National Technical University of Athens

School of Mechanical Engineering
Fluids Department
Laboratory of Thermal Turbomachines
Parallel CFD & Optimization Unit

An Implicit Grid Displacement Method based on a Rigid Body Motion Model

Diploma Thesis
by

Evangelos Christianos

Advisor:
Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2024

Acknowledgements

First, I would like to express my deepest gratitude to my professor, Dr. K. C. Giannakoglou, whose unwavering guidance, support, and trust have been instrumental in the successful completion of this thesis. His extensive knowledge, exemplary work, and constant availability significantly shaped the precision and caliber of my research. Working under his supervision has been both an honor and a valuable learning experience.

Special thanks are also due to Dr. V. Asouti and to Dr. M. Kontou, as well as to all the members of the PCOpt unit. Their expertise and willingness to assist have been invaluable.

I wish to convey my sincere appreciation to the people dearest to me, whose continual encouragement and understanding have been a source of strength throughout my life. Their steadfast belief in my abilities has fueled my determination, providing the necessary motivation to sustain myself through the challenges.

National Technical University of Athens
School of Mechanical Engineering
Fluids Department
Laboratory of Thermal Turbomachines
Parallel CFD & Optimization Unit

An Implicit Grid Displacement Method based on a Rigid Body Motion Model

Diploma Thesis

Evangelos Christianos

Advisor: Kyriakos C. Giannakoglou, Professor NTUA

Athens, 2024

Abstract

In recent years, there has been a growing interest in aerodynamic shape optimization, structural mechanics, and the transient analysis of unsteady phenomena. These fields often involve problems characterized by moving boundaries, making the adaptive grid displacement process crucial. This thesis explores the application of the Rigid Body Motion (RBM) technique on such challenges, displacing the grid nodes according to boundary deformations, drawing inspiration from physics' definition of a rigid body. A great advantage of this method is its ability to maintain the quality of the initially generated grid.

Primary aspects covered in this thesis include the presentation of the theoretical background and mathematical formulation of the RBM technique. Both the decoupled and coupled methods for optimizing grid node displacements are delineated, emphasizing the interdependence among nodes in the latter. A new approach involving the implicit solution of the coupled equations governing node displacement is proposed. Furthermore, a linearization strategy is developed to reduce computational costs, along with the sub-step displacement method to ensure computational feasibility.

A significant feature of the RBM method is its connectivity-agnostic nature, enabling its implementation across various grid types. The thesis considers a range of applications on two-dimensional grids, from simple geometric configurations like concentric squares, to more complex aerodynamic bodies such as airfoils and cascades. Additionally, scenarios in the three-dimensional space, including the deformation of a cube, the rotation of an aircraft, and the bending of a wing, are tested. The results

are analyzed, demonstrating the method's suitability for handling such problems.

Overall, this research underscores the RBM method's capacity to address challenges in computational fluid dynamics (CFD) simulations with moving boundaries, providing a cost-effective and reliable solution for adaptive grid displacement. The thesis concludes with a summary of outcomes and suggestions for future research directions.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Εργαστήριο Θερμικών Στροβιλομηχανών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης

Πεπλεγμένη Μέθοδος Μετατόπισης Πλέγματος βασισμένη στο Μοντέλο Κίνησης Απαραμόρφωτου Σώματος

Διπλωματική εργασία

Ευάγγελος Χριστιανός

Επιβλέπων: Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2024

Περίληψη

Τα τελευταία χρόνια παρατηρείται αυξανόμενο ενδιαφέρον στις περιοχές της βελτιστοποίησης αεροδυναμικής μορφής, της δομικής μηχανικής και της ανάλυσης χρονικά μη-μόνιμων φαινομένων. Σε αυτούς τους τομείς συχνά συναντώνται προβλήματα που χαρακτηρίζονται από μεταβαλλόμενα σύνορα, καθιστώντας τη διαδικασία προσαρμογής του υπολογιστικού πλέγματος απαραίτητη. Η διπλωματική αυτή εργασία εξετάζει την εφαρμογή της τεχνικής Κίνησης Απαραμόρφωτου Σώματος (ΚΑΣ) για την αντιμετώπιση τέτοιων προκλήσεων, μετατοπίζοντας τους κόμβους του πλέγματος σύμφωνα με τις συνθήκες στα όρια, εμπνευσμένη από τον ορισμό του απαραμόρφωτου σώματος στη φυσική. Ένα σημαντικό πλεονέκτημα αυτής της μεθόδου είναι η δυνατότητά της να διατηρεί την ποιότητα του αρχικού πλέγματος.

Οι κύριες πτυχές που καλύπτονται σε αυτήν τη διπλωματική εργασία περιλαμβάνουν την παρουσίαση του θεωρητικού υποβάθρου και της μαθηματικής διατύπωσης της τεχνικής ΚΑΣ. Περιγράφονται τόσο η αποσυζευγμένη όσο και η συζευγμένη μέθοδοι για τον υπολογισμό των βέλτιστων παραμέτρων μετατόπισης των κόμβων του πλέγματος, τονίζοντας την αλληλεξάρτηση μεταξύ των κόμβων στη δεύτερη. Προτείνεται μια νέα προσέγγιση, κατά την οποία οι συζευγμένες εξισώσεις που ελέγχουν τη μετατόπιση των κόμβων λύνονται πεπλεγμένα. Επιπλέον, αναπτύσσεται μια στρατηγική γραμμικοποίησης με σκοπό τη μείωση του υπολογιστικού κόστους, σε συνδυασμό με τη μέθοδο τμηματικής μετατόπισης για την εξασφάλιση της εφικτής εφαρμογής της.

Ένα σημαντικό χαρακτηριστικό της μεθόδου ΚΑΣ είναι ότι η διαδικασία προσδιορισμού των μετατοπίσεων των κόμβων είναι ανεξάρτητη από τη συνδεσιμότητα τους, επιτρέπο-

ντας έτσι την χρήση της σε διάφορους τύπους πλεγμάτων. Η παρούσα διπλωματική εργασία εξετάζει μια ποικιλία εφαρμογών σε διδιάστατα πλέγματα, από απλές γεωμετρικές διαμορφώσεις όπως ομόκεντρα τετράγωνα έως πιο πολύπλοκα αεροδυναμικά σώματα όπως αεροτομές και πτερυγώσεις συμπιεστών και στροβίλων. Επιπλέον, εξετάζονται σενάρια στον τριδιάστατο χώρο που αφορούν στην παραμόρφωση ενός κύβου, στην περιστροφή ενός αεροσκάφους και στην κάμψη μιας πτέρυγας. Τα αποτελέσματα αναλύονται, αποδεικνύοντας την καταλληλότητα της μεθόδου για την αντιμετώπιση τέτοιων προβλημάτων.

Συνολικά, αυτή η έρευνα υπογραμμίζει την ικανότητα της μεθόδου ΚΑΣ να αντιμετωπίζει προκλήσεις στις προσομοιώσεις υπολογιστικής ρευστοδυναμικής (ΥΡΔ) με μεταβαλλόμενα σύνορα, παρέχοντας μια αποδοτική και αξιόπιστη λύση για την προσαρμογή του πλέγματος. Η διπλωματική εργασία ολοκληρώνεται με μια σύνοψη των αποτελεσμάτων και προτάσεις για μελλοντική μελέτη.

Acronyms

NTUA	National Technical University of Athens
PCOpt	Parallel CFD & Optimization
CFD	Computational Fluid Dynamics
2D	Two-Dimensional
3D	Three-Dimensional
RBM	Rigid Body Motion
DoF	Degree of Freedom
PDE	Partial Differential Equation
Eq.	Equation
Fig.	Figure
w.r.t.	with respect to
std	standard deviation

ΕΜΠ	Εθνικό Μετσόβιο Πολυτεχνείο
ΕΘΣ	Εργαστήριο Θερμικών Στροβιλομηχανών
ΜΠΥΡ&Β	Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής & Βελτιστοποίησης
ΥΡΔ	Υπολογιστική Ρευστοδυναμική
2Δ	Διδιάστατο
3Δ	Τριδιάστατο

Contents

Chapter 1

Introduction

1.1 The Concept of Optimization

Optimization, as a fundamental concept, has roots in the quest for efficiency and improvement of various aspects of human life. The essence of optimization lies in the pursuit of optimal solutions to problems, aiming to enhance outcomes and resource utilization, spanning centuries. Over time, this idea evolved from practical challenges faced by ancient civilizations, such as maximizing agricultural yields or constructing functional structures, to becoming a formal discipline in mathematics. The digital age propelled optimization into diverse fields such as engineering, operations research, finance, and artificial intelligence, expanding its applications and methodologies. Today, optimization stands as a cornerstone in decision-making processes, continually evolving with the integration of cutting-edge technologies and interdisciplinary collaborations [1], [2].

The term *Optimization* in mathematics refers to the process of seeking the optimal or most suitable solution to a problem. In most engineering problems, the objective is to maximize or minimize certain quantities, by determining the most suitable value for each design variable. This pursuit aims to maximize desired outcomes or minimize undesired ones, presenting a universal applicability. The solutions usually have multiple dimensions, making the speed of the optimization method a crucial factor. During the search for potential solutions, support from a second computational tool is required, one that evaluates and scores each candidate solution based on the defined objectives [3], [4].

The main distinction among optimization methods is between:

- Deterministic: often referred to as Gradient-Based, these methods use the generalized concept of the derivative of the objective function. Typically, they

necessitate computing the derivatives concerning the design variables [5]. Adjoint methods, a representative of gradient-based optimization, compute the gradient of the objective function with respect to design variables, ensuring satisfaction of the fundamental equations of the problem, such as the Navier-Stokes equations utilized in flow analysis in the field of aerodynamics [6].

- Stochastic: these approaches incorporate elements of random search within the solution space to discover optimal solutions. Evolutionary algorithms, as an example of stochastic methods, solve problems by employing a process analogous to the adaptation found in nature. They handle populations of candidate solutions, and through generations, the population evolves, selecting parents based on their fitness and generating offsprings with potentially improved characteristics [7].

An example of useful optimization software employing evolutionary algorithms is the EASY (Evolutionary Algorithms SYstem) developed by PCOpt/NTUA, which utilizes meta-models and other innovative techniques to reduce computational costs [8], [9], [10].

Optimization methods exhibit further categorization, considering various forms like Single-/Multi-objective, Linear/Non-linear and Constrained/Unconstrained, each tailored to suit specific problems. From its fundamental concepts to the diverse categories that encompass its applications, optimization remains a driving force in shaping advancements across industries. Delving into the intricacies of optimization, one embarks on a journey towards refining solutions, maximizing benefits, and achieving excellence in every endeavor [2].

1.2 Optimization in Aerodynamics

Over the course of history, the examination of aerodynamic behavior analysis, along with the associated design and optimization procedures for aerodynamic bodies, was predominantly reliant on experimental methodologies. However, with the advent of electronic computers, a transformative era unfolded, allowing for the development of sophisticated numerical methods to analyze even the most intricate flow fields [11].

The passage of time reveals a continuous refinement in aerodynamic analysis, particularly in the realm of enhancing the performance of various aerodynamic bodies. Throughout the annals of aeronautics, from the rudimentary approaches of the past to the intricately detailed simulations of today, the quest for optimal aerodynamic performance has been a driving force in aeronautical engineering. The spectrum of applications is broad, ranging from rotor blades of turbo-machines and aircraft wings to the streamlined shapes of automobiles, as depicted in exemplary Figs. 1.1 and 1.2 [12]. In recent years, the focus has shifted towards not only comprehending

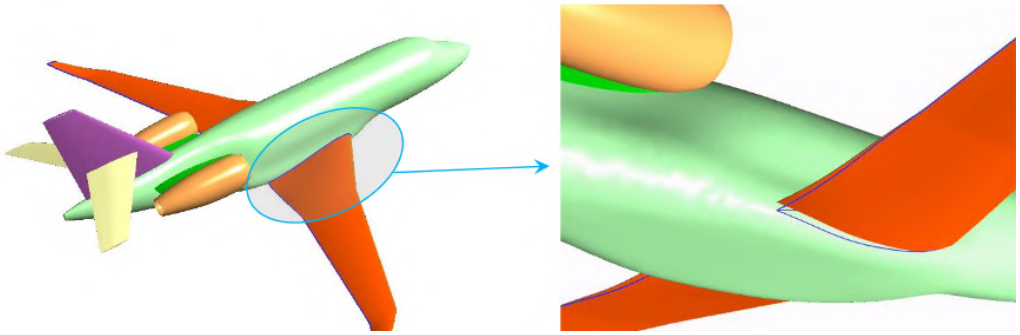
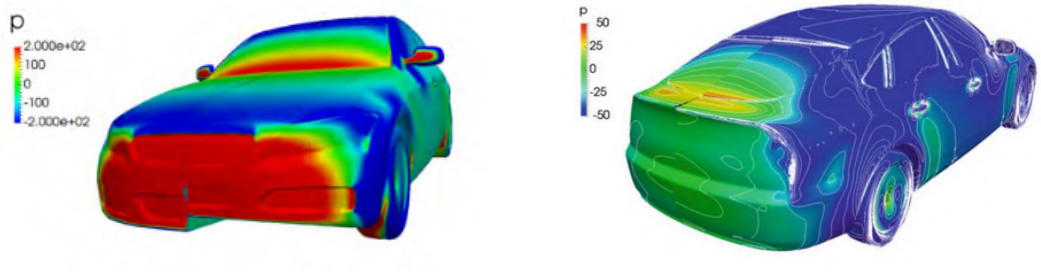


Figure 1.1: *Baseline and optimized shape of a wing of an aircraft. A change in geometry leads to improved aerodynamic behavior. From [14].*

aerodynamic principles but also harnessing computational power in order to further advance optimization methods [13].



(a) *(b)*

Figure 1.2: *Initial (left) and optimized (right) shape of an automobile. The pressure distribution is shown. (a): Front view. The optimization process suggests that lifting the bumper is beneficial for the performance of the car. (b): Rear view. Lowering the trunk leads to an increased pressure on the rear part of the car, which contributes to the reduction of the drag force. From [15].*

This collective array of methods forms the research domain known as *Aerodynamic Shape Optimization*, providing the broader context for the exploration within this thesis [9].

1.3 Need for Adaptive Grid Deformation

A key aspect of aerodynamic shape optimization is finding optimal shapes concerning one or more objectives. To ascertain the optimal solution, candidate solutions must be evaluated using a flow solver, such as Computational Fluid Dynamics (CFD) codes. These codes serve as numerical tools to analyze flow fields and assess, for instance, the performance of a candidate airfoil. Since these problems are solved

using computers, spatial discretization becomes imperative, necessitating a suitable computational grid.

However, the optimization process is iterative, meaning that in each optimization cycle, regardless of the method used, the geometry being studied is modified. Therefore, the grid must be adapted to the new geometry to support the solver in the new optimization cycle. This adaptation is performed at each time step for transient, e.g., aeroelasticity, problems, where fluid-structure interaction is observed, or at each optimization cycle for aerodynamic design optimization problems. After each adaptation, the previous flow quantities can be used as initial values for the new solution (provided that the grid topology is maintained), aiming to achieve faster convergence. It should be taken into account that the quality of the grid plays a critical role in influencing the accuracy and efficiency of numerical simulations. A high-quality grid can accelerate solver convergence and yield precise simulation results, albeit at a higher computational cost. On the contrary, a low-quality grid can compromise the entire procedure, rendering it useless. Achieving an optimal balance between computational cost and grid fineness becomes crucial.

Adaptive grid deformation, denoting the displacement of internal grid nodes in alignment with known displacements on the boundary, emerges as a vital process for solving optimization problems. Traditional grid reconstructing incurs escalated costs, particularly problematic for intricate applications with large 3D grids, a common scenario in industrial settings [16].

In this diploma thesis, the study centers on a grid displacement method known as the Rigid Body Motion Technique (RBM technique).

1.4 Grid Displacement Methods

The need to avoid recomputing the entire grid anew has led to the development of diverse adaptive grid deformation methods, each with its unique attributes and trade-offs. The selection of a method hinges on user requirements, as some prioritize superior grid quality at the expense of higher computational costs or vice versa [16].

The various adaptive grid deformation methods can be broadly categorized into four types:

- Partial Differential Equation (PDE) methods
- Algebraic methods
- Free Form Deformation (FFD) methods
- Physical Analogies

Methods of the first category compute grid movement through the numerical solution of PDEs with appropriate boundary conditions. While relatively straightforward,

they exhibit limited deformation flexibility and may require numerous small steps for significant deformations, making them mainly suitable for problems involving small deformations. The most common equation solved by these methods is the Laplace equation [16], [17].

The second category of grid deformation methods is algebraic methods. These methods define the displacement of each grid node through algebraic relationships dependent on boundary nodes and the relative position of each node to its displacement. Generally, algebraic grid movement techniques do not consider the topology of the grid, making them adaptable to any grid type. Examples include the Radial Basis Functions (RBF) interpolation method and the Delaunay graph mapping method. RBF offers ease of application and high-quality grids but its direct application to large 3D problems incurs high computational costs [16], [18], [19]. The Delaunay graph mapping method, for which the creation of a sparse graph of triangular (2D) or tetrahedral (3D) elements is required, is reliable and produces high-quality grids but it is also characterized by high computational costs [16].

The third category of grid deformation methods is a recent idea originating from the field of animated designs and digital characters. Free Form Deformation (FFD) is a geometric technique used to model small deformations in solid bodies. In this technique, modifying the geometry does not involve changing its geometric parameters but rather deforming the space within which the aerodynamic body resides through the deformation of a control grid. Thus, to modify the geometry of a body, the user or optimization method only needs to shift the control points of the control grid, simultaneously displacing all points of the aerodynamic body [20]. Notably, the Adaptive Grid Deformation using Harmonic Coordinates is an example within this category [21].

The majority of developed grid displacement methods belong to the last category, physical analogies. The three most widespread and reliable techniques in this category include the linear spring analogy method, the torsional spring analogy method, and the elastic analogy method [16]. In the linear spring analogy method, the whole computational domain is conceptualized as a network of linear springs connected at grid nodes, with deformation determined through the solution of static equilibrium equations for the entire system. Although this method is straightforward to apply, it faces stability issues with large displacements and fine grids, leading to the appearance of inverted cells in the grid [16]. The torsional spring analogy involves processing the movements of boundary nodes, calculating forces exerted by imaginary springs on nodes, and determining displacements of internal nodes, facilitating grid adaptation based on the shifted contour [22], [23], [24]. In the elastic analogy method, the entire computational domain is modeled as an elastic solid, with grid deformation dictated by classical laws of elastic solid theory [25]. While the elastic method provides considerable flexibility compared to spring methods, it comes with higher computational costs, a characteristic shared by all grid deformation methods inspired by physical analogies.

Another method that belongs in the physical analogy category is the ***Rigid Body Motion (RBM) technique***, presented in this diploma thesis. RBM operates by modifying the grid based on the motion of an undeformed body. Specifically, the elements of the grid, when displaced, are considered to imitate the motion of a rigid body. The method's operation is explained in detail in subsequent chapters.

These categories showcase the diverse landscape of adaptive grid deformation methods, each offering specific advantages and challenges, underscoring the importance of selecting an approach tailored to the particular demands of the aerodynamic optimization process [26].

1.5 The Rigid Body Motion Technique

The Rigid Body Motion (RBM) Technique, the focal point of this thesis, emerges as a potent tool for displacing internal grid nodes, leveraging known displacements of the boundary nodes of the shape under optimization.

In mechanics, a rigid body is conceptualized as a solid entity for which the fundamental condition holds that the distance between any two internal points remains constant during the body's motion in space (Fig. 1.3) [27]. Although many problems treat studied bodies as inherently solid, this oversimplified approach neglects crucial real-world factors affecting the body, such as stresses, vibrations, and material property variations.

This theory can be linked to the adaptive deformation of computational grids through the following rationale. Around each internal node of a 2D structured grid (although this example can seamlessly extend to unstructured and hybrid grids), there exist 8 (or 26 for 3D problems) nodes referred to as 'neighbors' of the internal node. The cell formed by these neighbors is treated as an ideal solid body, aspiring

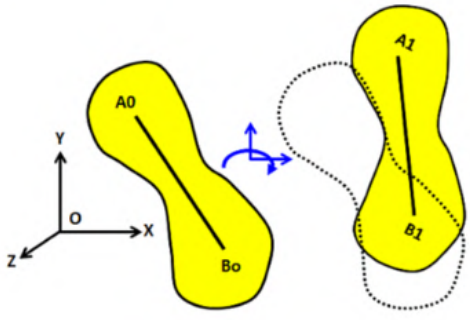


Figure 1.3: *Depiction of a rigid body motion in 2D. The distance between any two given points A and B of the body remains constant ($A_0B_0 = A_1B_1$).*

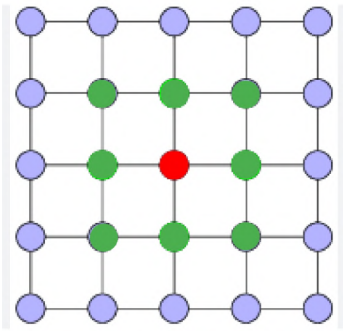


Figure 1.4: *Depiction of a 2D structured grid. The nodes in green color are the neighbors. The area defined by these nodes forms the cell, of which the representative is the node in red.*

to possess the previously described properties (Fig. 1.4). Therefore, the objective of the Rigid Body Motion (RBM) method is to displace the grid cells without altering their original shape. While achieving this for the entire grid is an ideal scenario, it is not practically attainable due to the unrestricted movement of boundaries. Consequently, the method strives for minimal deviation from this ideal state.

The method presented and implemented in this thesis falls within the category of Physical Analogies, as the cell around each internal grid node is modeled as an inherently solid body. The computation of the new position of internal nodes is determined based on the displacement of the boundary nodes of the grid. The primary aim is that the shape of elements in the final grid does not differ significantly from the initial grid, all while maintaining a low computational cost. These dual characteristics establish it as a reliable adaptive grid deformation method suitable for aerodynamic optimization problems.

The next chapters provide a detailed description of the rigid body motion technique for both 2D and 3D problems.

1.6 Structure of the Thesis

The upswing observed in the fields of Aerodynamic Shape Optimization and Aeroelasticity (or Hydroelasticity or, overall, Fluid Structure Interaction) constitutes a dynamic and evolving research area. This expansive and multidisciplinary field seeks to unravel the complexities of aerodynamic design, paving the way for innovative solutions and improved performance. Within this context, this thesis is positioned to contribute, adding to the collective knowledge aimed at optimizing aerodynamic shapes across diverse objectives and operating conditions.

This diploma thesis focuses on the implementation, programming, testing and evaluation of the RBM technique on structured, unstructured and hybrid, 2D and 3D

computational grids, aiming to draw useful conclusions about its effectiveness. The method is programmed in the *C++* programming language, while also using the *ParaView* [28] open-source software for the analysis and visualization of the grid. The structure of the thesis is outlined as follows:

Chapter 1 : Introduction to the whole concept and context, outline of the method that will be presented and description of the general aim of contribution of this thesis.

Chapter 2 : Presentation of the theoretical background and mathematical formulation of the RBM technique in two and three dimensions, a description of an enhancement of the method, definition of quality metrics for grids and explanation of the basic algorithm.

Chapter 3 : Application of the method to a variety of structured and unstructured 2D grids, presenting the results.

Chapter 4 : Application of the RBM technique to various 3D grids, with an analysis of the final results.

Chapter 5 : Summary and conclusions of the thesis, along with suggestions for future work.

Chapter 2

The RBM Technique

As explained in the Introduction, the *RBM technique* is a method employed to adapt a computational grid to the displacements of its boundaries. It is widely used within engineering disciplines, such as aerodynamics and structural mechanics, or their interaction, either in problems dealing with moving boundaries within a flow (aeroelasticity) or in optimization loops, when moving from cycle to cycle by changing the shape to be designed, where grids are required.

2.1 Theoretical Foundation

In *Physics*, a *rigid body* is defined as an assembly of particles that do not move relative to each other. Under this condition, a motion of a rigid body does not induce any deformation within the body [27]. The *RBM Technique*, as a *Grid Displacement Method*, derives from this definition and considers the properties of the elements of the grid to be analogous to those of a rigid body, during its displacement.

To better understand this method, Fig. 2.1 is presented, alongside pertinent definitions utilized throughout this thesis. Fig. 2.1 illustrates an elementary grid structure, used solely for the enhancement of comprehension of the method's operational principles.

A *node* is a zero-dimension entity which is inherently linked to the topology of the grid, accommodating specific Degrees of Freedom (DoFs), according to the attributes of the grid. An *edge* represents a segment which connects two nodes of the grid, determining the connectivity of the grid. A *face* represents the area defined by a closed loop of interconnected edges, in which no other nodes exist. These definitions can extend to 3D grids. Importantly, these entities lack inherent physical significance; rather, they facilitate the discretization of the domain for the calculation of

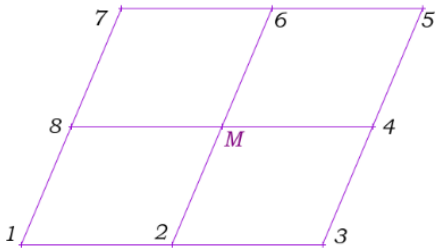


Figure 2.1: *A stencil of a 2D structured grid. Depicting the central node M and its neighbors, nodes 1 to 8. From [29].*

desired quantities.

Regarding the implementation of the RBM displacement model and as indicated in Fig. 2.1, it is desired that, given the displacements of the neighboring nodes *1* to *8*, the central node M is also displaced accordingly, following the motion (translation and rotation) that is dictated by the properties of a rigid body [29].

The nodes of a grid are classified in two main categories: *Internal* and *Boundary nodes*. Boundary nodes are the nodes, the displacement of which is user-defined, for example in the case of the nodes on the boundaries of the studied domain needing to be fixed at a certain position, or determined by an external tool, for example an optimization algorithm that is responsible for changing the coordinates of a subset of the nodes in order to achieve a user-defined function, therefore known in advance. Internal nodes are the rest of the nodes of the grid, on which the RBM technique is eventually applied.

While the central node M of Fig. 2.1 is certainly an internal node, any neighbor of it can be either an internal node as well or a boundary node. In case the neighbor is an internal node, its final position must be determined by making an initial guess, as its displacement is also a sought-after value. Conversely, if the neighbor is a boundary node, its final position is externally defined, known with certainty, and remains fixed throughout the entire solution process.

Assuming that the final positions of all neighbors of the central node M in Fig. 2.1 are either known or estimated, the central node M can also undergo displacement. In this general scenario, each neighbor's displacement is defined by a distinct set of parameters $(\Delta x, \Delta y, \theta)$. To align as closely as possible with the rigid body definition, the identification of the best-fit set $(\Delta x, \Delta y, \theta)$, that optimally describes the motion of the neighbors, is needed. This set is then applied to the central node. Achieving an ideal adherence to the properties of a rigid body is challenging, turning the problem into an optimization task. The objective function aims to ensure that the rigid body definition is respected to the fullest extent possible.

