocity distribution at different axes is presented. The maximum velocity appears at  $zz'$  axis and is equal to 36.10 *m/s*.

TABLE 6.8: NREL blade materials properties (NREL case).

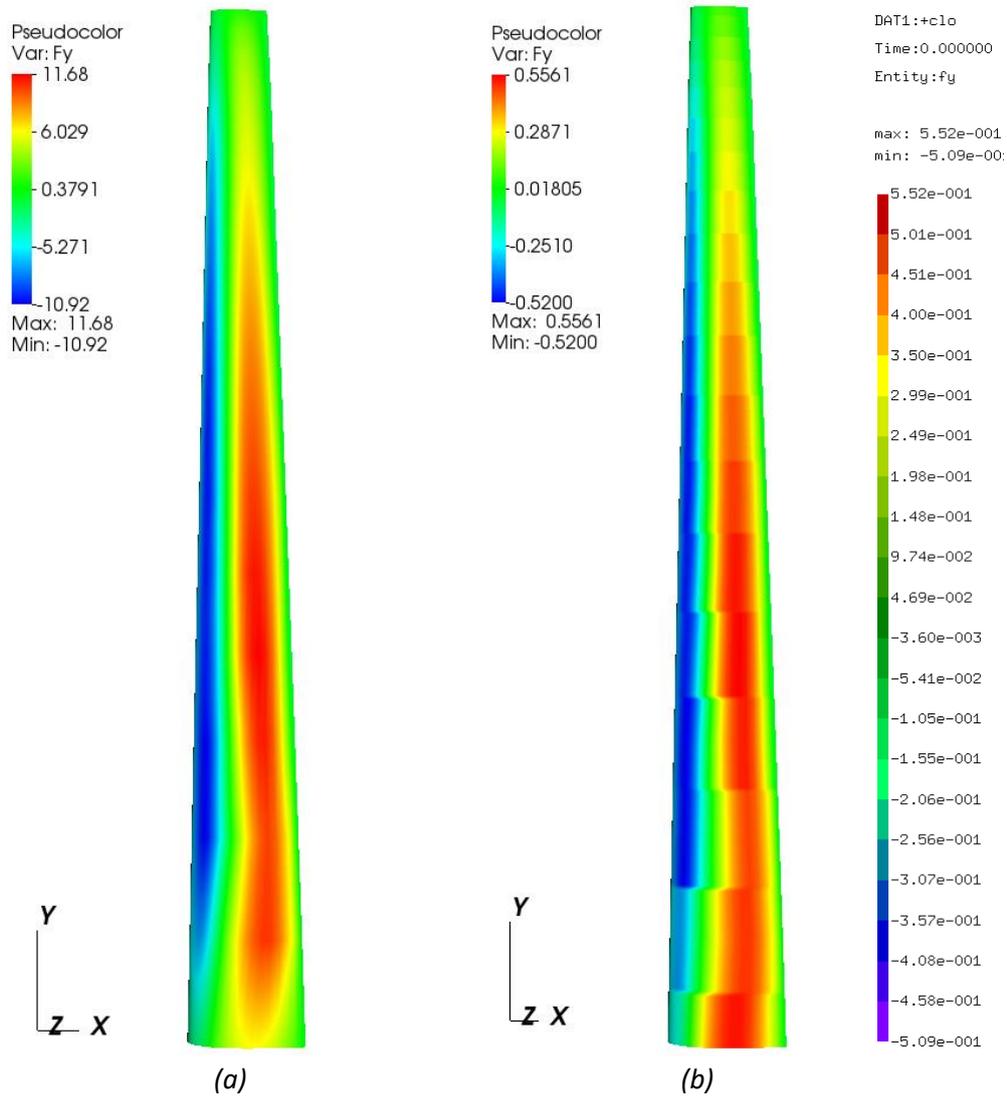
$E_{11}$ [GPa]	$E_{22}$ [GPa]	$E_{33}$ [GPa]	$\nu_{12}$	$\nu_{13}$	$\nu_{14}$	$G_{11}$ [GPa]	$G_{22}$ [GPa]	$G_{33}$ [GPa]	$\rho$ [kg/m <sup>3</sup> ]
44.65	12.96	12.96	0.500	0.500	0.500	4.32	4.32	4.32	7500



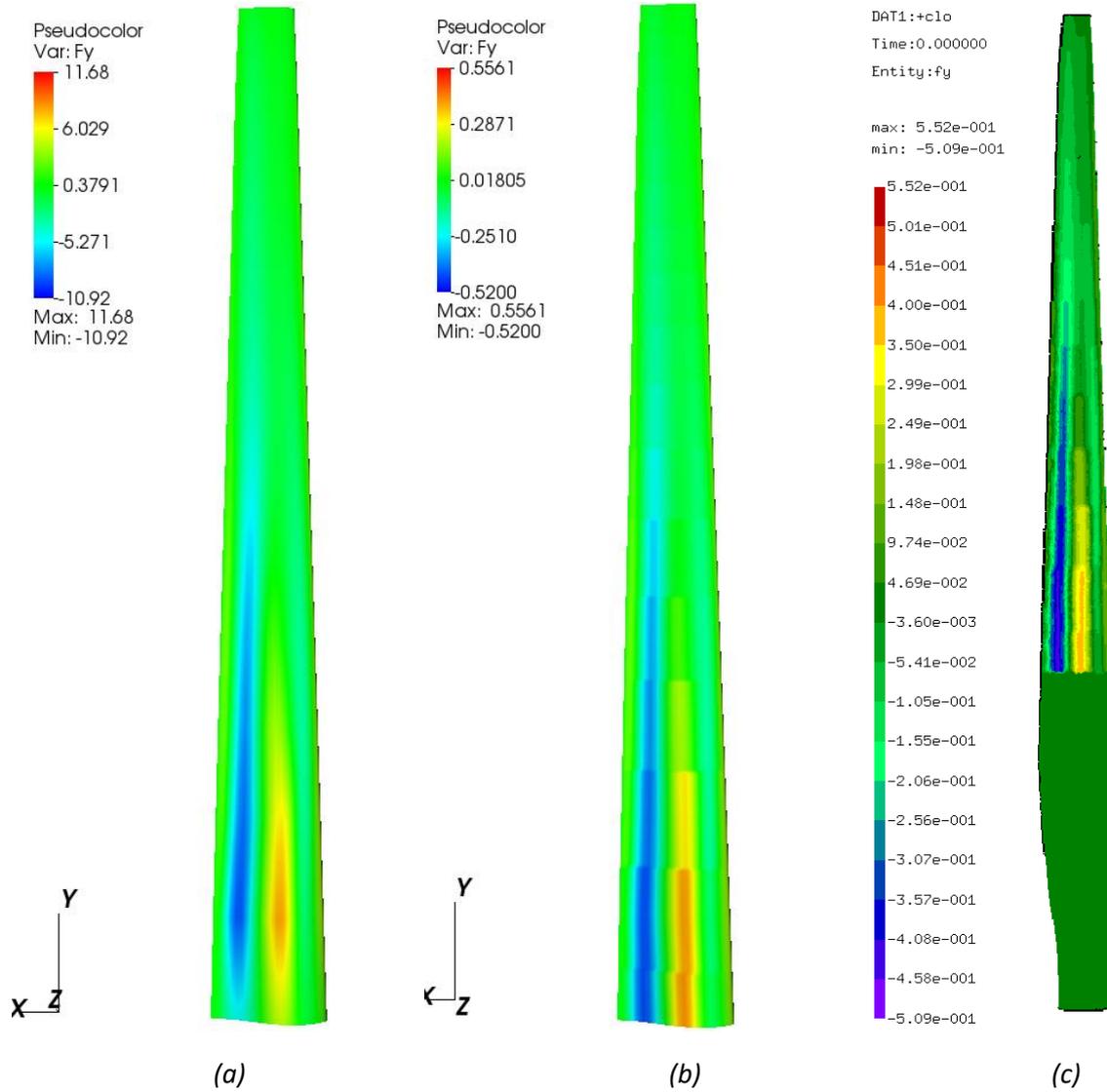
**FIGURE 6.41:** Distributions of loads on  $XX'$  axis at the blade during the first FSI iteration (front view): (a) coarse aerodynamic grid, (b) refined aerodynamic grid, (c) structural grid (NREL case).



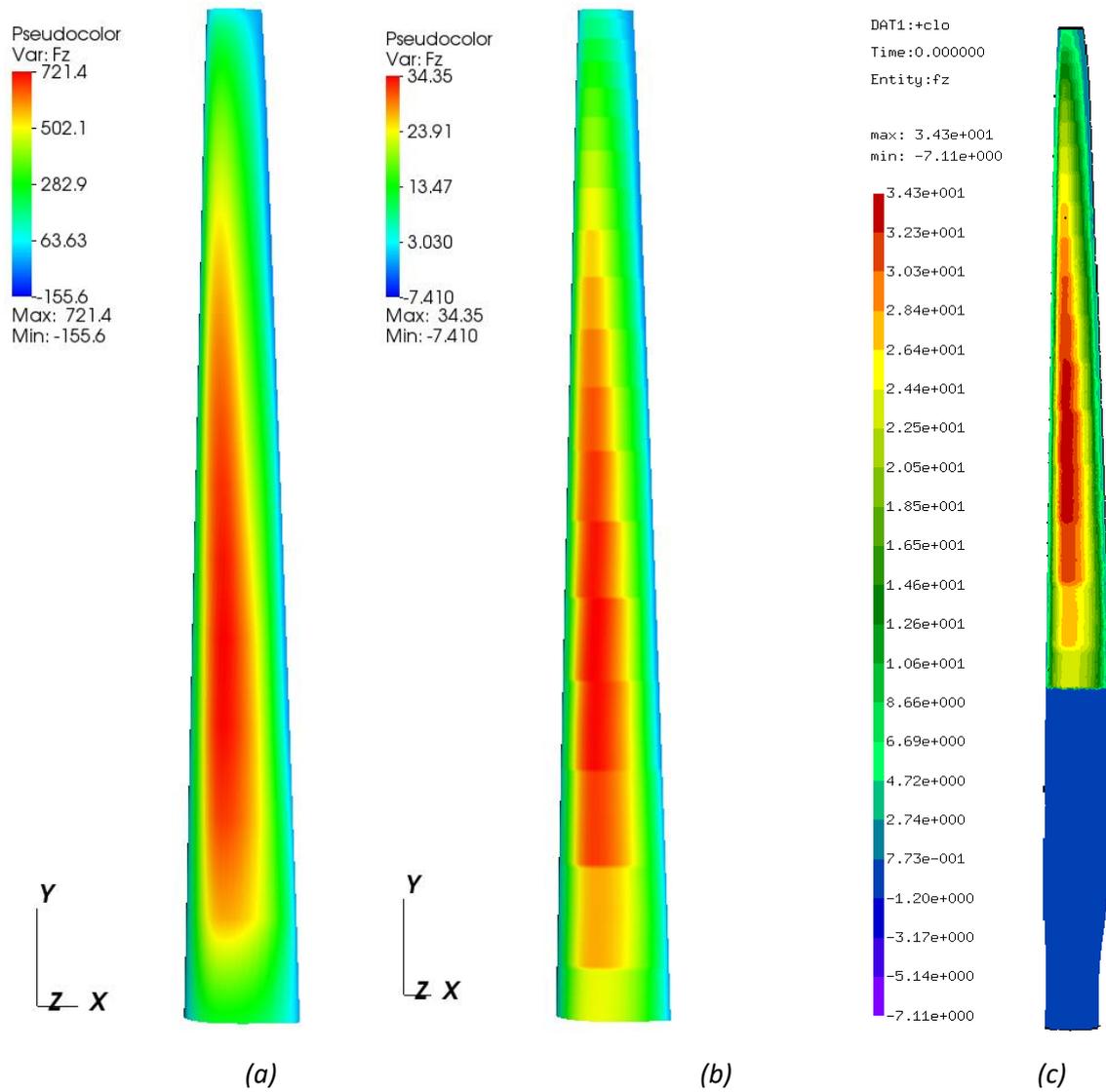
**FIGURE 6.42:** Distributions of loads on  $XX'$  axis at the blade during the first FSI iteration (back view): (a) coarse aerodynamic grid, (b) refined aerodynamic grid, (c) structural grid (NREL case).



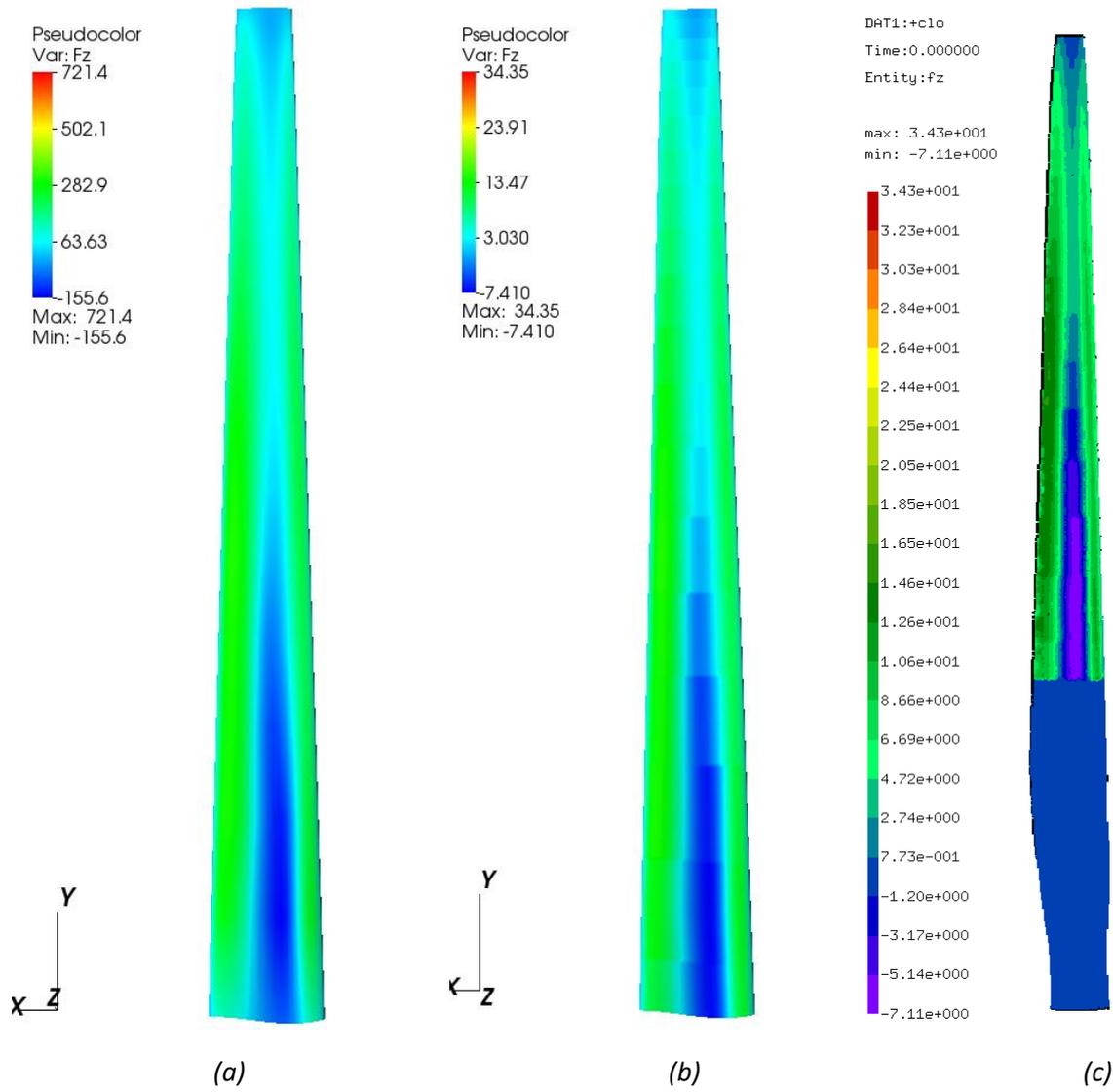
**FIGURE 6.43:** Distributions of loads on  $YY'$  axis at the blade during the first FSI iteration (front view): (a) coarse aerodynamic grid, (b) refined aerodynamic grid, (c) structural grid (NREL case).



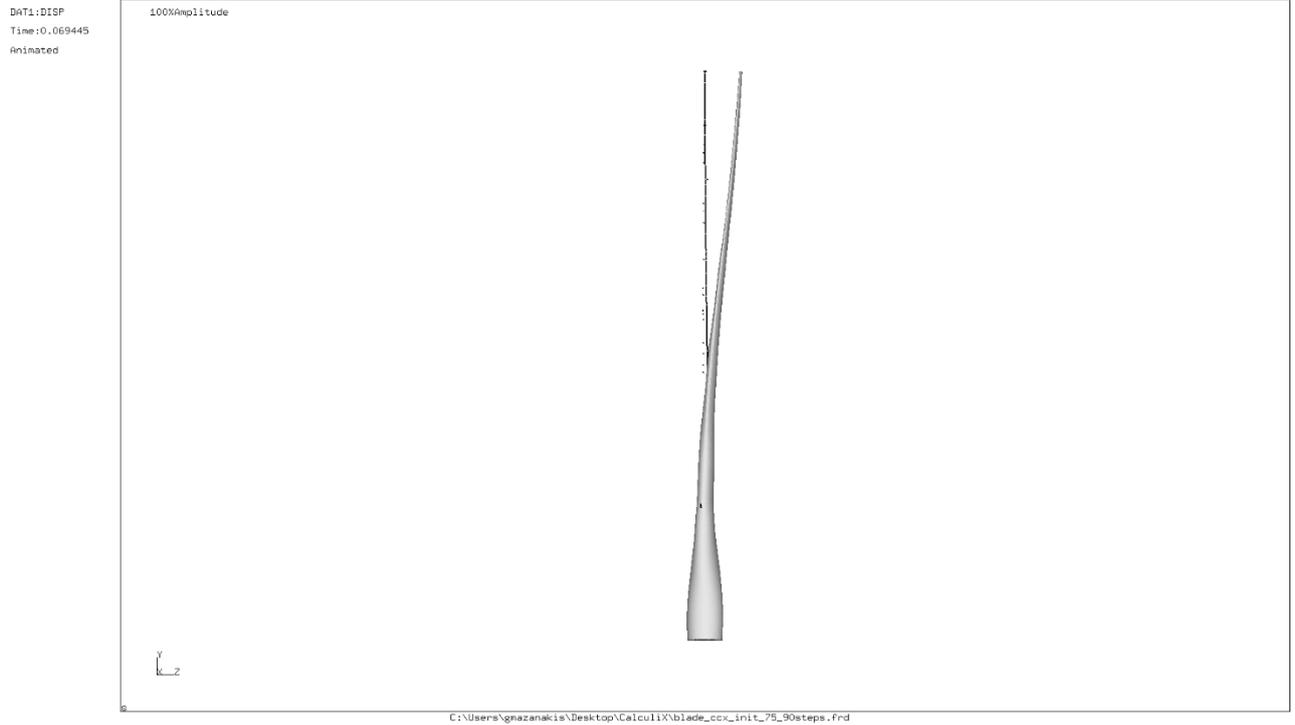
**FIGURE 6.44:** Distributions of loads on  $YY'$  axis at the blade during the first FSI iteration (back view): (a) coarse aerodynamic grid, (b) refined aerodynamic grid, (c) structural grid (NREL case).