2.2 Formulation of the Problem in 2D

The Objective Function

The mathematical expression for the displacement of a node under pure translation and rotation in 2D is:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{2.1}$$

In Eq. 2.1, (x, y), $(x^{'}, y^{'})$ denote the initial and final coordinates of the node respectively, $(\Delta x, \Delta y)$ denote the translation in the x and y axes respectively and θ denotes the angle of rotation around the z axis, with the global origin $(0, 0)$ as the center of rotation. If a different center of rotation is considered, the displacement from the initial (x, y) to the final coordinates $(x^{'}, y^{'})$ would be achieved by a different set of $(\Delta x, \Delta y, \theta)$, without affecting the outcome.

According to the assumption made that the grid is displaced by approximating as much as possible the motion of a rigid body, the *Objective Function (F)* that needs to be minimized for each *Internal* node i in a 2D grid is:

$$F_i \;=\; \frac{1}{2} \sum_{j \in \mathcal{N}(i)} [(x_j^{ideal} - x_j^{new})^2 + (y_j^{ideal} - y_j^{new})^2] \tag{2.2}$$

In Eq. 2.2, i denotes the central node of the stencil, j denotes a neighboring node of the central one and $\mathcal{N}(i)$ is the set of all the neighbors of the central node. The x_j^{ideal}, y_j^{ideal} denote the x, y coordinates of neighbor j respectively, after its displacement, supposing that the stencil indeed complies with the properties of a rigid body. The x_j^{new}, y_j^{new} denote the actual x, y coordinates of neighbor j respectively, after its displacement. Interpreting this objective, the aim is to minimize the disparity between the real final position (x_j^{new}, y_j^{new}) and the ideal one (x_j^{ideal}, y_j^{ideal}) which corresponds to a pure rigid motion.

The aim is to minimize the objective function (F) for every i, $i \in \mathcal{I}$, where $\mathcal{I}$ represents the set of internal nodes in the grid. This optimization task can be approached in two ways: decoupled or coupled with the corresponding objective functions of the other internal nodes in the grid. In the decoupled approach, the solution for node i is determined locally, not considering the type of neighboring nodes, internal or boundary, and assuming a fixed final position for all neighbors. In contrast, the coupled approach involves a more significant interdependence among the displacements of grid nodes. This allows for better flexibility in seeking the displacement of internal grid nodes.

To effectively implement the Coupled RBM Technique, it is essential to express the

corresponding objective function. The new objective function takes the interdependence among all the nodes of the grid into account. The *Total Objective Function* thus becomes:

$$F_{total} = \sum_{i \in \mathcal{I}} F_i \tag{2.3}$$

where $\mathcal{I}$ denotes the set of the Internal nodes (nodes of unknown displacement).

To ascertain the optimal set $(\Delta x, \Delta y, \theta)$ for each internal node i, it is necessary to compute the derivatives of the objective function with respect to the unknown quantities Δx_i, Δy_i, and θ_i, and set them to zero. For the decoupled method, this involves the derivatives of F_i as defined in Eq. 2.2, whereas, for the coupled method, it pertains to the derivatives of F_{total} as defined in Eq. 2.3.

In the decoupled method, the derivatives take the form:

$$\frac{\partial F_i}{\partial \Delta x_i} = 0 \tag{2.4}$$

$$\frac{\partial F_i}{\partial \Delta y_i} = 0 \tag{2.5}$$

$$\frac{\partial F_i}{\partial \theta_i} = 0 \tag{2.6}$$

Conversely, in the coupled method, they are represented as:

$$\frac{\partial F_{total}}{\partial \Delta x_i} = 0 \tag{2.7}$$

$$\frac{\partial F_{total}}{\partial \Delta y_i} = 0 \tag{2.8}$$

$$\frac{\partial F_{total}}{\partial \theta_i} = 0 \tag{2.9}$$

In case the decoupled method is utilized, the set of Eqs. 2.4, 2.5 and 2.6 forms a 3×3 non-linear system of equations (6×6 in 3D, where the unknowns are Δx_i, Δy_i, Δz_i, θ_{x_i}, θ_{y_i}, and θ_{z_i}). Solving this system provides the optimal set of quantities for displacing node i. If the coupled method is employed, Eqs. 2.7, 2.8, and 2.9 along with the corresponding equations for all internal nodes collectively form a system of equations, which, when solved, provides the optimal set of quantities for displacing all nodes simultaneously. The analytical expressions for these equations (derivatives) are detailed in the following section.

The terms of Eq. 2.2, based on Eq. 2.1, can be written as:

$$x_j^{ideal} = x_j \cos\theta_i + y_j \sin\theta_i + \Delta x_i \tag{2.10}$$

$$y_j^{ideal} = -x_j \sin\theta_i + y_j \cos\theta_i + \Delta y_i \tag{2.11}$$

$$x_j^{new} = \begin{cases} x_j \cos\theta_j + y_j \sin\theta_j + \Delta x_j & \text{, if } j \in \mathcal{IN}(i) \\ x_j^* & \text{, if } j \in \mathcal{BN}(i) \end{cases} \tag{2.12}$$

$$y_j^{new} = \begin{cases} -x_j \sin\theta_j + y_j \cos\theta_j + \Delta y_j & \text{, if } j \in \mathcal{IN}(i) \\ y_j^* & \text{, if } j \in \mathcal{BN}(i) \end{cases} \tag{2.13}$$

where $\mathcal{IN}(i)$ denotes the set of Internal Neighbors of the node i, $\mathcal{BN}(i)$ the set of Boundary Neighbors of the node i and the asterisk (*) indicates that these quantities are already known as boundary conditions.

2.3 Solution of the Optimization Problem

The solving process allows for two strategies, as previously outlined: either a decoupled method or a coupled method. In the decoupled method, the solution for each node is not affected by the type of the neighboring nodes, whether they are internal or boundary, while the coupled method considers a more pronounced interdependence among the grid nodes' displacements. The resolution of the system of equations, namely 2.4, 2.5, and 2.6 for the decoupled method or 2.7, 2.8, and 2.9 for the coupled method, is essential to determine the optimal values Δx, Δy, and θ, for displacing each internal node specifically.

The primary focus of this diploma thesis revolves around computing the nodal displacements of a grid by implicitly solving the corresponding equations of the coupled method. The purpose is to investigate if there is a gain in computational cost, compared to the decoupled method [29].

2.3.1 Decoupled Method

Analyzing firstly the Decoupled Method of the methodology discussed, in order to compute the displacement of the central node, the displacements of all the neighbors of it need to be known beforehand. If a neighboring node is a boundary node, then its displacement is indeed known. If, though, the neighbor is an internal node, its displacement may not be known at the time when the displacement of the central one is computed. As the internal nodes of the grid are sequentially displaced during each

iteration, the displacement of the neighbor becomes known if it has been displaced earlier in that iteration; otherwise, it remains unknown. To cope with this, the final position of this neighbor is considered to be the same as its initial position. Following this adjustment, the nodes are displaced successively, starting from the nodes close to the boundary ones. The main characteristic of this approach is that iterations are required, in each of which all the internal nodes are displaced, until the objective function F_i defined in Eq. 2.2 converges for every internal node or a different criterion is met.

Algorithm 1 Decoupled Method Algorithm

```
1: procedure DISPLACEMENT(nodes, edges, faces)
2:     for all InternalNodes do
3:         InternalNode.NewCoordinates ← InternalNode.OldCoordinates
4:     end for
5:     iteration ← 0
6:     while (not converged or stopping criterium not met) do
7:         for all InternalNodes do
8:             Update InternalNode.NewCoordinates
9:         end for
10:        Check Convergence
11:        iteration ← iteration + 1
12:    end while
13: end procedure
```

The system of equations required to be solved to determine the parameters for displacing each node i, previously introduced consisting of Eqs. 2.4, 2.5, and 2.6, is now presented in a more detailed form,

$$\frac{\partial F_i}{\partial \Delta x_i} = \sum_{j \in \mathcal{N}(i)} (x_j \cos\theta_i + y_j \sin\theta_i + \Delta x_i - x_j^*) = 0 \tag{2.14}$$

$$\frac{\partial F_i}{\partial \Delta y_i} = \sum_{j \in \mathcal{N}(i)} (-x_j \sin\theta_i + y_j \cos\theta_i + \Delta y_i - y_j^*) = 0 \tag{2.15}$$

$$\begin{aligned}\frac{\partial F_i}{\partial \theta_i} = & \sum_{j \in \mathcal{N}(i)} [(x_j \cos\theta_i + y_j \sin\theta_i + \Delta x_i - x_j^*)(-x_j \sin\theta_i + y_j \cos\theta_i)] + \\ & + \sum_{j \in \mathcal{N}(i)} [(-x_j \sin\theta_i + y_j \cos\theta_i + \Delta y_i - y_j^*)(-x_j \cos\theta_i - y_j \sin\theta_i)] = 0\end{aligned} \tag{2.16}$$

This 3×3 non-linear system of equations, regarding node i, is represented by Eqs. 2.14, 2.15, and 2.16, with unknowns Δx_i, Δy_i, and θ_i. Firstly, the Δx_i and Δy_i quantities are computed, according to Eqs. 2.17 and 2.18, derived by Eqs. 2.14 and 2.15:

$$\Delta x_i = \frac{1}{n} \sum_{j \in \mathcal{N}(i)} (x_j^* - x_j \cos\theta_i - y_j \sin\theta_i) \tag{2.17}$$

$$\Delta y_i = \frac{1}{n} \sum_{j \in \mathcal{N}(i)} (y_j^* + x_j \sin\theta_i - y_j \cos\theta_i) \tag{2.18}$$

where n equals the number of neighbors of node i.

Eq. 2.16 is written as:

$$\frac{\partial F_i}{\partial \theta_i} = A \sin\theta_i + B \cos\theta_i = 0 \tag{2.19}$$

where A, B equal to:

$$A = \sum_{j \in \mathcal{N}(i)} (x_j x_j^* + y_j y_j^* - x_j \Delta x_i - y_j \Delta y_i) \tag{2.20}$$

$$B = \sum_{j \in \mathcal{N}(i)} (x_j y_j^* - y_j x_j^* + y_j \Delta x_i - x_j \Delta y_i) \tag{2.21}$$

Eq. 2.19 is effectively solved using the *Newton-Raphson* iterative method [30], for which the steps are as follows:

$$\text{set :} \quad G_i = A \sin\theta_i + B \cos\theta_i = 0 \tag{2.22a}$$

$$\text{compute :} \quad G_i' = \frac{\partial G_i}{\partial \theta_i} = A \cos\theta_i - B \sin\theta_i \tag{2.22b}$$

$$\text{update angle :} \quad \theta_i^{new} = \theta_i^{old} - \frac{G_i}{G_i'} \tag{2.22c}$$

The steps of Eq. 2.22 iterate until $|\theta_i^{new} - \theta_i^{old}| < \epsilon$, $\epsilon << 1$. Then, Δx_i and Δy_i are recomputed based on the value of θ_i^{new}. The iterative process described by Eqs. 2.17-2.22 continues until convergence is achieved.

Post-solution, the node i undergoes displacement, and the system of equations for the next internal node is formulated. This iterative process spans all internal nodes, culminating in a displaced grid.

2.3.2 Coupled Method

The coupled approach emphasizes the significant interdependence among grid nodes' displacements by customizing the objective function accordingly. The Total Objective Function (F_{total}) aggregates individual objective functions for internal nodes (F_i), creating a system of equations. Resolving this system produces the optimal set of parameters for displacing each and every internal node, recognising at the same time that each node possesses a unique set of displacement parameters $(\Delta x_i, \Delta y_i, \theta_i)$. Within this approach, instead of solving a 3×3 non-linear system of equations for each internal node i individually and iteratively displacing the nodes, a non-linear system of equations associating all grid nodes is formulated and solved, facilitating a simultaneous transition of nodes from their initial to final positions.

Taking Eqs. 2.2 and 2.10-2.13 into consideration, the total objective function becomes:

$$\begin{aligned} F_{total} \; = & \frac{1}{2} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{IN}(i)} [(x_j \cos\theta_i + y_j \sin\theta_i + \Delta x_i) - (x_j \cos\theta_j + y_j \sin\theta_j + \Delta x_j)]^2 + \\ & + \frac{1}{2} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{BN}(i)} [(x_j \cos\theta_i + y_j \sin\theta_i + \Delta x_i) - (x_j^*)]^2 + \\ & + \frac{1}{2} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{IN}(i)} [(-x_j \sin\theta_i + y_j \cos\theta_i + \Delta y_i) - (-x_j \sin\theta_j + y_j \cos\theta_j + \Delta y_j)]^2 + \\ & + \frac{1}{2} \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{BN}(i)} [(-x_j \sin\theta_i + y_j \cos\theta_i + \Delta y_i) - (y_j^*)]^2 \end{aligned} \tag{2.23}$$

In order to minimize F_{total}, the derivatives of Eq. 2.23 with respect to the unknown quantities Δx_i, Δy_i and θ_i need to be computed and set to zero, for every $i \in \mathcal{I}$.

$$\begin{aligned} \frac{\partial F_{total}}{\partial \Delta x_i} \; = & \sum_{j \in \mathcal{IN}(i)} [(x_j \cos\theta_i + y_j \sin\theta_i + \Delta x_i) - (x_j \cos\theta_j + y_j \sin\theta_j + \Delta x_j)] - \\ & - \sum_{j \in \mathcal{IN}(i)} [(x_i \cos\theta_j + y_i \sin\theta_j + \Delta x_j) - (x_i \cos\theta_i + y_i \sin\theta_i + \Delta x_i)] + \\ & + \sum_{j \in \mathcal{BN}(i)} [(x_j \cos\theta_i + y_j \sin\theta_i + \Delta x_i) - (x_j^*)] \; = \; 0 \end{aligned} \tag{2.24}$$

$$\begin{aligned}\frac{\partial F_{total}}{\partial \Delta y_i} &= \sum_{j\in\mathcal{IN}(i)} [(-x_j \sin\theta_i + y_j \cos\theta_i + \Delta y_i) - (-x_j \sin\theta_j + y_j \cos\theta_j + \Delta y_j)] - \\ &- \sum_{j\in\mathcal{IN}(i)} [(-x_i \sin\theta_j + y_i \cos\theta_j + \Delta y_j) - (-x_i \sin\theta_i + y_i \cos\theta_i + \Delta y_i)] + \\ &+ \sum_{j\in\mathcal{BN}(i)} [(-x_j \sin\theta_i + y_j \cos\theta_i + \Delta y_i) - (y_j^*)] \;=\; 0\end{aligned} \tag{2.25}$$

$$\begin{aligned}\frac{\partial F_{total}}{\partial \theta_i} &= \sum_{j\in\mathcal{IN}(i)} \{[(x_j \cos\theta_i + y_j \sin\theta_i + \Delta x_i) - (x_j \cos\theta_j + y_j \sin\theta_j + \Delta x_j)] \cdot \\ &\qquad \cdot (-x_j \sin\theta_i + y_j \cos\theta_i)\} \; - \\ &- \sum_{j\in\mathcal{IN}(i)} \{[(x_i \cos\theta_j + y_i \sin\theta_j + \Delta x_j) - (x_i \cos\theta_i + y_i \sin\theta_i + \Delta x_i)] \cdot \\ &\qquad \cdot (-x_i \sin\theta_i + y_i \cos\theta_i)\} \; + \\ &+ \sum_{j\in\mathcal{BN}(i)} \big\{[(x_j \cos\theta_i + y_j \sin\theta_i + \Delta x_i) - (x_j^*)] \cdot \\ &\qquad \cdot (-x_j \sin\theta_i + y_j \cos\theta_i)\} \; + \\ &+ \sum_{j\in\mathcal{IN}(i)} \{[(-x_j \sin\theta_i + y_j \cos\theta_i + \Delta y_i) - (-x_j \sin\theta_j + y_j \cos\theta_j + \Delta y_j)] \cdot \\ &\qquad \cdot (-x_j \cos\theta_i - y_j \sin\theta_i)\} \; - \\ &- \sum_{j\in\mathcal{IN}(i)} \{[(-x_i \sin\theta_j + y_i \cos\theta_j + \Delta y_j) - (-x_i \sin\theta_i + y_i \cos\theta_i + \Delta y_i)] \cdot \\ &\qquad \cdot (-x_i \cos\theta_i - y_i \sin\theta_i)\} \; + \\ &+ \sum_{j\in\mathcal{BN}(i)} \big\{[(-x_j \sin\theta_i + y_j \cos\theta_i + \Delta y_i) - (y_j^*)] \cdot \\ &\qquad \cdot (-x_j \cos\theta_i - y_j \sin\theta_i)\} \; = \; 0\end{aligned} \tag{2.26}$$

It is evident that the unknown quantities Δx_i, Δy_i, and θ_i appear in both the summation related to node i and the summation across each neighboring node j, hence leading to the form of Eqs. 2.24-2.26.

Eqs. 2.24, 2.25 and 2.26, from now on named $Eq1$, $Eq2$ and $Eq3$, simplified, result in:

$$
\begin{aligned}
Eq1 \;=\; \frac{\partial F_{total}}{\partial \Delta x_i} \;=\; & \sum_{j \in \mathcal{IN}(i)} [(x_i + x_j)(\cos\theta_i - \cos\theta_j) + (y_i + y_j)(\sin\theta_i - \sin\theta_j) + \\
& + \; 2(\Delta x_i - \Delta x_j)] + \\
& + \sum_{j \in \mathcal{BN}(i)} (x_j \cos\theta_i + y_j \sin\theta_i + \Delta x_i - x_j^*) \;=\; 0
\end{aligned}
\tag{2.27}
$$

$$
\begin{aligned}
Eq2 \;=\; \frac{\partial F_{total}}{\partial \Delta y_i} \;=\; & \sum_{j \in \mathcal{IN}(i)} [(x_i + x_j)(\sin\theta_j - \sin\theta_i) + (y_i + y_j)(\cos\theta_i - \cos\theta_j) + \\
& + \; 2(\Delta y_i - \Delta y_j)] + \\
& + \sum_{j \in \mathcal{BN}(i)} (-x_j \sin\theta_i + y_j \cos\theta_i + \Delta y_i - y_j^*) \;=\; 0
\end{aligned}
\tag{2.28}
$$

$$
\begin{aligned}
Eq3 \;=\; \frac{\partial F_{total}}{\partial \theta_i} \;=\; & \sum_{j \in \mathcal{IN}(i)} \{(\Delta x_j - \Delta x_i)\,[(x_i + x_j)\sin\theta_i - (y_i + y_j)\cos\theta_i] \; + \\
& + (\Delta y_j - \Delta y_i)\,[(y_i + y_j)\sin\theta_i + (x_i + x_j)\cos\theta_i] \; + \\
& + (\sin\theta_i \cos\theta_j - \cos\theta_i \sin\theta_j)(x_i^2 + y_i^2 + x_j^2 + y_j^2)\} \; + \\
& + \sum_{j \in \mathcal{BN}(i)} [(\Delta x_i - x_j^*)\,(-x_j \sin\theta_i + y_j \cos\theta_i) \; + \\
& + \; (\Delta y_i - y_j^*)\,(-x_j \cos\theta_i - y_j \sin\theta_i)] \;=\; 0
\end{aligned}
\tag{2.29}
$$

The derivatives of Eqs. 2.27, 2.28, 2.29, with respect to the unknown quantities Δx_i, Δy_i, θ_i and Δx_j, Δy_j, θ_j, $j \in \mathcal{IN}(i)$, which will be the entries to the Jacobian matrix of the non-linear system of equations, are computed:

$$
\frac{\partial Eq1}{\partial \Delta x_i} \;=\; \sum_{j \in \mathcal{IN}(i)} 2 + \sum_{j \in \mathcal{BN}(i)} 1 \tag{2.30}
$$

$$
\frac{\partial Eq1}{\partial \Delta y_i} \;=\; 0 \tag{2.31}
$$

$$\begin{aligned}\frac{\partial Eq1}{\partial \theta_i} &= \sum_{j\in\mathcal{IN}(i)} [-(x_i+x_j)\sin\theta_i + (y_i+y_j)\cos\theta_i] + \\ &+ \sum_{j\in\mathcal{BN}(i)} (-x_j\sin\theta_i + y_j\cos\theta_i)\end{aligned} \tag{2.32}$$

$$\frac{\partial Eq2}{\partial \Delta x_i} = 0 \tag{2.33}$$

$$\frac{\partial Eq2}{\partial \Delta y_i} = \sum_{j\in\mathcal{IN}(i)} 2 + \sum_{j\in\mathcal{BN}(i)} 1 \tag{2.34}$$

$$\begin{aligned}\frac{\partial Eq2}{\partial \theta_i} &= \sum_{j\in\mathcal{IN}(i)} [-(x_i+x_j)\cos\theta_i - (y_i+y_j)\sin\theta_i] + \\ &+ \sum_{j\in\mathcal{BN}(i)} (-x_j\cos\theta_i - y_j\sin\theta_i)\end{aligned} \tag{2.35}$$

$$\begin{aligned}\frac{\partial Eq3}{\partial \Delta x_i} &= \sum_{j\in\mathcal{IN}(i)} [(y_i+y_j)\cos\theta_i - (x_i+x_j)\sin\theta_i] + \\ &+ \sum_{j\in\mathcal{BN}(i)} (y_j\cos\theta_i - x_j\sin\theta_i)\end{aligned} \tag{2.36}$$

$$\begin{aligned}\frac{\partial Eq3}{\partial \Delta y_i} &= \sum_{j\in\mathcal{IN}(i)} [-(y_i+y_j)\sin\theta_i - (x_i+x_j)\cos\theta_i] + \\ &+ \sum_{j\in\mathcal{BN}(i)} (-x_j\cos\theta_i - y_j\sin\theta_i)\end{aligned} \tag{2.37}$$

$$\begin{aligned}\frac{\partial Eq3}{\partial \theta_i} &= \sum_{j\in\mathcal{IN}(i)} \{(\Delta x_j - \Delta x_i)\,[(x_i+x_j)\cos\theta_i + (y_i+y_j)\sin\theta_i] \; + \\ &\quad + (\Delta y_j - \Delta y_i)\,[(y_i+y_j)\cos\theta_i - (x_i+x_j)\sin\theta_i] \; + \\ &\quad + (\sin\theta_i\sin\theta_j + \cos\theta_i\cos\theta_j)(x_i^2+y_i^2+x_j^2+y_j^2)\} \; + \\ &+ \sum_{j\in\mathcal{BN}(i)} [(\Delta x_i - x_j^*)\,(-x_j\cos\theta_i - y_j\sin\theta_i) \; + \\ &\quad + \; (\Delta y_i - y_j^*)\,(x_j\sin\theta_i - y_j\cos\theta_i)]\end{aligned} \tag{2.38}$$

The following Eqs. 2.39-2.47 result from differentiation with respect to the specific neighbor node j, hence no summations appear in these equations.

$$\frac{\partial Eq1}{\partial \Delta x_j} = -2 \tag{2.39}$$

$$\frac{\partial Eq1}{\partial \Delta y_j} = 0 \tag{2.40}$$

$$\frac{\partial Eq1}{\partial \theta_j} = (x_i + x_j)\sin\theta_j - (y_i + y_j)\cos\theta_j \tag{2.41}$$

$$\frac{\partial Eq2}{\partial \Delta x_j} = 0 \tag{2.42}$$

$$\frac{\partial Eq2}{\partial \Delta y_j} = -2 \tag{2.43}$$

$$\frac{\partial Eq2}{\partial \theta_j} = (x_i + x_j)\cos\theta_j + (y_i + y_j)\sin\theta_j \tag{2.44}$$

$$\frac{\partial Eq3}{\partial \Delta x_j} = (x_i + x_j)\sin\theta_i - (y_i + y_j)\cos\theta_i \tag{2.45}$$

$$\frac{\partial Eq3}{\partial \Delta y_j} = (x_i + x_j)\cos\theta_i + (y_i + y_j)\sin\theta_i \tag{2.46}$$

$$\frac{\partial Eq3}{\partial \theta_j} = (-\sin\theta_i \sin\theta_j - \cos\theta_i \cos\theta_j)(x_i^2 + y_i^2 + x_j^2 + y_j^2) \tag{2.47}$$

Finally, the system of equations to be solved comprises Eq. 2.27-2.29, while the derivatives with respect to the quantities for the central node are given in Eqs. 2.30-2.38 and the derivatives with respect to the quantities of each (internal) neighbor in Eqs. 2.39-2.47.

Algorithm 2 Coupled Method Algorithm

```
1: procedure DISPLACEMENT(nodes, edges, faces)
2:     for all InternalNodes do
3:         Compute InternalNode.derivatives
4:         Insert to GeneralEqsSystem
5:     end for
6:     Solve GeneralEqsSystem
7:     for all InternalNodes do
8:         Displace InternalNode
9:     end for
10: end procedure
```

2.4 System Linearization via Approximation

In this section, an approach aimed at replacing the non-linear system discussed earlier with a linear counterpart is introduced. This substitution serves the purpose of reducing the computational cost of the resolution. The precision and credibility of this model is also assessed. It must be noted that this strategy is irrelevant to the linearization of non-linear systems within the algorithms of numerical methods.

This strategy finds its foundation in the observation of small displacements occurring in the nodes of the grid within real-world applications. Considering, for instance, the iterative assessment of various node positions during an optimization procedure associated with the aerodynamic performance of a body, the displacements of these nodes are typically small in comparison to the overall size of the grid. Consequently, it is reasonable to make the assumption that the angle of rotation (around the z axis in 2D in the case of a grid spanning across the x-y plane), denoted as θ, tends towards zero.

$$\theta \to 0 \implies \begin{cases} \sin\theta \to \theta \\ \cos\theta \to 1 \end{cases} \tag{2.48}$$

The procedure outlined below specifically addresses the equations of the coupled method. Nevertheless, the presumption articulated here is equally relevant when considering the equations developed within the context of the decoupled method's approach.

Following the assumption of Eq. 2.48, Eq. 2.23 becomes:

$$
\begin{aligned}
F_{total}^{approx} &= \frac{1}{2}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{IN}(i)}[(y_j\theta_i+\Delta x_i)-(y_j\theta_j+\Delta x_j)]^2 + \\
&+\frac{1}{2}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{BN}(i)}[(x_j+y_j\theta_i+\Delta x_i)-(x_j^*)]^2 + \\
&+\frac{1}{2}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{IN}(i)}[(-x_j\theta_i+\Delta y_i)-(-x_j\theta_j+\Delta y_j)]^2 + \\
&+\frac{1}{2}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{BN}(i)}[(-x_j\theta_i+y_j+\Delta y_i)-(y_j^*)]^2
\end{aligned}
\tag{2.49}
$$

Similarly to the procedure followed earlier, Eqs. 2.27-2.29 become:

$$
\begin{aligned}
Eq1 \;=\; \frac{\partial F_{total}^{approx}}{\partial \Delta x_i} \;&=\; \sum_{j\in\mathcal{IN}(i)}[(y_i+y_j)(\theta_i-\theta_j)\; + 2(\Delta x_i-\Delta x_j)] + \\
&+\sum_{j\in\mathcal{BN}(i)}(x_j+y_j\theta_i+\Delta x_i-x_j^*) \;=\; 0
\end{aligned}
\tag{2.50}
$$

$$
\begin{aligned}
Eq2 \;=\; \frac{\partial F_{total}^{approx}}{\partial \Delta y_i} \;&=\; \sum_{j\in\mathcal{IN}(i)}[(x_i+x_j)(\theta_j-\theta_i)+2(\Delta y_i-\Delta y_j)] + \\
&+\sum_{j\in\mathcal{BN}(i)}(-x_j\theta_i+y_j+\Delta y_i-y_j^*) \;=\; 0
\end{aligned}
\tag{2.51}
$$

$$
\begin{aligned}
Eq3 \;=\; \frac{\partial F_{total}^{approx}}{\partial \theta_i} \;&=\; \sum_{j\in\mathcal{IN}(i)}[(\Delta x_i-\Delta x_j)(y_i+y_j) + \\
&\qquad +\; (\Delta y_j-\Delta y_i)(x_i+x_j) + \\
&\qquad +\; (\theta_i-\theta_j)(x_i^2+y_i^2+x_j^2+y_j^2)] + \\
&+\sum_{j\in\mathcal{BN}(i)}[(\Delta x_i-x_j^*)y_j - \\
&\qquad -\; (\Delta y_i-y_j^*)x_j + \\
&\qquad +\; (x_j^2+y_j^2)\theta_i] \;=\; 0
\end{aligned}
\tag{2.52}
$$

Eqs. 2.30-2.47 of the derivatives become:

$$
\frac{\partial Eq1}{\partial \Delta x_i} \;=\; \sum_{j\in\mathcal{IN}(i)} 2 + \sum_{j\in\mathcal{BN}(i)} 1 \tag{2.53}
$$

$$\frac{\partial Eq1}{\partial \Delta y_i} = 0 \tag{2.54}$$

$$\frac{\partial Eq1}{\partial \theta_i} = \sum_{j \in \mathcal{IN}(i)} (y_i + y_j) + \sum_{j \in \mathcal{BN}(i)} y_j \tag{2.55}$$

$$\frac{\partial Eq2}{\partial \Delta x_i} = 0 \tag{2.56}$$

$$\frac{\partial Eq2}{\partial \Delta y_i} = \sum_{j \in \mathcal{IN}(i)} 2 + \sum_{j \in \mathcal{BN}(i)} 1 \tag{2.57}$$

$$\frac{\partial Eq2}{\partial \theta_i} = \sum_{j \in \mathcal{IN}(i)} [-(x_i + x_j)] + \sum_{j \in \mathcal{BN}(i)} (-x_j) \tag{2.58}$$

$$\frac{\partial Eq3}{\partial \Delta x_i} = \sum_{j \in \mathcal{IN}(i)} (y_i + y_j) + \sum_{j \in \mathcal{BN}(i)} y_j \tag{2.59}$$

$$\frac{\partial Eq3}{\partial \Delta y_i} = \sum_{j \in \mathcal{IN}(i)} [-(x_i + x_j)] + \sum_{j \in \mathcal{BN}(i)} (-x_j) \tag{2.60}$$

$$\frac{\partial Eq3}{\partial \theta_i} = \sum_{j \in \mathcal{IN}(i)} (x_i^2 + y_i^2 + x_j^2 + y_j^2) + \sum_{j \in \mathcal{BN}(i)} (x_j^2 + y_j^2) \tag{2.61}$$

$$\frac{\partial Eq1}{\partial \Delta x_j} = -2 \tag{2.62}$$

$$\frac{\partial Eq1}{\partial \Delta y_j} = 0 \tag{2.63}$$

$$\frac{\partial Eq1}{\partial \theta_j} = -(y_i + y_j) \tag{2.64}$$

$$\frac{\partial Eq2}{\partial \Delta x_j} = 0 \tag{2.65}$$

$$\frac{\partial Eq2}{\partial \Delta y_j} = -2 \tag{2.66}$$

$$\frac{\partial Eq2}{\partial \theta_j} = (x_i + x_j) \tag{2.67}$$

$$\frac{\partial Eq3}{\partial \Delta x_j} = -(y_i + y_j) \tag{2.68}$$

$$\frac{\partial Eq3}{\partial \Delta y_j} = (x_i + x_j) \tag{2.69}$$

$$\frac{\partial Eq3}{\partial \theta_j} = -(x_i^2 + y_i^2 + x_j^2 + y_j^2) \tag{2.70}$$

2.5 Formulation of the Problem in 3D

In 3D, the equation that describes the displacement of a node from its initial coordinates (x, y, z) to its final ($x^{'}$, y' , z') is:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} \tag{2.71}$$

The rotation matrix R equals to the multiplication of the three individual matrices regarding rotation around each axis:

$$R = R_z R_y R_x \tag{2.72}$$

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix} \tag{2.73a}$$

$$R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{2.73b}$$

$$R_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 \\ \sin\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.73c}$$

In Eq. 2.71, $(\Delta x, \Delta y, \Delta z)$ denote the translation in the x, y and z axes respectively, while in Eq. 2.73, $(\theta_x, \theta_y, \theta_z)$ denote the angle of rotation around the x, y and z axes respectively, with the global origin $(0, 0, 0)$ as the center of rotation.

Eventually, R of Eq. 2.71, used to describe the rotation of the node, results in the following matrix, consisting of the terms:

$$R = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix} \tag{2.74a}$$

$$\begin{aligned} R_{11} &= \cos\theta_y \cos\theta_z \\ R_{12} &= -\cos\theta_x \sin\theta_z + \sin\theta_x \sin\theta_y \cos\theta_z \\ R_{13} &= \sin\theta_x \sin\theta_z + \cos\theta_x \sin\theta_y \cos\theta_z \\ R_{21} &= \cos\theta_y \sin\theta_z \\ R_{22} &= \cos\theta_x \cos\theta_z + \sin\theta_x \sin\theta_y \sin\theta_z \\ R_{23} &= -\sin\theta_x \cos\theta_z + \cos\theta_x \sin\theta_y \sin\theta_z \\ R_{31} &= -\sin\theta_y \\ R_{32} &= \sin\theta_x \cos\theta_y \\ R_{33} &= \cos\theta_x \cos\theta_y \end{aligned} \tag{2.74b}$$

Therefore, the new coordinates (x', y', z') of a node in 3D are:

$$\begin{aligned} x' = {} & x\cos\theta_y \cos\theta_z + \\ & + \; y(-\cos\theta_x \sin\theta_z + \sin\theta_x \sin\theta_y \cos\theta_z) + \\ & + \; z(\sin\theta_x \sin\theta_z + \cos\theta_x \sin\theta_y \cos\theta_z) + \\ & + \; \Delta x \end{aligned} \tag{2.75}$$

$$\begin{aligned} y^{'} &= x\cos\theta_y \sin\theta_z + \\ &+ y(\cos\theta_x \cos\theta_z + \sin\theta_x \sin\theta_y \sin\theta_z) + \\ &+ z(-\sin\theta_x \cos\theta_z + \cos\theta_x \sin\theta_y \sin\theta_z) + \\ &+ \Delta y \end{aligned} \tag{2.76}$$

$$\begin{aligned} z^{'} &= -x\sin\theta_y + \\ &+ y\sin\theta_x \cos\theta_y + \\ &+ z\cos\theta_x \cos\theta_y + \\ &+ \Delta z \end{aligned} \tag{2.77}$$

Based on the assumption that the grid undergoes displacement, aiming to closely imitate the motion of a rigid body, the *Objective Function (F)* that needs to be minimized for each *Internal* node i in a 3D grid is:

$$F_i^{3D} = \frac{1}{2} \sum_{j \in \mathcal{N}(i)} [(x_j^{ideal} - x_j^{new})^2 + (y_j^{ideal} - y_j^{new})^2 + (z_j^{ideal} - z_j^{new})^2] \tag{2.78}$$

In alignment with the previously mentioned distinction regarding 2D grids, when employing the coupled method, the objective function expands to:

$$F_{total}^{3D} = \sum_{i \in \mathcal{I}} F_i^{3D} \tag{2.79}$$

where $\mathcal{I}$ denotes the set of the Internal nodes (nodes of unknown displacement).

Building upon the methodology applied to formulate equations for scenarios involving 2D grids, the non-linear system's equations for a 3D problem are outlined as follows:

For the decoupled method:

$$\frac{\partial F_i^{3D}}{\partial \Delta x_i} = 0 \tag{2.80}$$

$$\frac{\partial F_i^{3D}}{\partial \Delta y_i} = 0 \tag{2.81}$$

$$\frac{\partial F_i^{3D}}{\partial \Delta z_i} = 0 \tag{2.82}$$

$$\frac{\partial F_i^{3D}}{\partial \theta_{x_i}} = 0 \tag{2.83}$$

$$\frac{\partial F_i^{3D}}{\partial \theta_{y_i}} = 0 \tag{2.84}$$

$$\frac{\partial F_i^{3D}}{\partial \theta_{z_i}} = 0 \tag{2.85}$$

For the coupled method:

$$\frac{\partial F_{total}^{3D}}{\partial \Delta x_i} = 0 \tag{2.86}$$

$$\frac{\partial F_{total}^{3D}}{\partial \Delta y_i} = 0 \tag{2.87}$$

$$\frac{\partial F_{total}^{3D}}{\partial \Delta z_i} = 0 \tag{2.88}$$

$$\frac{\partial F_{total}^{3D}}{\partial \theta_{x_i}} = 0 \tag{2.89}$$

$$\frac{\partial F_{total}^{3D}}{\partial \theta_{y_i}} = 0 \tag{2.90}$$

$$\frac{\partial F_{total}^{3D}}{\partial \theta_{z_i}} = 0 \tag{2.91}$$

The entries to the Jacobian matrix of the system consist of the derivatives of Eqs., either 2.80-2.85 for the decoupled method or 2.86-2.91 for the coupled method, with respect to the quantities Δx_i, Δy_i, Δz_i, θ_{x_i}, θ_{y_i} and θ_{z_i} for the central node and with respect to the quantities Δx_j, Δy_j, Δz_j, θ_{x_j}, θ_{y_j} and θ_{z_j}, $j \in \mathcal{IN}(i)$, for its neighbors.

2.5.1 Linearization via Approximation in 3D

Similar to the approach adopted for 2D cases, the linearization of the problem in 3D occurs by assuming that the displacements of the boundaries are small. Consequently, the angles of rotation θ_x, θ_y, and θ_z all tend towards zero. This assumption aligns with the notion that incremental changes in real-world displacements result in relatively minor angular variations, justifying the linear approximation for a resolution of reduced computational cost.

$$\theta_x \to 0 \implies \begin{cases} \sin\theta_x \to \theta_x \\ \cos\theta_x \to 1 \end{cases} \tag{2.92a}$$

$$\theta_y \to 0 \implies \begin{cases} \sin\theta_y \to \theta_y \\ \cos\theta_y \to 1 \end{cases} \tag{2.92b}$$

$$\theta_z \to 0 \implies \begin{cases} \sin\theta_z \to \theta_z \\ \cos\theta_z \to 1 \end{cases} \tag{2.92c}$$

The subsequent steps of the procedure closely mirror those presented for 2D, tailored to suit the demands of the 3D context.

2.6 Sub-Step Displacement Method

In most real-world applications, the displacements of grid boundary nodes are typically influenced by external tools, such as optimization algorithms seeking optimal geometries based on specified criteria. While these displacements are typically small relative to the grid size, this study also considers more extreme scenarios. To accommodate the use of the linearized mathematical formulations in such cases, the Sub-Step Displacement Method is implemented [31]. This method gradually displaces grid nodes from their initial to their final coordinates, ensuring that each step induces a small displacement relative to the overall grid size. While applicable in real-world scenarios, this approach increases the total computational cost, as it requires the resolution of a system of equations at each step.

It is important to note that the selection of the number and magnitude of steps depends heavily on the user. This can be pre-defined or computed based on specific criteria. In this thesis, unless otherwise specified, the number of steps is computed based on the total displacement of each node in relation to the size of its stencil. More precisely, the number of steps for each node is determined as the minimum number of times the displacement needs to be divided to ensure that the displacement of each step is smaller than the shortest distance to any neighbor. Ultimately, the number of steps applied is the maximum value computed among all boundary nodes. This strategy reduces the number of inverted cells in the final grid. The magnitude of displacement remains consistent for each step, tailored to the unique characteristics of the problem at hand.

Fig. 2.2 illustrates a fundamental example of the discussed method. Typically, a larger number of sub-steps is required to ensure the validity of the assumption that the angle of rotation is small, but the user should carefully weigh this requirement against the associated increase in computational cost. The determination of the

Figure 2.2: *Example of displacement of the NACA4415 Airfoil using the Sub-Step Displacement Method. The airfoil is rotated and the total displacement has been achieved in 3 sub-steps.*

optimal number of sub-steps depends on various factors, including the user's preferences for final grid quality versus computational expenditure, their prior experience and expertise, and the magnitude of the boundary displacement imposed.

2.7 Grid Quality Metrics

The quality of a grid is tightly connected to the precision of the outcome, as it significantly impacts the accuracy and efficiency of numerical simulations. A grid of high quality often leads to quicker solver convergence and precise simulation results. While increasing grid fineness typically enhances quality, it also substantially escalates computational costs. Thus, the goal is to strike an optimal balance between computational expense and the level of fineness. Conversely, a low-quality grid not only results in inaccurate simulations but can potentially cause solver errors, particularly due to instability introduced by poorly shaped elements. Ill-shaped elements can lead to ill-conditioned matrices, slowing down or even causing divergence in iterative solvers.

Given this, the use of grid quality metrics becomes indispensable in grid generation. A fundamental requirement for grid quality is the absence of inverted elements. Beyond this, it is desirable for elements to exhibit good shape and size, minimizing truncation errors. Ideally, defining good shape and size should be linked to solution error, indicating that overall grid quality is satisfactory when errors are below an acceptable threshold. In practice, the solution is often determined after the initial grid

generation. In the absence of comprehensive knowledge about the solution, a priori quality metrics are employed to control the quality of produced grids. This involves the assumption that the geometric properties of the grid somehow correspond to downstream simulation accuracy. Consequently, various geometrically-based quality metrics have been developed, considering aspects like size, orientation, shape, and skewness of grid elements [32], [33], [34].

It must be noted that the bibliography encompasses a variety of quality metrics, and no single metric comprehensively addresses all the aspects necessary for a cell to be deemed acceptable. Opinions on what constitutes an acceptable grid may vary. In this thesis, the metrics employed to assess the final grid in comparison to the initial one are detailed shortly thereafter [29], [35], [36], [37], [38], [39], [40], [41], [42], [43], [44].

2.7.1 Quality Metrics of a 2D Grid

Typically, 2D grids consist predominantly of triangles and/or quadrilaterals, with the possibility of incorporating various polygons featuring five or more edges. This thesis introduces specific metrics tailored for triangles and quadrilaterals, considering their prevalence, along with two metrics designed to evaluate any shape that may emerge in the grid.

The mean-ratio metric for triangles

The ideal shape for triangular grid elements is considered to be the equilateral triangle. The selected shape metric should remain consistent under various transformations like translation, reflection, rotation, and uniform scaling. The mean-ratio metric q, chosen for its manageable algebraic form, is employed to assess the quality of triangular elements.

To evaluate the quality (q_e) of a triangular element, defining the node orientation is crucial, opting for a counterclockwise direction to ensure a positive Jacobian determinant for acceptable triangles. Each triangular element comprises three nodes with coordinates (x_i, y_i), (x_j, y_j), and (x_k, y_k). The first step involves the computation of the Jacobian matrix for each vertex of the triangular element,

$$A_i = \begin{bmatrix} x_j - x_i & x_k - x_i \\ y_j - y_i & y_k - y_i \end{bmatrix} \tag{2.93a}$$

$$A_j = \begin{bmatrix} x_k - x_j & x_i - x_j \\ y_k - y_j & y_i - y_j \end{bmatrix} \tag{2.93b}$$

$$A_k = \begin{bmatrix} x_i - x_k & x_j - x_k \\ y_i - y_k & y_j - y_k \end{bmatrix} \tag{2.93c}$$

It should be noted that the Jacobian matrix is not independent of the node for which it is computed. However, the determinant of matrix A_n, $n = i, j, k$, equals to two times the area of the triangular element and remains the same regardless of which of the three Jacobian matrices is used [29].

The Jacobian matrix W for the ideal element, an equilateral triangle with an edge size of 1 (metric unit), is given by:

$$W = \begin{bmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{bmatrix} \tag{2.94}$$

Another relevant matrix, denoted as S_n, is introduced:

$$S_n = A_n W^{-1} = A_n \frac{2\sqrt{3}}{3} \begin{bmatrix} \sqrt{3}/2 & -1/2 \\ 0 & 1 \end{bmatrix} \tag{2.95}$$

The matrix S_n becomes the identity matrix when the random and ideal triangular elements have the same shape and size. If the random and ideal triangles have the same shape but different sizes, then S_n equals to a positive multiple of the identity matrix, indicating the scale of the random element.

The shape metric of node n is defined as:

$$\mu_n = \frac{\det(S_n)}{2\|S_n\|_F^2} \tag{2.96}$$

In Eq. 2.96, the determinant of matrix S_n, denoted as $\det(S_n)$, serves as an indicator in the assessment of triangular quality. A negative value implies an inverted triangle, while a value of zero indicates a degenerate triangle.

The term $\|S_n\|_F$ in Eq. 2.96 represents the *Frobenius norm*. It corresponds to the square root of the trace of the product of the transpose matrix of S_n and S_n, denoted as $\sqrt{tr(S_n^T S_n)}$, where trace (tr) signifies the sum of the diagonal elements. It is important to note that this term is often disregarded, resulting in a dimensionless final outcome.

Thus, the quality q_e of a triangular element is defined as twice the average of the shape metric of the nodes comprising the element, as shown in Eq. 2.97.

$$q_e = 2\bar{\mu}, \quad \bar{\mu} = \frac{1}{3}\sum^{n} \mu_n, \quad n = i, j, k \tag{2.97}$$

The rationale behind doubling the average of the shape metric in the triangular element's quality computation is to simplify the denominator in Eq. 2.96. This

adjustment ensures that when an element is identical to the ideal one, its quality is normalized to 1.

By following the outlined procedure, the quality of an individual triangular element within the grid is determined. However, to evaluate the overall quality of the grid, it is essential to compute the quality for every triangular element that constitutes it. The mean (Eq. 2.98a) and standard deviation (Eq. 2.98b) of the individual qualities serve to characterize the displaced grid's quality. Additionally, knowledge of the minimum (Eq. 2.98c) and maximum (Eq. 2.98d) values encountered in the grid proves beneficial.

$$\bar{q} = \frac{\sum_{t=1}^{N}(q_e)_t}{N} \quad , N \text{ equals the number of triangles} \tag{2.98a}$$

$$\sigma(q) = \sqrt{\frac{1}{N}\sum_{t=1}^{N}[\,(q_e)_t - \bar{q}\,]^2} \tag{2.98b}$$

$$\mathrm{Min}(q) = \min_{t=1}^{N}[(q_e)_t] \tag{2.98c}$$

$$\mathrm{Max}(q) = \max_{t=1}^{N}[(q_e)_t] \tag{2.98d}$$

The shape metric for quadrilaterals

For a grid comprising quadrilaterals, a suitable metric, denoted as F_{shape}, is introduced to assess the quality of the quadrilateral's shape. Here, the ideal element is defined as a square with an edge size of 1 (metric unit).

Considering the quadrilateral element Q_{ijkm} with its nodes arranged in a counterclockwise order, the matrices of Eq. 2.99 are constructed:

$$A_i = \begin{bmatrix} x_j - x_i & x_m - x_i \\ y_j - y_i & y_m - y_i \end{bmatrix} \tag{2.99a}$$

$$A_j = \begin{bmatrix} x_k - x_j & x_i - x_j \\ y_k - y_j & y_i - y_j \end{bmatrix} \tag{2.99b}$$

$$A_k = \begin{bmatrix} x_m - x_k & x_j - x_k \\ y_m - y_k & y_j - y_k \end{bmatrix} \tag{2.99c}$$

$$A_m = \begin{bmatrix} x_i - x_m & x_k - x_m \\ y_i - y_m & y_k - y_m \end{bmatrix} \tag{2.99d}$$

Subsequently, the matrices $S_n = A_n^T A_n$ are assembled:

$$S_i = \begin{bmatrix} S_{jj} & S_{jm} \\ S_{mj} & S_{mm} \end{bmatrix} \tag{2.100a}$$

$$S_j = \begin{bmatrix} S_{kk} & S_{ki} \\ S_{ik} & S_{ii} \end{bmatrix} \tag{2.100b}$$

$$S_k = \begin{bmatrix} S_{mm} & S_{mj} \\ S_{jm} & S_{jj} \end{bmatrix} \tag{2.100c}$$

$$S_m = \begin{bmatrix} S_{ii} & S_{ik} \\ S_{ki} & S_{kk} \end{bmatrix} \tag{2.100d}$$

In Eq. 2.100a, the term S_{jj} represents the squared distance between node i and node j, S_{mm} represents the squared distance between node i and node m, while S_{jm} and S_{mj} denote the product of the distance from node i to node j and the distance from node i to node m, multiplied by the cosine of the angle between them. Similarly, the same interpretation applies to the other matrices constructed, where the relevant terms represent squared distances and products of distances involving the nodes of the quadrilateral element Q_{ijkm}.

Eventually, the formula to compute the corresponding metric is given by:

$$F_{shape} = \frac{8}{\sum^{n}[tr(S_n)/\det(A_n)]} \quad , n = i, j, k, m \tag{2.101}$$

As explained earlier for triangular elements, the metric F_{shape} takes the value of 1 if and only if the quadrilateral is a square with an edge size of 1. A negative value indicates that the quadrilateral is inverted and, consequently, unacceptable. A value of zero suggests that the quadrilateral is degenerate.

By following this process, the metric is computed for each quadrilateral. To assess the grid comprehensively, it is necessary to determine the mean, standard deviation, minimum, and maximum values, as discussed earlier.

The non-orthogonality metric

The non-orthogonality ($n - o$) metric is defined as the angle (measured in degrees) between the line connecting the centers of two neighboring cells and the perpendicular to the common edge of the cells (Fig. 2.3). The ideal value is 0 degrees.

The skewness metric

The skewness (s) metric is defined as the distance between the intersection, of the line connecting the centers of two neighboring cells and their common edge, and the

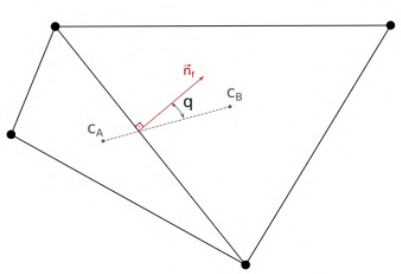


Figure 2.3: *The non-orthogonality of the side between two neighboring cells is defined as the angle (in degrees) between the line C_AC_B connecting the centers of the cells and the perpendicular vector to the common edge of the cells $\overrightarrow{n_f}$. From [35].*

center of their common edge, normalized by the distance of the centers of the two cells (Fig. 2.4). The ideal value is 0 (metric units).

2.7.2 Quality Metrics of a 3D Grid

For 3D grids, the metrics of non-orthogonality and skewness introduced in the previous section, which are independent of the type of elements, are employed. The definitions of these metrics extend to the 3D space.

The non-orthogonality metric

The non-orthogonality $(n - o)$ metric is defined as the angle (measured in degrees) between the line connecting the centers of two neighboring cells and the perpendicular to the common face of the cells (Fig. 2.5). The ideal value is 0 degrees.

The skewness metric

The skewness (s) metric is defined as the distance between the intersection, of the line connecting the centers of two neighboring cells and their common face, and the center of their common face, normalized by the distance of the centers of the two cells (Fig. 2.6). The ideal value is 0 (metric units).

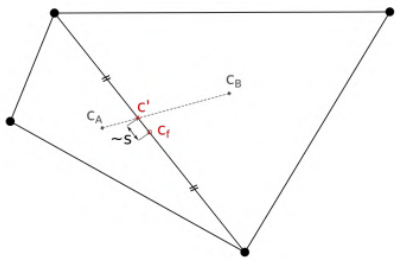


Figure 2.4: *The skewness of the side between two neighboring cells is defined as the distance between the intersection c', of the line C_AC_B connecting the centers of the cells and their common edge, and the center c_f of their common edge, normalized by the distance C_AC_B. From [35].*

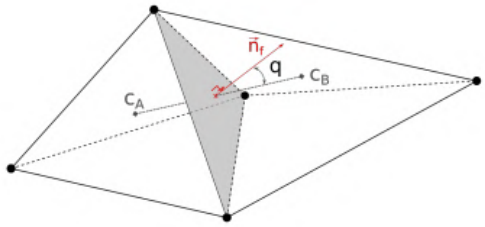


Figure 2.5: *The non-orthogonality of the surface between two neighboring cells is defined as the angle (in degrees) between the line C_AC_B connecting the centers of the cells and the perpendicular vector to the common face of the cells $\overrightarrow{n_f}$. From [35].*

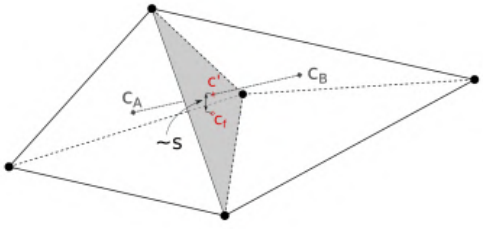


Figure 2.6: *The skewness of the surface between two neighboring cells is defined as the distance between the intersection c', of the line C_AC_B connecting the centers of the cells and their common face, and the center c_f of their common face, normalized by the distance C_AC_B. From [35].*

2.8 The RBM Algorithmic Framework

Implementing the aforementioned method involves a step-by-step algorithm for the application of the *RBM* technique, specifically employed to displace the internal nodes of the grid. In brief, the key steps of this algorithm are as follows:

Step 1 : User Input Files

The user is required to provide the input files containing essential information about the grid. These files should include, at a minimum:

- The Number of Nodes in the grid.
- The Initial Coordinates of all the nodes, specified as (x, y) for 2D or (x, y, z) for 3D.
- Node Categorization data (a Flag) designating nodes into one of the three categories: those remaining at fixed positions throughout the procedure, those with displacements determined by external conditions (known beforehand), and those with unknown, and therefore sought, displacements.
- The Number of Elements in the grid (specified as faces in 2D or cells in 3D), along with their specific type (e.g., triangle, quadrilateral, etc., in 2D or tetrahedron, pyramid, prism, hexahedron in 3D).
- Node sequences comprising each element, specified in the appropriate order.
- The New Coordinates for nodes with known displacements (second category mentioned above).

This structured approach ensures that the necessary information is provided for a successful execution of the algorithm.

Step 2 : User-defined Settings and Grid Parameters

The user must also input various settings prior to running the code. These encompass both the main characteristics of the grid and the fundamental parameters dictating the execution process. The user should:

- Specify the number of dimensions of the grid, either 2D or 3D.
- Define the method to be implemented in solving the problem, choosing between the decoupled and coupled methods.
- Define the type of equations to be employed in solving the problem, choosing between non-linear or linearized formulations as detailed earlier.
- Specify the format of the input files, which can be in the form of .ele/.nod files, .hyb/.nod/.patch files, or the 'default' format used for algorithm execution.
- Choose whether to implement the sub-step displacement method.
- Set the convergence criteria for the numerical method's solution process.

This structured approach ensures that the user provides essential information for configuring the algorithm according to specific requirements.

Step 3 : Pre-Processing of the Input Files

The user-provided input files undergo processing, resulting in the generation of three .dat files. These files encapsulate all the vital information outlined in `Step 1`, formatted in the 'default' format utilized by the subsequent stages of the code.