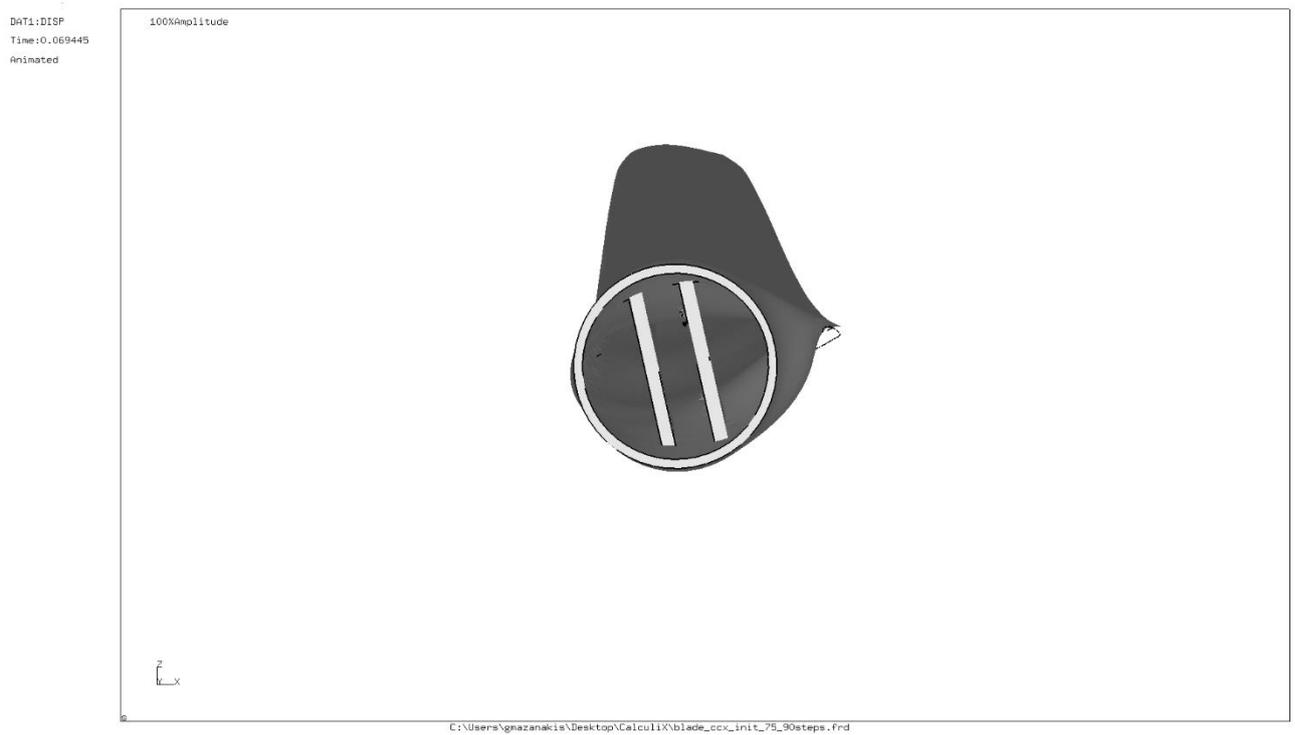
**FIGURE 6.45:** Distributions of loads on ZZ' axis at the blade during the first FSI iteration (front view): (a) coarse aerodynamic grid, (b) refined aerodynamic grid, (c) structural grid (NREL case).



**FIGURE 6.46:** Distributions of loads on ZZ' axis at the blade during the first FSI iteration (back view): (a) coarse aerodynamic grid, (b) refined aerodynamic grid, (c) structural grid (NREL case).

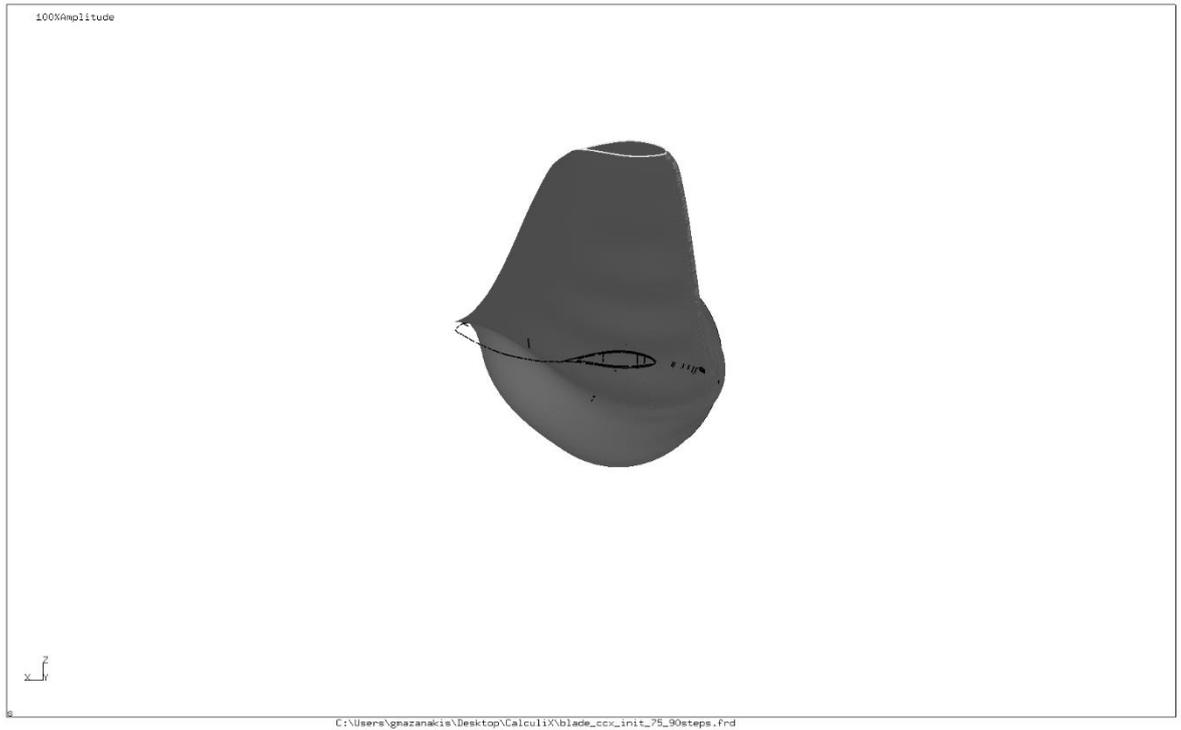


**FIGURE 6.47:** Blade's deformation at YZ plane during the first FSI iteration (NREL case).



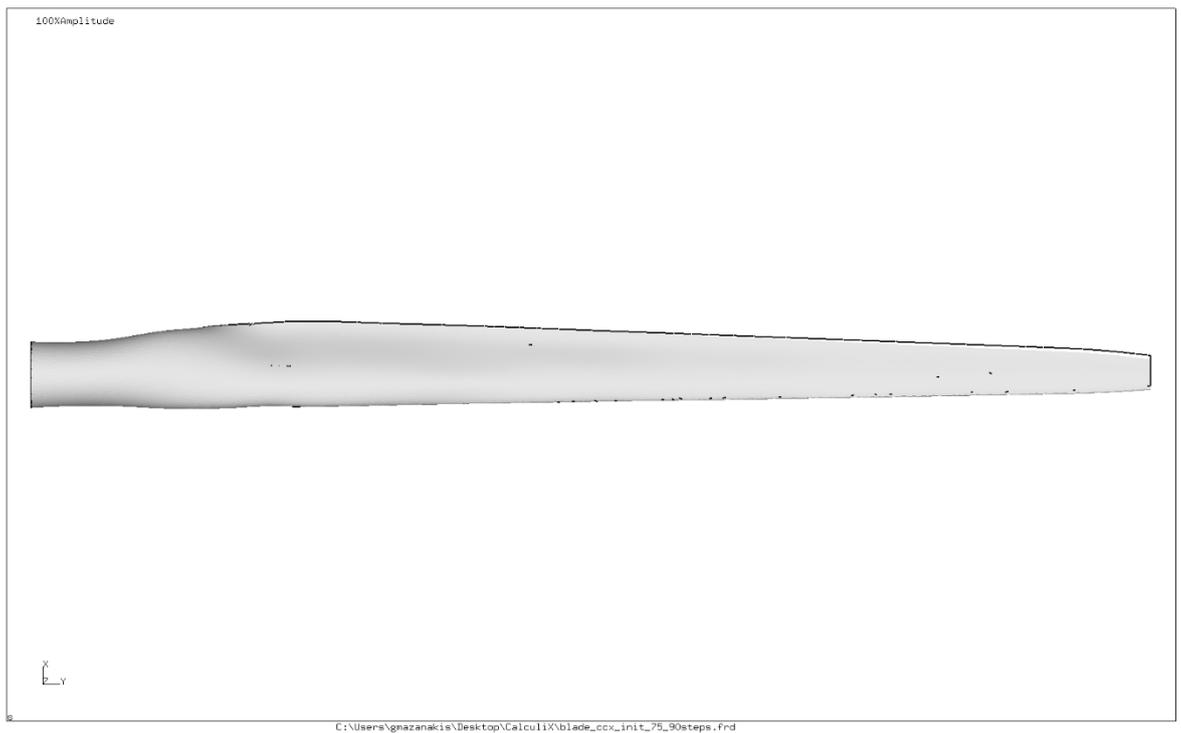
**FIGURE 6.48:** Blade's deformation at ZX plane during the first FSI iteration (NREL case).

DAT1:DISP  
Time:0.069445  
Animated

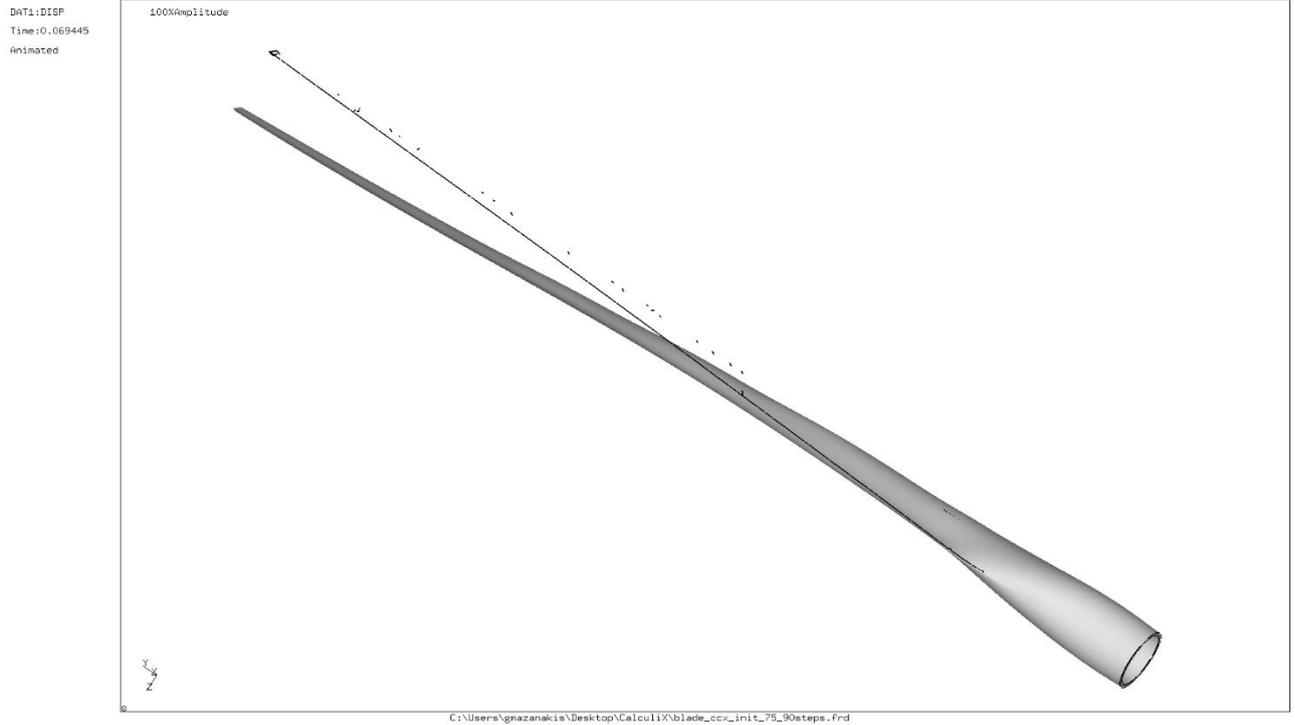


**FIGURE 6.49:** Blade's deformation at XZ plane during the first FSI iteration (NREL case).

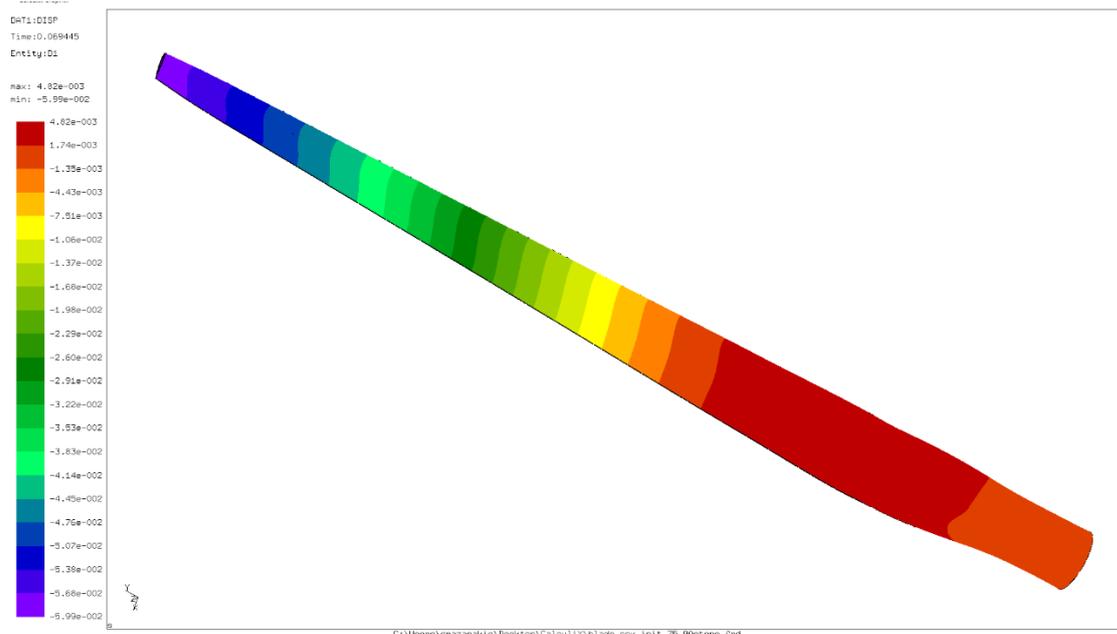
DAT1:DISP  
Time:0.069445  
Animated



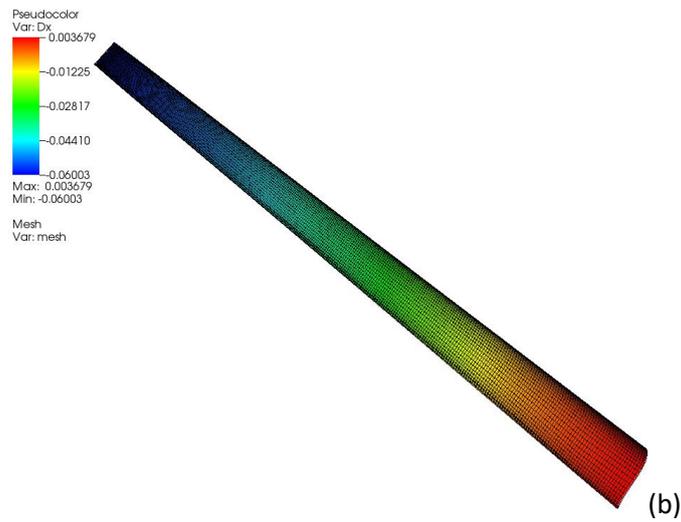
**FIGURE 6.50:** Blade's deformation at YX plane during the first FSI iteration (NREL case).



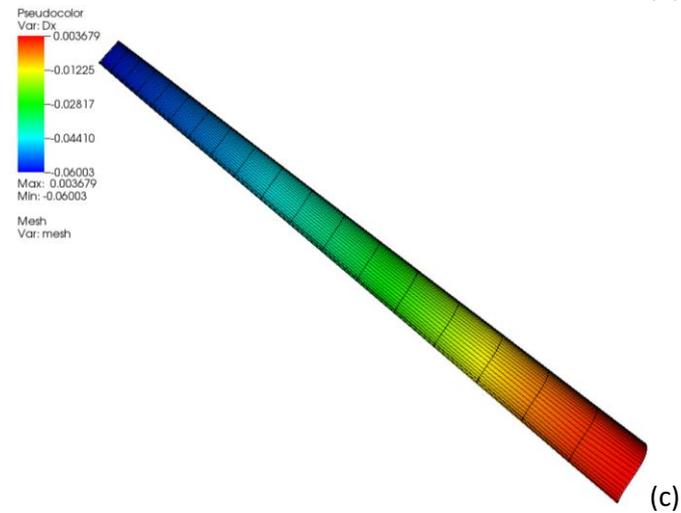
**FIGURE 6.51:** 3D Blade's deformation during the first FSI iteration (NREL case).



(a)

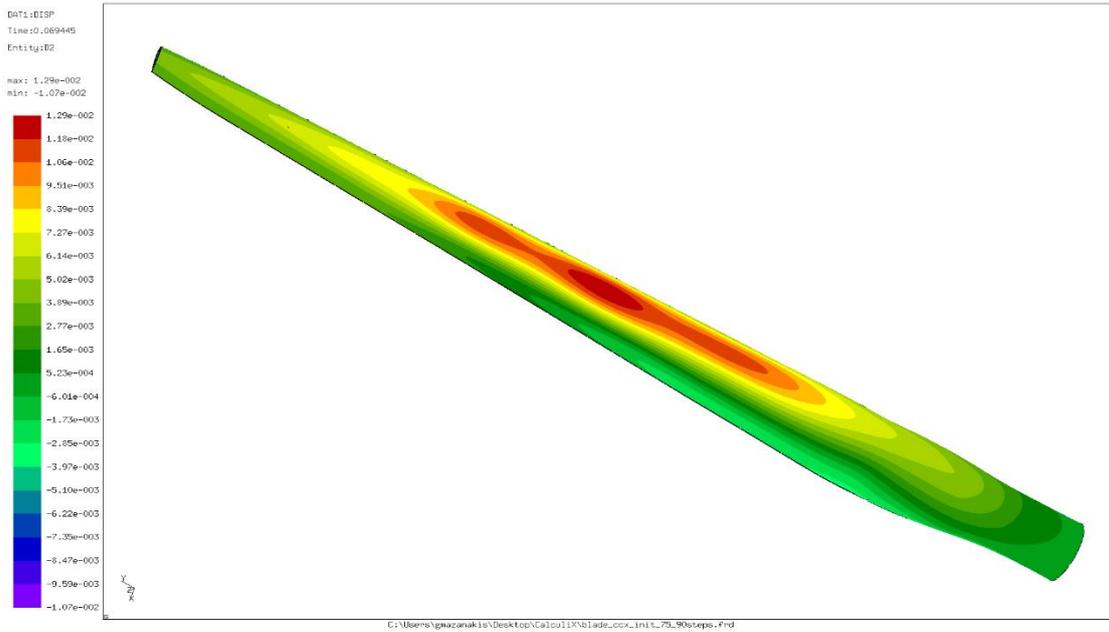


(b)

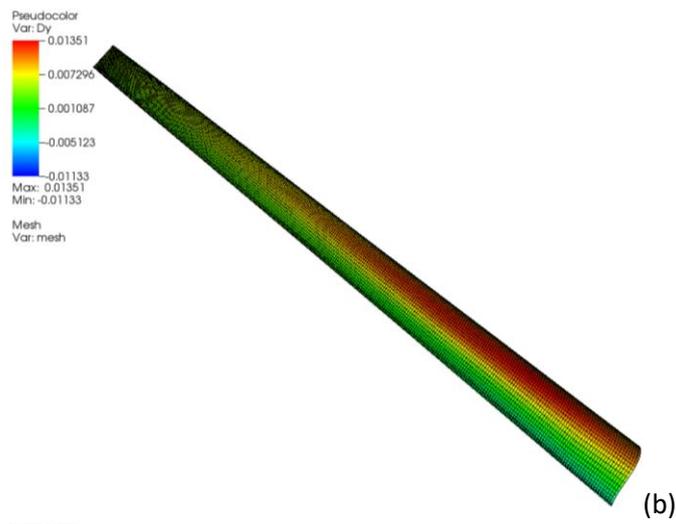


(c)

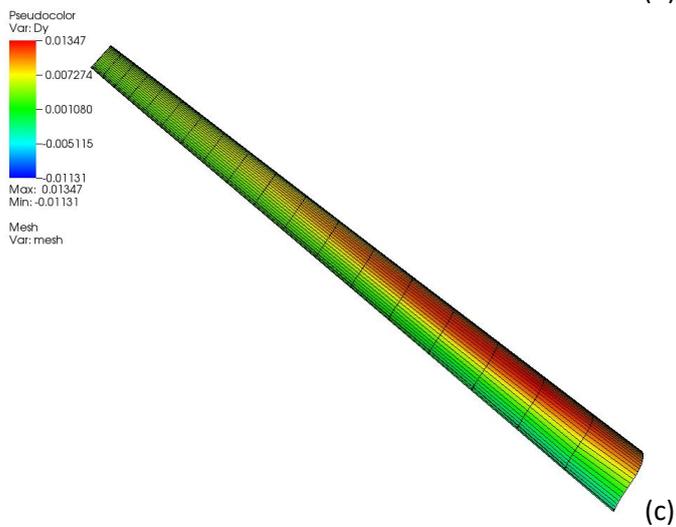
**FIGURE 6.52:** Distributions of the deformation at the blade during the first FSI iteration at  $XX'$  axis: (a) structural grid, (b) refined aerodynamic grid, (c) coarse aerodynamic grid (NREL case).