Step 4 : Data Structure Construction

This step involves reading the information from the three .dat files, determining the topology and connectivity of nodes, and storing any additional relevant data for future use. The culmination of this process is the construction of a robust data structure, forming the foundational basis upon which subsequent steps will depend.

Step 5 : Solution Process

In this stage, the equations governing the displacement of nodes, based on the known displacements of certain nodes, are formulated. Subsequently, a system of equations, either linear or non-linear, is assembled and solved, employing a numerical method aligned with the convergence criteria specified in `Step 2`. In case sub-steps for the entire required displacement are anticipated, this procedure iterates for the displacements of each sub-step.

Step 6 : Post-Processing

Upon achieving resolution, the node positions are updated to their final ones. Following this, quality metrics for both the initial and final grid are computed for comparison and procedural evaluation. Finally, output files are generated, facilitating the assessment and visualization of the final grid.

Note : It is noteworthy that the implemented algorithm remains consistent across both 2D and 3D cases, with appropriate adjustments made to accommodate variations in cell information and equations.

Chapter 3

2D Grid Displacement: Applications

Having presented the theoretical background of the proposed method, a series of tests follow for its assessment. This necessitates the implementation of the RBM technique, with a primary focus on assessing the efficacy of the Coupled Method. The Sub-Step Displacement Method is also tried where enhancement is deemed necessary. Occasional comparisons with the Decoupled Method are made.

All codes, including both the decoupled and the coupled method, along with their non-linear and linearized mathematical formulations, have been developed from scratch. The initial geometric configuration of the grids and aerodynamic body cases analyzed herein are provided by PCOpt/NTUA. Some of the grids, initially designed in a pseudo-3D format, as frequently employed in OpenFOAM [45] computations where the use of 2D grids is impractical, were converted by a developed supplementary code into 2D grids. All code executions were conducted on a single processor on a personal computer.

A main characteristic of these codes is their connectivity-agnostic nature, meaning that the type, fineness and structure of the grid, as well as the type of elements constituting it, do not influence the solution procedure itself. For example, the fact that in unstructured grids the number of neighbors many vary between nodes is insignificant. In 2D, any type of element can be handled, while in 3D, tetrahedra, pyramids, prisms and hexaedra are accommodated, as commonly encountered in typical applications.

In practical optimization scenarios, it is expected that the initial geometry of the shape undergoing optimization undergoes small changes. The deformations illustrated in the examined cases are intentionally designed to stress-test the methods

and ascertain their limits. Notably, instances featuring reversed cells are included for completeness. Despite their extreme features, these cases provide valuable insights into the method's robustness.

3.1 Two Concentric Squares

Initiating the methodological testing with simpler examples offers a preliminary insight into the behavior of the proposed method. It serves as a valuable initial assessment, revealing the efficacy of the method and providing an anticipation of the outcomes to be expected in more complex applications within the field of aerodynamics.

The first application of the RBM method is demonstrated through the deformation analysis of two squares. The initial grid comprises two squares with a shared center: an outer square defining the computational domain and a smaller inner square. The outer square spans 101 nodes on each side, with all grid elements forming squares, each with an edge size of 10 metric units. Consequently, the total edge size of the outer square measures 1000 metric units, while of the inner square measures 200 metric units, constituting one-fifth of the outer square. The grid encompasses a total of 9840 nodes, forming 9600 quadrilaterals.

In the broader context of the envisioned applications for the RBM tool, the positions of boundary nodes, which predominantly define the geometry of the aerodynamic body and the outer boundary, are typically determined by an external tool, such as an optimization algorithm. In these cases, boundary nodes undergo displacement according to algorithm- or user-defined functions. Subsequently, the RBM method is implemented to displace the internal nodes to their final positions, utilizing the methodology and equations outlined in the preceding chapter. Recall that the connectivity of nodes remains identical in the initial and final grids.

3.1.1 Rotating Squares

The initial application under consideration involves the implementation of the RBM technique on a 2D structured grid composed exclusively of quadrilaterals, as mentioned above, particularly squares. This ideal design ensures the initial grid's high quality based on metrics utilized for comparison and evaluation.

In this scenario, the deformation is externally induced by rotating the inner square by a 10-degree counter-clockwise angle around its center. This rotation is enforced by adjusting the coordinates of each boundary node forming the inner square. Simultaneously, the nodes constituting the outer square remain fixed, granting freedom exclusively to the internal nodes between the two squares. These internal nodes are the ones on which the RBM technique is applied, resulting in their displacement.

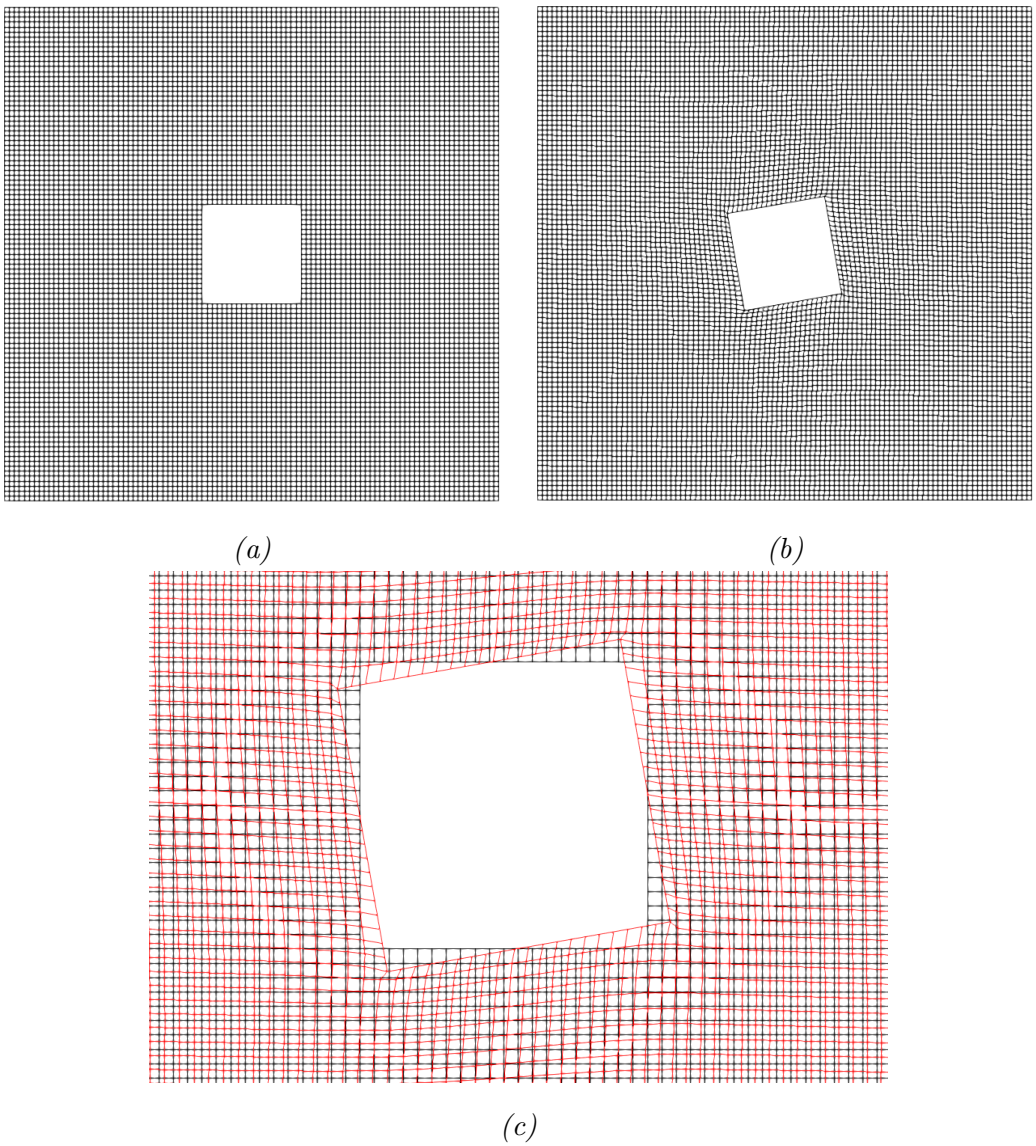
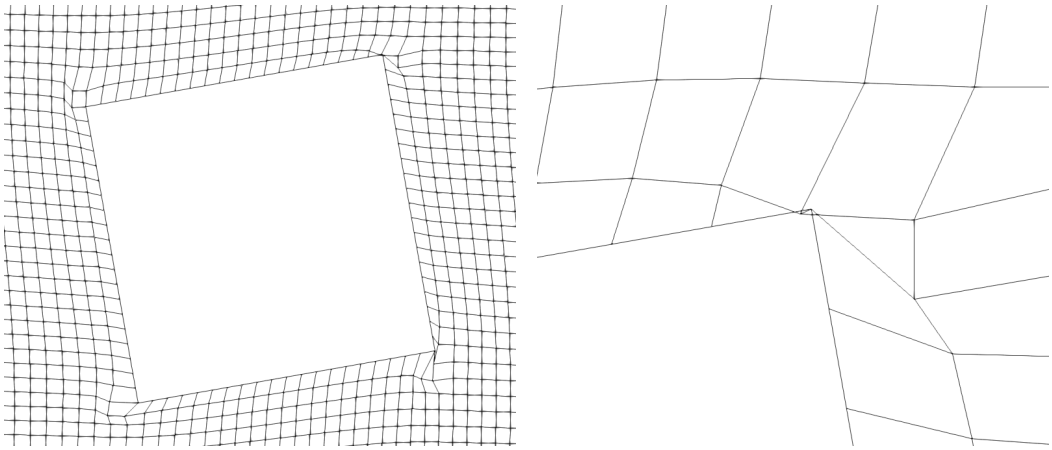


Figure 3.1: *Rotating Squares. The inner square undergoes a 10-degree rotation around its center. (a): Initial grid. (b): Final grid. (c): Combined presentation of the initial and final grids focused on the inner square; black color stands for the initial and red for the final grid. Result of the non-linear coupled method.*

Fig. 3.1 visually presents the initial and final grid configurations, demonstrating the impact of the non-linear coupled method. Notably, the displacement is more pronounced near the boundary nodes of the inner square and less prominent towards the outer side. Importantly, this rotation of the inner square by 10 degrees does not induce the appearance of reversed cells in the final grid.

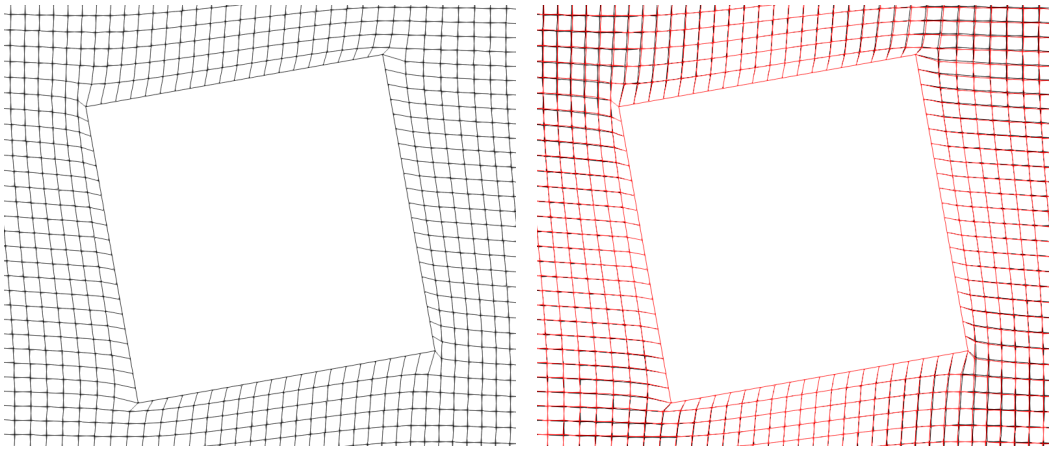
Fig. 3.2 depicts the outcome of the implementation of the linearized equations of the coupled method. Inverted cells occur in the top-right corner of the inner square, rendering the final grid unsuitable for CFD simulations. To address this issue, the



(a) *(b)*

Figure 3.2: *Rotating Squares. (a): Result from the implementation of the linearized equations of the coupled method. (b): Focus on the top-right corner.*

sub-step displacement method is employed. More specifically, the total imposed displacement is sub-divided, so that the inner square is rotated by 1 degree at each step. The result, after completing all 10 sub-steps, is displayed in Fig. 3.3. It is evident that applying this enhancement leads to no inverted cells occurring in the final grid. Additionally, the grid is compared with the result obtained from the implementation of the non-linear equations, revealing that the linearized formulations provide a final grid remarkably close to the one derived from the non-linear coupled method.



(a) *(b)*

Figure 3.3: *Rotating Squares. (a): Result from the implementation of the sub-step displacement method on the linearized equations of the coupled method. (b): Comparison between this resulting grid (black color) and the final grid obtained from the implementation of the non-linear equations of the coupled method (red color).*

Metric		Initial	Final		
			Decoupled	Coupled	
				Non-linear	Linearized
F_{shape}	mean	1	0.997002	0.997001	0.996634
	std	0	0.009780	0.011795	0.018526
	min	1	0.805632	0.684565	0.674755
	max	1	1	1	1
$N-O$	mean	0	4.5892	1.9313	1.9083
	std	0	0.0024	2.5939	2.8019
	min	0	0.0001	0.0053	0.0047
	max	0	18.3247	24.6579	40.1184
$Skewness$	mean	0	0.000431	0.000548	0.002319
	std	0	0.001621	0.002369	0.004300
	min	0	0	0	0.000001
	max	0	0.048137	0.070856	0.075720

Table 3.1: *Rotating Squares: Quality Metrics*

Further insight is provided in Table 3.1, indicating similar metrics for the coupled and the decoupled method, both maintaining values close to those of the initial grid — a desirable outcome. It should be mentioned that the non-linear and linearized formulations under the decoupled method yield identical results.

Analysis of the results from this case leads to the conclusion that, while the implementation of the non-linear coupled method produces a final grid of sufficient quality, utilizing the linearized equations is not straightforward. Even a relatively modest rotation of 10 degrees violates the assumption of small angle rotation which the linearization strategy is based upon. The rotation should be executed in smaller increments in order to maintain the validity of this assumption, ensuring the applicability of the linearized equations. Eventually, given the absence of inverted cells following the assisting employment of the sub-step displacement method, the application of the linearized equations is deemed effective, though more costly, producing satisfactory results.

3.1.2 Translated Squares

In this scenario, the inner square undergoes pure translation to a new position, devoid of any rotational transformation. Each boundary node forming the inner square experiences a displacement of 50 metric units along the x-axis and 25 metric units along the y-axis. As with the previous case, the nodes along the outer boundary maintain their original positions. The aim of this case is to examine the limitations of the linearized coupled method.

In Fig. 3.4, the initial and final grids are depicted, providing a closer inspection of

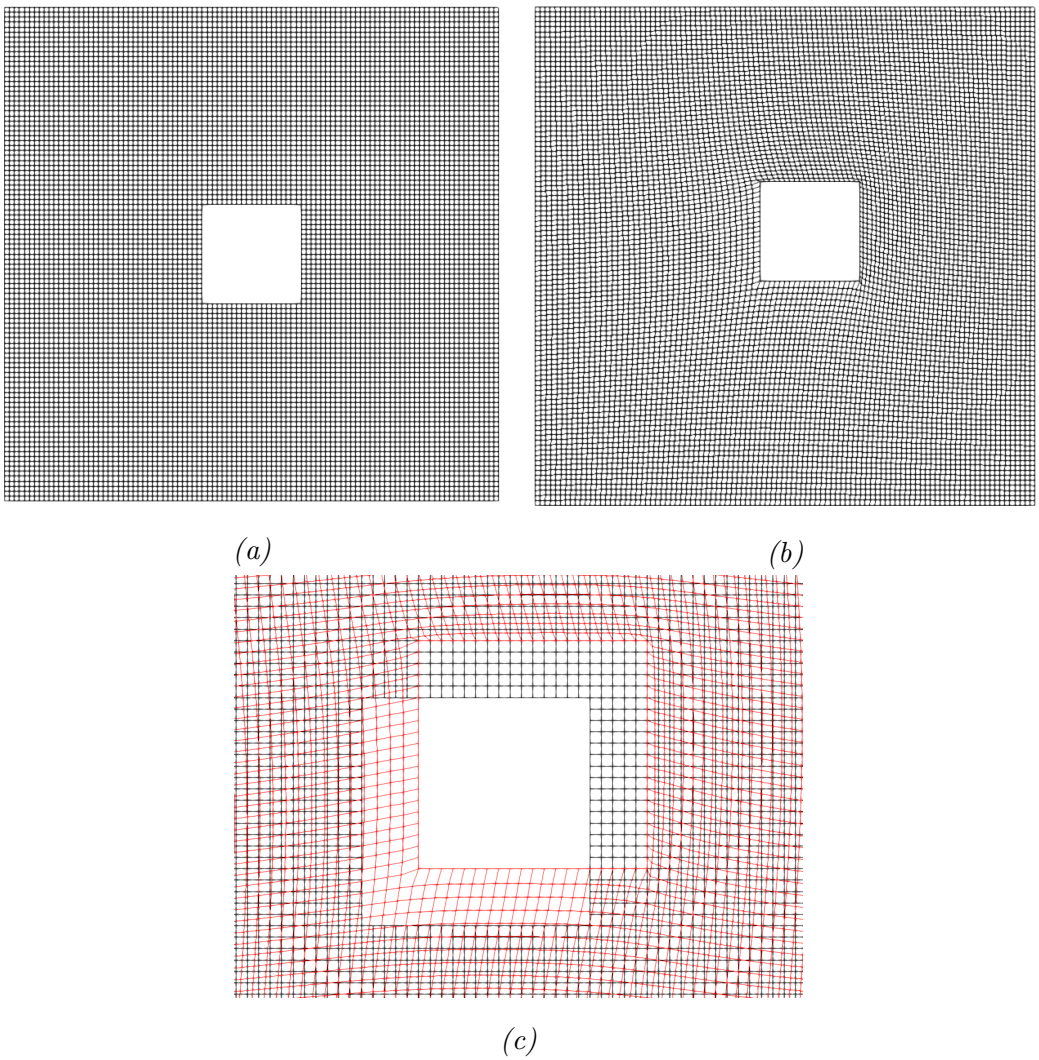


Figure 3.4: *Translated Squares. The inner square experiences a displacement of 50 metric units along the x-axis and 25 metric units along the y-axis. (a): Initial grid. (b): Final grid. (c): Combined presentation of the initial and final grids focused on the inner square; black color stands for the initial and red for the final grid. Result of the non-linear coupled method.*

displacement magnitudes. This particular result corresponds to the implementation of the non-linear coupled method. Notably, the cells of the grid adjacent to the right and top sides of the inner square (aligned with the direction of its movement) have been narrowed, while those near the left and bottom sides have been elongated. Importantly, this displacement of the inner square does not induce the appearance of reversed cells in the final grid.

Approaching this scenario with the implementation of the linearized equations of the coupled method leads to results unsuitable for use, e.g., from a CFD solver code, as shown in Fig. 3.5. The shortcomings observed highlight the imperative

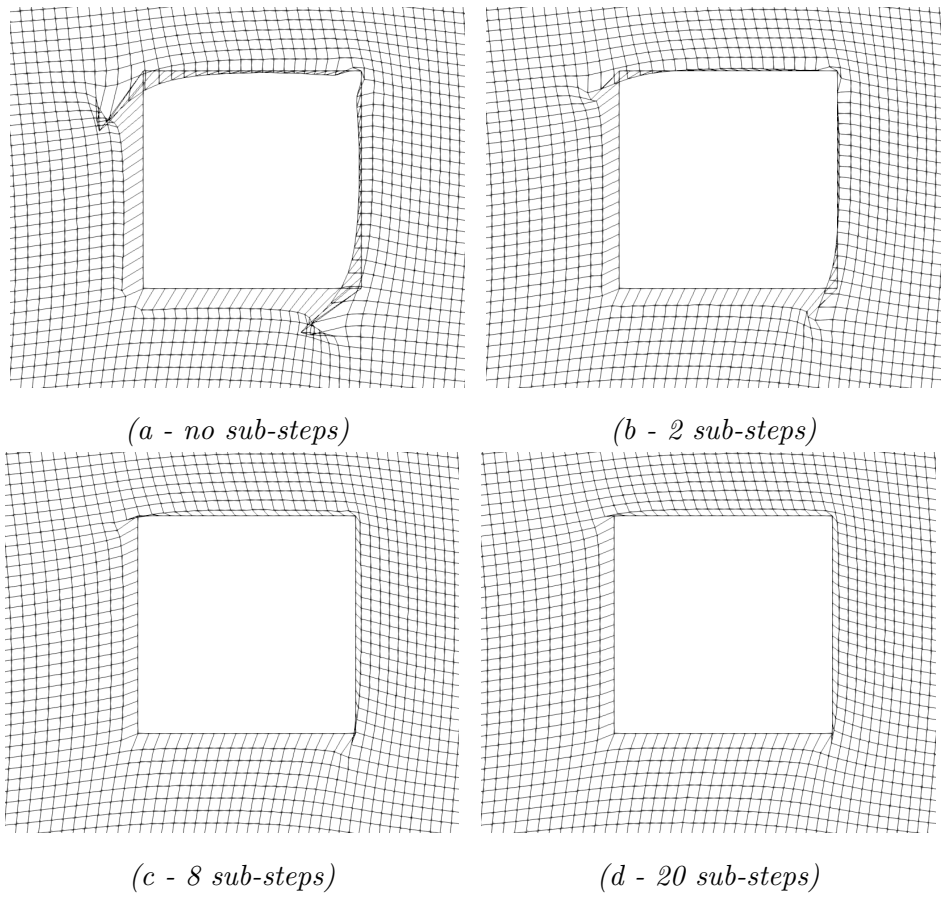


Figure 3.5: *Translated Squares. Implementation of the linearized coupled method, assisted by the sub-step displacement method. The figures represent the resulting grid according to the number of steps taken.*

need for employing the sub-step displacement method. The results of various trials are also depicted in Fig. 3.5. With the utilization of 20 sub-steps, the grid quality improves and reversed cells disappear. However, it is essential to acknowledge that this enhancement comes at the expense of greater computational cost, measured in time units required to obtain the result. The increase in computational time is nearly proportional to the number of steps taken. Consequently, the decision-making process should carefully weigh the desired grid's quality against the computational cost, in the general case of implementing the sub-step displacement method.

Comparatively, as shown in Fig. 3.6, where the outcomes of the decoupled and coupled methods are illustrated, minimal discrepancies are detected between the resulting grids. The coupled method appears to displace nodes emphasizing on the interconnections of all grid nodes. On the other hand, the decoupled method seems to focus more locally, primarily adjusting nodes near the boundary while leaving further nodes relatively unchanged. This observation aligns with the nature of the methodology followed in each method, specifically with the different formulations of

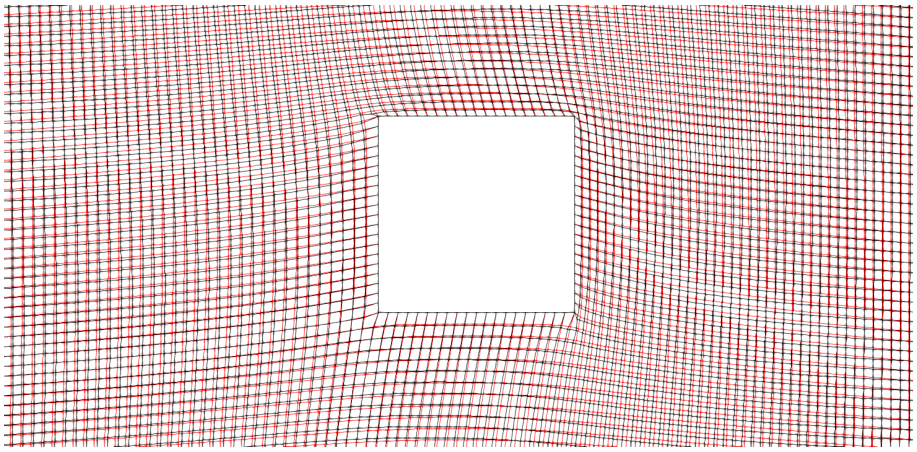


Figure 3.6: *Translated Squares. Results from the implementation of the coupled (black color) and decoupled (red color) methods.*

Metric		**Initial**	**Final**		
			Decoupled	**Coupled**	
				Non-linear	**Linearized**
F_{shape}	mean	1	0.982138	0.988662	0.988998
	std	0	0.057681	0.037137	0.037967
	min	1	0.466518	0.684565	0.777164
	max	1	1	1	1
$N-O$	mean	0	3.5523	4.4511	4.1373
	std	0	2.6969	2.5939	3.0386
	min	0	0	0.0082	0.0052
	max	0	65.1358	64.1750	64.4376
$Skewness$	mean	0	0.001846	0.000971	0.000845
	std	0	0	0	0.001563
	min	0	0	0	0.000004
	max	0	1.578900	1.382751	1.786970

Table 3.2: *Translated Squares: Quality Metrics*

the objective function. It is important to note that both methods produce final grids of comparable quality to the initial, making them viable for practical application.

The metrics listed in Table 3.2 also support the claim that the resulting grids respect the quality level of the initial one. As in the previous case, the decoupled method, whether utilizing the non-linear or linearized equations, yields identical outcomes.

It becomes apparent that even in cases where the imposed boundary displacement lacks rotation, the linearized equations of the coupled method cannot be utilized without consideration. Specifically, since the system of equations considers the angle of rotation for each node as a variable, the solution process may yield non-zero

rotation values. This can lead to an unsuitable final grid. To address this issue, the sub-step displacement method must be employed, segmenting the total boundary displacement into incremental steps.

3.1.3 Free Outer Boundary

As observed in the previous case of the Translated Squares, the resulting grids obtained from the implementation of the coupled and the decoupled methods may differ. In order to further investigate the possibly different outcomes between the coupled and the decoupled method, another case is set-up and studied. This case represents a different scenario, not seen in optimization loops.

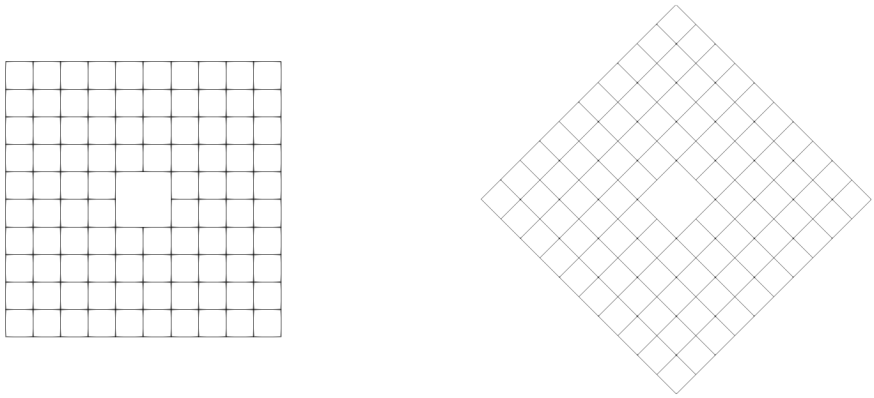
In this scenario, the outer square spans 11 nodes on each side, with all grid elements forming squares, each with an edge size of 10 metric units. Consequently, the total edge size of the outer square measures 100 metric units, while of the inner square measures 20 metric units, constituting one-fifth of the outer square. The grid encompasses a total of 120 nodes, forming 96 quadrilaterals.

The deformation is externally induced by rotating the inner square by a 45-degree counter-clockwise angle around its center. As previously, this rotation is enforced by adjusting the coordinates of each boundary node forming the inner square. The distinguishing aspect of this example is that the nodes forming the outer square are free to be displaced, too.

Typically, the boundaries defining the computational domain are specified at the initial grid generation and do not change throughout computational stages, such as optimization cycles in an optimization procedure or time-steps in an aeroelasticity problem. The outer nodes of the grid are therefore constrained from moving, as they define critical boundaries such as inlets, outlets, or periodicity conditions. This holds particularly true for scenarios involving external aerodynamic analyses.

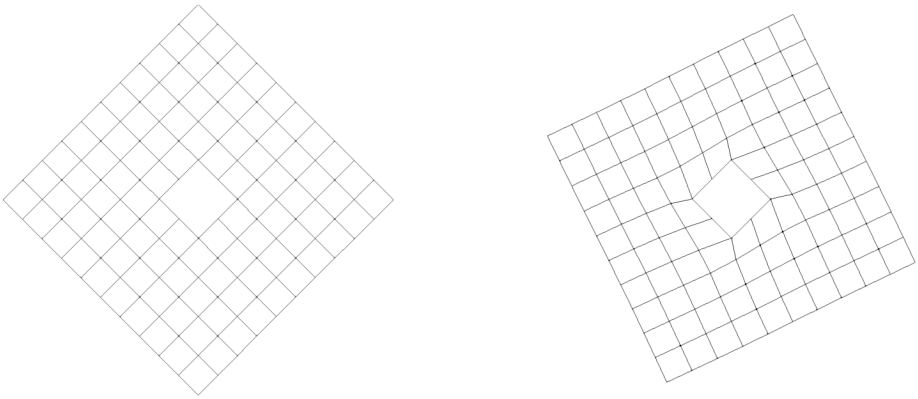
However, in this case, the RBM technique is applied on all nodes, except for the ones forming the inner square. The initial and the ideal resulting grid are illustrated in Fig. 3.7.

In Fig. 3.8, the resulting grids obtained from the implementation of the coupled and the decoupled methods are depicted. It is obvious that each method has produced a different final grid. The coupled method appears to grasp the context of interdependence among all grid nodes, perceiving the grid as an entity. In contrast, the decoupled method seems to adopt a more localized approach, predominantly adjusting nodes near the boundary while leaving more distant nodes relatively unaffected, hence confining the disturbance provoked by the external displacement of the boundary nodes. This divergence reflects the distinct methodologies inherent in each approach, particularly their differing formulations of the objective function.



(a) *(b)*

Figure 3.7: *Rotating Squares with Free Outer Boundary. The inner square undergoes a 45-degree rotation around its center. The nodes of the outer square are free to be displaced. (a): Initial grid. (b): Ideal final grid.*



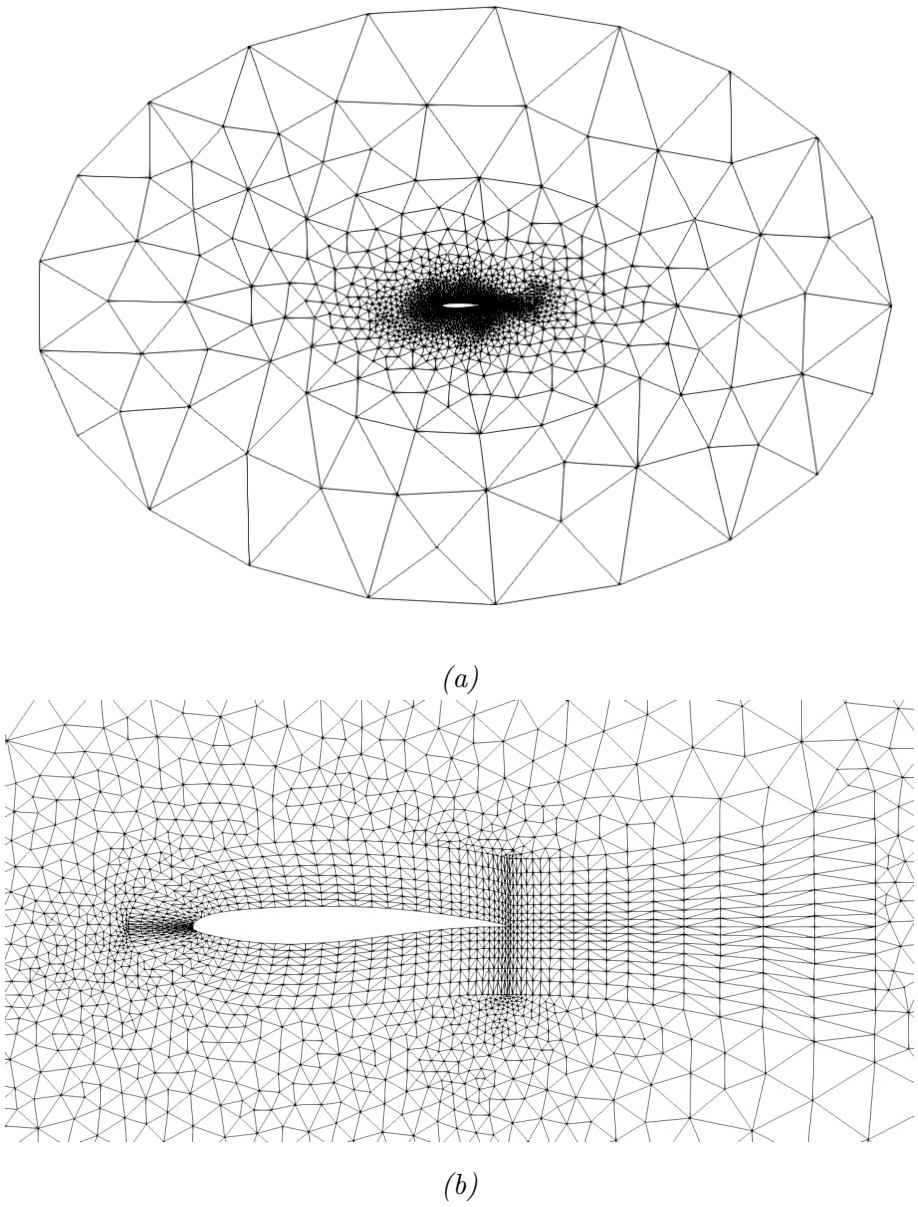
(a) *(b)*

Figure 3.8: *Rotating Squares with Free Outer Boundary. The inner square undergoes a 45-degree rotation around its center. The nodes of the outer square are free to be displaced. Results from the implementation of the coupled (a) and decoupled (b) methods.*

3.2 Isolated Airfoil

The next application lies closer to the realm of aerodynamics, focusing on the case of an isolated airfoil. In Fig. 3.9, the initial grid is presented, providing an overview of the computational domain, along with a zoomed-in view for a more detailed examination of the region around the airfoil.

This 2D grid is hybrid with a structured and denser arrangement near the aerodynamic body's wall. This strategic densification aims to enhance simulation preci-



(a)

(b)

Figure 3.9: *(a): 2D hybrid grid around an Isolated Airfoil. (b): Focus on the Airfoil.*

sion, particularly for critical quantities. In contrast, the grid becomes unstructured and coarser towards the outer boundary, known as the farfield. The overall grid comprises 2242 nodes and 4383 triangular elements. The airfoil's chord length is standardized to 1 metric unit, aligning with a common practice for dimensionless study of aerodynamic quantities.

The mathematical formula for the displacement of a given point around a center of rotation (x_c, y_c) is generally expressed by:

$$x^{new} = (x^{old} - x_c)\cos\phi + (y^{old} - y_c)\sin\phi + \Delta x + x_c \tag{3.1a}$$
$$y^{new} = -(x^{old} - x_c)\sin\phi + (y^{old} - y_c)\cos\phi + \Delta y + y_c \tag{3.1b}$$

In practical scenarios, the parameter ϕ typically assumes values between 3 and 5 degrees. For this particular application, two distinct rotation angles have been examined: 5 and 10 degrees, executed around two distinct centers of rotation. The rotation is applied exclusively to the boundary nodes forming the airfoil, simulating an optimization procedure. Concurrently, the nodes in the farfield are deliberately kept fixed at their initial positions.

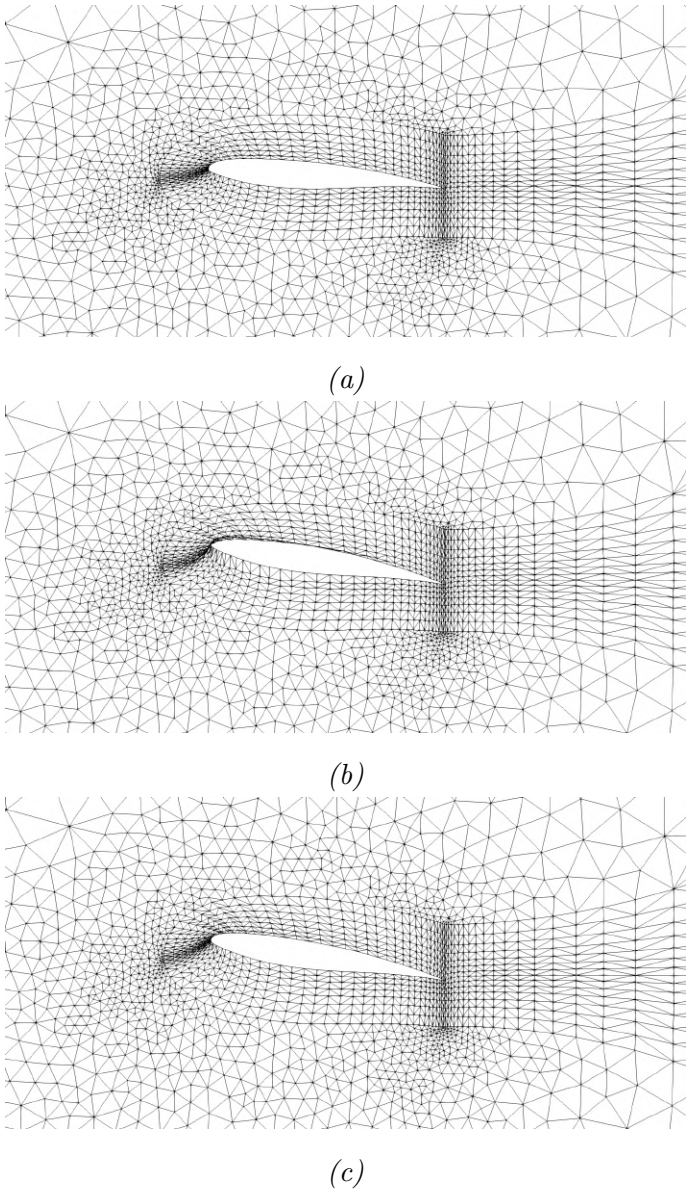
Rotation around the trailing edge

As initial test of this application, a clockwise rotation around the z-axis (the grid spans across the x-y plane) is applied at angles of 5 and 10 degrees with the trailing edge as the chosen center of rotation.

The resulting displaced grids showcased in Fig. 3.10 stem from the implementation of the linearized equations of the coupled method of the RBM technique. It is observed that when a 5-degree rotation is applied, the grid displacement occurs smoothly, without any instances of reversed cells. However, when a larger angle of rotation equal to 10 degrees is applied, the coupled method struggles to achieve an acceptable grid displacement. Consequently, the sub-step displacement method is employed, ultimately yielding a satisfactory final grid suitable for applications such as those of CFD flow analysis. Table 3.3, regarding the case of a 10-degree rotation, reinforces this observation, indicating quality metrics that closely align with those of the initial grid, signifying sufficiency for practical use.

It is crucial to highlight that the application of the non-linear equations is also explored, yielding metrics of high quality, but at the same time requiring more computational time. Specifically, for a 5-degree rotation, the non-linear coupled method needs 33 seconds to achieve convergence, the decoupled method needs 4 seconds, while the linearized coupled method achieves comparable quality metrics in just 2 seconds. Therefore, implicitly solving the linearized equations of the coupled method demonstrates superior efficiency in this case.

However, in the case of a 10-degree rotation, both the decoupled and coupled methods featuring linearized equations produce resulting grids containing reversed cells, in the number of 16 and 36 respectively. Despite their rapid generation in 12 seconds, these outcomes are deemed practically useless, especially in applications where high-fidelity models are required. In contrast, the coupled method aided by the enhancement of the sub-step displacement process requires 68 seconds but produces a grid free of reversed cells, maintaining high-quality metrics. This reinforces the



(a)

(b)

(c)

Figure 3.10: *Isolated Airfoil. Rotation of the airfoil clock-wise by 5 degrees (a) and 10 degrees (b, c) with the trailing edge as the center of rotation. In (c) the sub-step displacement method has been employed. Results from implementing the linearized formulation of equations of the coupled method.*

value of the sub-step displacement method for resolving challenges posed by larger displacements of the boundary.

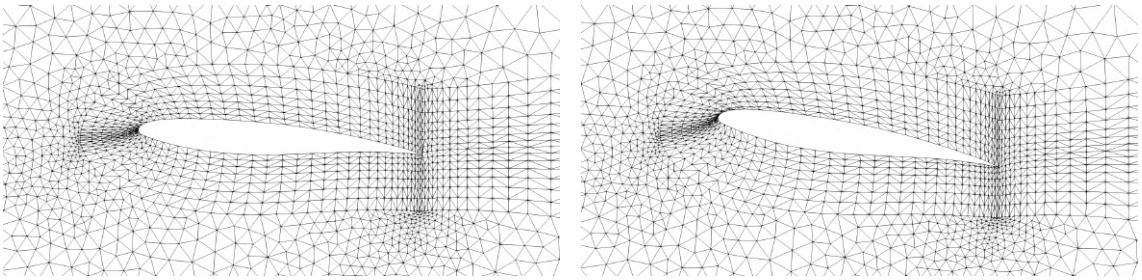
Rotation around the aerodynamic center

The rotation angles of 5 and 10 degrees are further tested, with the center of rotation positioned at the aerodynamic center of the airfoil $(c/4, 0)$. This rotation is once

Metric		Initial	Final	
			Decoupled	Coupled (sub-step)
q	mean	0.114109	0.114527	0.114309
	std	0.583948	0.584957	0.583978
	min	0.000043	0.000001	0.000035
	max	7.715670	7.717010	7.741460
$N-O$	mean	11.1667	12.7817	12.0144
	std	10.5519	14.2340	11.8213
	min	0.0444	0.1017	0.3107
	max	49.1254	85.9531	69.7260
$Skewness$	mean	0.104589	0.159747	0.112232
	std	0.060032	0.175077	0.087810
	min	0.001421	0.002240	0.002190
	max	0.826552	0.840400	0.891540

Table 3.3: *Isolated Airfoil: Quality Metrics. The airfoil is rotated by 10 degrees clock-wise with the trailing edge as the center of rotation.*

again executed clockwise around the z-axis.



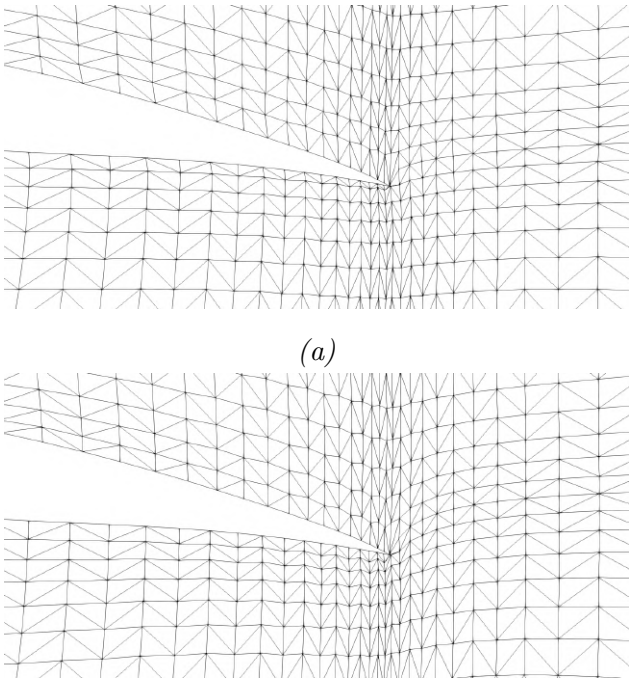
(a) *(b)*

Figure 3.11: *Isolated Airfoil. Rotation of the airfoil clock-wise by 5 (a) and 10 (b) degrees around the aerodynamic center. Results from the implementation of the linearized equations of the coupled method.*

In Fig. 3.11, the results obtained from implementing the coupled method for both rotation angles are displayed. In the case of a 10-degree rotation, the sub-step method is again necessary to be employed to prevent the occurrence of reversed cells near the trailing edge.

Fig. 3.12 highlights the distinctions near the trailing edge between the decoupled and coupled methods' results. Notably, in the final grid produced by the coupled method's approach, grid nodes appear to remain closer to their initial positions, resulting in a finer grid on the pressure side of the airfoil.

Additionally, similar to the previous case, using the linearized formulation of equations proves to be significantly faster, requiring only 2 seconds to achieve conver-



(a)

(b)

Figure 3.12: *Isolated Airfoil. Rotation of the airfoil clock-wise by 10 degrees around the aerodynamic center. The resulting grids from the implementation of the coupled (a) and the decoupled (b) methods.*

gence, while the non-linear coupled approach needs 10 seconds and the decoupled method needs 4 seconds.

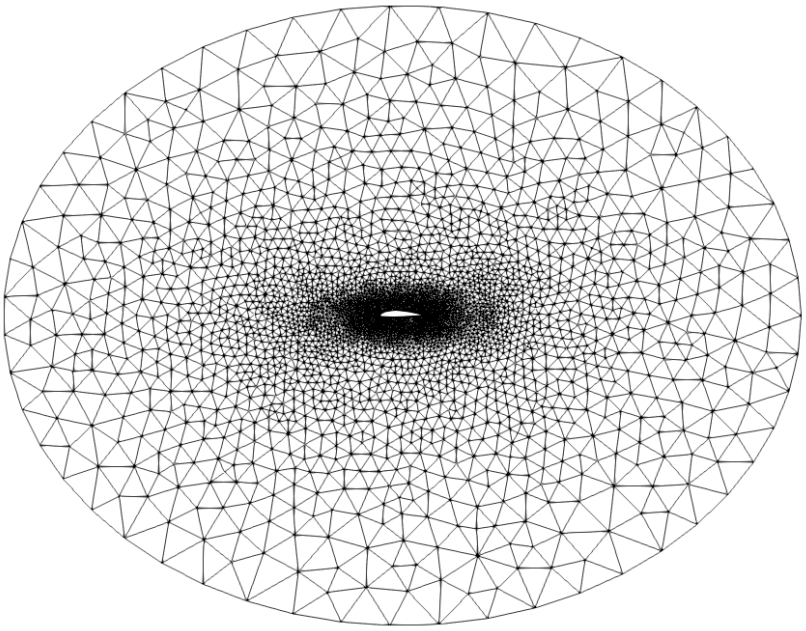
Therefore, the conclusions drawn from the Two Concentric Squares case are further validated in the aerodynamics domain. Once again, it is evident that the sub-step displacement method is indispensable, particularly to support the linearized coupled method in managing more significant displacements, albeit with increased computational cost. Additionally, in terms of computational efficiency, employing the linearized equations of the coupled method leads to faster convergence compared to both the non-linear coupled and the decoupled methods. Furthermore, the contrast between the results obtained from the coupled and decoupled methods is reaffirmed.

3.2.1 The NACA4415 Airfoil

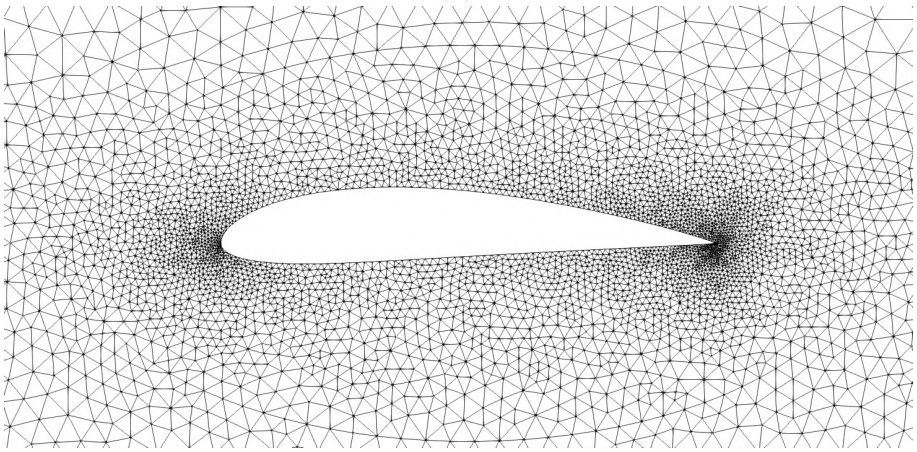
Exploring the behavior of the RBM method, the NACA4415 airfoil serves as a distinct case study of an isolated airfoil. The grid surrounding this airfoil is revealed in Fig. 3.13. This unstructured 2D grid comprises 13024 nodes and 12773 triangles.

This case study aims to illustrate the different outcomes of the coupled and decoupled methods, particularly within a finer grid region of an example from the field of

aerodynamics. Additionally, with a larger number of nodes and higher grid fineness compared to the previous case, it serves to test the RBM method's capabilities in slightly larger problem domains.



(a)

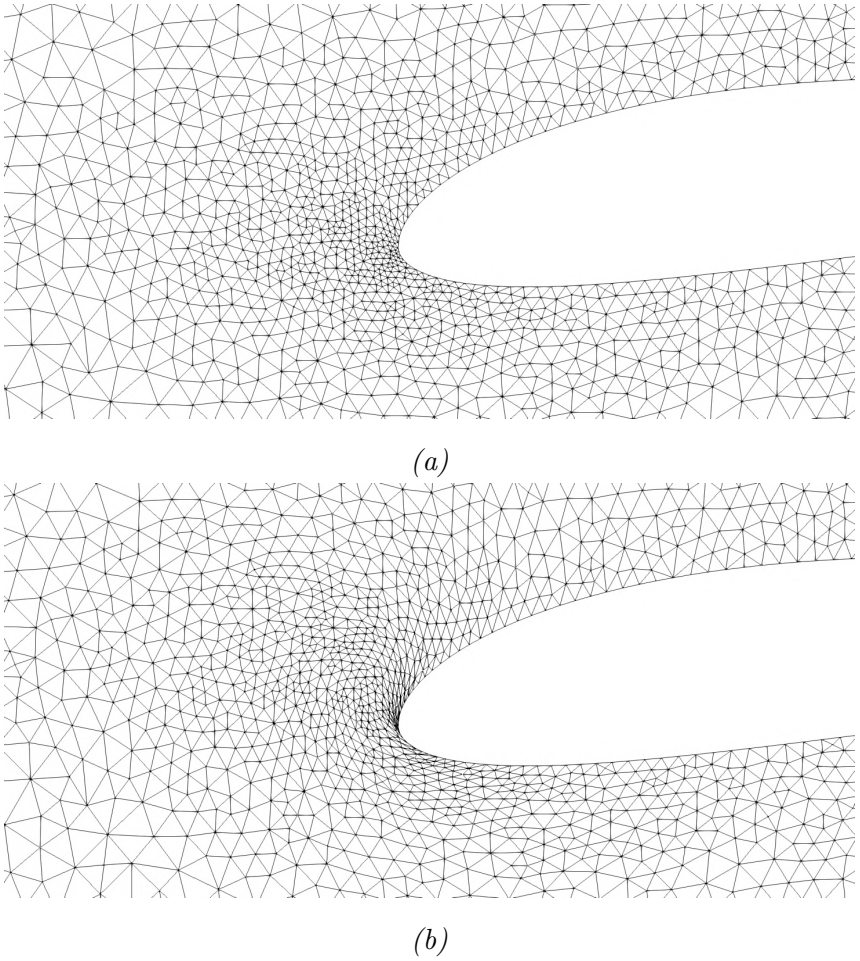


(b)

Figure 3.13: *(a): 2D unstructured grid around the NACA4415 Airfoil. (b): Focus on the Airfoil.*

Initially designed as a pseudo-3D grid consisting solely of prisms for OpenFOAM applications, it is modified using a tailored code to retain one of the identical sides, facilitating ease of handling and faster results production. To assess the RBM tool in this case, the boundary nodes of the airfoil are subjected to a 4-degree counter-clockwise rotation, with the trailing edge as the selected center of rotation.

The results, illustrated in Fig. 3.14, focus on the leading edge for a detailed comparison of the final grids. It becomes evident that the coupled method displaces grid nodes comprehensively, providing a broader view of the entire grid. In contrast, the decoupled method places more emphasis on nodes in proximity to the airfoil's boundary, with less displacement observed further away. This aligns with the underlying rationale of the methodology. Crucially, both final grids exhibit quality metrics comparable to the values of the initial grid, rendering them suitable for practical use. Furthermore, the absence of reversed cells in both final grids underscores their suitability for downstream applications.



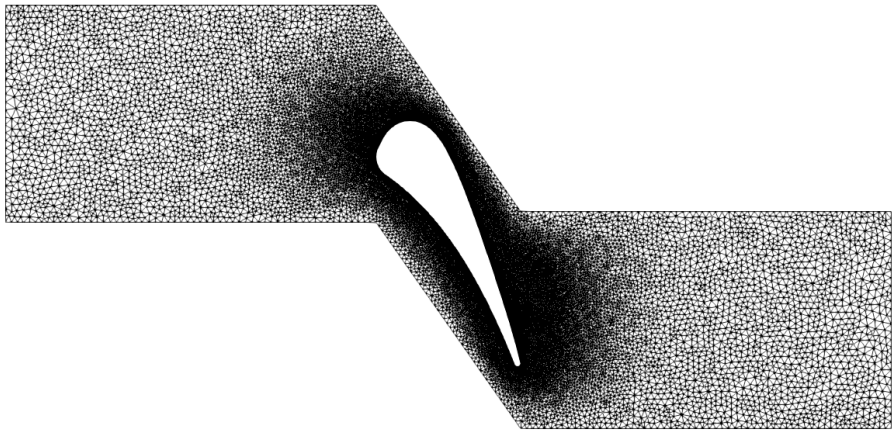
(a)

(b)

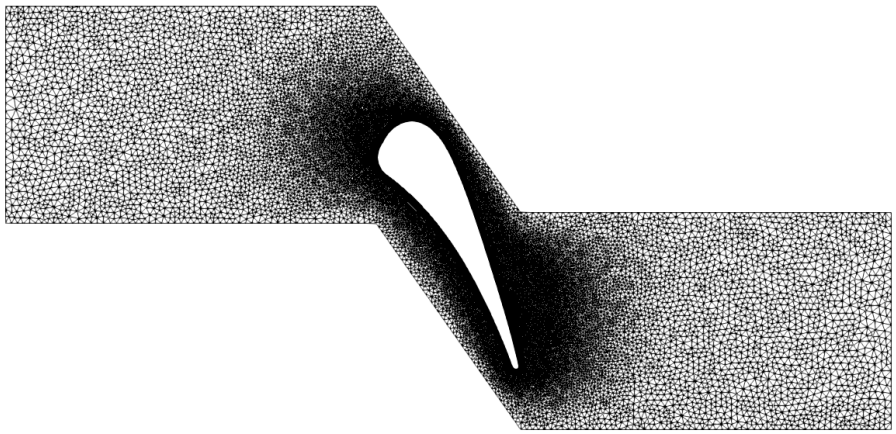
Figure 3.14: *The NACA4415 Airfoil. The airfoil is rotated by 4 degrees counter-clock-wise with the trailing edge as the center of rotation. Presentation of the resulting grids of the coupled (a) and decoupled (b) methods with emphasis on the leading edge.*

3.3 The C3X Turbine Cascade

In this particular case, the geometrical configuration of the C3X turbine cascade is undertaken to discretize the surrounding domain. The motivation behind this application lies in the need to assess the effectiveness of the RBM coupled method in handling fine grids, constituted by a substantially greater number of nodes. The grid illustrated in Fig. 3.15 serves this purpose well, boasting 190124 nodes and 128652 elements.