(a)

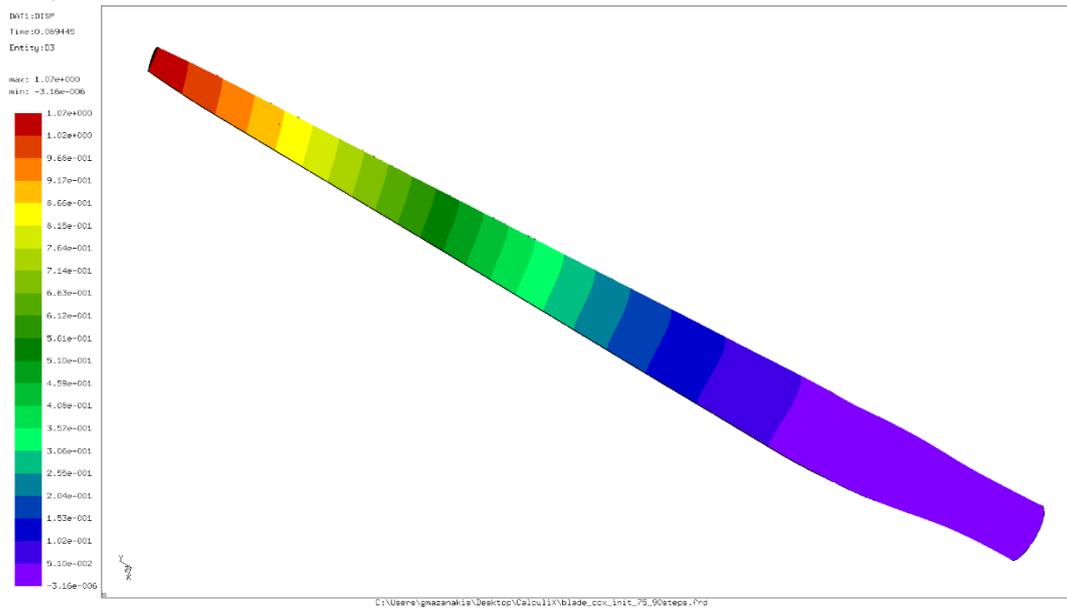


(b)

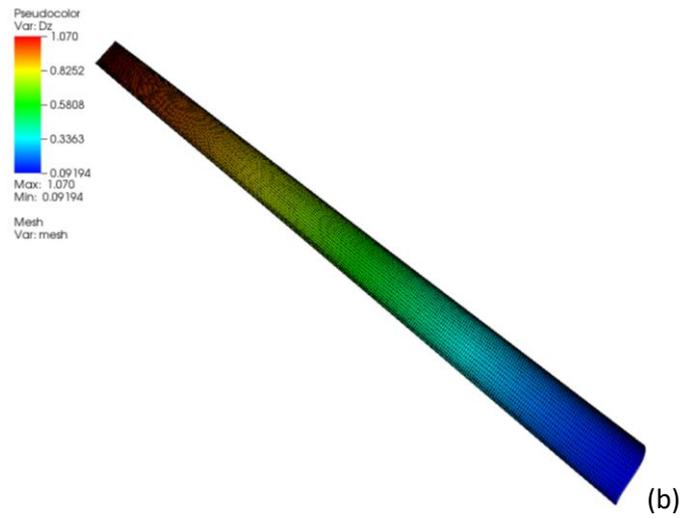


(c)

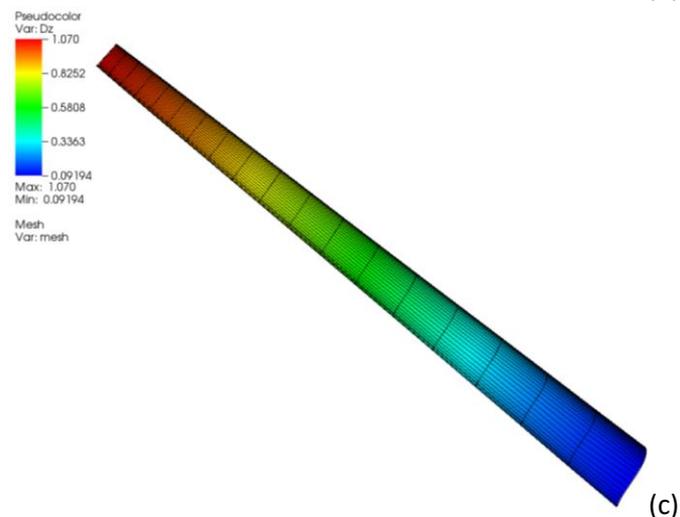
**FIGURE 6.53:** Distributions of the deformation at the blade during the first FSI iteration at  $YY'$  axis: (a) structural grid, (b) refined aerodynamic grid, (c) coarse aerodynamic grid (NREL case).



(a)

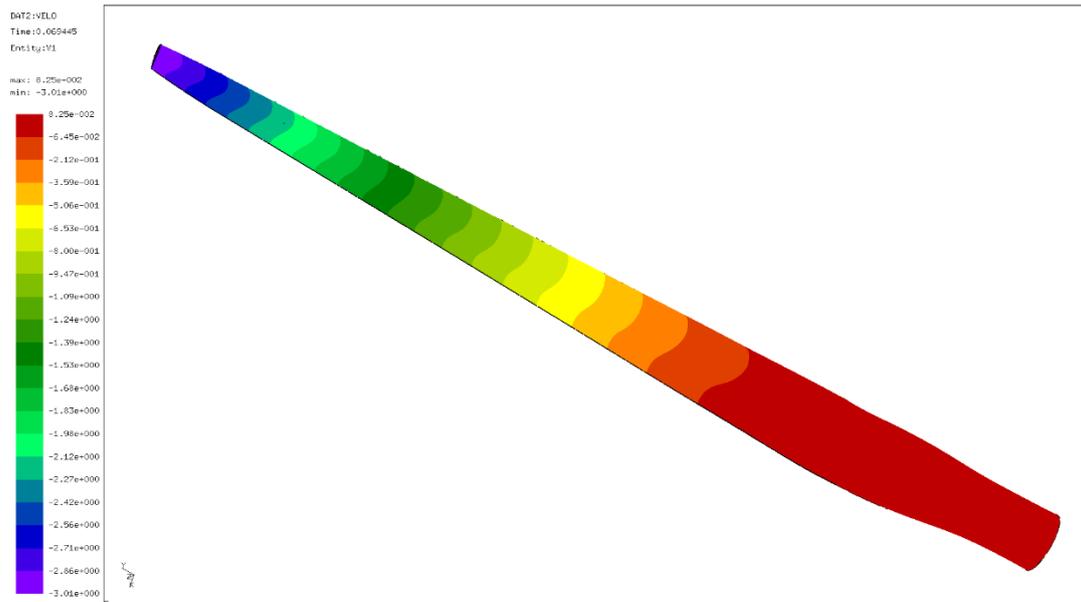


(b)

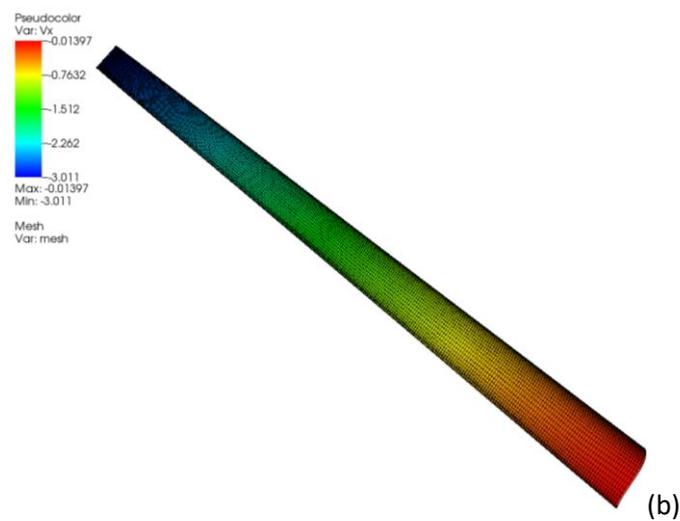


(c)

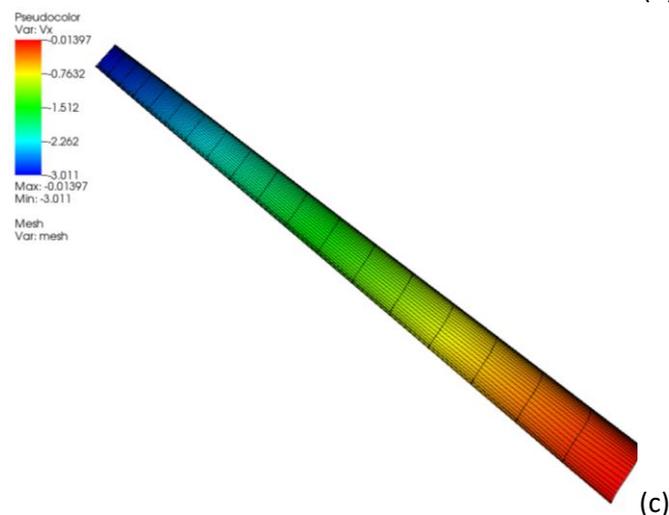
**FIGURE 6.54:** Distributions of the deformation at the blade during the first FSI iteration at  $ZZ'$  axis: (a) structural grid, (b) refined aerodynamic grid, (c) coarse aerodynamic grid (NREL case).



(a)

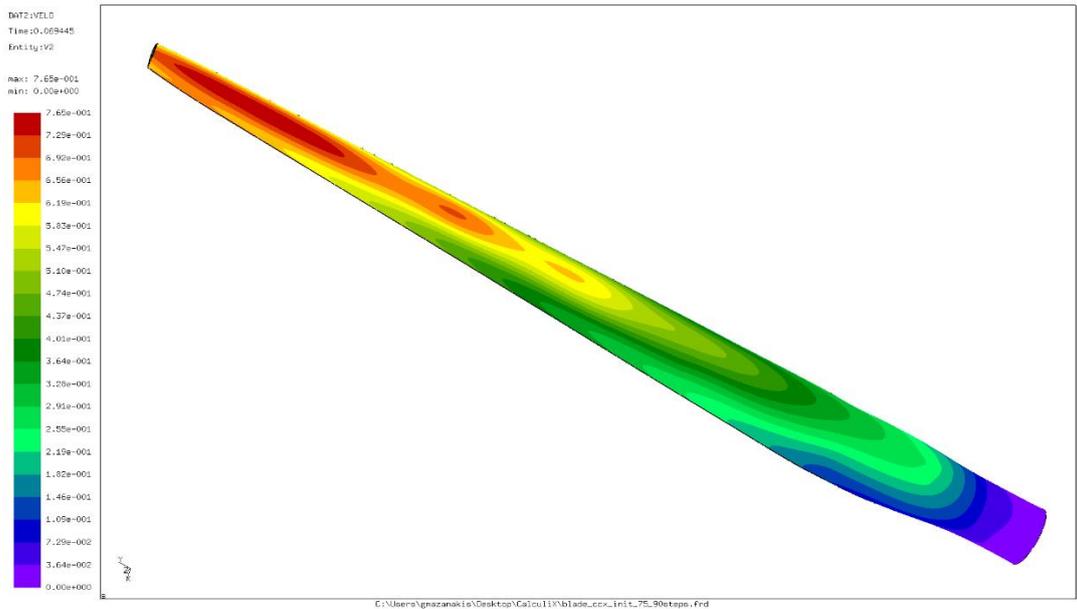


(b)

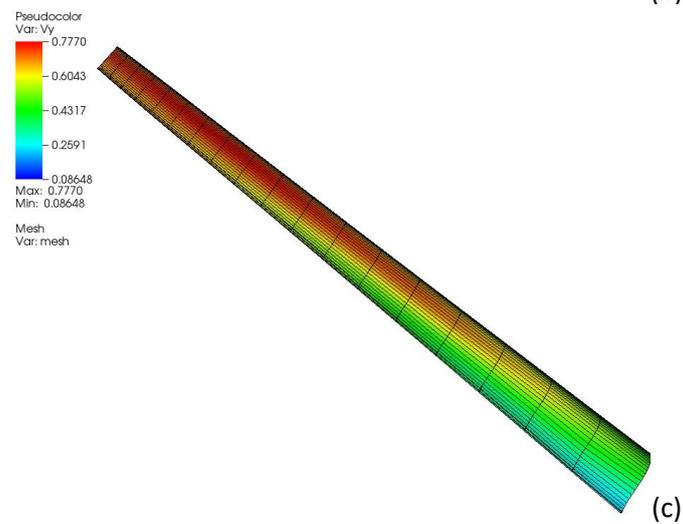
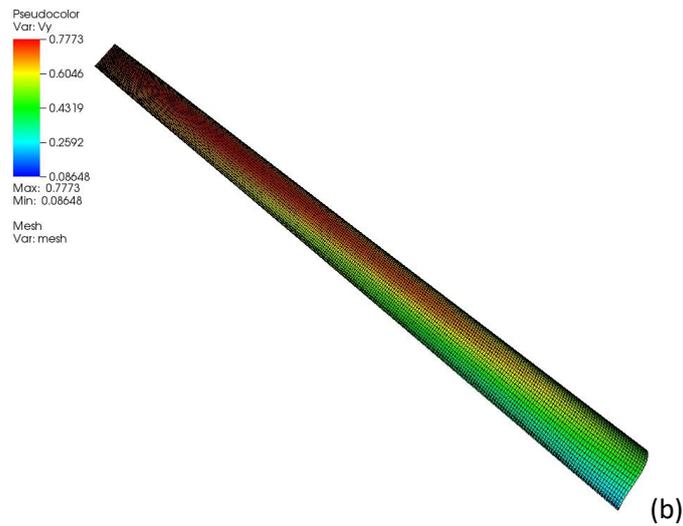


(c)

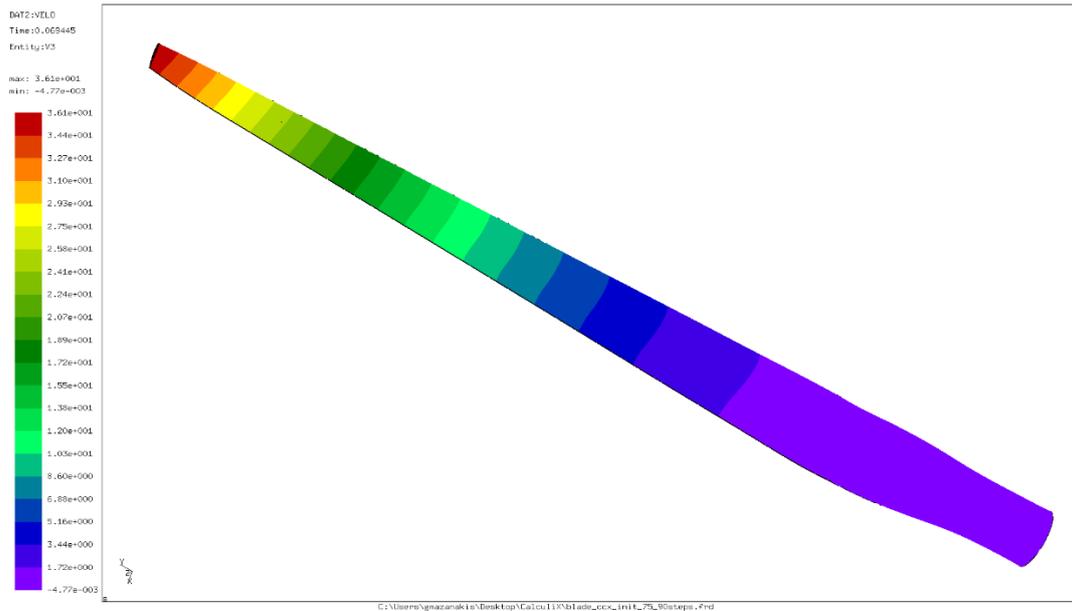
**FIGURE 6.55:** Distributions of the velocities at the blade during the first FSI iteration at  $XX'$  axis: (a) structural grid, (b) refined aerodynamic grid, (c) coarse aerodynamic grid (NREL case).



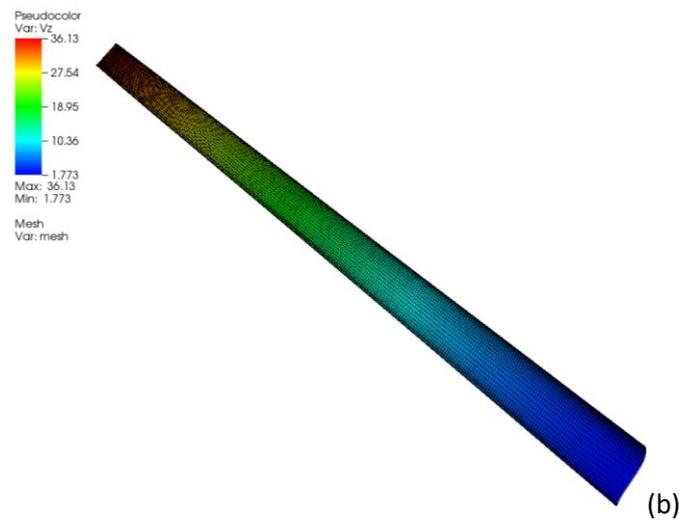
(a)



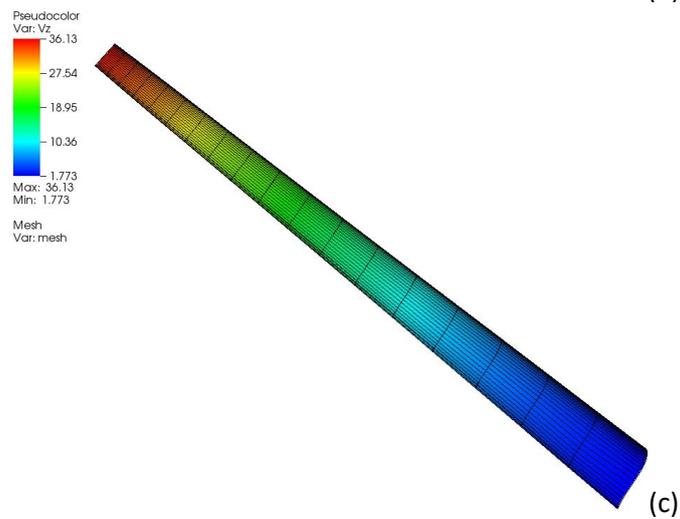
**FIGURE 6.56:** Distributions of the velocities at the blade during the first FSI iteration at  $YY'$  axis: (a) structural grid, (b) refined aerodynamic grid, (c) coarse aerodynamic grid (NREL case).