(a)



(b)

Figure 3.15: *C3X turbine cascade. The airfoil is slightly rotated. (a) Initial grid. (b): Final grid. Result from the implementation of the linearized equations of the coupled method.*

Initially designed as a pseudo-3D grid for the OpenFOAM software environment, featuring 68622 prisms and 60030 hexahedra, this grid undergoes transformation into a 2D grid with 68622 triangular and 60030 quadrilateral elements through a dedicated pre-processing code. Designed as a hybrid grid, it strategically employs

structured quadrilateral elements near the cascade's wall and unstructured triangular elements further away, a choice made to mitigate the already considerable computational demands associated with the high-fineness and the great number of nodes constituting the grid.

In this challenging test, the airfoil undergoes a slight 1-degree clockwise rotation around its aerodynamic center. The primary objective is to evaluate the method's capability to manage such fine grids, ideally without the occurrence of reversed cells in the final grid. Additionally, this case serves to test the method's capacity to handle and solve a significantly larger system of equations compared to previous tests.

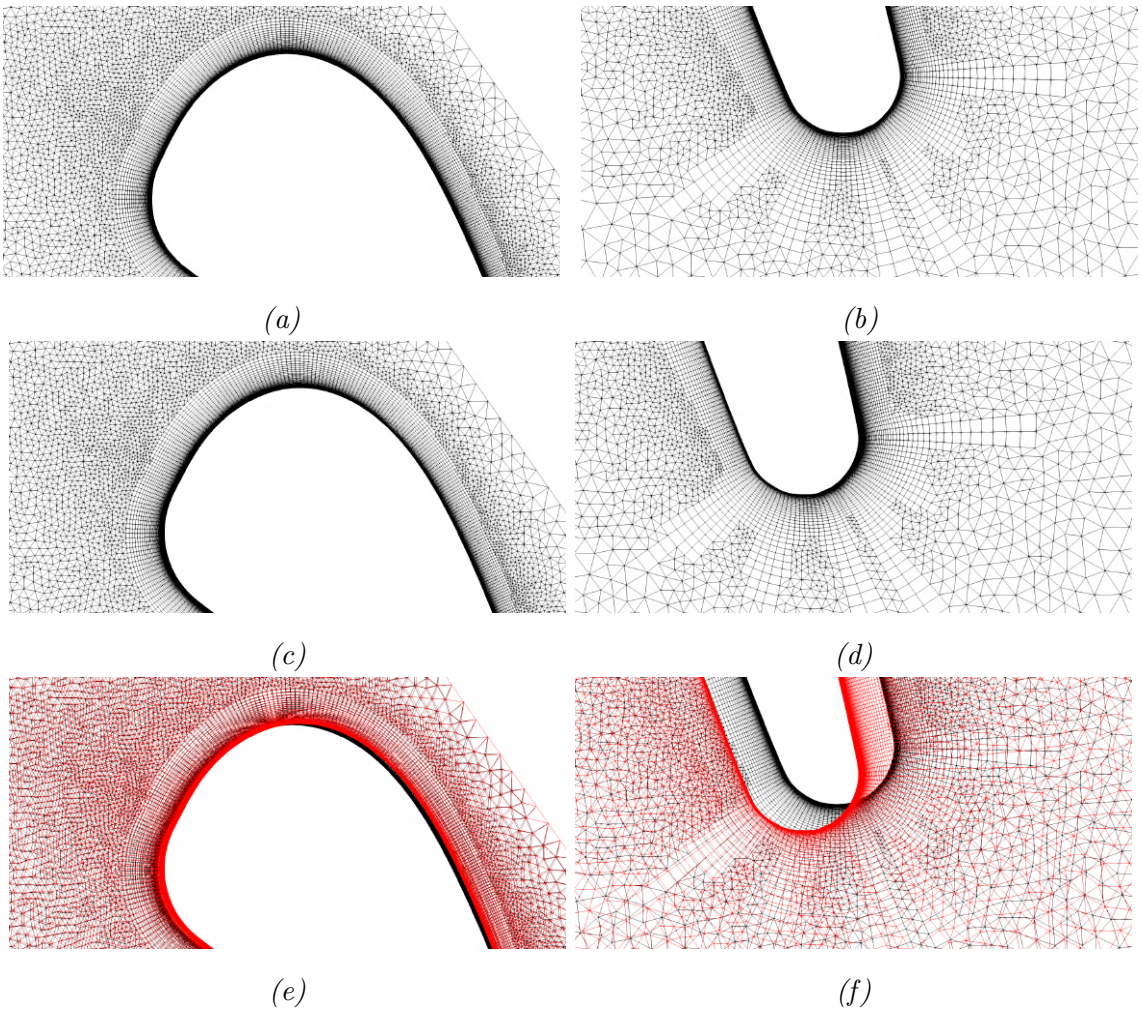


Figure 3.16: *C3X turbine cascade. The airfoil is slightly rotated. Illustration of the initial (a),(b) and final (c),(d) grids, emphasizing on the leading (a),(c),(e) and trailing (b),(d),(f) edges of the cascade. In (e),(f) both the initial (black color) and the final (red color) grids are depicted.*

It is crucial to note the computational intensity of this process, with a substan-

Metric		Initial	Final
$N-O$	mean	6.3245	8.7421
	std	10.5519	14.2819
	min	0.0010	0.0073
	max	32.2275	89.6606

Table 3.4: *C3X turbine cascade: Quality Metrics*

tial computational cost. It should be mentioned that the linearized formulation of equations was employed to alleviate the already high computational demands.

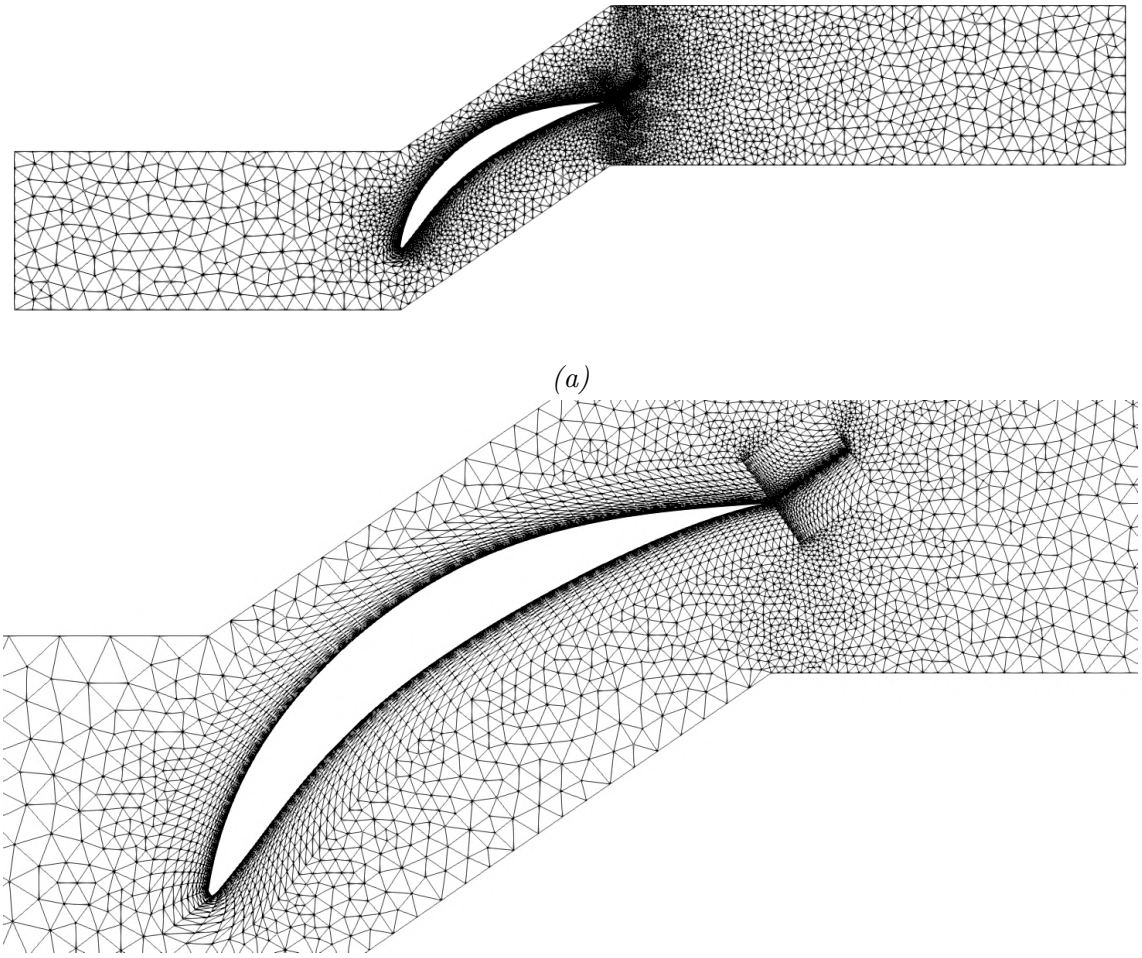
Results presented in Fig. 3.16, with specific attention to the leading and trailing edges, indicate that the method adeptly maintains high fineness levels near the cascade's wall, with no reversed cells evident in the final grid. Supporting this observation, the metric of non-orthogonality provided in Table 3.4 aligns closely with the values of the initial grid, meeting the desired criteria. Overall, despite the high fineness of the grid, the results are deemed satisfactory, affirming the displaced grid's qualification for further use.

3.4 Compressor Cascade

In this application, rather than treating the entire aerodynamic body uniformly and displacing all nodes forming it with the same parameters $(\Delta x, \Delta y, \theta)$, distinct parts of the body experience a different displacement. This procedure mimics the process of aerodynamic shape optimization, aimed at refining the shape to optimize a specific function.

Metric		Initial	Final
q	mean	0.000596	0.000626
	std	0.000067	0.000118
	min	0.000001	0.000002
	max	0.007523	0.008916
$N-O$	mean	17.4561	21.7731
	std	0.4456	0.8432
	min	0.0749	0.0750
	max	55.3218	85.5535
$Skewness$	mean	6.3245	8.7421
	std	1.3551	1.4282
	min	0.0010	0.0015
	max	10.4442	10.7766

Table 3.5: *Compressor cascade: Quality Metrics*



(a)

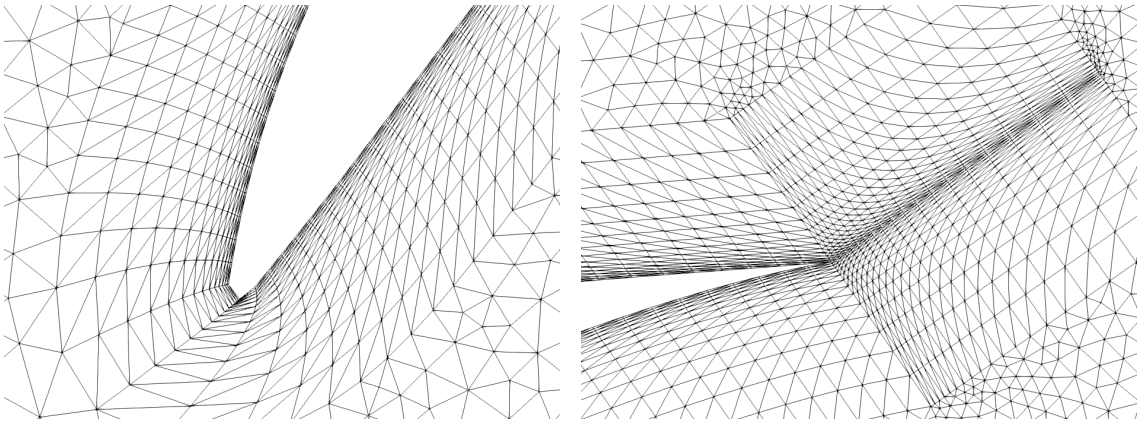
(b)

Figure 3.17: *(a): 2D grid around a Cascade of a Compressor. (b): Focus on the Cascade.*

This test employs a 2D hybrid grid surrounding a compressor cascade. This grid, depicted in Fig. 3.17, comprises 4983 nodes and 9632 triangular elements. It is characterized by high fineness and structure near the cascade's wall and trailing edge, facilitating detailed analysis of turbulent flows. Conversely, as distance from the wall increases, the configuration transitions to a coarser grid.

In order to imitate the change in shape, simulating an aerodynamic shape optimization problem, the curvature of the cascade is reduced by lifting both the leading and the trailing edge, shown in Fig. 3.18. To handle this case, the linearized formulation of equations of the coupled method is used.

The outcome, depicted in Fig. 3.19, indicates that the method works efficiently,



(a) *(b)*

Figure 3.18: *Compressor Cascade. Focus on the leading (a) and trailing (b) edges of the cascade.*

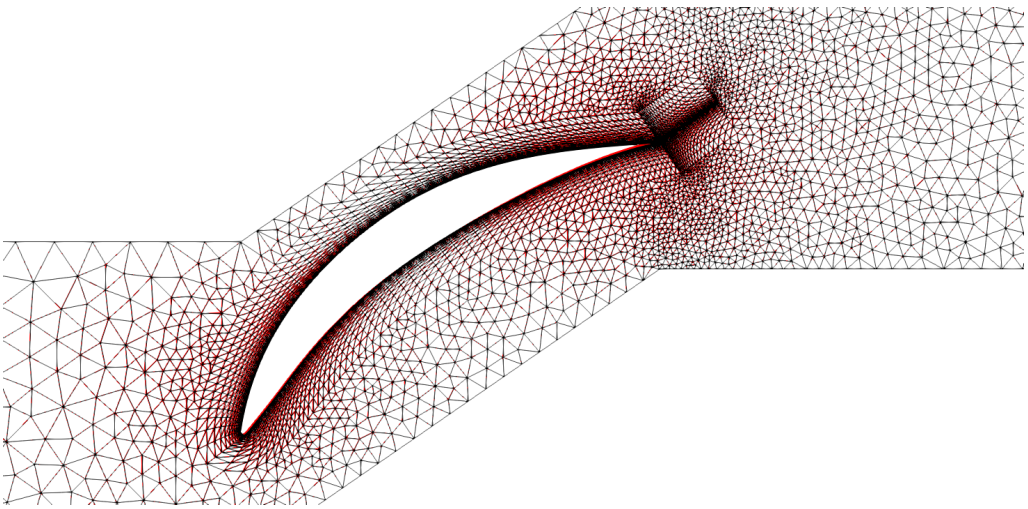


Figure 3.19: *Compressor Cascade. Change in the shape of the cascade. Depiction of the initial (black color) and final (red color) grids.*

producing a final grid preserving high fineness in critical areas and free of inverted cells. Additionally, Table 3.5 indicates the suitability of the resulting grid for practical use, as evidenced by the comparison of quality metrics between the initial and final grids. These findings highlight the method's capability to effectively handle cases involving changes in the shape of aerodynamic bodies, as encountered in an aerodynamic shape optimization process.

Chapter 4

3D Grid Displacement: Applications

Following the testing on 2D grid displacements, the implementation of the RBM coupled method is extended to 3D applications to assess its efficacy. This involves utilizing the appropriate equations tailored for the 3D space and evaluating the outcomes to ensure the method's reliability and functionality.

4.1 Deformation of a Cube

The first case under investigation involves the deformation of a cube, selected for its geometric simplicity yet diverse element composition. Comprising a blend of tetrahedra, pyramids, prisms, and hexahedra, this choice aims to present the efficient handling of various element types by the RBM method. The cube houses 1340 nodes forming 3740 elements. The initial configuration is depicted in Fig. 4.1.

As part of the deformation process, the cube undergoes distortion via a rotation around the y-axis, proportionate to the y-coordinate of each plane. This rotational transformation is mathematically defined as:

$$x^{new} = x^{old} \cos\theta_y + z^{old} \sin\theta_y \tag{4.1a}$$

$$y^{new} = y^{old} \tag{4.1b}$$

$$z^{new} = -x^{old} \sin\theta_y + z^{old} \cos\theta_y \tag{4.1c}$$

$$\theta_y = \alpha \cdot y^{old} + \beta \tag{4.1d}$$

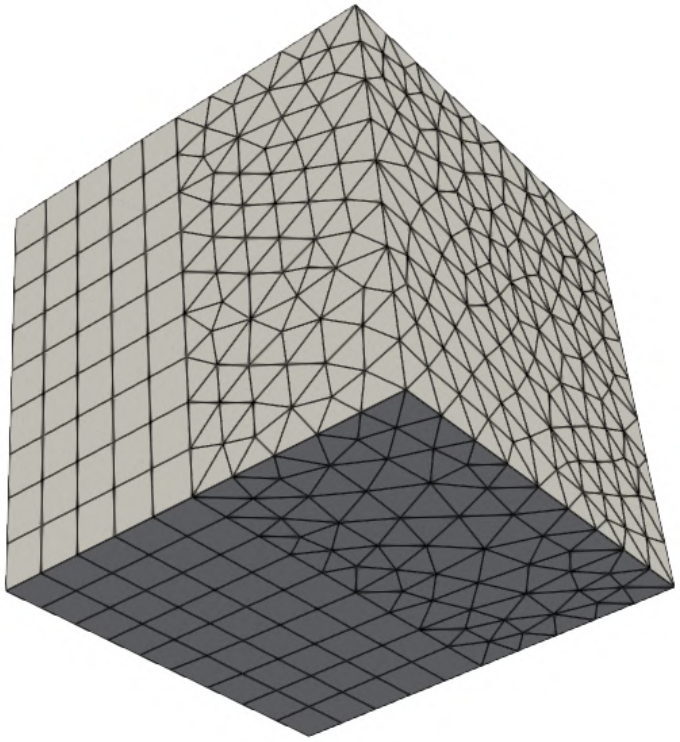


Figure 4.1: *Visualization of the Cube with grid overlay.*

Given the relatively coarse grid, a maximum twist of 30 degrees is applied to the front plane to assess the RBM coupled method's performance in 3D. The resulting grid is displayed in Fig. 4.2, with detailed internal configurations showcased in Fig. 4.3.

The absence of reversed cells in the final grid indicates the RBM coupled method's capability to handle 3D cases effectively. It should be mentioned that the linearized equations are employed for this grid displacement. Moreover, the sub-step displacement method needs to be utilized, dividing the total 30-degree rotation into incremental steps of 1 degree. Despite this, the method achieves rapid convergence, completing the total grid deformation within 48 seconds.

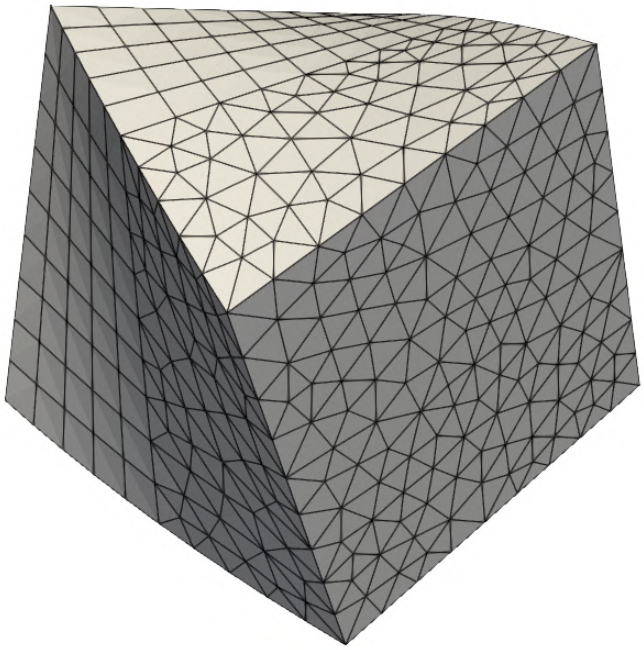


Figure 4.2: *Resulting grid after the deformation of the cube.*

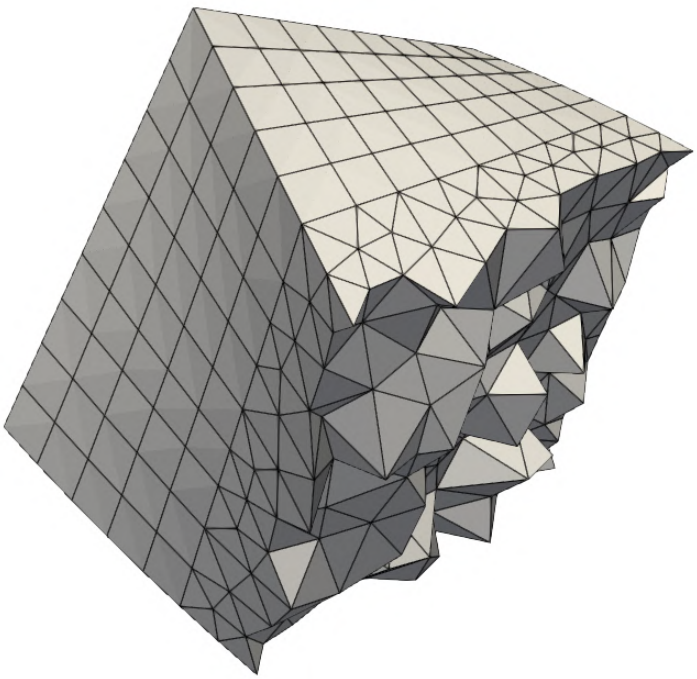


Figure 4.3: *Resulting grid after the deformation of the cube. Internal configuration displayed.*

4.2 Rotation of an Aircraft

The next test case focuses on the pitching motion of a small passenger aircraft. Around this aircraft, a computational grid is designed and displaced. The grid utilized in this scenario consists of 45387 nodes forming 255944 tetrahedra. The initial configuration of the grid is illustrated in Figs. 4.4 and 4.5. It exhibits intricate and finely detailed geometric features, posing a significant challenge to the RBM method's adaptability to complex grid configurations.

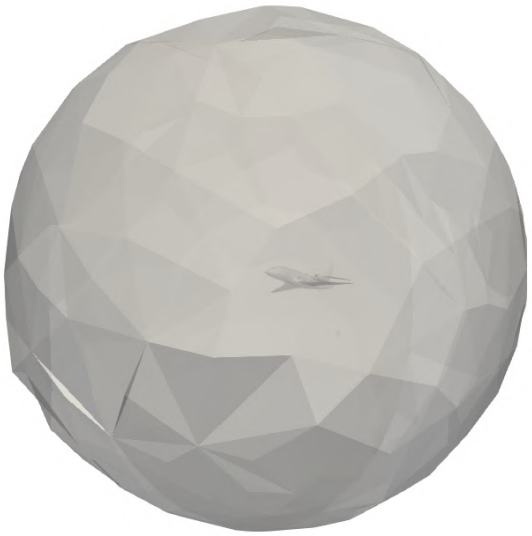
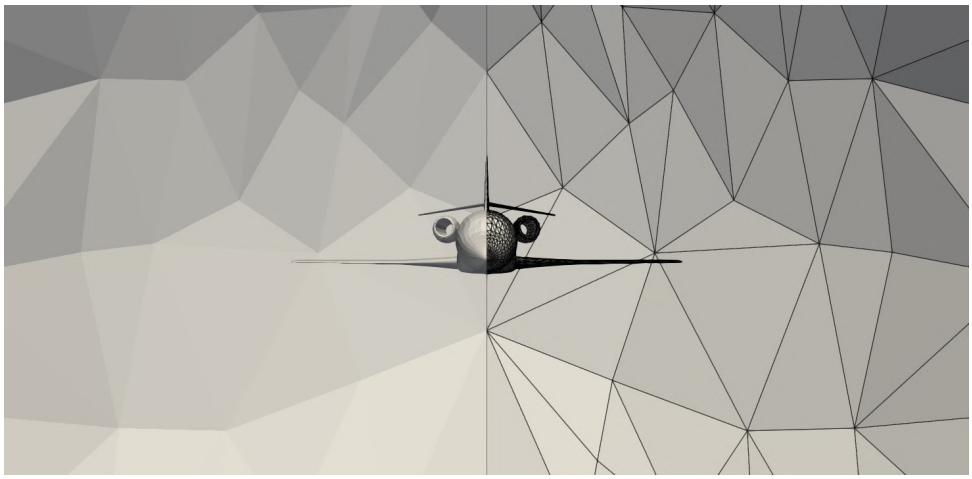


Figure 4.4: *Computational domain around an Aircraft.*

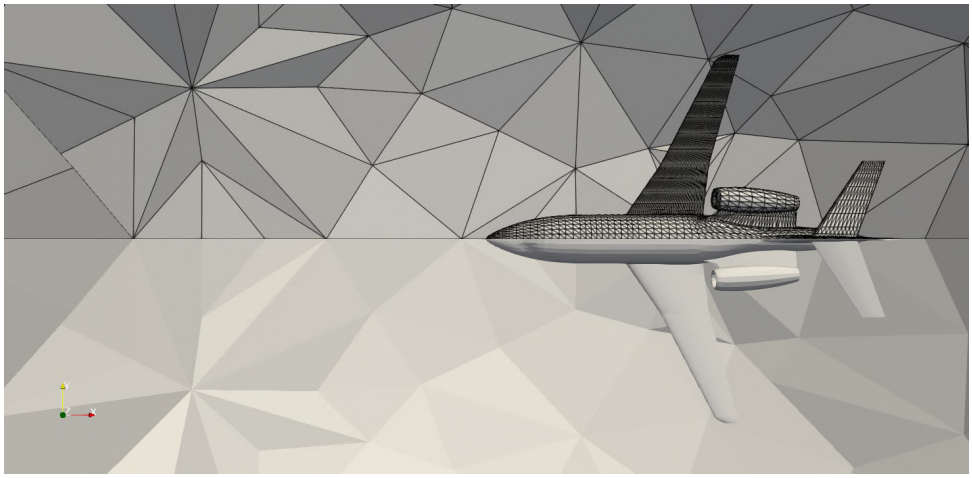
For this test, the aircraft undergoes a pitch of 3 degrees, simulating the initiation of a descent. The goal is to assess the RBM method's ability to displace the grid accurately, enabling its use in subsequent computational fluid dynamics (CFD) simulations. Such simulations could be integral to aircraft design procedures or studies evaluating different flight conditions.

Metric		**Initial**	**Final**
$N-O$	mean	5.3241	8.4567
	std	1.3331	2.2845
	min	0.0040	0.0042
	max	42.9898	53.0104
Skewness	mean	7.4231	8.9854
	std	2.5001	3.1905
	min	0.0438	0.0439
	max	60.7893	72.7638

Table 4.1: *Small passenger Aircraft: Quality Metrics*



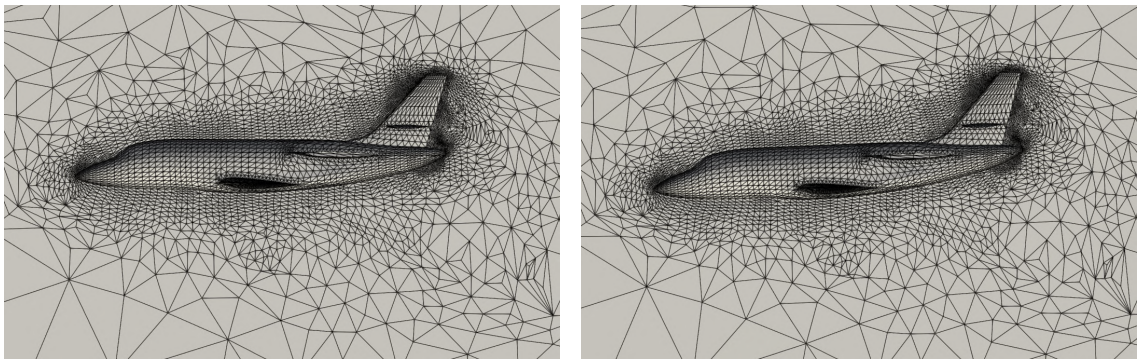
(a)



(b)

Figure 4.5: *(a): Front view of the aircraft with grid visualization on its left side. (b): Top view of the aircraft with grid visualization on the upper half.*

The resulting grid is obtained through the implementation of the linearized equations of the coupled method. Fig. 4.6 depicts the initial and final grids, showcasing the RBM method's ability to maintain grid quality by preserving the fineness level in critical areas essential for computing significant quantities. To achieve satisfactory results, the sub-step displacement method needs to be employed, pitching the aircraft by 0.5 degrees at each step, totaling 6 steps. Furthermore, the absence of inverted cells in the resulting grid is evident, as examined in Fig. 4.7, which focuses on specific fine regions of the grid. Additionally, Table 4.1 confirms the displaced grid's suitability for use in CFD solvers based on the assessment of appropriate quality metrics.



(a) *(b)*

Figure 4.6: *Comparison of the initial (a) and final (b) grids surrounding the aircraft, viewed from the side, after pitching by 3 degrees.*

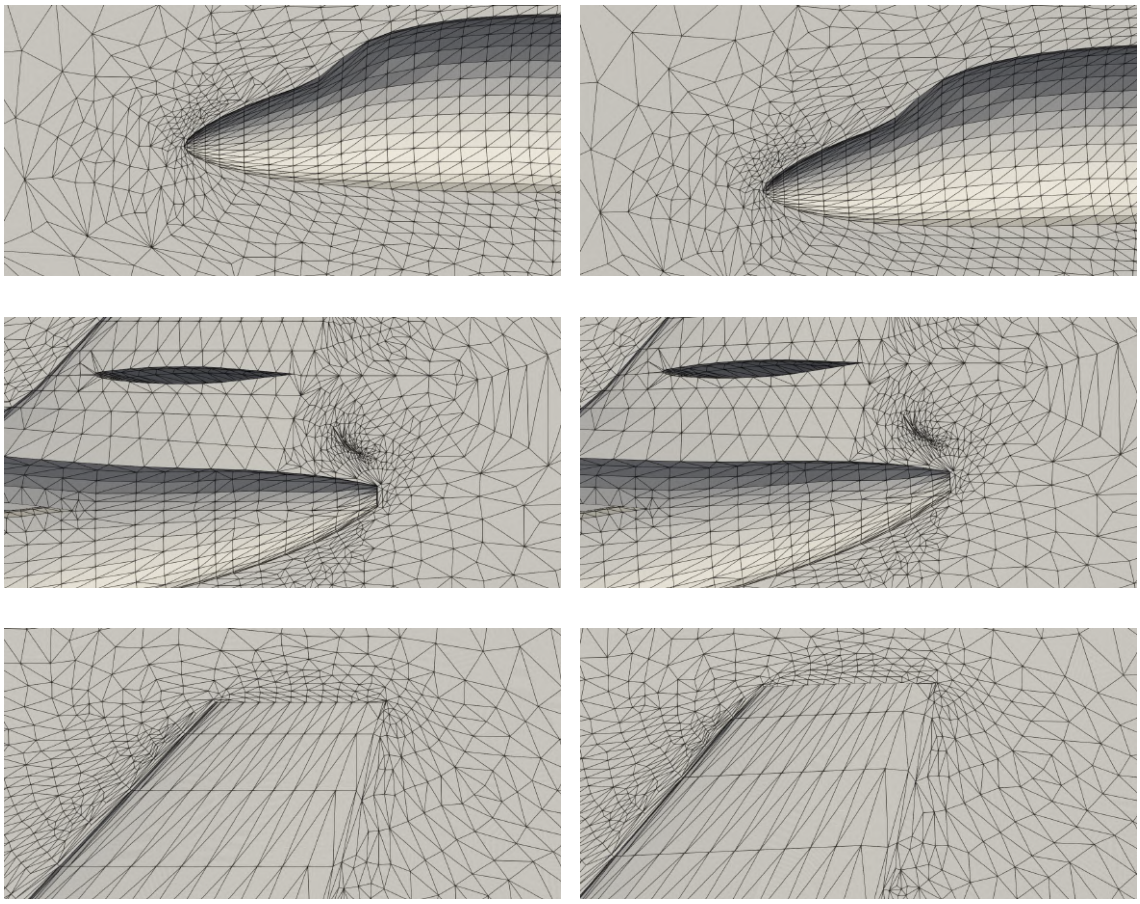


Figure 4.7: *Focus on specific fine regions of the grid. Left: initial grid. Right: final grid. Top: nose of the aircraft. Middle: tail of the aircraft. Bottom: rudder of the aircraft.*

4.3 Bending of the ONERA M6 Wing

To simulate a process in 3D, where the shape of the aerodynamic body is subject to change due to an optimization procedure or a fluid-structure interaction, a computational grid accommodating the ONERA M6 wing is utilized. The entire computational domain is defined as a hexahedron, inside of which the wing is modeled. This grid comprises 72791 nodes, forming 341797 tetrahedra, making it significantly larger and finer than the one used in the previous application involving the small passenger aircraft. The substantial increase in the number of nodes provides a valuable opportunity to assess the method's performance in large 3D scenarios.

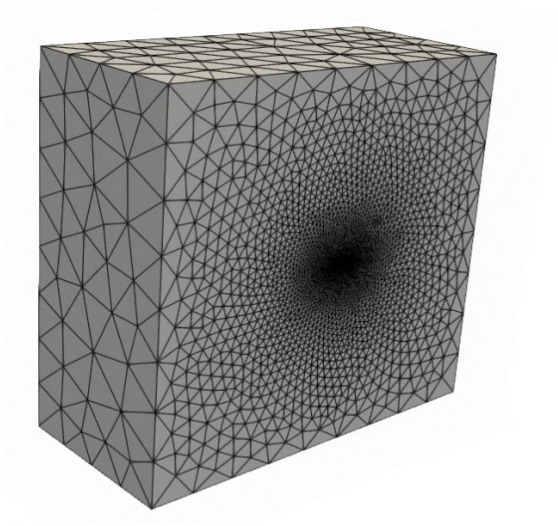
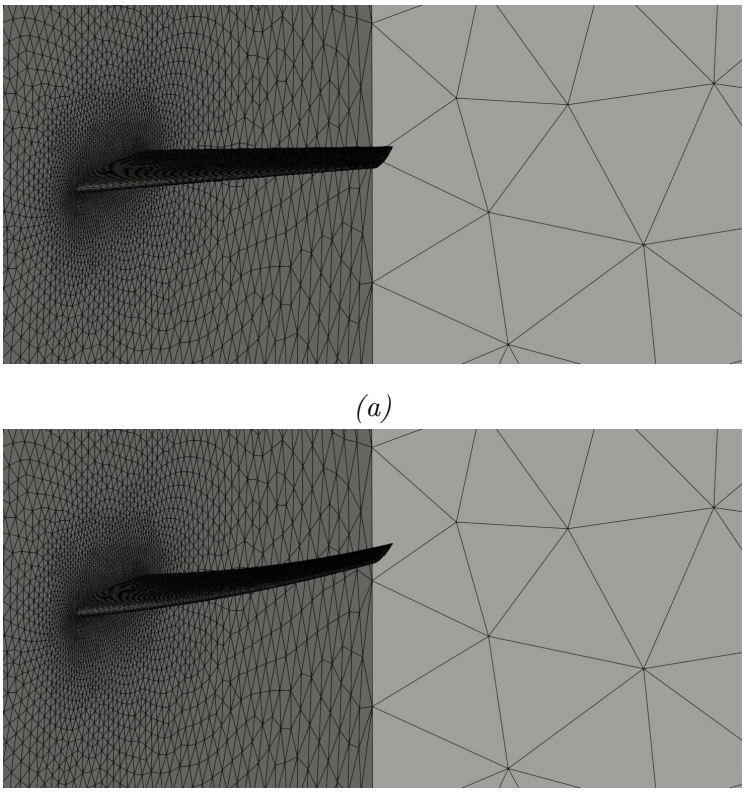


Figure 4.8: *Computational domain around the ONERA M6 Wing.*

The complete computational domain containing the ONERA M6 Wing is illustrated in Fig. 4.8, while Fig. 4.9 provides a detailed view of the wing itself along with the grid designed around it.

Metric		**Initial**	**Final**
$N-O$	mean	20.5861	22.8184
	std	4.2118	4.4960
	min	1.2393	1.2393
	max	87.3420	89.5102
Skewness	mean	48.5234	49.5417
	std	1.2354	1.5336
	min	0.1398	0.1398
	max	60.8676	62.95672

Table 4.2: *ONERA M6 Wing: Quality Metrics*



(a)

(b)

Figure 4.9: *Focus on the ONERA M6 Wing. (a): Initial grid. (b): Final grid.*

For this test, the ONERA M6 wing is subjected to bending, following the mathematical formula:

$$x^{new} = x^{old} \tag{4.2a}$$

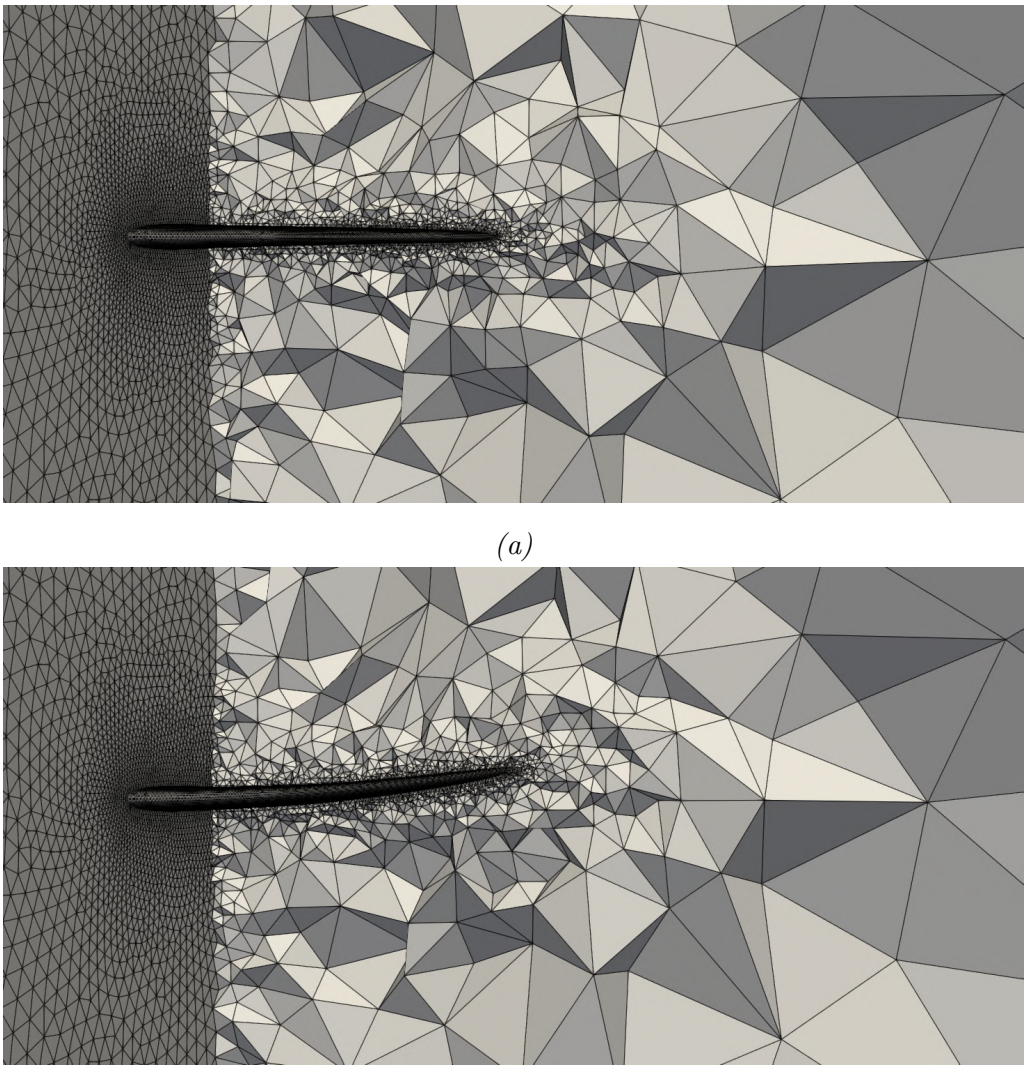
$$y^{new} = y^{old} + \alpha(z^{old})^2 \tag{4.2b}$$

$$z^{new} = z^{old} \tag{4.2c}$$

The α variable is set to a value of 0.1, and the grid is displaced using the linearized equations of the coupled method. The resulting grid is depicted in Fig. 4.9.

The RBM method demonstrates remarkable proficiency in managing the displacement of the fine grid surrounding the ONERA M6 Wing, as illustrated in Fig. 4.10. It effectively prevents the formation of inverted cells while preserving the quality and fineness of the initial grid. This achievement is particularly noteworthy given the high demands imposed by the great number of nodes of the grid and the deformed aerodynamic profile of the wing.

Despite the considerable computational cost associated with this task, the RBM method proves to be an efficient solution, delivering satisfactory results. The necessity of employing a significant number of sub-steps is evident, especially given the fineness of the grid. This approach underscores the method's reliability in handling 3D fine grid configurations and resolving larger systems of equations. The successful adaptation of the grid through the RBM method enables the accurate analysis of aerodynamic phenomena.



(a)

(b)

Figure 4.10: *Grid visualization on a specific plane. (a): Initial grid. (b): Final grid.*

Chapter 5

Summary and Conclusions

5.1 Summary

The need for adapting an existing grid to displaced boundaries arises in many applications (e.g., shape optimization, aeroelasticity problems). However, reconstructing the grid anew each time this adjustment is necessary can be prohibitively expensive for large-scale industrial applications. Therefore, automated algorithms are employed instead, as a cost-effective alternative to displace the nodes of the grid in alignment with the updated boundaries.

This work relies heavily on the recent PhD thesis presented by A. G. Liatsikouras [31], conducted in collaboration with the PCOpt/NTUA unit and the ESI Group. The utilization of external closed-source libraries prompts the development of an in-house tool, leading to the undertaking of this diploma thesis project.

Within this diploma thesis, a new approach to the RBM adaptive grid displacement method is developed and mathematically grounded. This method is inspired by the motion of a rigid body as defined in mechanics. To determine the configuration of the adapted grid, the displacement parameters $(\Delta x, \Delta y, \theta)$ need to be computed for each node. The new approach involves coupling the equations that govern the displacement of each node and implicitly solving the non-linear system of equations formed. The procedure of the problem's resolution using the decoupled equations is also presented. A strategy to linearize these equations, by approximating the nodes' rotation angle, is introduced and applied in both the coupled and decoupled methods. Additionally, an enhancement technique known as the sub-step displacement method is developed to assist in the node displacement process.

Software is programmed in *C++* and is capable of handling all types of grids (2D and 3D, structured and unstructured). This software is tested in a plethora of ap-

plications to evaluate the robustness of the method. Initial applications involve simpler geometric configurations (squares, cube) constituting coarse grids and examining their displacement to certify that the developed software produces sufficient results. Subsequently, tests are conducted on aerodynamic body cases, such as the rotation of airfoils and the deformation of a compressor's cascade. The case of rotating the C3X turbine cascade is also used to verify that the discussed method can handle finer and larger grids. To ascertain the establishment of the new approach of the RBM technique, the cases of the rotation of an aircraft and the bending of the ONERA M6 wing are implemented in 3D.

In all applications, the quality of the adapted grid is evaluated based on quality metrics tailored for 2D and 3D grids. Furthermore, the outcomes of the distinct methodologies employed are compared to gain valuable insights. The resulting grids are confirmed to meet the requirements for practical utilization in computational fluid dynamics (CFD) simulations. This extensive testing serves as a validation of the proposed model.

5.2 Conclusions

Drawing insights from diverse applications of the RBM method, it becomes evident that it stands as an efficient technique for adjusting grid nodes to align with new boundaries. Across various computational grid deformations and subsequent RBM method implementations, it becomes apparent that the new approach of implicitly solving the system of coupled equations dictating the displacement of each node yields sufficient outcomes, preserving the quality of the initial grid. This is due to the objective set for the development of the method, which treats each grid element as a rigid body, aiming to achieve the smallest deviation possible between its initial and final shape.

The analysis of the outcomes reveals notable distinctions between the various methods. While the non-linear coupled method achieves convergence at slower rates, it produces grids of superior quality. On the other hand, the utilization of linearized equations within the coupled method proves to be the swiftest in achieving convergence.

It is worth noting that both the coupled and decoupled methods, including their non-linear variants, may encounter issues related to the occurrence of inverted cells when dealing with significant boundary displacements. To overcome this obstacle, the integration of the sub-step displacement method is indispensable, resulting in higher-quality results devoid of inverted cells. It is observed that, even in cases without any imposed rotation of the boundary, the employment of the sub-step displacement method remains essential, particularly when the displacement magnitude exceeds the cell size. It is recommended to carefully consider the number of sub-steps about to be taken, in order to ensure that each step adequately avoids the risk of creating

inverted cells at the time when the boundary displacement is applied.

With this adaptation, there emerges a preference for employing the linearized coupled method, complemented by the sub-step displacement method, due to its robustness. Nevertheless, it must be noted that there are scenarios where the decoupled method may outperform the coupled approach, especially when comparing their non-linear formulations.

Moreover, the resulting grids obtained from the decoupled and coupled methods differ. The decoupled method prioritizes the displacement of nodes closest to the boundary, while the coupled method considers the broader context of the grid and the interconnections between its nodes.

In conclusion, the new approach of the RBM technique emerges as a reliable and effective means of adaptive grid displacement. Demonstrating its capabilities through various grid types, it consistently delivers satisfactory outcomes. Thus, it can serve as a valuable tool in addressing CFD problems with moving boundaries.

5.3 Recommendations for Future Study

The method investigated in this thesis has proven to be highly effective. Nonetheless, there are opportunities for further continuation and improvement.

Firstly, another categorization of the grid nodes could be introduced for nodes of which some of the coordinates may be subject to change. Specifically, this category could include nodes with specific DoFs, allowed to be displaced only along certain axes, while their coordinates on other axes remain constant. This refinement could enhance control over node displacements and decrease the dimensions of the system to be solved.

Additionally, in this work, the system of coupled equations is solved utilizing the Portable Extensible Toolkit for Scientific Computation (PETSc) library (more details in Appendix A). Given its standalone nature, there is potential for further exploration of its capabilities, leveraging advanced numerical analysis algorithms to achieve even faster convergence and resolution of the system. Exploring and adjusting various settings related to its operation could enhance its robustness in tackling a wide range of problems effectively.

Finally, a significant improvement of the method would involve the parallelization of the software. Parallelization can greatly accelerate code execution by distributing tasks among multiple processors [46]. While the management of computational resources is not examined in this thesis, it presents an important area for future investigation.

Appendix A

Resolution of the System of Equations

As described in Chapter 2, in the context of implementing the coupled method, the interdependence of equations governing the displacement of each node leads to the formation of a system of equations. The resolution of this system is imperative for obtaining the displacement parameters $(\Delta x, \Delta y, \theta)$ for every internal node of the grid. Achieving precise resolution of this system of equations is therefore essential for accurate grid adaptation.

To address this, an external library known as the Portable Extensible Toolkit for Scientific Computation (PETSc) is employed [47], [48]. PETSc facilitates the resolution process by accepting specific inputs, thereby simplifying the procedure for users.

The input required from PETSc should include the following:

- The Equations of the system to be solved.
- An Initial Guess of the solution.
- The Jacobian matrix (optional).
- The Convergence Criteria.

Initially, users define the system's dimensions. Setting up the system's equations constitutes a fundamental aspect of the problem. Users may also supply the Jacobian matrix, representing derivatives of the equations w.r.t. the unknown quantities. While PETSc can compute this matrix, if not provided by the user, using Finite Differences, this incurs increased computational time, thus user-supplied Jacobians are encouraged. PETSc utilizes a sparse matrix format to conserve memory space. Regarding initialization, an initial guess of zero for all unknown variables is adopted in this thesis, assuming that boundary displacement or deformation is relatively small compared to the grid's overall size, thereby expecting the final node positions to be close to their initial ones. Lastly, users must specify convergence criteria and

relevant tolerances for PETSc to adhere to, assessing convergence at each iteration and continuing until convergence or meeting a stopping criterion.

PETSc offers more customizable settings, including the selection of the numerical analysis method and the type of preconditioner employed, allowing users to tailor options to their needs and preferences.

Overall, the PETSc library serves as a useful tool for simplifying the numerical resolution of systems of equations for users.

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Μηχανολόγων Μηχανικών
Τομέας Ρευστών
Εργαστήριο Θερμικών Στροβιλομηχανών
Μονάδα Παράλληλης Υπολογιστικής Ρευστοδυναμικής
& Βελτιστοποίησης

Πεπλεγμένη Μέθοδος Μετατόπισης Πλέγματος βασισμένη στο Μοντέλο Κίνησης Απαραμόρφωτου Σώματος

Εκτενής Περίληψη Διπλωματικής Εργασίας

Ευάγγελος Χριστιανός

Επιβλέπων:
Κυριάκος Χ. Γιαννάκογλου, Καθηγητής ΕΜΠ

Αθήνα, 2024

Εισαγωγή

Η έννοια της βελτιστοποίησης, θεμελιώδης για την επίτευξη στόχων αποδοτικότητας, έχει διευρυνθεί σημαντικά στην ψηφιακή εποχή, διεισδύοντας σε διάφορους τομείς όπως η μηχανική, οι οικονομικές επιστήμες και η τεχνητή νοημοσύνη. Αυτή η εξέλιξη έχει οδηγήσει σε συνεχή ανάπτυξη και στον τομέα της αεροδυναμικής, ειδικότερα στην προσπάθεια βελτίωσης της αεροδυναμικής συμπεριφοράς των υπό μελέτη σωμάτων.

Κεντρικό σημείο της διαδικασίας αεροδυναμικής βελτιστοποίησης είναι η αξιολόγηση υποψήφιων λύσεων χρησιμοποιώντας εργαλεία επίλυσης ροών, όπως κώδικες Υπολογιστικής Ρευστοδυναμικής (ΥΡΔ), οι οποίοι απαιτούν ένα κατάλληλο υπολογιστικό πλέγμα. Λόγω της επαναληπτικής φύσης της διαδικασίας βελτιστοποίησης, η προσαρμοστική παραμόρφωση του υπολογιστικού πλέγματος κατέχει εξαιρετική σημασία, μετατοπίζοντας τους εσωτερικούς κόμβους του πλέγματος βάσει γνωστών παραμορφώσεων των ορίων.

Η διπλωματική αυτή εργασία επικεντρώνεται στην εξερεύνηση της τεχνικής της Κίνησης Απαραμόρφωτου Σώματος (ΚΑΣ), η οποία αποτελεί μια μέθοδο προσαρμοστικής παραμόρφωσης υπολογιστικών πλεγμάτων. Η τεχνική ΚΑΣ βασίζεται στον ορισμό του απαραμόρφωτου σώματος στη μηχανική, αντιμετωπίζοντας τα στοιχεία του πλέγματος ως απαραμόρφωτα σώματα. Η νέα προσέγγιση της τεχνικής ΚΑΣ η οποία παρουσιάζεται σε αυτήν τη διπλωματική εργασία αφορά στην πεπλεγμένη επίλυση των συζευγμένων εξισώσεων που διέπουν τη μετατόπιση των κόμβων του πλέγματος, τονίζοντας με αυτόν τον τρόπο την αλληλεξάρτηση τους.

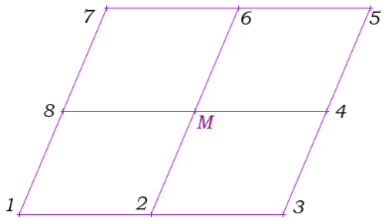
Διατύπωση του Προβλήματος

Σχετικά με την υλοποίηση του μοντέλου μετατόπισης ΚΑΣ, και όπως φαίνεται στο Σχήμα 1, επιθυμείται, δεδομένων των μετατοπίσεων των γειτονικών κόμβων *1* έως *8*, η κατάλληλη μετατόπιση του κεντρικού κόμβου M, εναρμονισμένη με τις ιδιότητες ενός απαραμόρφωτου σώματος. Η επίτευξη ιδανικής προσαρμογής για κάθε εσωτερικό κόμβο του πλέγματος αποτελεί ένα ιδανικό σενάριο, καθιστώντας την όλη διαδικασία ένα πρόβλημα βελτιστοποίησης.

Η μαθηματική έκφραση για τη μετατόπιση ενός κόμβου στον διδιάστατο χώρο είναι:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \tag{1}$$

Σύμφωνα με την υπόθεση πως το πλέγμα μετατοπίζεται προσεγγίζοντας όσο το δυνατόν περισσότερο την κίνηση ενός απαραμόρφωτου σώματος, η *Συνάρτηση Κόστους* F που ζητείται να ελαχιστοποιηθεί για κάθε κόμβο i, $i \in \mathcal{I}$, όπου $\mathcal{I}$ αναπαριστά το σύνολο των εσωτερικών κόμβων στο πλέγμα, είναι:



Σχήμα 1: *Πρότυπο 2Δ δομημένου πλέγματος. Αναπαράσταση του κεντρικού κόμβου Μ και των γειτονικών του, κόμβοι 1 έως 8.*

$$F_i \;=\; \frac{1}{2}\sum_{j\in\mathcal{N}(i)}[(x_j^{ideal}-x_j^{new})^2+(y_j^{ideal}-y_j^{new})^2] \tag{2}$$

Στην εξίσωση 2, ο δείκτης i υποδηλώνει τον κεντρικό κόμβο, ο δείκτης j υποδηλώνει έναν γειτονικό κόμβο του κεντρικού, ενώ το $\mathcal{N}(i)$ αποτελεί το σύνολο όλων των γειτονικών κόμβων του κεντρικού. Τα x_j^{ideal}, y_j^{ideal} υποδηλώνουν τις x, y συντεταγμένες του γείτονα j αντίστοιχα, μετά τη μετατόπισή του, υποθέτοντας πως πράγματι μετατοπίζεται σύμφωνα με τις ιδιότητες ενός στερεού σώματος. Τα x_j^{new}, y_j^{new} υποδηλώνουν τις πραγματικές x, y συντεταγμένες του γείτονα j αντίστοιχα, μετά τη μετατόπισή του.