(a)



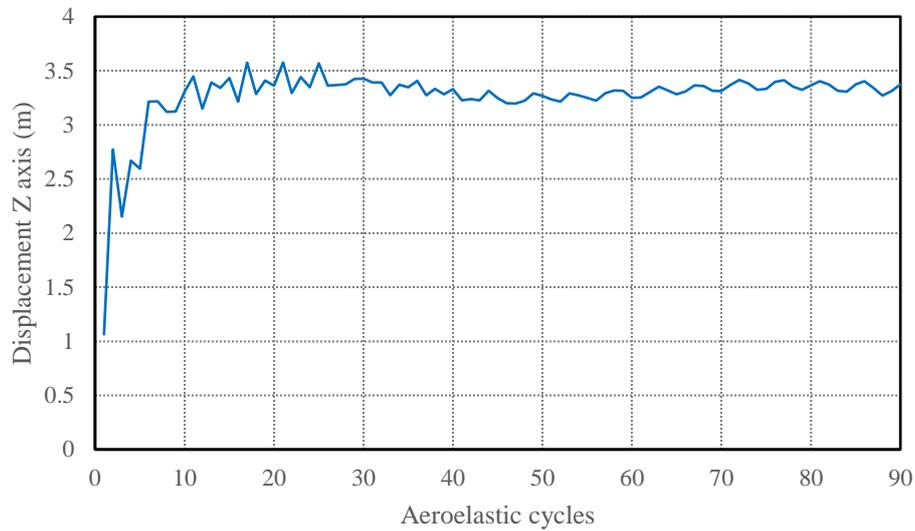
(b)



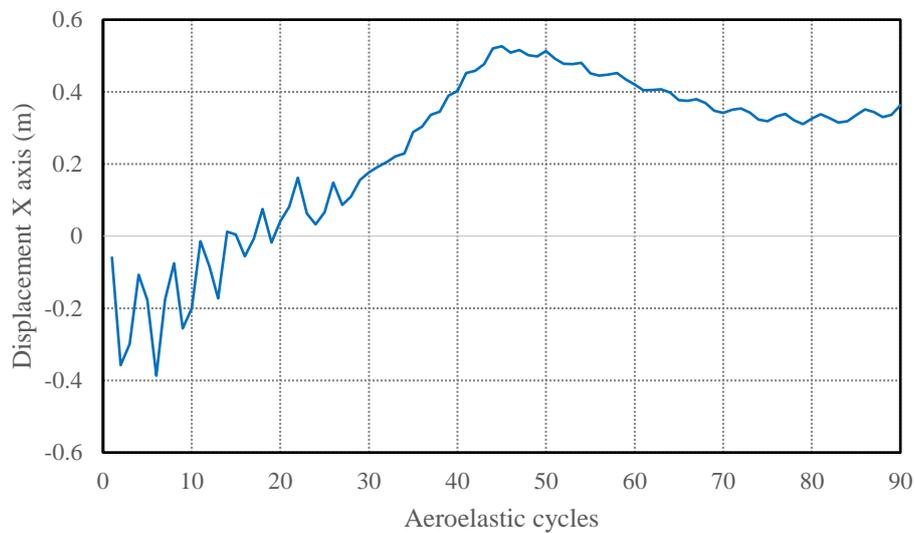
(c)

**FIGURE 6.57:** Distributions of the velocities at the blade during the first FSI iteration at  $ZZ'$  axis: (a) structural grid, (b) refined aerodynamic grid, (c) coarse aerodynamic grid (NREL case).

The results of the aeroelastic simulation for a full rotation of the blade are presented in *FIGURE 6.58* and *FIGURE 6.59*. The full rotation is conducted in 90 aerodynamic cycles, while the maximum values of the displacements along  $Z$  and  $X$  axis are 3.57 m and 0.53 m, respectively.



**FIGURE 6.58:** Displacements of the blade tip along the  $Z$  axis (NREL case).



**FIGURE 6.59:** Displacements of the blade tip along the  $X$  axis (NREL case).

## 6.6 Conclusions

In this Chapter a partitioned FSI methodology was reported. The whole procedure involved the coupling of CFD and CSD solvers employing the Radial Basis Functions (RBFs) Partition of Unity (PoU) method. With this approach the conservation of energy, momentum and force is ensured over the interface of the flow and structural meshes as a result of the radial functions' properties. The use of the PoU methodology improved the efficiency of the data transfer procedure providing simultaneously a physical formulation of the force distribution. The FSI methodology was applied on 4 benchmark test cases; the numerical results revealed the capabilities of the proposed methodology to accurately and efficiently perform such simulations.

# Chapter 7: Conclusions and Future Work

This final Chapter summarizes the findings and results of this thesis. In particular, Section 7.1 gives a summary of the study and highlights the main results and contributions of the thesis, while Section 7.2 indicates future research aspects that could be considered to extend the investigation results.

## 7.1 Concluding remarks

In this thesis the development and validation of methodologies and computational tools, allowing for the design and numerical simulation of wind turbines, was presented. However, the methodologies and the corresponding software, developed in this work, have broader applications (than the design and simulation of wind turbines), as it was already revealed in the previous Chapters. The developed methodologies and computational tools can be used (along with other numerical simulation, mesh generation and post-processing software) in the cycle of design, simulation and evaluation of wind turbine geometries.

The second Chapter of this Thesis focused on the description of the introduced modifications and additional capabilities to the "T4T" software (developed within the Turbomachines & Fluid Dynamics Laboratory of the Technical University of Crete), providing the ability to define wind turbine blades with internal structure in a full parametric way. Variable material thickness and different material properties can be applied in different blade regions in an interactive manner, through a user friendly graphical interface. After the interactive definition of the blade digital model (including both 3D surface and internal structure) the software may be used in an automated way (batch mode) to produce several candidate geometries in an optimization cycle, while retaining its topology unchanged.

Regarding the use of an optimization methodology in the aforementioned wind turbine design loop, an asynchronous version of a parallel, metamodel-assisted Differential Evolution algorithm was developed, with its details reported in Chapter 3 of this Thesis. The evaluation and testing of this computational tool is not reported in this Thesis; however, details can be found in (Stroflas, *et al.*, 2018).

At next the development of a specialized surface reconstruction methodology was reported, developed for the geometry definition of a wind turbine blade as a single NURBS surface, from a target set of data points provided in the form of a surface triangular mesh. For the parameterization of the blade surface the dedicated blade geometry modelling software "T4T" was used. The shape reconstruction of the blade surface was formulated as an optimization procedure, which was realized with the parallel, synchronous/asynchronous, metamodel-assisted Differential Evolution algorithm. The blade design variables, used in "T4T", are automatically modified inside the optimization procedure to produce alternative candidate blade designs, within prescribed limits. The cost function to be minimized is the sum of the squared distance between each given data point and the resulting NURBS surface of the candidate blade. The proposed methodology was found to provide reconstructed surfaces with a small error with respect to the given set of surface data points, while the resulting parameterization has a physical meaning.

The major drawback of the proposed methodology is the large computational time, associated with the evaluation of each candidate solution. The computation of the normal distances of the given data points from the free-form surface is a demanding calculation, and different alternatives should be examined.

In Chapter 5, a methodology for the acceleration of an RBF-based mesh deformation procedure via the reduction of the surface points was proposed. An agglomeration strategy was used to this end, modified from an already developed one for a multigrid methodology applied in a CFD code (Lygidakis & Nikolos, 2014b; Lygidakis *et al.*, 2016). As to the authors' best knowledge, although agglomeration multigrid techniques are well-established in numerical simulations, the corresponding fusion strategy has not been used so far with mesh-deformation algorithms.

It was demonstrated to be a highly efficient scheme; it requires no more than two seconds to generate four coarser resolutions of a surface including approximately 200,000 nodes. Moreover, significant improvements to the computational performance of the mesh-deformation algorithm can be succeeded, especially for large grid sizes. Besides improving efficiency of the RBF-based mesh deformation technique, the aforementioned scheme was revealed to preserve the grid quality, despite the significant decrease of the number of surface nodes, used as RBF-centers. Due to the incorporated constraints, the nodes near the edges of the deforming surface are preserved during the fusion process, outlining the studied geometry and maintaining its topology unaffected. However, depending on the examined test case, some of the aforementioned limitations can be applied in a looser way, increasing further the efficiency of the proposed approach without significantly reducing its accuracy. Based on the presented validation results, the proposed methodology appears to significantly enhance the mesh deformation algorithm in terms of computational efficiency, while maintaining simultaneously its accuracy at a competitive level. Indicative CFD results on the deformed grids (produced with and without the proposed agglomeration methodology) revealed no difference at all.

Finally in Chapter 6 a partitioned FSI methodology was reported based on the Radial Basis Functions (RBFs) Partition of Unity (PoU) method. With this approach the conservation of energy, momentum and force is ensured over the interface of the flow and structural mesh as a result of the radial functions properties. The use of the PoU methodology improved the efficiency of the data transfer procedure providing simultaneously a physical formulation of the force distribution.

Considering the above, the main contributions of this Thesis can be summarized as follows:

- The development of a methodology (and the corresponding software) for the RBF-based mesh deformation, highly accelerated using a novel agglomeration strategy. The developed methodology renders the RBF-based mesh deformation practically applicable in real-world problems. This is considered as the main contribution of this Thesis.
- The development of a methodology (and the corresponding software) for FSI simulations, based on the RBFs Partition of Unity (PoU) method for the exchange of data between the interacting flow and structural domains and grids. The methodology is independent of the grid types and the analyses software used in the computational procedure.

- The introduction of new tools in the “T4T” blade design software, for the parametric design of the external surface and internal geometry of wind turbine blades. The software can be a part of an automated design optimization procedure.
- The development of an asynchronous parallel version of the existing synchronous serial metamodel-assisted Differential Evolution algorithm. This methodology is currently used with success (in the context of a different Ph.D. Thesis) for the design optimization of wind turbines (among other applications).
- The development of an optimization-based methodology for the reverse-engineering of wind turbine blades, based on “T4T” and the developed DE algorithm.

Based on the above, the contribution of this Thesis is in the development of methodologies and computational tools for the synergetic design, optimization and simulation of wind turbine blades. However, their application is not limited to the aforementioned engineering field.

## 7.2 Further research

There are various ways to extend the work presented in this thesis. Few of them are listed below:

- Further evaluation and testing of the FSI methodology.
- Further development and refinement of the “T4T” software.
- Integration of the newly developed and the already available computational tools (design, analysis, optimization, post-processing) within a unified Computational Engineering software, with open architecture, so as to allow easy modifications and introduction of new tools and capabilities.
- Integration of the already developed (within the TurboLab – TUC) Blade Element Momentum (BEM) methodology, for wind turbines and propellers, with the computational tools developed in this Thesis. This is an ongoing work.
- Development (and integration in the aforementioned computational procedure) of alternative fast simulation tools for the flow field around the wind turbine based on a) axisymmetric CFD simulations (already developed) and b) Artificial Neural Networks.

The current Thesis has contributed in the development of methodologies and computational tools that have already been used in the work of other researchers within the Turbomachines & Fluid Dynamics Laboratory of the Technical University of Crete (TurboLab – TUC).

# Appendix A

## A.1 Free Form Deformation

An alternative very powerful method to deform computational grids is based on Free Form Deformation (FFD). Despite the several versions of the FFD technique appearing in the literature, there is a standard procedure that is followed by most of the researchers. In general, the implementation of an FFD technique can be decomposed into four main steps, which are common to all of its variant (Lamousin & Waggenpack, 1994). These essential steps are the following (for the 3D version of FFD):

- **1<sup>st</sup> Step  $\Rightarrow$  Construction of the parametric lattice and definition of the parametric space:** First, a parametric lattice is constructed with the topology of a structured mesh within a 3D volume that surrounds the object to be deformed or a specific part of it. The lattice is defined by an ordered mesh of control points that are arranged in the form of a structured mesh (forming successive rectangles), while in order to define the properties of the 3D lattice, B-Spline or Non-Uniform Rational B-Splines (NURBS) functions are used. The user defines the number of the control points to the 3-dimensional coordinate system, the degree of the curve (NURBS) in each direction, as well as the associated weights of all control points of the parametric mesh. Thus, each point within the solid  $(x, y, z)$  can be mapped to a parametric coordinate set  $(u, v, w)$ . In each direction of the parametric mesh, a particular type of a NURBS polynomial of a certain degree is applied, based on the following constrains:

$$1 \leq p \leq a, 1 \leq m \leq b, 1 \leq n \leq c \quad (\text{A.1})$$

where,  $p, m, n$  define the degree of the respective basis functions in the  $u, v$  and  $w$  parameter directions, whereas  $a, b$  and  $c$  represent the number of control points in the  $u, v$  and  $w$  parameter directions, respectively, starting the numbering from zero. The corresponding knot vectors are given as

$$\mathbf{U} = (u_0, u_1, \dots, u_q), \quad q = a + p + 1 \quad (\text{A.2})$$

$$\mathbf{V} = (v_0, v_1, \dots, v_r), \quad r = b + m + 1 \quad (\text{A.3})$$

$$\mathbf{W} = (w_0, w_1, \dots, w_s), \quad s = c + n + 1 \quad (\text{A.4})$$

The knot vector  $\mathbf{U}$  on the  $u$  parameter direction is defined according to the following equation (Knot vectors  $\mathbf{V}$  and  $\mathbf{W}$  are similarly computed):

$$u_i = \begin{cases} 0 & 0 \leq i \leq p \\ i - p & p < i \leq (q - p - 1) \\ q - 2p & (q - p - 1) < i \leq q \end{cases} \quad (\text{A.5})$$

NURBS basis functions  $N$  on each direction of the parameter volume are then derived, using the standard recursive formula of (Cox, 1972) and (de Boor, 1972) (provided here for the  $u$  direction):

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (\text{A.6})$$

$$N_{i,0}(u) = \begin{cases} 1, & \text{if } u_i \leq u \leq u_{i+1} \\ 0, & \text{elsewhere.} \end{cases} \quad (\text{A.7})$$

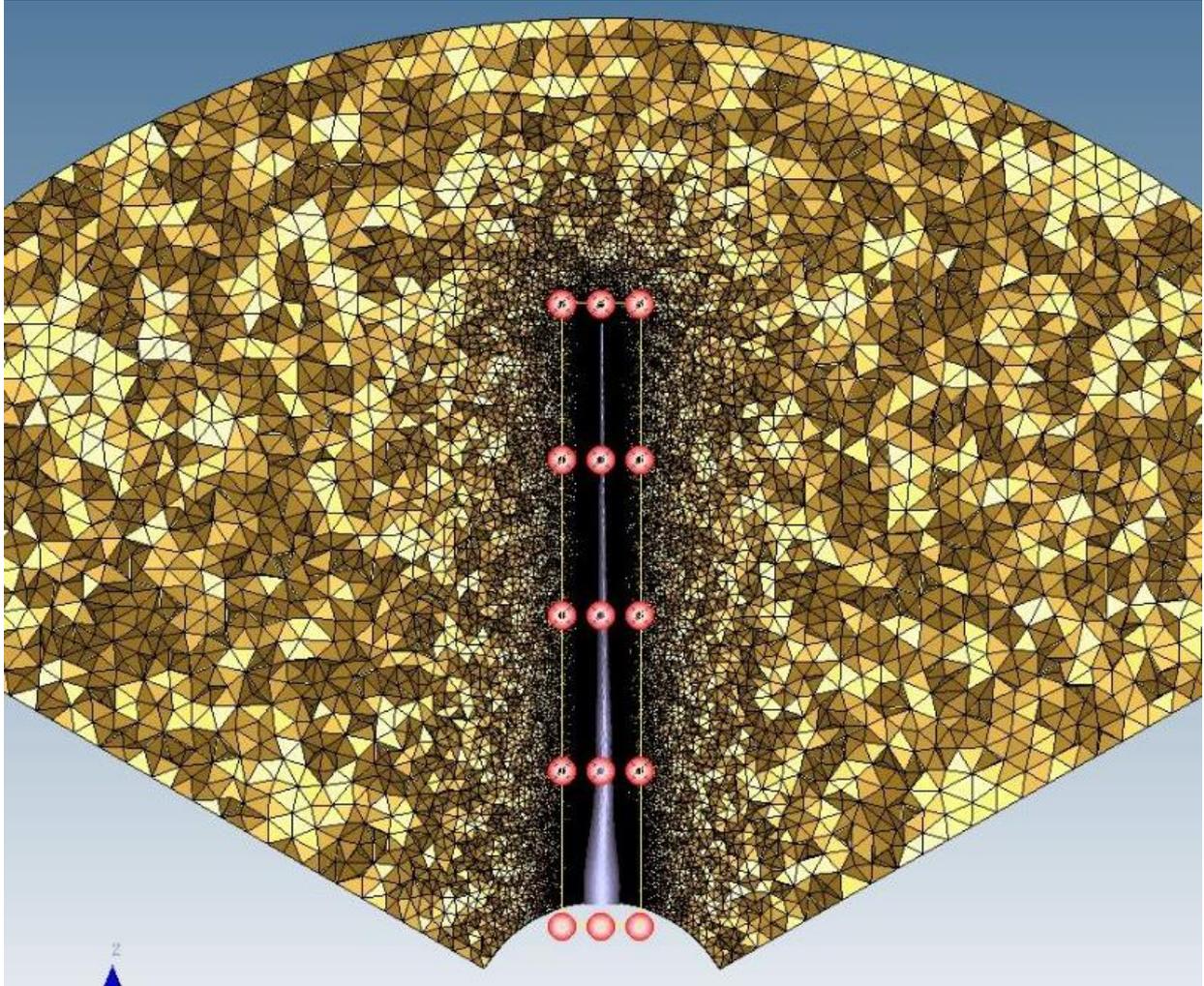
Finally, the Cartesian coordinates  $\mathbf{R}$  of a point within the 3D volume with parametric coordinates  $(u, v, w)$  is computed using a formula derived from the standard formula for NURBS curves or surfaces according to following equation (Yu & Soni, 1999):

$$\mathbf{R}(u, v, w) = \frac{\sum_{i=0}^a \sum_{j=0}^b \sum_{k=0}^c P_{i,j,k} G_{ijk} N_{i,p} N_{j,m} N_{k,m}}{\sum_{i=0}^a \sum_{j=0}^b \sum_{k=0}^c G_{ijk} N_{i,p} N_{j,m} N_{k,m}} \quad (\text{A.8})$$

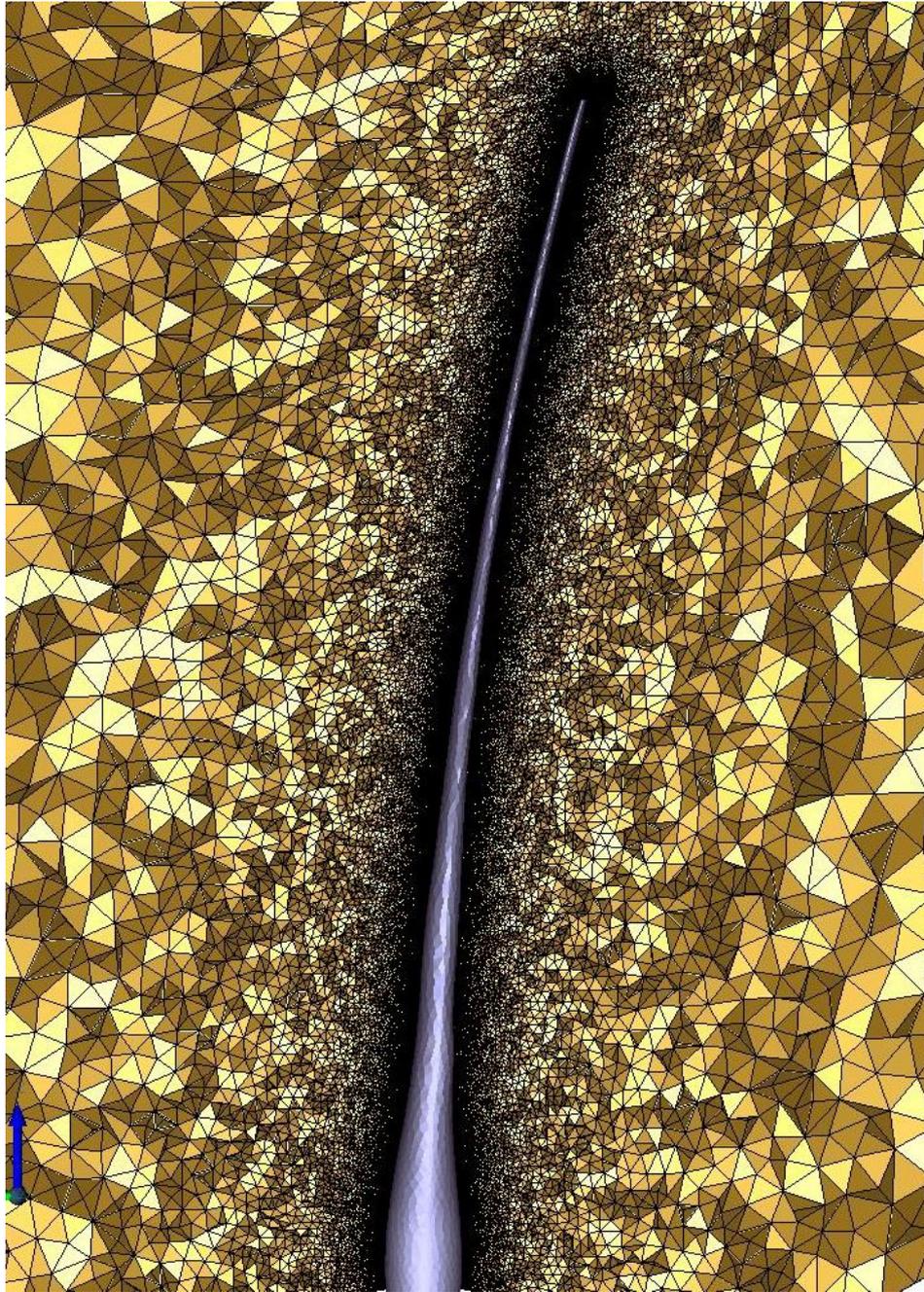
- **2<sup>nd</sup> Step  $\Rightarrow$  Embedding the object within the lattice:** The object to be deformed is usually represented as a set of ordered points, with known Cartesian coordinates  $(x, y, z)$ . However, in order to deform the object, the Cartesian coordinates must be first transformed into parametric ones  $(u, v, w)$ . In essence, their calculation is performed using an iterative methodology. In this work, the Octree algorithm was used to approximate the parametric coordinates that correspond to the Cartesian coordinates  $(x, y, z)$  of each point of the object to be deformed. According to the fundamental principles of the Octree method, for each point of the object, the following algorithm is repeatedly applied (for the 3D case).
  - i. The parametric volume is consecutively divided into eight equal subvolumes.
  - ii. The Cartesian coordinates  $(x, y, z)$  of each subvolume vertex are calculated using equation (A.8).
  - iii. Subsequently, these coordinates are compared to the Cartesian coordinates  $(x, y, z)$  of the object's point under consideration, in order to specify the subvolume in which the corresponding point lies.
  - iv. Finally, the latter subvolume is divided into eight new equal subvolumes and steps (ii)-(iv) can be iterated for a predefined number of subdivisions or until the required accuracy is achieved. Essentially, the desired parametric coordinates  $(u, v, w)$  of the point to be considered are defined as the parametric coordinates of the center of the subvolume that results from the last subdivision (Patrikalakis & Maekawa, 2002).
- **3<sup>rd</sup> Step  $\Rightarrow$  Deforming the parametric volume:** Herein, the deformation process of the parametric volume is accomplished by altering the coordinates of the control points of the 3D volume. However, depending on the variant of FFD technique that is implemented, the deformation mechanisms vary; for example, in cases where the FFD is used based on the non-uniform rational B-splines (NURBS-based FFD) the deformation of the considered object is achieved either by altering the coordinates of the control points of the lattice or by additional modification of the weights of the control points.
- **4<sup>th</sup> Step  $\Rightarrow$  Evaluating the effects of the deformation on the embedded object:** Evaluating any deformation's effect on the embedded object is a straightforward process to calculate the new coordinates of all the points that form the object under consideration, using equation (A.8). At this point it is important to highlight that during the deformation procedure, the parametric coordinates  $(u, v, w)$  of each point of the embedded object do not alter, contrary to their Cartesian coordinates  $(x, y, z)$ , which are deformed due to the modification of the control lattice.

## A.2 Wind Turbine Blade

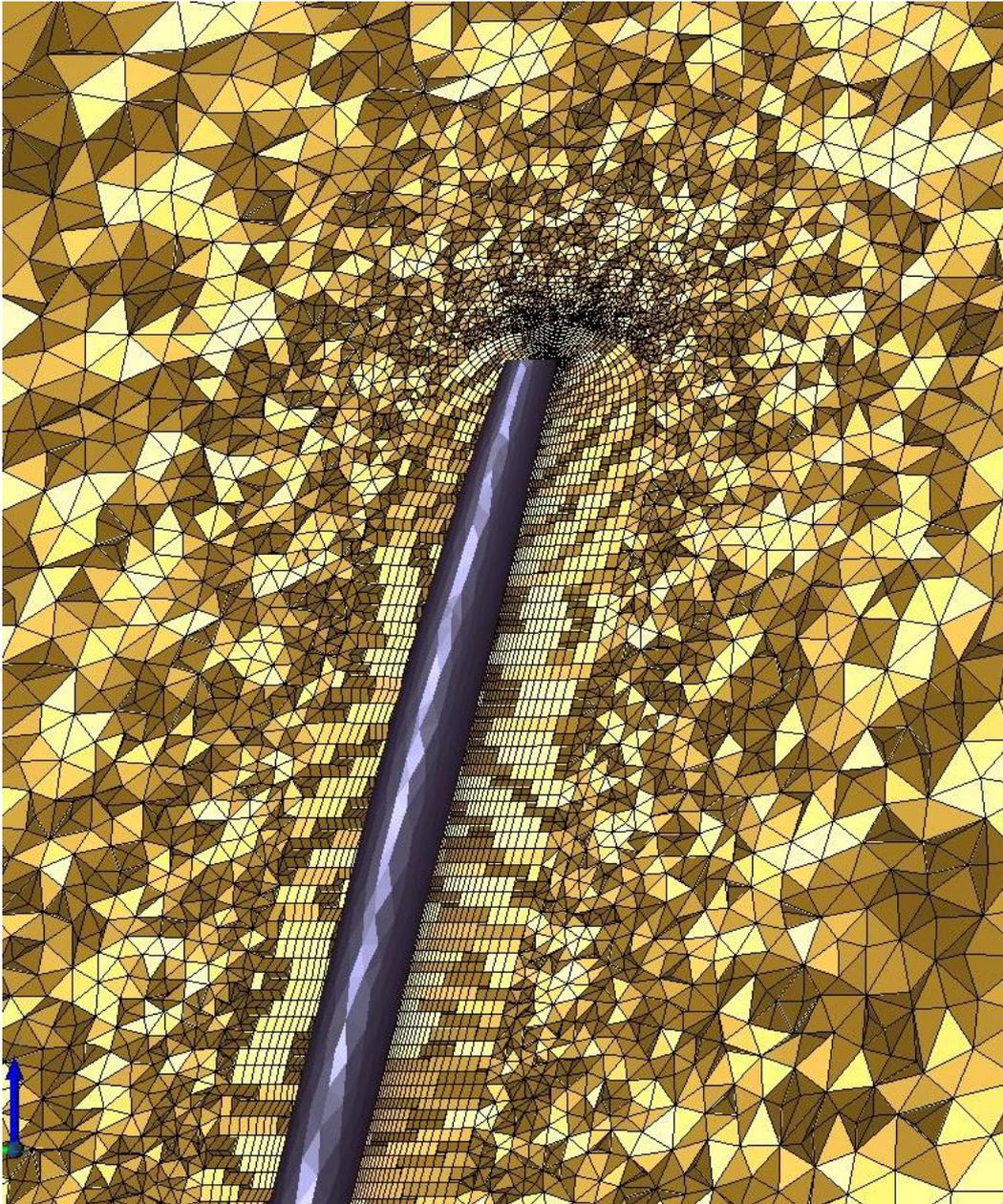
The following benchmark test case concerns the parabolic deformation of the NREL offshore 5-MW baseline wind turbine using FFD. The method has been modeled in a way to deform everything that is embedded in the computational lattice i.e. to deform simultaneously the CFD and CSD grids. The deformation was applied using 45 control points with unit weights *FIGURE A.1 - A.4*.



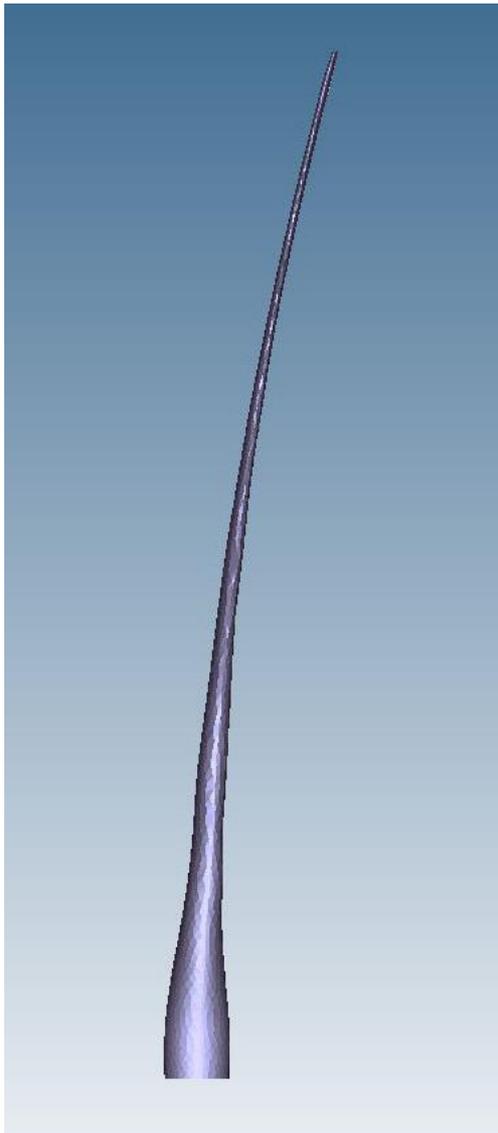
*FIGURE A.1: Initial grids (yellow – CFD grid), (grey – CSD grid) of NREL 5-MW turbine blade.*



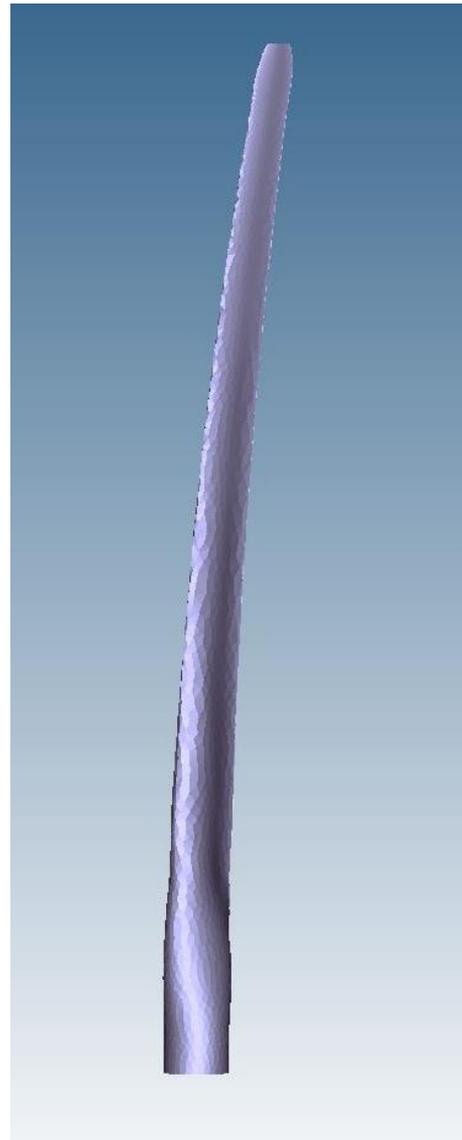
*FIGURE A.2: Deformed grids (yellow – CFD grid), (grey – CSD grid) of NREL 5-MW turbine blade.*



*FIGURE A.3: Close-up view at the tip region of the deformed grids (yellow – CFD grid), (grey – CSD grid) of NREL 5-MW turbine blade.*



(a)



(b)

**FIGURE A.4:** Deformed CSD grid of NREL 5-MW turbine blade.

# Appendix B

Four criteria have been employed in this work to evaluate the quality of the produced 3D unstructured hybrid grids: a) the aspect ratio, b) the skewness, c) the Jacobian and d) the orthogonality of each cell of the grid. The aspect ratio is a measure of the stretching of a cell. In this work, the metric is used to evaluate the quality only for the tetrahedral elements and it is defined as the normalized ratio of the maximum cell's length to the radius of the inscribed sphere of the cell (Frey & George, 2000). The second quality metric is the skewness of a cell, which corresponds to the distance between each face centroid and the face integration point, normalized by the magnitude of the face area vector (Jasak, 1996). In general, highly skewed cells should be avoided, as they can yield less accurate results and may lead to numerical instabilities. The maximum skewness should not exceed 0.95, with a considerably lower average value. The Jacobian metric is the minimum determinant of the Jacobian matrix evaluated at each corner and the center of the cell, divided by the corresponding edge lengths (Knupp, 2003). A value of unity corresponds to a good mesh quality, while negative deviation to concave elements. Finally, in order to determine the orthogonality metric of a cell, the vectors from the cell centroid to the centroids of its faces and the vectors from the cell centroid to the centroids of its adjacent cells, as well as the corresponding face area vectors are used ("ANSYS Fluent User's Guide," 2012). The orthogonality ranges from 0.0 (poor orthogonality) to 1.0 (good orthogonality). For all types of cells the minimum value of the orthogonality metric should be higher than 0.01, but with a considerably higher average value.

## B.1 NASA Common Research Model (CRM)

The quality of the un-deformed and deformed grids, considering the four pre-described quality metrics, are presented in **Error! Reference source not found.** - TABLE B.4 and FIGURE B.1: Comparison of the quality metrics: (a) aspect ratio, (b) skewness, (c) Jacobian and (d) orthogonality, for the un-deformed and the deformed grids, using agglomerated control areas (Level 4, NASA CRM case).

. The quality of the un-deformed three-dimensional grid is compared with both the deformed grids: the one deformed using the proposed (Level 4) agglomeration procedure, and the one deformed without the agglomeration procedure (Full RBF).

**TABLE B.1:** Aspect Ratio of the un-deformed and deformed grids, with (Level 4) and without agglomeration of RBF-centers (NASA CRM case).

<i>Metric value range</i>	<b>Aspect Ratio</b>					
	<b>Un-deformed</b>		<b>Deformed (Full-RBF)</b>		<b>Deformed (Level 4 agglomeration)</b>	
	<i># of Elements</i>	<i>% of Elements</i>	<i># of Elements</i>	<i>% of Elements</i>	<i># of Elements</i>	<i>% of Elements</i>
[0.0-0.1]	29	0.001	36	0.001	36	0.001
[0.1-0.2]	1,408	0.052	1,663	0.062	1,662	0.062
[0.2-0.3]	4,964	0.185	5,738	0.214	5,739	0.214
[0.3-0.4]	11,761	0.438	14,083	0.524	14,077	0.524
[0.4-0.5]	28,805	1.073	41,440	1.543	41,436	1.543
[0.5-0.6]	151,441	5.639	207,394	7.722	207,408	7.723
[0.6-0.7]	630,027	23.459	676,045	25.172	675,990	25.170
[0.7-0.8]	1,031,771	38.417	1,005,009	37.421	1,005,022	37.421
[0.8-0.9]	707,585	26.346	640,305	23.841	640,336	23.842
[0.9-1.0]	117,913	4.390	93,991	3.500	93,998	3.500

**TABLE B.2:** Skewness of the un-deformed and deformed grids, with (Level 4) and without agglomeration of RBF-centers (NASA CRM case).

<i>Metric value range</i>	<b>Skewness</b>					
	<b>Un-deformed</b>		<b>Deformed (Full-RBF)</b>		<b>Deformed (Level 4 agglomeration)</b>	
	<i># of Elements</i>	<i>% of Elements</i>	<i># of Elements</i>	<i>% of Elements</i>	<i># of Elements</i>	<i>% of Elements</i>
[0.0-0.1]	5,852,969	53.304	5,881,037	53.560	5,881,329	53.562
[0.1-0.2]	3,354,803	30.641	3,370,864	30.790	3,370,662	30.788
[0.2-0.3]	1,150,252	10.506	1,130,645	10.327	1,130,551	10.327
[0.3-0.4]	437,232	3.993	412,402	3.767	412,471	3.768
[0.4-0.5]	106,225	0.970	107,546	0.982	107,537	0.982
[0.5-0.6]	38,839	0.355	36,310	0.332	36,275	0.331
[0.6-0.7]	15,738	0.144	16,325	0.149	16,295	0.149
[0.7-0.8]	4,666	0.043	4,561	0.042	4,556	0.042
[0.8-0.9]	3,169	0.029	3,681	0.034	3,678	0.034

[0.9-1.0]	2,145	0.020	2,667	0.024	2,684	0.025
-----------	-------	-------	-------	-------	-------	-------

TABLE B.3: Jacobian of the un-deformed and deformed grids, with (Level 4) and without agglomeration of RBF-centers (NASA CRM case).

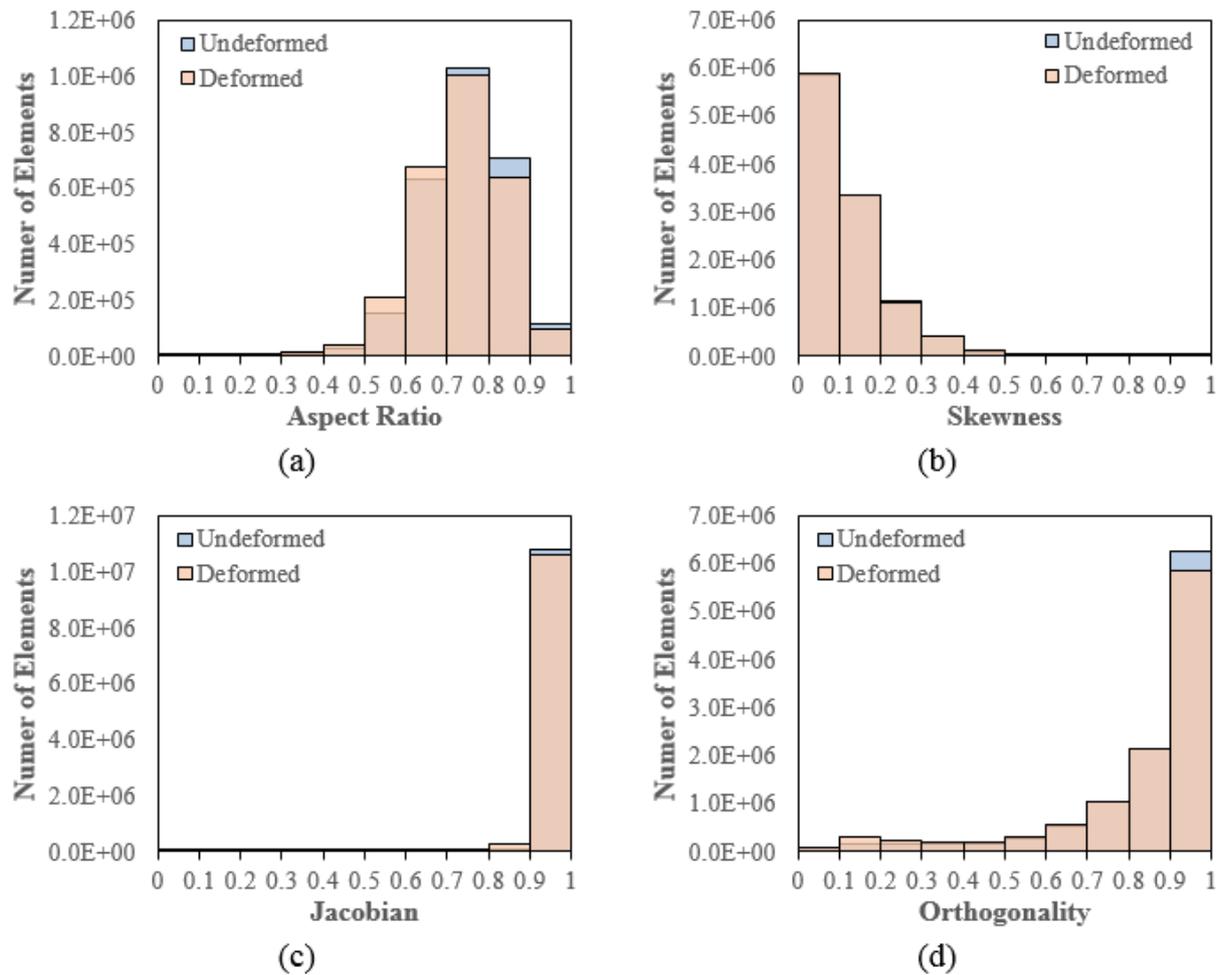
Metric value range	Jacobian					
	Un-deformed		Deformed (Full-RBF)		Deformed (Level 4 agglomeration)	
	# of Elements	% of Elements	# of Elements	% of Elements	# of Elements	% of Elements
[0.0-0.1]	0	0.000	25	0.000	27	0.000
[0.1-0.2]	0	0.000	31	0.000	33	0.000
[0.2-0.3]	0	0.000	167	0.002	161	0.001
[0.3-0.4]	0	0.000	807	0.007	811	0.007
[0.4-0.5]	23	0.000	1,746	0.016	1,732	0.016
[0.5-0.6]	2,500	0.023	7,943	0.072	7,959	0.073
[0.6-0.7]	16,427	0.150	33,071	0.302	33,196	0.303
[0.7-0.8]	37,983	0.346	86,285	0.787	86,012	0.784
[0.8-0.9]	81,594	0.744	252,025	2.298	252,437	2.302
[0.9-1.0]	10,827,511	98.737	10,583,863	96.516	10,583,595	96.513

TABLE B.4: Orthogonality of the un-deformed and deformed grids, with (Level 4) and without agglomeration of RBF-centers (NASA CRM case).

Metric value range	Orthogonality					
	Un-deformed		Deformed (Full-RBF)		Deformed (Level 4 agglomeration)	
	# of Elements	% of Elements	# of Elements	% of Elements	# of Elements	% of Elements
[0.0-0.1]	83,866	0.765	99,804	0.910	99,294	0.905
[0.1-0.2]	152,398	1.390	321,696	2.934	321,456	2.931
[0.2-0.3]	162,419	1.481	233,218	2.127	234,971	2.143
[0.3-0.4]	150,437	1.372	193,094	1.761	193,462	1.764
[0.4-0.5]	170,087	1.551	214,036	1.952	213,402	1.946
[0.5-0.6]	258,575	2.358	310,928	2.835	311,134	2.837
[0.6-0.7]	519,542	4.738	576,773	5.260	577,206	5.264
[0.7-0.8]	1,057,326	9.642	1,026,938	9.365	1,026,809	9.364
[0.8-0.9]	2,161,584	19.712	2,139,164	19.507	2,137,839	19.495

[0.9-1.0] | 6,249,804 | 56.992 | 5,850,387 | 53.350 | 5,850,465 | 53.351

Comparing the values of all four quality metrics, it is evident that the quality of the deformed grids exhibits small changes after the deformation, despite the large displacement of the wing surface. Moreover, the quality of the two deformed grids (with and without the agglomeration of the RBF-centers) is practically the same, demonstrating that the proposed methodology improves the computational efficiency of the procedure without deteriorating the quality of the deformed grid.



**FIGURE B.1:** Comparison of the quality metrics: (a) aspect ratio, (b) skewness, (c) Jacobian and (d) orthogonality, for the un-deformed and the deformed grids, using agglomerated control areas (Level 4, NASA CRM case).

## B.2 Wind Turbine Blade

For the second test case a similar evaluation was performed, considering all four quality metrics. In TABLE B.5: Aspect Ratio of the un-deformed and deformed grids, with agglomeration (Level 4) of RBF-centers (NREL case). TABLE B.5 - TABLE B.8 and FIGURE B.2, the quality of the un-deformed grid is compared with the deformed one using the proposed methodology for the agglomeration of the RBF-centers (Level 4). Once more the results reveal that the quality of the grid slightly changes between the un-deformed and the deformed one implementing the agglomeration procedure.

**TABLE B.5:** Aspect Ratio of the un-deformed and deformed grids, with agglomeration (Level 4) of RBF-centers (NREL case).

<i>Metric value range</i>	<i>Aspect Ratio</i>			
	<i>Un-deformed</i>		<i>Deformed (Level 4 agglomeration)</i>	
	<i># of Elements</i>	<i>% of Elements</i>	<i># of Elements</i>	<i>% of Elements</i>
<i>[0.0-0.1]</i>	0	0.000	0	0.000
<i>[0.1-0.2]</i>	105	0.001	118	0.001
<i>[0.2-0.3]</i>	1,592	0.012	1,879	0.014
<i>[0.3-0.4]</i>	16,507	0.126	20,401	0.155
<i>[0.4-0.5]</i>	135,834	1.033	164,652	1.253
<i>[0.5-0.6]</i>	676,252	5.144	789,138	6.003
<i>[0.6-0.7]</i>	2,564,567	19.509	2,811,323	21.386
<i>[0.7-0.8]</i>	5,368,309	40.838	5,268,240	40.077
<i>[0.8-0.9]</i>	3,854,647	29.323	3,620,235	27.540
<i>[0.9-1.0]</i>	527,538	4.013	469,365	3.571

**TABLE B.6:** Skewness of the un-deformed and deformed grids, with agglomeration (Level 4) of RBF-centers (NREL case).

<i>Metric value range</i>	<i>Skewness</i>			
	<i>Un-deformed</i>		<i>Deformed (Level 4 agglomeration)</i>	
	<i># of Elements</i>	<i>% of Elements</i>	<i># of Elements</i>	<i>% of Elements</i>
<i>[0.0-0.1]</i>	4,518,006	22.737	4,539,663	22.847
<i>[0.1-0.2]</i>	10,151,833	51.092	10,172,925	51.198
<i>[0.2-0.3]</i>	4,250,873	21.394	4,219,995	21.238
<i>[0.3-0.4]</i>	796,002	4.006	784,931	3.950
<i>[0.4-0.5]</i>	122,168	0.615	120,897	0.608
<i>[0.5-0.6]</i>	22,015	0.111	22,196	0.112
<i>[0.6-0.7]</i>	6,178	0.031	6,408	0.032
<i>[0.7-0.8]</i>	2,292	0.012	2,351	0.012
<i>[0.8-0.9]</i>	815	0.004	843	0.004

[0.9-1.0]	443	0.002	416	0.002
-----------	-----	-------	-----	-------

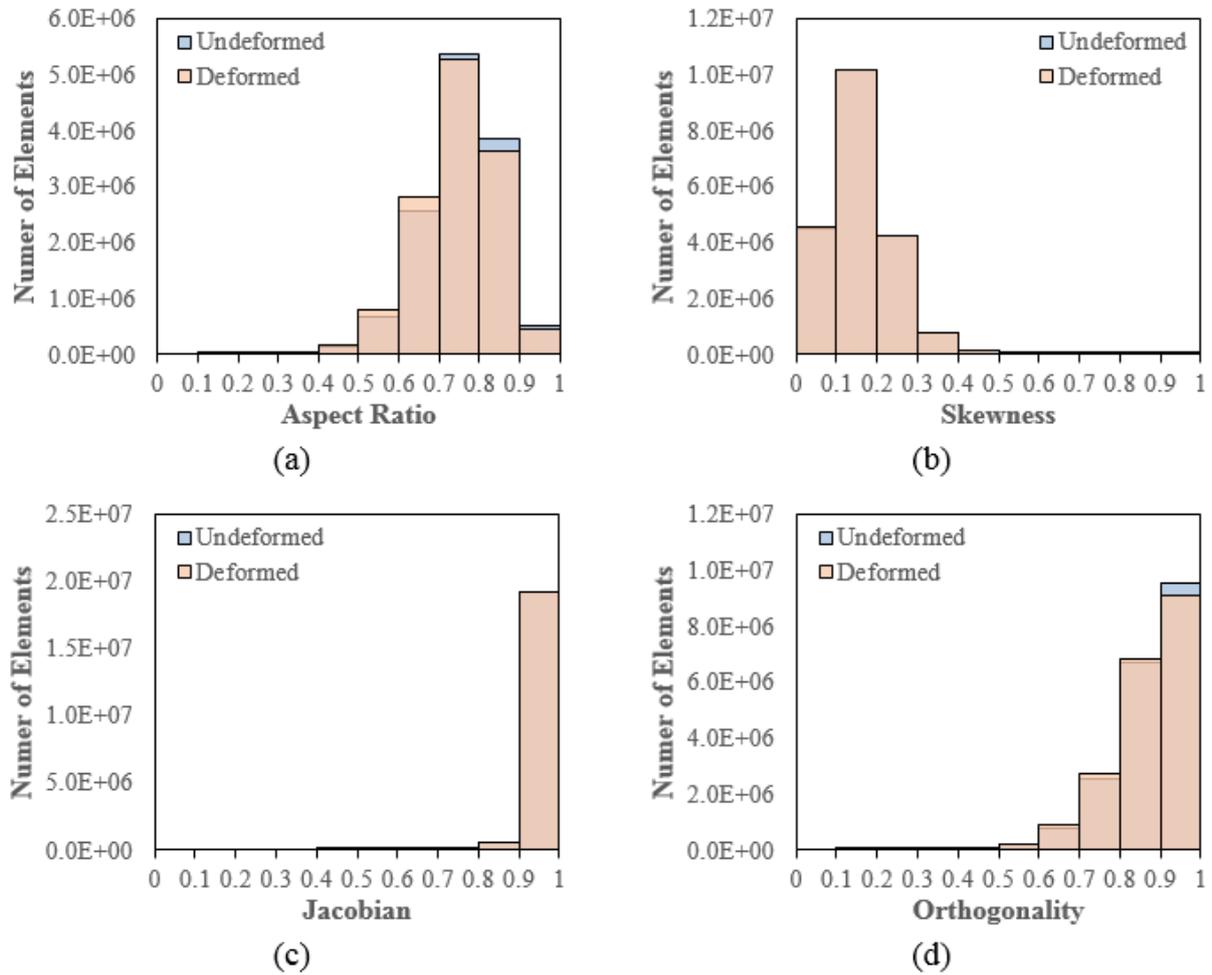
**TABLE B.7:** *Jacobian of the un-deformed and deformed grids, with agglomeration (Level 4) of RBF-centers (NREL case).*

<i>Metric value range</i>	<i>Jacobian</i>			
	<i>Un-deformed</i>		<i>Deformed (Level 4 agglomeration)</i>	
	<i># of Elements</i>	<i>% of Elements</i>	<i># of Elements</i>	<i>% of Elements</i>
[0.0-0.1]	0	0.000	0	0.000
[0.1-0.2]	0	0.000	0	0.000
[0.2-0.3]	0	0.000	0	0.000
[0.3-0.4]	2	0.000	2	0.000
[0.4-0.5]	65	0.000	65	0.000
[0.5-0.6]	695	0.003	695	0.003
[0.6-0.7]	7,443	0.037	7,445	0.037
[0.7-0.8]	49,845	0.251	49,848	0.251
[0.8-0.9]	640,199	3.222	640,288	3.222
[0.9-1.0]	19,172,376	96.486	19,172,282	96.486

**TABLE B.8:** *Orthogonality of the un-deformed and deformed grids, with agglomeration (Level 4) of RBF-centers (NREL case).*

<i>Metric value range</i>	<i>Orthogonality</i>			
	<i>Un-deformed</i>		<i>Deformed (Level 4 agglomeration)</i>	
	<i># of Elements</i>	<i>% of Elements</i>	<i># of Elements</i>	<i>% of Elements</i>
[0.0-0.1]	0	0.000	0	0.000
[0.1-0.2]	45	0.000	57	0.000
[0.2-0.3]	750	0.004	837	0.004
[0.3-0.4]	5,513	0.028	6,351	0.032
[0.4-0.5]	36,705	0.185	43,968	0.221
[0.5-0.6]	204,586	1.030	234,966	1.182
[0.6-0.7]	809,723	4.075	898,881	4.524
[0.7-0.8]	2,524,317	12.704	2,744,047	13.810
[0.8-0.9]	6,722,494	33.831	6,847,592	34.461

[0.9-1.0] | 9,566,492    48.144    9,093,926    45.766



**FIGURE B.2:** Comparison of quality metrics: (a) aspect ratio, (b) skewness, (c) Jacobian and (d) orthogonality, for the un-deformed and the deformed grids, using agglomerated control areas (Level 4, NREL case).

# References

- ABBOTT, I.H. & VON DOENHOFF, A.E. (1959) *Theory of Wing Sections: Including a Summary of Airfoil Data*. New York, NY: Dover Publications.
- ABD-ELHADY, A.M., SABIHA, N.A., & IZZULARAB, M.A. (2014) Experimental evaluation of air-termination systems for wind turbine blades. *Electric Power Systems Research*, **107**, 133–143.
- ACKERMANN, T. & SÖDER, L. (2002) An overview of wind energy-status 2002. *Renewable and Sustainable Energy Reviews*, **6**, 67–127.
- AHREM, R., BECKERT, A., & WENDLAND, H. (2006) A meshless spatial coupling scheme for large-scale fluid-structure-interaction problems. *CMES*, **12**.
- ALBERSMANN, F., DRERUP, P., MEHNEN, J., & WEINERT, K.W. (1998) New Solutions For Surface Reconstruction From Discrete Point Data By Means Of Computational Intelligence (Report No. SFB-531-CI--22/98). Dortmund University, Germany.
- ALLEN, C. (2007) Parallel universal approach to mesh motion and application to rotors in forward flight. *International Journal for Numerical Methods in Engineering*, **69**, 2126–2149.
- ANSYS (2013) *Engineering Simulation & 3D Design Software*. URL <http://www.ansys.com>.
- ANSYS Fluent User's Guide (2012) .
- ASOUTI, V.G. & GIANNAKOGLU, K.C. (2009) Aerodynamic optimization using a parallel asynchronous evolutionary algorithm controlled by strongly interacting demes. *Engineering Optimization*, **41**, 241–257.
- ASOUTI, V.G., KAMPOLIS, I.C., & GIANNAKOGLU, K.C. (2009) A grid-enabled asynchronous metamodel-assisted evolutionary algorithm for aerodynamic optimization. *Genet Program Evolvable Mach*, **10**, 373.
- BARHAK, J. & FISCHER, A. (2002) Adaptive reconstruction of freeform objects with 3D SOM neural network grids. *Computers & Graphics*, **26**, 745–751.
- BARR, A.H. (1984) Global and Local Deformations of Solid Primitives. *11th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '84*. New York, NY, USA: ACM, pp. 21–30.
- BATINA, J.T. (1990) Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA Journal*, **28**, 1381–1388.
- BAXEVANOU, C.A., CHAVIAROPOULOS, P.K., VOUTSINAS, S.G., & VLACHOS, N.S. (2008) Evaluation study of a Navier–Stokes CFD aeroelastic model of wind turbine airfoils in classical flutter. *Journal of Wind Engineering and Industrial Aerodynamics*, **96**, 1425–1443.
- BEATSON, R.K., CHERRIE, J.B., & MOUAT, C.T. (1999) Fast fitting of radial basis functions: Methods based on preconditioned GMRES iteration. *Advances in Computational Mathematics*, **11**, 253–270.
- BEATSON, R.K. & NEWSAM, G.N. (1992) Fast evaluation of radial basis functions: I. *Computers & Mathematics with Applications*, **24**, 7–19.
- BECKERT, A. (2000) Coupling fluid (CFD) and structural (FE) models using finite interpolation elements. *Aerospace Science and Technology*, **4**, 13–22.
- BECKERT, A. & WENDLAND, H. (2001) Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science and Technology*, **5**, 125–134.
- BEIELSTEIN, T., MEHNEN, J., SCHÖNEMANN, L., SCHWEFEL, H.-P., SURMANN, T., WEINERT, K., & WIESMANN, D. (2003) Design of Evolutionary Algorithms and Applications in Surface Reconstruction. *Advances in Computational Intelligence: Theory and Practice, Natural Computing Series* (Schwefel, H.-P., Wegener, I., & Weinert, K. eds). Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 145–193.
- BERG, J.C. & RESOR, B.R. (2012) Numerical Manufacturing And Design Tool (NuMAD v2.0) for Wind Turbine Blades: User's Guide ( No. SAND2012-7028). Sandia National Laboratories.
- BHARDWAJ, M.K., KAPANIA, R.K., REICHENBACH, E., & GURUSWAMY, G.P. (1998) Computational Fluid Dynamics/Computational Structural Dynamics Interaction Methodology for Aircraft

- Wings. *AIAA Journal*, **36**, 2179–2186.
- BLAZEK, J. (2001) *Computational Fluid Dynamics: Principles and Applications*. Kidlington: Elsevier Science.
- BLOM, F.J. (2000) Considerations on the spring analogy. *International Journal for Numerical Methods in Fluids*, **32**, 647–668.
- BOER, A. de, ZUIJLEN, A.H. van, & BIJL, H. (2009) Radial Basis Functions for Interface Interpolation and Mesh Deformation. *Advanced Computational Methods in Science and Engineering, Lecture Notes in Computational Science and Engineering* (Koren, B. & Vuik, K. eds). Springer Berlin Heidelberg, pp. 143–178.
- BOSSANYI, E.A. (2003) GH Bladed, theory manual ( No. 282/BR/009).
- BOTTASSO, C.L., DETOMI, D., & SERRA, R. (2005) The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering*, **194**, 4244–4264.
- BRANDT, A. & LIVNE, O. (2011) *Multigrid Techniques*. , Classics in Applied Mathematics. Society for Industrial and Applied Mathematics.
- BRAUN, A.L. & AWRUCH, A.M. (2009) Aerodynamic and aeroelastic analyses on the CAARC standard tall building model using numerical simulation. *Computers & Structures*, **87**, 564–581.
- BROEK, D. (1982) *Elementary engineering fracture mechanics*. Springer Netherlands.
- BULDER, B.H., BARHORST, S.A.M., SCHEPERS, J.G., & HAGG, F. (2001) Theory and User Manual: BLADOPT ( No. ECN-C-01-011). Energy Research Centre of the Netherlands (ECN).
- BUMSUK, K., KIM, W., LEE, S., BAE, S., & LEE, Y. (2013) Developement and verification of a performance based optimal design software for wind turbine blades. *Renewable Energy*, **54**, 166–172.
- CARRÉ, G., FOURNIER, L., & LANTERI, S. (2000) Parallel linear multigrid algorithms for the acceleration of compressible flow calculations. *Computer Methods in Applied Mechanics and Engineering*, **184**, 427–448.
- CEBRAL, J.R. & LÖHNER, R. (1997) Conservative Load Projection and Tracking for Fluid-Structure Problems. *AIAA Journal*, **35**, 687–692.
- CEZE, M. & FIDKOWSKI, K.J. (2013) Drag Prediction Using Adaptive Discontinuous Finite Elements. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. Grapevine, Texas, USA.
- CHARALAMPOUS, K.G., STROFYLAS, G.A., MAZANAKIS, G.I., & NIKOLOS, I.K. (2015) Wind Turbine Blades Parametric Design Using Grasshopper. *8th GRACM, International Congress on Computational Mechanics*. Volos, Greece.
- CHEN, P. & JADIC, I. (1998) Interfacing of Fluid and Structural Models via Innovative Structural Boundary Element Method. *AIAA Journal*, **36**, 282–287.
- CHORIN, A.J. (1997) A Numerical Method for Solving Incompressible Viscous Flow Problems. *Journal of Computational Physics*, **135**, 118–125.
- CIZMAS, P.G.A. & GARGOLOFF, J.I. (2008) Mesh Generation and Deformation Algorithm for Aeroelasticity Simulations. *Journal of Aircraft*, **45**, 1062–1066.
- CORDERO-GRACIA, M., GÓMEZ, M., PONSIN, J., & VALERO, E. (2012) An interpolation tool for aerodynamic mesh deformation problems based on octree decomposition. *Aerospace Science and Technology*, 35th ERF: Progress in Rotorcraft Research, **23**, 93–107.
- COX, K. & ECHTERMEYER, A. (2012) Structural Design and Analysis of a 10MW Wind Turbine Blade. *Energy Procedia*, Selected papers from Deep Sea Offshore Wind R&D Conference, Trondheim, Norway, 19-20 January 2012, **24**, 194–201.
- COX, M.G. (1972) The Numerical Evaluation of B-Splines. *IMA J Appl Math*, **10**, 134–149.
- DAGNEW, A.K., BITSUAMLAK, G., & MERRICK, R. (2009) Computational evaluation of wind pressures on tall buildings. *11th Americas Conference on Wind Engineering*. Presented at the 11th Americas Conference on Wind Engineering, San Juan, Puerto Rico.
- DAGNEWA, A.K. & BITSUAMLAKB, G.T. (2010) LES evaluation of wind pressures on a standard tall building with and without a neighboring building. *5th International Symposium on Computational Wind Engineering, (CWE2010)*. Chapel Hill, North Carolina, USA.
- DAGUM, L. & MENON, R. (1998a) OpenMP: an industry standard API for shared-memory programming. *IEEE Computational Science and Engineering*, **5**, 46–55.

- DAGUM, L. & MENON, R. (1998b) OpenMP: An Industry-Standard API for Shared-Memory Programming. *IEEE Comput. Sci. Eng.*, **5**, 46–55.
- DE BOER, A., VAN DER SCHOOT, M.S., & BIJL, H. (2007) Mesh deformation based on radial basis function interpolation. *Computers & Structures*, Fourth MIT Conference on Computational Fluid and Solid Mechanics, **85**, 784–795.
- DE BOER, A., VAN ZUIJLEN, A.H., & BIJL, H. (2007) Review of coupling methods for non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, Domain Decomposition Methods: recent advances and new challenges in engineering, **196**, 1515–1525.
- DE BOOR, C. (1972) On calculating with B-splines. *Journal of Approximation Theory*, **6**, 50–62.
- DERVILIS, N., CHOI, M., TAYLOR, S.G., BARTHORPE, R.J., PARK, G., FARRAR, C.R., & WORDEN, K. (2014) On damage diagnosis for a wind turbine blade using pattern recognition. *Journal of Sound and Vibration*, **333**, 1833–1850.
- DHONDT, G. (2014) CalculiX CrunchiX USER'S MANUAL, version 2.7.
- EMMERICH, M., GIOTIS, A., ÖZDEMİR, M., BÄCK, T., & GIANNAKOGLU, K. (2002) Metamodel—Assisted Evolution Strategies. *Parallel Problem Solving from Nature — PPSN VII, Lecture Notes in Computer Science* (Guervós, J.J.M., Adamidis, P., Beyer, H.-G., Schwefel, H.-P., & Fernández-Villacañes, J.-L. eds). Berlin, Heidelberg: Springer, pp. 361–370.
- EMMERICH, M.T.M., GIANNAKOGLU, K.C., & NAUJOKS, B. (2006) Single- and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, **10**, 421–439.
- FARHAT, C., DEGAND, C., KOOBUS, B., & LESOINNE, M. (1998) An improved method of spring analogy for dynamic unstructured fluid meshes. *39th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference and Exhibit*. American Institute of Aeronautics and Astronautics.
- FARHAT, C., LESOINNE, M., & LE TALLEC, P. (1998) Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, **157**, 95–114.
- FARIN, G. (1992) *Curves and Surfaces for Computer-Aided Geometric Design - A Practical Guide*, 3rd Edition ed. Academic Press.
- FELIPPA, C.A. & PARK, K.C. (1980) Staggered transient analysis procedures for coupled mechanical systems: Formulation. *Computer Methods in Applied Mechanics and Engineering*, **24**, 61–111.
- FERZIGER, J.H. & PERIC, M. (2002) *Computational Methods for Fluid Dynamics*, 3rd ed. Berlin Heidelberg: Springer-Verlag.
- FREY, P.J. & GEORGE, P.L. (2000) *Mesh Generation*. Oxford & Paris: Hermes Science Publishing.
- FUCHS, L. (1986) A local mesh-refinement technique for incompressible flows. *Computers & Fluids*, **14**, 69–81.
- GÁLVEZ, A. & IGLESIAS, A. (2012) Particle swarm optimization for non-uniform rational B-spline surface reconstruction from clouds of 3D data points. *Information Sciences, Swarm Intelligence and Its Applications*, **192**, 174–192.
- GÁLVEZ, A., IGLESIAS, A., & PUIG-PEY, J. (2012) Iterative two-step genetic-algorithm-based method for efficient polynomial B-spline surface reconstruction. *Information Sciences, Nature-Inspired Collective Intelligence in Theory and Practice*, **182**, 56–76.
- GEUZAINÉ, C. & REMACLE, J.-F. (2014) GMSH: A finite element mesh generator with built-in pre- and post-processing facilities, Version 2.8.4. URL <http://geuz.org/gmsh/>.
- GIANNAKOGLU, K.C. (2002) Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Sciences*, **38**, 43–76.
- GIANNAKOGLU, K.C., GIOTIS, A.P., & KARAKASIS, M.K. (2001) Low-cost genetic optimization based on inexact pre-evaluations and the sensitivity analysis of design parameters. *Inverse Problems in Engineering*, **9**, 389–412.
- GIANNAKOGLU, K.C., PAPADIMITRIOU, D.I., & KAMPOLIS, I.C. (2006) Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels. *Computer Methods in Applied Mechanics and Engineering*, **195**, 6312–6329.
- GOIŃSKI, A. & OBUCHOWICZ, A. (2007) 3-D surface reconstruction by the means of evolutionary algorithms. *Evolutionary Computation and Global Optimization*, vol. z. 160. Warsaw

- University of Technology Publishing House, pp. 107–114.
- GOURA, G.S.L., BADCOCK, K.J., WOODGATE, M.A., & RICHARDS, B.E. (2001) A data exchange method for fluid-structure interaction problems. *Aeronautical Journal*, **105**, 215–221.
- GRIFFITH, D.T. & ASHWILL, T.D. (2011) THE SANDIA 100-METER ALL-GLASS BASELINE WIND TURBINE BLADE: SNL100-00 ( No. SAND2011-3779). Sandia National Laboratories.
- GUENNEBAUD, G. & JACOB, B. (2010) Eigen v3. URL <http://eigen.tuxfamily.org>.
- GUENNEBAUD, G., JACOB, B., & OTHERS (2010) Eigen. *Eigen v3*. URL <http://eigen.tuxfamily.org>.
- Gurit (2013) *Wind Turbine Blade Aerodynamics, Wind Energy Handbook*. URL <https://www.gurit.com/>.
- GURUSWAMY, G.P. & BYUN, C. (1993) Fluid-structural interactions using Navier-Stokes flow equations coupled with shell finite element structures. *24th Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, Orlando.
- HAMEED, M.S. & AFAQ, S.K. (2013) Design and analysis of a straight bladed vertical axis wind turbine blade using analytical and numerical techniques. *Ocean Engineering*, **57**, 248–255.
- HANSEN, A.C. & LAINO, D.J. (1998) User's Guide to the Wind Turbine Dynamics Computer Programs YawDyn and AeroDyn for ADAMS® ( No. XAF-4-14076-02). National Renewable Energy Laboratory (NREL).
- HANSEN, M.O.L., SØRENSEN, J.N., VOUTSINAS, S., SØRENSEN, N., & MADSEN, H.A. (2006) State of the art in wind turbine aerodynamics and aeroelasticity. *Progress in Aerospace Sciences*, **42**, 285–330.
- HELENBROOK, B.T. (2003) Mesh deformation using the biharmonic operator. *International Journal for Numerical Methods in Engineering*, **56**, 1007–1021.
- HENRIQUES, J.C.C., MARQUES DA SILVA, F., ESTANQUEIRO, A.I., & GATO, L.M.C. (2009) Design of a new urban wind turbine airfoil using a pressure-load inverse method. *Renewable Energy*, **34**, 2728–2734.
- HOUNJET, M.H.L. & MEIJER, J.J. (1995) Evaluation of Elastomechanical and Aerodynamic Data Transfer Methods for Non-planar Configurations in Computational Aeroelastic Analysis. *International Forum on Aeroelasticity and Structural Dynamics*. pp. 11.1-11.24.
- HSU, S.-Y. & CHANG, C.-L. (2007) Mesh deformation based on fully stressed design: The method and 2-D examples. *International Journal for Numerical Methods in Engineering*, **72**, 606–629.
- HUANG, S., LI, Q.S., & XU, S. (2007) Numerical evaluation of wind effects on a tall steel building by CFD. *Journal of Constructional Steel Research*, **63**, 612–627.
- ILKILIÇ, C., AYDIN, H., & BEHÇET, R. (2011) The current status of wind energy in Turkey and in the world. *Energy Policy*, Special Section on Offshore wind power planning, economics and environment, **39**, 961–967.
- INOMATA, N., TSUCHIYA, K., & YAMADA, S. (1999) Measurement of stress on blade of NEDO's 500 kW prototype wind turbine. *Renewable Energy*, Renewable Energy Efficiency, Policy and the Environment, **16**, 912–915.
- ISKE, A. (2003) Radial basis functions: Basics, advanced topics and meshfree methods for transport problems. *Rendiconti del Seminario Matematico*, **3**.
- ITO, Y. (2013) Challenges in unstructured mesh generation for practical and efficient computational fluid dynamics simulations. *Computers & Fluids*, International Workshop on Future of CFD and Aerospace Sciences, **85**, 47–52.
- JAKOBSSON, S. & AMOIGNON, O. (2007) Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Computers & Fluids*, **36**, 1119–1136.
- JASAK, H. (1996) Error analysis and estimation for the finite volume method with applications to fluid flows (Ph.D. Thesis).
- JENSEN, F.M. (2008) Ultimate strength of a large wind turbine blade (PhD Thesis).
- JENSEN, F.M., FALZON, B.G., ANKERSEN, J., & STANG, H. (2006) Structural testing and numerical simulation of a 34m composite wind turbine blade. *Composite Structures*, Fifteenth International Conference on Composite Materials, **76**, 52–61.
- JIAO, X. & HEATH, M.T. (2004) Overlaying surface meshes, part II: Topology preservation and feature matching. *International Journal of Computational Geometry & Applications*, **14**, 403–419.
- JIMACK, P. & KIRBY, R. (2011) Towards the development of an hp-refinement strategy based upon error estimate sensitivity. *Computers & Fluids - COMPUT FLUIDS*, **46**.
- JONKMAN, J., BUTTERFIELD, S., MUSIAL, W., & SCOTT, G. (2009) Definition of a 5-MW Reference

- Wind Turbine for Offshore System Development ( No. NREL/TP-500-38060). National Renewable Energy Laboratory (NREL).
- JONKMAN, J.M. & BUHL JR., M.L. (2005) Fast User's Guide ( No. NREL/EL-500-38230). National Renewable Energy Laboratory (NREL).
- JØRGENSEN, E.R., BORUM, K.K., MCGUGAN, M., THOMSEN, C.L., JENSEN, F.M., DEBEL, C.P., & SØRENSEN, B.F. (2004) Full scale testing of wind turbine blade to failure - flapwise loading (Report No. 87-550-3183-8). , Denmark. Forskningscenter Risoe. Risoe-R.
- KALLINDERIS, Y. & AHN, H.T. (2005) Incompressible Navier–Stokes method with general hybrid meshes. *Journal of Computational Physics*, **210**, 75–108.
- KAMOUN, B., AFUNGCHUI, D., & ABID, M. (2006) The inverse design of the wind turbine blade sections by the singularities method. *Renewable Energy*, **31**, 2091–2107.
- KAMPOLIS, I.C. & GIANNAKOGLU, K.C. (2009) Distributed evolutionary algorithms with hierarchical evaluation. *Engineering Optimization*, **41**, 1037–1049.
- KAPSOLIS, D., TSIKAS, K., TROMPOUKIS, X., ASOUTI, V., & GIANNAKOGLU, K. (2018) Evolutionary multi-objective optimization assisted by metamodels, kernel PCA and multi-criteria decision making techniques with applications in aerodynamics. *Applied Soft Computing*, **64**, 1–13.
- KARAKASIS, M.K. & GIANNAKOGLU, K.C. (2006) On the use of metamodel-assisted, multi-objective evolutionary algorithms. *Engineering Optimization*, **38**, 941–957.
- KARAKASIS, M.K., KOUBOGIANNIS, D.G., & GIANNAKOGLU, K.C. (2007) Hierarchical distributed metamodel-assisted evolutionary algorithms in shape optimization. *International Journal for Numerical Methods in Fluids*, **53**, 455–469.
- KIM, W.-W. & MENON, S. (1999) An unsteady incompressible Navier–Stokes solver for large eddy simulation of turbulent flows. *International Journal for Numerical Methods in Fluids*, **31**, 983–1017.
- KNUPP, P.M. (2003) Algebraic mesh quality metrics for unstructured initial meshes. *Finite Elements in Analysis and Design*, **39**, 217–241.
- KOINI, G.N., SARAOKINOS, S.S., & NIKOLOS, I.K. (2009) A software tool for parametric design of turbomachinery blades. *Advances in Engineering Software*, **40**, 41–51.
- KOJEKINE, N., SAVCHENKO, V., & HAGIWARA, I. (2004) Surface Reconstruction Based on Compactly Supported Radial Basis Functions. *Geometric Modeling: Techniques, Applications, Systems and Tools* (Sarraz, M. ed). Dordrecht: Springer Netherlands, pp. 217–231.
- KONG, C., BANG, J., & SUGIYAMA, Y. (2005) Structural investigation of composite wind turbine blade considering various load cases and fatigue life. *Energy*, International Symposium on CO2 Fixation and Efficient Utilization of Energy (CandE 2002) and the International World Energy System Conference (WESC-2002), **30**, 2101–2114.
- KONG, C., KIM, T., HAN, D., & SUGIYAMA, Y. (2006) Investigation of fatigue life for a medium scale composite wind turbine blade. *International Journal of Fatigue*, The Third International Conference on Fatigue of Composites, **28**, 1382–1388.
- KOOIJMAN, H.J.T., LINDENBURG, C., WINKELAAR, D., & VAN DER HOOFT, E.L. (2003) DOWEC 6 MW PRE-DESIGN: Aero-elastic modelling of the DOWEC 6 MW pre-design in PHATAS ( No. DOWEC-F1W2-HJK-01-046/9).
- KOUSKOURIS, P.E. & NIKOLOS, I. (2012) Evaluation of Radial Basis Functions for the Deformation of Unstructured Meshes. *8th International Conference on Engineering Computational Technology*, vol. 100. Presented at the Civil-Comp Proceedings.
- KOWOLLIK, D.S.C., HAUPT, M., & HORST, P. (2013) Mesh deformation with exact surface reconstruction using a reduced radial basis function approach. *V International Conference on Computational Methods for Coupled Problems in Science and Engineering*. pp. 467–478.
- KRAUSE, F.-L., FISCHER, A., GROSS, N., & BARHAK, J. (2003) Reconstruction of Freeform Objects with Arbitrary Topology Using Neural Networks and Subdivision Techniques. *CIRP Annals*, **52**, 125–128.
- KROLL, N., HEINRICH, R., KRUEGER, W., & NAGEL, B. (2008) Fluid-Structure Coupling for Aerodynamic Analysis and Design: A DLR Perspective (Invited). *46th AIAA Aerospace Sciences Meeting and Exhibit*. Reno, Nevada, USA: American Institute of Aeronautics and Astronautics.

- LACO, L.I. (2012) Structural response and dynamics of fluid structure-control interaction in wind turbine blades (Ph.D. Thesis).
- LAINO, D.J. & HANSEN, A.C. (2002) User's guide to the Wind Turbine Aerodynamics Computer Software Aerodyn ( No. TCX-9-29209-01). National Renewable Energy Laboratory (NREL).
- LALLEMAND, M.H. (1988a) Schémas décentrés multigrilles pour la résolution des équations d'Euler en éléments finis (PhD Thesis).
- LALLEMAND, M.H. (1988b) Etude de Schemas Runge-Kutta a 4 pas pour la Resolution Multigrille des Equations d' Euler 2D (Raport de Recherche). INRIA.
- LAMOUSIN, H.J. & WAGGENSPACK, N.N. (1994) NURBS-based free-form deformations. *IEEE Computer Graphics and Applications*, **14**, 59–65.
- LAURENT-GENGOUX, P. & MEKHILEF, M. (1993) Optimization of a NURBS representation. *Computer-Aided Design*, **25**, 699–710.
- LEE, J.-W., LEE, J.-S., HAN, J.-H., & SHIN, H.-K. (2012) Aeroelastic analysis of wind turbine blades based on modified strip theory. *Journal of Wind Engineering and Industrial Aerodynamics*, **110**, 62–69.
- LEE, S.G., PARK, S.J., LEE, K.S., & CHUNG, C. (2012) Performance prediction of NREL (National Renewable Energy Laboratory) Phase VI blade adopting blunt trailing edge airfoil. *Energy, Asia-Pacific Forum on Renewable Energy 2011*, **47**, 47–61.
- LEE-RAUSCH, E.M., HAMMOND, D.P., NIELSEN, E.J., PIRZADEH, S.Z., & RUMSEY, C.L. (2010) Application of the FUN3D Solver to the 4th AIAA Drag Prediction Workshop. *28th AIAA Applied Aerodynamics Conference*. Chicago, Illinois, USA.
- LEFRANÇOIS, E. (2008) A simple mesh deformation technique for fluid–structure interaction based on a submesh approach. *International Journal for Numerical Methods in Engineering*, **75**, 1085–1101.
- LEUNG, D.Y.C. & YANG, Y. (2012) Wind energy development and its environmental impact: A review. *Renewable and Sustainable Energy Reviews*, **16**, 1031–1039.
- LEVY, D.W., LAFLIN, K.R., TINOCO, E.N., VASSBERG, J.C., MANI, M., RIDER, B., RUMSEY, C.L., WAHLS, R.A., MORRISON, J.H., BRODERSEN, O.P., CRIPPA, S., MAVRIPLIS, D.J., & MURAYAMA, M. (2013) Summary of Data from the Fifth Computational Fluid Dynamics Drag Prediction Workshop. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. Grapevine, Texas, USA.
- LIU, X., QIN, N., & XIA, H. (2006) Fast dynamic grid deformation based on Delaunay graph mapping. *Journal of Computational Physics*, **211**, 405–423.
- LIU, X., WANG, L., & TANG, X. (2013) Optimized linearization of chord and twist angle profiles for fixed-pitch fixed-speed wind turbine blades. *Renewable Energy*, **57**, 111–119.
- LIU, Y., GUO, Z., & LIU, J. (2012) RBFs-MSA Hybrid Method for Mesh Deformation. *Chinese Journal of Aeronautics*, **25**, 500–507.
- LIVNE, O.E. & WRIGHT, G.B. (2005) Fast multilevel evaluation of 1-D piecewise smooth radial basis function Expansions. *SIAM Conference on Geometric Design and Computing*. Phoenix, Arizona, USA.
- LÖHNER, R. & YANG, C. (1996) Improved ALE mesh velocities for moving bodies. *Communications in Numerical Methods in Engineering*, **12**, 599–608.
- LOMBARDI, M., PAROLINI, N., & QUARTERONI, A. (2013) Radial basis functions for inter-grid interpolation and mesh motion in FSI problems. *Computer Methods in Applied Mechanics and Engineering*, **256**, 117–131.
- LUKE, E., COLLINS, E., & BLADES, E. (2012) A fast mesh deformation method using explicit interpolation. *Journal of Computational Physics*, **231**, 586–601.
- LUO, H., BAUM, J.D., & LOHNER, R. (2005) High-Reynolds Number Viscous Flow Computations Using an Unstructured-Grid Method. *Journal of Aircraft*, **42**, 483–492.
- LYGIDAKIS, G. & NIKOLOS, I. (2015) Numerical Analysis of Flow Over the NASA Common Research Model Using the Academic Computational Fluid Dynamics Code Galatea. *Journal of Fluids Engineering*, **137**.
- LYGIDAKIS, G., SARAOKINOS, S., & NIKOLOS, I. (2014) A Parallel Agglomeration Multigrid Method for Incompressible Flow Simulations. *9th International Conference on Engineering Computational Technology*. Stirlingshire, UK: Civil-Comp Press.

- LYGIDAKIS, G.N. & NIKOLOS, I.K. (2013) Using a High-Order Spatial/Temporal Scheme and Grid Adaptation with a Finite-Volume Method for Radiative Heat Transfer. *Numerical Heat Transfer, Part B: Fundamentals*, **64**, 89–117.
- LYGIDAKIS, G.N. & NIKOLOS, I.K. (2014a) Using a Parallel Spatial/Angular Agglomeration Multigrid Scheme to Accelerate the FVM Radiative Heat Transfer Computation—Part I: Methodology. *Numerical Heat Transfer, Part B: Fundamentals*, **66**, 471–497.
- LYGIDAKIS, G.N. & NIKOLOS, I.K. (2014b) Using a Parallel Spatial/Angular Agglomeration Multigrid Scheme to Accelerate the FVM Radiative Heat Transfer Computation—Part II: Numerical Results. *Numerical Heat Transfer, Part B: Fundamentals*, **66**, 498–525.
- LYGIDAKIS, G.N. & NIKOLOS, I.K. (2014c) Using the DLR-F6 Aircraft Model for the Evaluation of the Academic CFD Code “Galatea.” *ASME International Mechanical Engineering Congress and Exposition*. Montreal, Quebec, Canada.
- LYGIDAKIS, G.N., SARAOKINOS, S.S., & NIKOLOS, I.K. (2016) Comparison of different agglomeration multigrid schemes for compressible and incompressible flow simulations. *Advances in Engineering Software, Civil-Comp - Part 2*, **101**, 77–97.
- LYNCH, D.R. & O’NEILL, K. (1980) Elastic grid deformation for moving boundary problems in two space dimensions. *3rd International Conference in finite elements in water resources*. Mississippi : School of Engineering, University of Mississippi ; Southampton.
- MA, W. & KRUTH, J. (1995) Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. *Computer-Aided Design*, **27**, 663–675.
- MA, W. & KRUTH, J.-P. (1998) NURBS curve and surface fitting for reverse engineering. *Int J Adv Manuf Technol*, **14**, 918–927.
- MALHOTRA, P. (2010) Advanced Blade Testing Methods for Wind Turbines (Master Thesis).
- MALHOTRA, P., HYERS, R.W., MANWELL, J.F., & MCGOWAN, J.G. (2012) A review and design study of blade testing systems for utility-scale wind turbines. *Renewable and Sustainable Energy Reviews*, **16**, 284–292.
- MAMAN, N. & FARHAT, C. (1995) Matching fluid and structure meshes for aeroelastic computations: A parallel approach. *Computers & Structures*, **54**, 779–785.
- MANDELL, J.F., REED, R.M., & SAMBORSKY, D.D. (1992) Fatigue of Fiberglass Wind Turbine Blade Materials (CONTRACTOR REPORT No. SAND92–7005).
- MARTEN, D. & WENDLER, J. (2013) QBlade Guidelines v0.6.
- MARTIN, H.R. (2011) Development of a Scale Model Wind Turbine for Testing of Offshore Floating Wind Turbine Systems (M.Sc Thesis).
- Mathworks (n.d.) *Matlab*. URL <https://www.mathworks.com>.
- MAVRIPLIS, D.J. & VENKATAKRISHNAN, V. (1995) Agglomeration multigrid for two-dimensional viscous flows. *Computers & Fluids*, **24**, 553–570.
- MAVRIPLIS, D.J. & VENKATAKRISHNAN, V. (1996) A 3d Agglomeration Multigrid Solver for the Reynolds–Averaged Navier–Stokes Equations on Unstructured Meshes. *International Journal for Numerical Methods in Fluids*, **23**, 527–544.
- MELBOURNE, W.H. (1980) Comparison of measurements on the CAARC standard tall building model in simulated model wind flows. *Journal of Wind Engineering and Industrial Aerodynamics*, **6**, 73–88.
- MELVILLE, R. (2000) Dynamic aeroelastic simulation of complex configurations using overset grid systems. *Fluids 2000 Conference and Exhibit, Fluid Dynamics and Co-located Conferences*. American Institute of Aeronautics and Astronautics.
- MENTER, F.R. (1994) Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, **32**, 1598–1605.
- MICHLER, A.K. (2011) Aircraft control surface deflection using RBF-based mesh deformation. *International Journal for Numerical Methods in Engineering*, **88**, 986–1007.
- MURAYAMA, M., YAMAMOTO, K., HASHIMOTO, A., ISHIDA, T., UENO, M., TANAKA, K., & ITO, Y. (2013) Summary of JAXA Studies for the Fifth AIAA CFD Drag Prediction Workshop Using UPACS and FaSTAR. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. Grapevine, Texas, USA.
- NIKOLOS, I. (2011) Surrogate modeling in evolutionary based engineering design optimization. *Computational Science, Engineering & Technology Series, Saxe-Coburg Publications*, 173–

- 203.
- NIKOLOS, I.K. (2004) Inverse design of aerodynamic shapes using differential evolution coupled with artificial neural network. *ERCOFTAC Conference in Design Optimization: Methods and Applications*. Athens, Greece.
- NIKOLOS, I.K. (2013) On the use of multiple surrogates within a differential evolution procedure for high-lift airfoil design. *International Journal of Advanced Intelligence Paradigms*, **5**, 319–341.
- NISHIKAWA, H. & DISKIN, B. (2011) Development and Application of Parallel Agglomerated Multigrid Method for Complex Geometries. *20th AIAA Fluid Dynamics Conference*. Presented at the 20th AIAA Fluid Dynamics Conference, Honolulu, Hawaii, USA.
- NISHIKAWA, H., DISKIN, B., THOMAS, J., & HAMMOND, D. (2013) Recent Advances in Agglomerated Multigrid. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. Presented at the 51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition, Grapevine, Texas.
- NREL 5MW Rotor Geometry - NWTC (n.d) . URL <https://wind.nrel.gov/forum/wind/viewtopic.php?f=2&t=440>.
- OBASAJU, E.D. (1992) Measurement of forces and base overturning moments on the CAARC tall building model in a simulated atmospheric boundary layer. *Journal of Wind Engineering and Industrial Aerodynamics*, **40**, 103–126.
- OpenCASCADE (2015) *Open CASCADE Technology, 3D modeling & numerical simulation*. URL <http://www.opencascade.org>.
- PARK, M.A., LAFLIN, K.R., CHAFFIN, M.S., POWELL, N., & LEVY, D.W. (2013) CFL3D, FUN3D, and NSU3D Contributions to the Fifth Drag Prediction Workshop. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. Grapevine, Texas, USA.
- PATRIKALAKIS, N.M. & MAEKAWA, T. (2002) *Shape Interrogation for Computer Aided Design and Manufacturing*. Berlin Heidelberg: Springer-Verlag.
- PÉREZ-ARRIBAS, F. & TREJO-VARGAS, I. (2012) Computer-aided design of horizontal axis turbine blades. *Renewable Energy*, **44**, 252–260.
- PETERS, J. (2000) Patching Catmull-Clark meshes. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques, SIGGRAPH '00*. USA: ACM Press/Addison-Wesley Publishing Co., pp. 255–258.
- PIEGL, L. & TILLER, W. (1987) Curve and surface constructions using rational B-splines. *Computer-Aided Design*, **19**, 485–498.
- PIEGL, L. & TILLER, W. (1995) *The NURBS Book*. , Monographs in Visual Communication. Berlin Heidelberg: Springer-Verlag.
- PIPERNO, S., FARHAT, C., & LARROUTUROU, B. (1995) Partitioned procedures for the transient solution of coupled aeroelastic problems Part I: Model problem, theory and two-dimensional application. *Computer Methods in Applied Mechanics and Engineering*, **124**, 79–112.
- POTSDAM, M. & GURUSWAMY, G. (2001) A parallel multiblock mesh movement scheme for complex aeroelastic applications. *39th Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics.
- PRICE, K., STORN, R.M., & LAMPINEN, J.A. (2005) *Differential Evolution: A Practical Approach to Global Optimization*. , Natural Computing Series. Berlin Heidelberg: Springer-Verlag.
- QUARANTA, G., MASARATI, P., & MANTEGAZZA, P. (2005) A Conservative Mesh-Free Approach for Fluid Structure Interface Problems. *COUPLED PROBLEMS 2005*. Presented at the Computational Methods for Coupled Problems in Science and Engineering, Santorini, pp. 24–27.
- RENDALL, T. & ALLEN, C. (2009a) Parallel efficient mesh motion using radial basis functions with application to multi-bladed rotors. *International Journal for Numerical Methods in Engineering*, **81**, 89–105.
- RENDALL, T. & ALLEN, C. (2010) Reduced surface point selection options for efficient mesh deformation using radial basis functions. *Journal of Computational Physics*, **229**, 2810–2820.
- RENDALL, T.C.S. & ALLEN, C.B. (2008) Unified fluid–structure interpolation and mesh motion using radial basis functions. *International Journal for Numerical Methods in Engineering*, **74**, 1519–1559.

- RENDALL, T.C.S. & ALLEN, C.B. (2009b) Efficient mesh motion using radial basis functions with data reduction algorithms. *Journal of Computational Physics*, **228**, 6231–6249.
- RENDALL, T.C.S. & ALLEN, C.B. (2009c) Improved radial basis function fluid–structure coupling via efficient localized implementation. *Int. J. Numer. Meth. Engng.*, **78**, 1188–1208.
- RESOR, B.R. (2013) Definition of a 5MW/61.5m wind turbine blade reference model ( No. SAND2013-2569). Sandia National Laboratories.
- RIZIOTIS, V.A. & VOUTSINAS, S.G. (1997) GAST: A general aerodynamic and structural prediction tool for wind turbines. Presented at the EWEC, Dublin.
- ROE, P.L. (1981) Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, **43**, 357–372.
- ROSSI, R., COTELA, J., LAFONTAINE, N.M., DADVAND, P., & IDELSOHN, S.R. (2013) Parallel adaptive mesh refinement for incompressible flow problems. *Computers & Fluids*, Selected contributions of the 23rd International Conference on Parallel Fluid Dynamics ParCFD2011, **80**, 342–355.
- SADEGHI, M. & LIU, F. (2005) Coupled Fluid-Structure Simulation for Turbomachinery Blade Rows. *43rd Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics*.
- SAMBORSKY, D.D. & MANDELL, J.F. (1996) Fatigue Resistant Fiberglass Laminates For Wind Turbine Blades. *1996 ASME Wind Energy Symposium*. Presented at the ASME, pp. 46–51.
- SARAKINOS, S.S., LYGIDAKIS, G.N., & NIKOLOS, I.K. (2014) Evaluation of a Parallel Agglomeration Multigrid Finite-Volume Algorithm, Named Galatea-I, for the Simulation of Incompressible Flows on 3D Hybrid Unstructured Grids. *International Mechanical Engineering Congress and Exposition (IMECE 2014)*. Presented at the ASME 2014 International Mechanical Engineering Congress and Exposition, Montreal, Canada.
- SAVIO, E., DE CHIFFRE, L., & SCHMITT, R. (2007) Metrology of freeform shaped parts. *CIRP Annals*, **56**, 810–835.
- SAYED, M.A., KANDIL, H.A., & SHALTOT, A. (2012) Aerodynamic analysis of different wind-turbine-blade profiles using finite-volume method. *Energy Conversion and Management*, IREC 2011, The International Renewable Energy Congress, **64**, 541–550.
- SCALABRIN, L. & FELIX, R. (2013) Grid Assessment Using the NASA Common Research Model (CRM) Wind Tunnel Data. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. Grapevine, Texas, USA.
- SCLAFANI, A., VASSBERG, J., MANI, M., WINKLER, C., DORGAN, A., OLSEN, M., & CODER, J. (2013) DPW-5 Analysis of the CRM in a Wing-Body Configuration Using Structured and Unstructured Meshes. *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. Grapevine, Texas, USA.
- SINGH, V.K., THOMAS, T., & WARUDKAR, V. (2013) Structural design of a wind turbine blade: A review. *International Journal of ChemTech Research*, **5**, 2443–2448.
- SMITH, B.F. (1997) Domain Decomposition Methods for Partial Differential Equations. *Parallel Numerical Algorithms, ICASE/LaRC Interdisciplinary Series in Science and Engineering* (Keyes, D.E., Sameh, A., & Venkatakrishnan, V. eds). Springer Netherlands, pp. 225–243.
- SMITH, M., CESNIK, C., HODGES, D., & MORAN, K. (1995) An evaluation of computational algorithms to interface between CFD and CSD methodologies. *American Institute of Aeronautics and Astronautics, AIAA 96-1400*, 745–755.
- SMITH, M.J., HODGES, D.H., & CESNIK, C.E.S. (2000) Evaluation of Computational Algorithms Suitable for Fluid-Structure Interactions. *Journal of Aircraft*, **37**, 282–294.
- SOBOYEJO, W. (2002) *Mechanical Properties of Engineered Materials*, 1st ed. CRC Press.
- STORN, R. & PRICE, K. (1995) Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization Over Continuous Spaces. *Journal of Global Optimization*, **23**.
- STORN, R. & PRICE, K. (1997) Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces. *Journal of Global Optimization*, **11**, 341–359.
- STROFYLAS, G.A., LYGIDAKIS, G.N., & NIKOLOS, I.K. (2015) Accelerating RBF-Based Mesh Deformation by Implementing an Agglomeration Strategy. *ASME 2015 International Mechanical Engineering Congress and Exposition, IMECE2015-50902*. Presented at the ASME 2015 International Mechanical Engineering Congress and Exposition, Houston, Texas:

- American Society of Mechanical Engineers.
- STROFYLAS, G.A., MAZANAKIS, G.I., & NIKOLOS, I.K. (2014) Wind Turbine Blade Structure Parameterization Using T4T. vol. 11. Presented at the ASME 2014 International Mechanical Engineering Congress and Exposition, American Society of Mechanical Engineers, p. V011T14A026-V011T14A026.
- STROFYLAS, G.A. & NIKOLOS, I.K. (2013) Wind Turbine Blade Design Using T4T. *10th HSTAM International Congress on Mechanics*. Chania, Greece, pp. 25–27.
- STROFYLAS, G.A. & NIKOLOS, I.K. (2015) Reverse Engineering of a Wind Turbine Blade Surface using Differential Evolution. *Proceedings of the 4th International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering*. Stirlingshire, UK.
- STROFYLAS, G.A., PORFYRI, K.N., NIKOLOS, I.K., DELIS, A.I., & PAPAGEORGIOU, M. (2018) Using synchronous and asynchronous parallel Differential Evolution for calibrating a second-order traffic flow model. *Advances in Engineering Software*, **125**, 1–18.
- THEPMONGKORN, S. & KWOK, K.C.S. (2002) Wind-induced responses of tall buildings experiencing complex motion. *Journal of Wind Engineering and Industrial Aerodynamics, Bluff Body Aerodynamics and Applications*, **90**, 515–526.
- THÉVENAZ, P., BLU, T., & UNSER, M. (2000) Interpolation revisited. *IEEE Trans Med Imaging*, **19**, 739–758.
- THOMAS, J.L., DISKIN, B., & NISHIKAWA, H. (2011) A critical study of agglomerated multigrid methods for diffusion on highly-stretched grids. *Computers & Fluids, Implicit Solutions of Navier–Stokes Equations. Special Issue Dedicated to Drs. W.R. Briley and H. McDonald*, **41**, 82–93.
- TIANHANG, X., HAISONG, A., & CHAO, T. (2008) A New Dynamic Mesh Generation Method for Large Movements of Flapping wings with Complex Geometries. *ACTA Aeronautica Et Astronautica Sinica*, **29**, 41–47.
- TSIAKAS, K., GAGLIARDI, F., TROMPOUKIS, X., & GIANNAKOGLU, K. (2016) Shape Optimization of Turbomachinery Rows Using a Parametric Blade Modeller and the Continuous Adjoint Method Running on GPUs. *VII European Congress on Computational Methods in Applied Sciences and Engineering*. Presented at the ECCOMAS, Crete.
- TSIAKAS, K., TROMPOUKIS, X., ASOUTI, V., & GIANNAKOGLU, K. (2015) Shape Optimization of Wind Turbine Blades Using the Continuous Adjoint Method and Volumetric NURBS on a GPU Cluster. *11th International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems*. Presented at the EUROGEN, Glasgow, UK.
- TSIAKAS, K.T., TROMPOUKIS, X.S., ASOUTI, V.G., & GIANNAKOGLU, K.C. (2013) Design-Optimization of a Compressor Blading on a GPU Cluster. *10th International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control with Applications to Industrial and Societal Problems*. Presented at the EUROGEN, Las Palmas de Gran Canaria, Spain.
- VÁRADY, T., MARTIN, R.R., & COX, J. (1997) Reverse engineering of geometric models—an introduction. *Computer-Aided Design, Reverse Engineering of Geometric Models*, **29**, 255–268.
- VEERS, P.S., LANGE, C.H., & WINTERSTEIN, S.R. (1993) FAROW: A tool for fatigue and reliability of wind turbines. *Windpower '93 Conference*. San Francisco, California: American Wind Energy Society.
- WAGNER, T., MICHELITSCH, T., & SACHAROW, A. (2007) On the Design of Optimisers for Surface Reconstruction. *9th Annual Conference on Genetic and Evolutionary Computation*. Presented at the GECCO 2007: Genetic and Evolutionary Computation Conference, London, England, pp. 2195–2202.
- WANG, Q., CHEN, J., PANG, X., LI, S., & GUO, X. (2013) A new direct design method for the medium thickness wind turbine airfoil. *Journal of Fluids and Structures*, **43**, 287–301.
- WANG, Z. & PRZEKWAŚ, A. (1994) Unsteady flow computation using moving grid with mesh enrichment. *32nd Aerospace Sciences Meeting and Exhibit, American Institute of Aeronautics and Astronautics*.
- WARDLAW, R.L. & MOSS, G.F. (1970) A standard tall building model for the comparison of simulated natural winds in wind tunnels ( No. Report CC-662 Tech25).

- WEINERT, K., SURMANN, T., & MEHNEN, J. (2001) Evolutionary Surface Reconstruction Using CSG-NURBS-hybrids. *3rd Annual Conference on Genetic and Evolutionary Computation, GECCO'01*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 1456–1456.
- WEISS, V., ANDOR, L., RENNER, G., & VÁRADY, T. (2002) Advanced surface fitting techniques. *Computer Aided Geometric Design*, **19**, 19–42.
- WENDLAND, H. (2002) Fast evaluation of radial basis functions: methods based on partition of unity. *Approximation Theory X: Wavelets, Splines, and Applications* (Chui, C.K., Schumaker, L.L., & Stoeckler, J. eds). Vanderbilt University Press, pp. 473–483.
- WITTEVEEN, J. (2010) Explicit and Robust Inverse Distance Weighting Mesh Deformation for CFD. *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*. American Institute of Aeronautics and Astronautics.
- XIAOJUN WU, YU, M., & WANG QI XIA (2005) Implicit fitting and smoothing using radial basis functions with partition of unity. *9th International Conference on Computer Aided Design and Computer Graphics (CAD-CG'05)*. Presented at the Computer Aided Design and Computer Graphics (CAD-CG'05), pp. 139–148.
- XIE, W.-C., ZOU, X.-F., YANG, J.-D., & YANG, J.-B. (2012) Iteration and optimization scheme for the reconstruction of 3D surfaces based on non-uniform rational B-splines. *Computer-Aided Design*, **44**, 1127–1140.
- YANG, B. & SUN, D. (2013) Testing, inspecting and monitoring technologies for wind turbine blades: A survey. *Renewable and Sustainable Energy Reviews*, **22**, 515–526.
- YANG, J., PENG, C., XIAO, J., ZENG, J., XING, S., JIN, J., & DENG, H. (2013) Structural investigation of composite wind turbine blade considering structural collapse in full-scale static tests. *Composite Structures*, **97**, 15–29.
- YU, T.-Y. & SONI, B.K. (1999) NURBS in structured Grid Generation. *Handbook of Grid Generation*. CRC Press LLC.
- YU, Y. (1999) Surface Reconstruction from Unorganized Points Using Self-Organizing Neural Networks. *IEEE Visualization 99*. pp. 61–64.
- YUMER, M.E. & KARA, L.B. (2012) Surface creation on unstructured point sets using neural networks. *Computer-Aided Design*, **44**, 644–656.
- ZHANG, W., ZOU, Z., & YE, J. (2012) Leading-edge redesign of a turbomachinery blade and its effect on aerodynamic performance. *Applied Energy*, (1) Green Energy; (2) Special Section from papers presented at the 2nd International Energy 2030 Conf, **93**, 655–667.
- ZHANG, Z., GIL, A.J., HASSAN, O., & MORGAN, K. (2008) The simulation of 3D unsteady incompressible flows with moving boundaries on unstructured meshes. *Computers & Fluids*, Special Issue Dedicated to Professor M.M. Hafez on the Occasion of his 60th Birthday, **37**, 620–631.