Για να εξασφαλιστεί το βέλτιστο σύνολο $(\Delta x, \Delta y, \theta)$ για κάθε εσωτερικό κόμβο i, είναι απαραίτητο να υπολογιστούν οι παράγωγοι της συνάρτησης κόστους ως προς τις άγνωστες ποσότητες Δx_i, Δy_i, και θ_i, και να τεθούν ίσες με μηδέν.

Αποσυζευγμένη Μέθοδος

Αναλύοντας πρώτα την Αποσυζευγμένη Μέθοδο επίλυσης του προβλήματος, για να υπολογιστεί η μετατόπιση του κεντρικού κόμβου, πρέπει να είναι γνωστές εκ των προτέρων οι μετατοπίσεις όλων των γειτονικών του. Λύνεται ένα 3×3 μη-γραμμικό σύστημα εξισώσεων, παρέχοντας το βέλτιστο σύνολο τιμών των ποσοτήτων για τη μετατόπιση του κόμβου i. Βάσει αυτού, οι κόμβοι μετατοπίζονται διαδοχικά, ξεκινώντας από τους κόμβους που βρίσκονται κοντά στο σύνορο. Το κύριο χαρακτηριστικό αυτής της μεθόδου είναι ότι απαιτούνται επαναλήψεις, σε καθεμία από τις οποίες όλοι οι εσωτερικοί κόμβοι μετατοπίζονται.

Συζευγμένη Μέθοδος

Για την αποτελεσματική εφαρμογή της Συζευγμένης Μεθόδου επίλυσης του προβλήματος, είναι απαραίτητο να εκφραστεί η αντίστοιχη συνάρτηση κόστους. Η νέα συνάρτηση λαμβάνει υπόψη την αλληλεξάρτηση μεταξύ όλων των κόμβων του πλέγματος. Η *Συνολική Συνάρτηση Κόστους* γίνεται:

$$F_{total} \;=\; \sum_{i\in\mathcal{I}} F_i \tag{3}$$

Στην περίπτωση εφαρμογής της συζευγμένης μεθόδου, οι αντίστοιχες εξισώσεις για όλους τους εσωτερικούς κόμβους συνθέτουν ένα σύστημα εξισώσεων, το οποίο, όταν επιλυθεί, παρέχει το βέλτιστο σύνολο τιμών των ποσοτήτων για την ταυτόχρονη μετατόπιση όλων των κόμβων.

Γραμμικοποίηση του Προβλήματος

Ακόμη, παρουσιάζεται μια προσέγγιση με στόχο την αντικατάσταση του μη-γραμμικού συστήματος με το γραμμικό αντίστοιχο. Αυτή η αντικατάσταση εξυπηρετεί τον σκοπό της μείωσης του υπολογιστικού κόστους της επίλυσης, βασίζεται δε στην παρατήρηση μικρών μετατοπίσεων των κόμβων του πλέγματος σε πραγματικές εφαρμογές. Ως εκ τούτου, είναι λογική η υπόθεση πως η γωνία περιστροφής τείνει προς το μηδέν,

$$\theta \to 0 \implies \begin{cases} \sin\theta \to \theta \\ \cos\theta \to 1 \end{cases} \tag{4}$$

Μέθοδος Τμηματικής Μετατόπισης

Για να είναι εφικτή η χρήση των γραμμικοποιημένων εξισώσεων σε εφαρμογές μεγαλύτερων μετατοπίσεων του πλέγματος, αναπτύσσεται η Μέθοδος Τμηματικής Μετατόπισης. Σύμφωνα με αυτήν τη μέθοδο, οι κόμβοι μετατοπίζονται σταδιακά, διασφαλίζοντας ότι κάθε βήμα προκαλεί μια μικρή μετατόπιση, τέτοια ώστε η υπόθεση μικρής γωνίας περιστροφής να είναι έγκυρη. Παρόλο που είναι εφαρμόσιμη σε πραγματικά σενάρια, αυτή η στρατηγική αυξάνει το συνολικό υπολογιστικό κόστος, καθώς απαιτεί την επίλυση ενός συστήματος εξισώσεων σε κάθε βήμα.

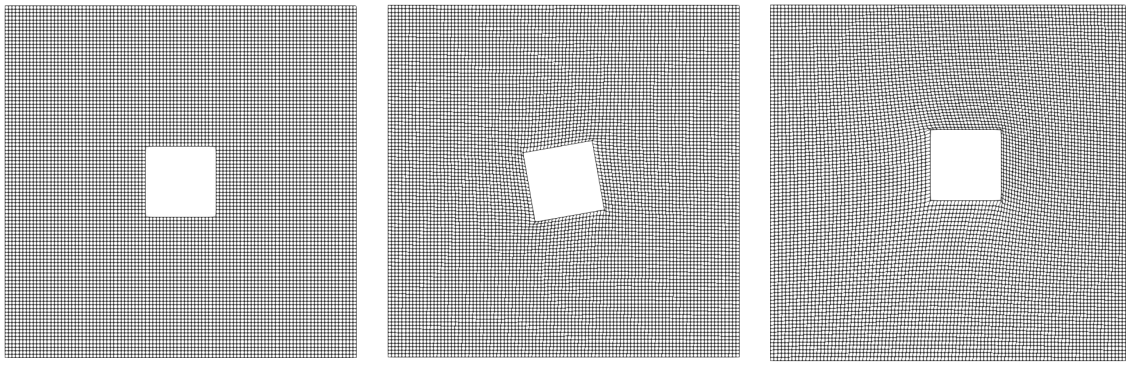
Παραμόρφωση 2Δ Πλεγμάτων: Εφαρμογές

Ομόκεντρα Τετράγωνα

Η πρώτη εφαρμογή της μεθόδου ΚΑΣ σχετίζεται με την παραμόρφωση δύο ομόκεντρων τετραγώνων. Εφαρμόζεται αρχικά μια ανθωρολογιακή περιστροφή του εσωτερικού τετραγώνου κατά 10 μοίρες γύρω από το κέντρο του. Όταν χρησιμοποιείται η μη-γραμμική συζευγμένη μέθοδος, δεν παρουσιάζονται ανεστραμμένα κελιά. Ωστόσο, όταν εφαρμόζονται οι γραμμικοποιημένες εξισώσεις της συζευγμένης μεθόδου, εμφανίζονται ανεστραμμένα κελιά, επομένως απαιτείται η εφαρμογή της μεθόδου τμηματικής μετατόπισης. Η συνολική επιβληθείσα μετατόπιση διαιρείται σε 10 βήματα και τα αποτελέσματα είναι ικανοποιητικά, καθώς το πλέγμα δεν εμφανίζει ανεστραμμένα κελιά.

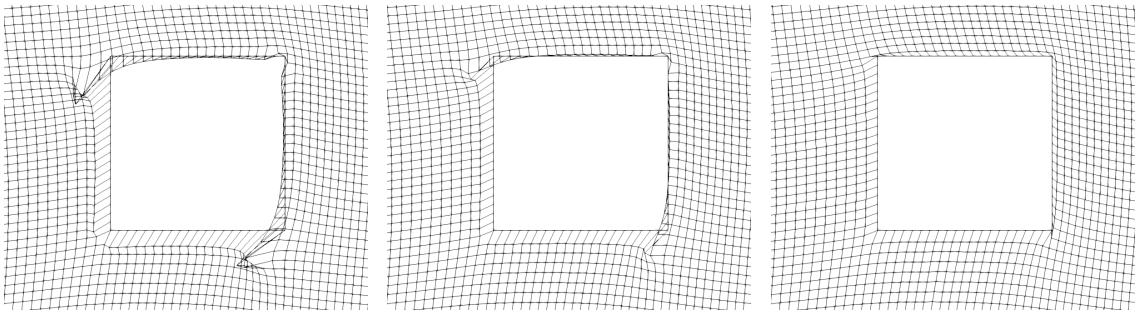
Στη συνέχεια, το εσωτερικό τετράγωνο μετατοπίζεται σε μια νέα θέση χωρίς επιβολή περιστροφής. Η εφαρμογή των γραμμικοποιημένων εξισώσεων της συζευγμένης μεθόδου οδηγεί σε αποτελέσματα ακατάλληλα για χρήση, για παράδειγμα, από έναν κώδικα ΥΡΔ. Γίνεται προφανές ότι ακόμη και σε περιπτώσεις όπου η επιβληθείσα με-

τατόπιση των συνόρων δεν περιλαμβάνει περιστροφή, καθώς οι εξισώσεις προς επίλυση αντιλαμβάνονται τη γωνία περιστροφής για κάθε κόμβο ως μία μεταβλητή, η τελική λύση μπορεί να συμπεριλαμβάνει μη μηδενικές τιμές της γωνίας περιστροφής. Με τη χρήση 20 βημάτων, η ποιότητα του πλέγματος βελτιώνεται και τα ανεστραμμένα κελιά εξαφανίζονται. Ωστόσο, αυτή η βελτίωση συνεπάγεται μεγαλύτερο υπολογιστικό κόστος.



(α) *(β)* *(γ)*

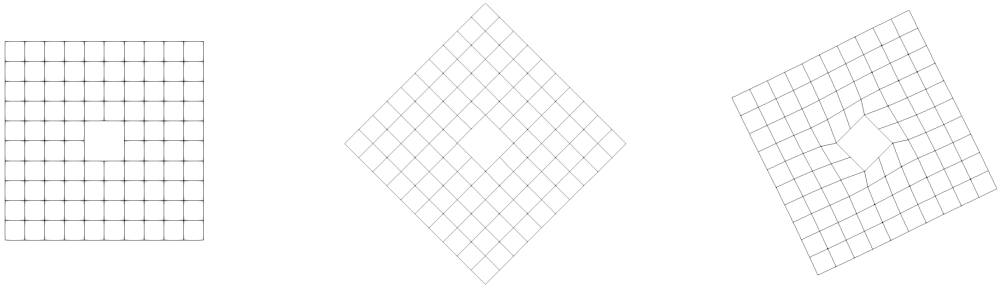
Σχήμα 2: *Ομόκεντρα Τετράγωνα. (α): Αρχικό πλέγμα. (β): Τελικό πλέγμα μετά την περιστροφή του εσωτερικού τετραγώνου. (γ): Τελικό πλέγμα μετά τη μεταφορά του εσωτερικού τετραγώνου.*



(1 βήμα) *(2 βήματα)* *(20 βήματα)*

Σχήμα 3: *Ομόκεντρα Τετράγωνα. Εφαρμογή της Μεθόδου Τμηματικής Μετατόπισης.*

Στη συνέχεια, μελετάται ένα διαφορετικό σενάριο κατά το οποίο το εσωτερικό τετράγωνο περιστρέφεται κατά 45 μοίρες ανθωρολογιακά γύρω από το κέντρο του. Το ιδιαίτερο χαρακτηριστικό αυτού του παραδείγματος είναι ότι οι κόμβοι που σχηματίζουν το εξωτερικό τετράγωνο είναι ελεύθεροι να μετατοπιστούν. Κάθε μέθοδος παράγει ένα διαφορετικό τελικό πλέγμα. Η συζευγμένη μέθοδος φαίνεται να αντιλαμβάνεται το πλαίσιο της αλληλεξάρτησης μεταξύ όλων των κόμβων του πλέγματος. Αντίθετα, η αποσυζευγμένη μέθοδος φαίνεται να υιοθετεί μια πιο τοπική προσέγγιση, προσαρμόζοντας κυρίως τους κόμβους κοντά στα σύνορα και αφήνοντας τους πιο απομακρυσμένους σχετικά ανεπηρέαστους. Αυτή η διαφοροποίηση αντικατοπτρίζει τις διακριτές διατυπώσεις της συνάρτησης κόστους στις δύο μεθόδους.

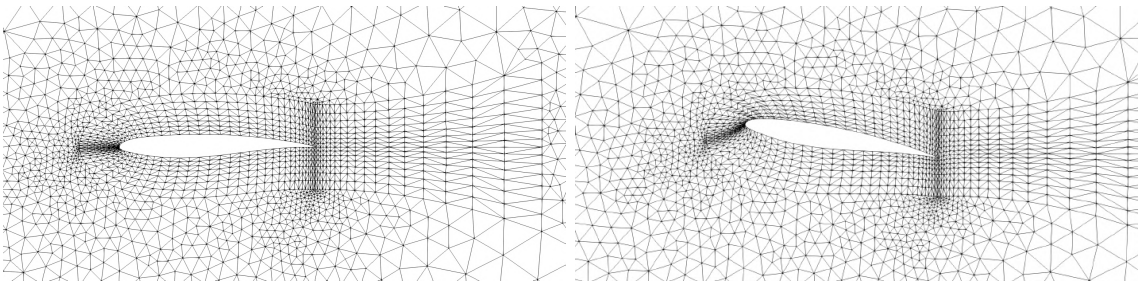


(α) *(β)* *(γ)*

Σχήμα 4: *Ομόκεντρα Τετράγωνα. (α): Αρχικό πλέγμα. (β): Αποτέλεσμα της Συζευγμένης Μεθόδου. (γ): Αποτέλεσμα της Αποσυζευγμένης Μεθόδου.*

Μεμονωμένη Αεροτομή

Για τη συγκεκριμένη εφαρμογή, μελετήθηκε η περιστροφή μιας μεμονωμένης αεροτομής. Παρατηρείται ότι, όταν εφαρμόζεται περιστροφή 5 μοιρών, η μετατόπιση του πλέγματος συμβαίνει ομαλά, χωρίς την εμφάνιση ανεστραμμένων κελιών. Ωστόσο, όταν εφαρμόζεται μια μεγαλύτερη γωνία περιστροφής ίση με 10 μοίρες, η συζευγμένη μέθοδος αντιμετωπίζει δυσκολία στο να επιτύχει μια αποδεκτή μετατόπιση του πλέγματος. Ως αποτέλεσμα, χρησιμοποιείται η μέθοδος τμηματικής μετατόπισης, προσφέροντας τελικά ένα ικανοποιητικό τελικό πλέγμα. Επίσης, εξερευνήθηκε η χρήση των μη-γραμμικών εξισώσεων, οδηγώντας σε μετρικές υψηλής ποιότητας, αλλά απαιτώντας περισσότερο υπολογιστικό χρόνο. Συνεπώς, η εφαρμογή των γραμμικοποιημένων εξισώσεων της συζευγμένης μεθόδου επιδεικνύει αυξημένη αποδοτικότητα σε αυτήν την περίπτωση, διασφαλίζοντας ταχύτερη σύγκλιση σε σύγκριση τόσο με τη μη-γραμμική συζευγμένη όσο και την αποσυζευγμένη μέθοδο.



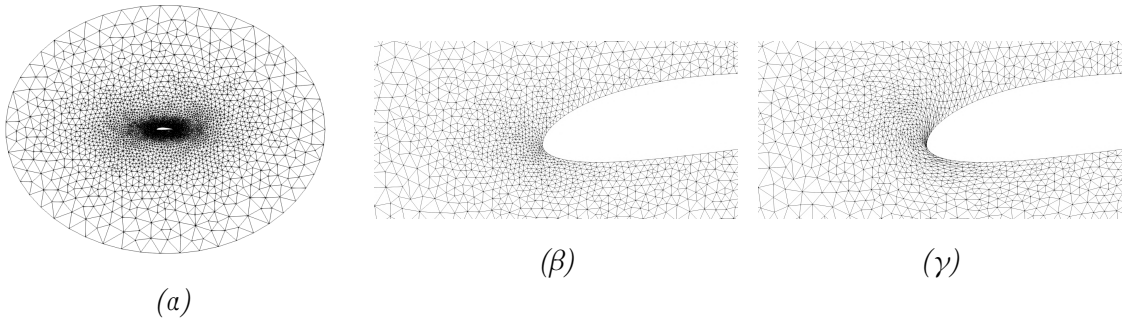
(α) *(β)*

Σχήμα 5: *Μεμονωμένη Αεροτομή. (α): Αρχικό πλέγμα. (β): Περιστροφή της αεροτομής κατά 10 μοίρες.*

Αεροτομή NACA4415

Η μελέτη αυτής της περίπτωσης στοχεύει στο να επιδείξει τα διαφορετικά αποτελέσματα της συζευγμένης και αποσυζευγμένης μεθόδου. Για να αξιολογηθεί η τεχνική ΚΑΣ,

η αεροτομή υπόκειται σε ανθωρολογιακή περιστροφή 4 μοιρών. Εστιάζοντας στην ακμή πρόσπτωσης, γίνεται εμφανές ότι η συζευγμένη μέθοδος μετατοπίζει τους κόμβους διαθέτοντας μια ευρύτερη αντίληψη ολόκληρου του πλέγματος. Αντίθετα, η αποσυζευγμένη μέθοδος δίνει μεγαλύτερη έμφαση στην μετατόπιση των κόμβων κοντά στα όρια της αεροτομής. Με αυτόν τον τρόπο, επιβεβαιώνεται η διαφοροποίηση μεταξύ των αποτελεσμάτων που προκύπτουν από τη συζευγμένη και την αποσυζευγμένη μέθοδο.

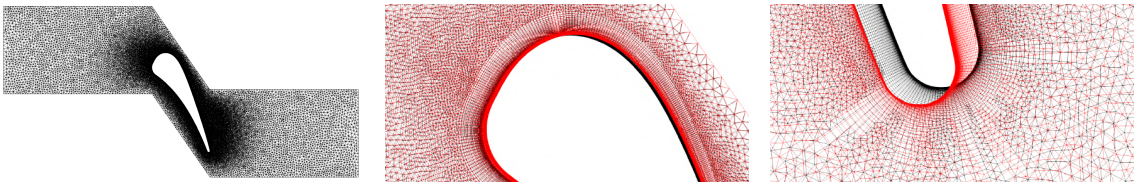


(α) *(β)* *(γ)*

Σχήμα 6: *Αεροτομή NACA4415. (α): Αρχικό πλέγμα. (β): Αποτέλεσμα της Συζευγμένης Μεθόδου. (γ): Αποτέλεσμα της Αποσυζευγμένης Μεθόδου.*

2Δ Πτερύγωση Στροβίλου

Έπειτα, το πλέγμα γύρω από τη 2Δ πτερύγωση του στροβίλου C3X χρησιμοποιείται για την αξιολόγηση της συζευγμένης μεθόδου ΚΑΣ στη διαχείριση πυκνών πλεγμάτων. Επιπρόσθετα, αυτή η περίπτωση χρησιμεύει για τον έλεγχο της ικανότητας της μεθόδου ως προς την επίλυση ενός σημαντικά μεγαλύτερου συστήματος εξισώσεων. Χρησιμοποιείται η γραμμικοποιημένη διατύπωση των εξισώσεων για να μετριαστούν οι ήδη υψηλές υπολογιστικές απαιτήσεις. Τα αποτελέσματα δείχνουν ότι η μέθοδος διατηρεί επαρκώς υψηλά επίπεδα πυκνότητας κοντά στον τοίχο του πτερυγίου, χωρίς να εμφανίζονται ανεστραμμένα κελιά στο τελικό πλέγμα.



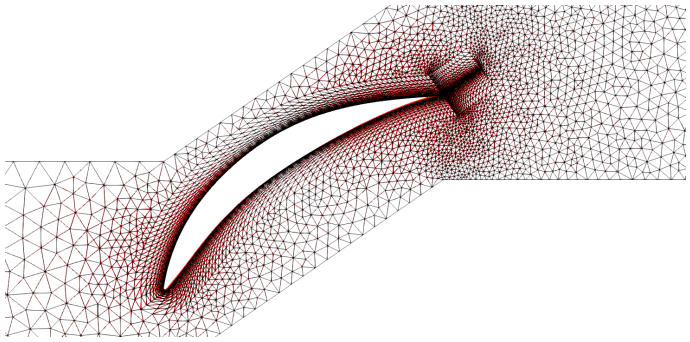
(α) *(β)* *(γ)*

Σχήμα 7: *2Δ Πτερύγωση Στροβίλου. (α): Αρχικό πλέγμα. (β): Εστίαση στην ακμή πρόσπτωσης. (γ): Εστίαση στην ακμή εκφυγής.*

2Δ Πτερύγωση Συμπιεστή

Σε αυτήν την εφαρμογή, αντί της ομοιόμορφης αντιμετώπισης του αεροδυναμικού σώματος μετατοπίζοντας με το ίδιο σύνολο παραμέτρων $(\Delta x, \Delta y, \theta)$ όλους τους κόμβους που

το σχηματίζουν, διάφορα τμήματα του σώματος υπόκεινται σε διαφορετική μετατόπιση. Η καμπυλότητα της αεροτομής του πτερυγίου ενός συμπιεστή μειώνεται ανασηκώνοντας τόσο την ακμή πρόσπτωσης όσο και την ακμή εκφυγής. Η λειτουργία της μεθόδου είναι αποτελεσματική, παράγοντας ένα τελικό πλέγμα που διατηρεί υψηλή πυκνότητα σε κρίσιμες περιοχές και είναι απαλλαγμένο από ανεστραμμένα κελιά.



Σχήμα 8: *2Δ Πτερύγωση Συμπιεστή. Μεταβολή της καμπυλότητας της αεροτομής του πτερυγίου.*

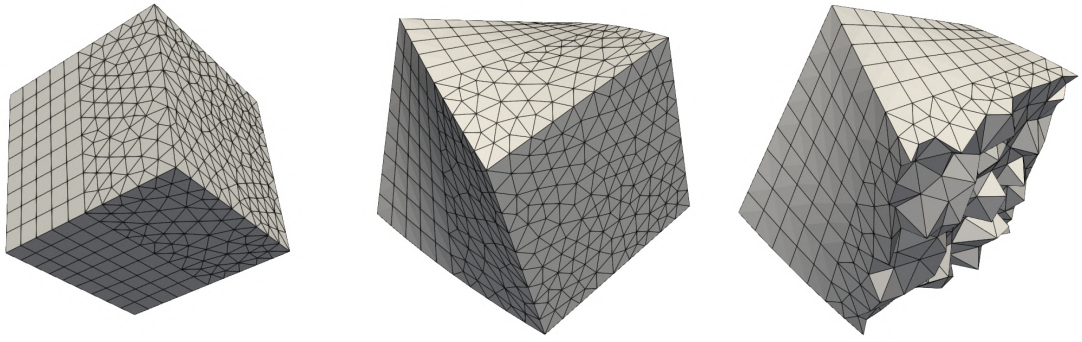
Παραμόρφωση 3Δ Πλεγμάτων: Εφαρμογές

Παραμόρφωση Κύβου

Η πρώτη 3Δ περίπτωση που εξετάζεται αφορά στην παραμόρφωση ενός κύβου, επιλεγμένου λόγω της γεωμετρικής του απλότητας, αλλά και της ποικιλομορφίας των στοιχείων που περιλαμβάνει. Καθώς αποτελείται από ένα μίγμα τετραέδρων, πυραμίδων, πρισμάτων και εξαπλεύρων, αυτή η επιλογή στοχεύει στην παρουσίαση του αποτελεσματικού χειρισμού των διαφόρων τύπων στοιχείων από τη μέθοδο ΚΑΣ. Ο κύβος παραμορφώνεται υπό μία στρέβλωση γύρω από τον άξονα y. Παρόλο που χρησιμοποιούνται οι γραμμικοποιημένες εξισώσεις υποβοηθούμενες από τη μέθοδο τμηματικής μετατόπισης, η διαδικασία ολοκληρώνεται εντός 48 δευτερολέπτων, επιτυγχάνοντας γρήγορη σύγκλιση, δεδομένου ότι η εκτέλεση του κώδικα πραγματοποιήθηκε σε έναν επεξεργαστή ενός προσωπικού υπολογιστή.

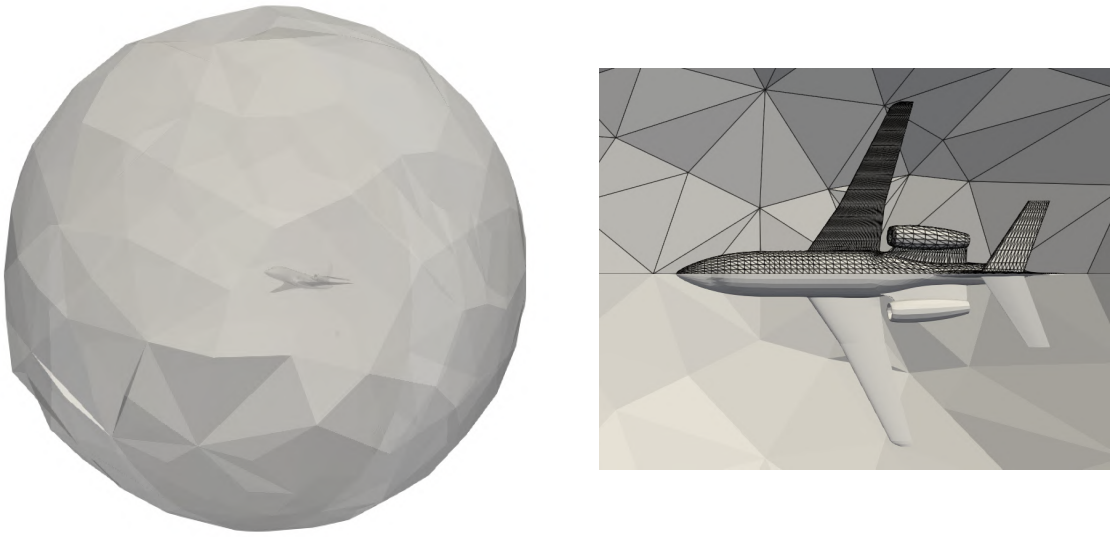
Περιστροφή Αεροσκάφους

Η επόμενη περίπτωση επικεντρώνεται στην κλίση ενός μικρού επιβατικού αεροσκάφους. Γύρω από το αεροσκάφος, το υπολογιστικό πλέγμα εμφανίζει ιδιαίτερες γεωμετρικές λεπτομέρειες, αποτελώντας μια σημαντική πρόκληση για την προσαρμοστικότητα της μεθόδου ΚΑΣ σε πολύπλοκες δομές πλέγματος. Για αυτήν τη δοκιμή, το αεροσκάφος υφίσταται μια κλίση 3 μοιρών γύρω από τον διαμήκη άξονά του, προσομοιώνοντας την έναρξη μιας κατάβασης. Το τελικό πλέγμα προκύπτει μέσω της εφαρμογής των γραμμικοποιημένων εξισώσεων της συζευγμένης μεθόδου, επιδεικνύοντας την ικανότητα της να διατηρεί την ποιότητα και την πυκνότητα του πλέγματος σε κρίσιμες περιοχές.



(α) *(β)* *(γ)*

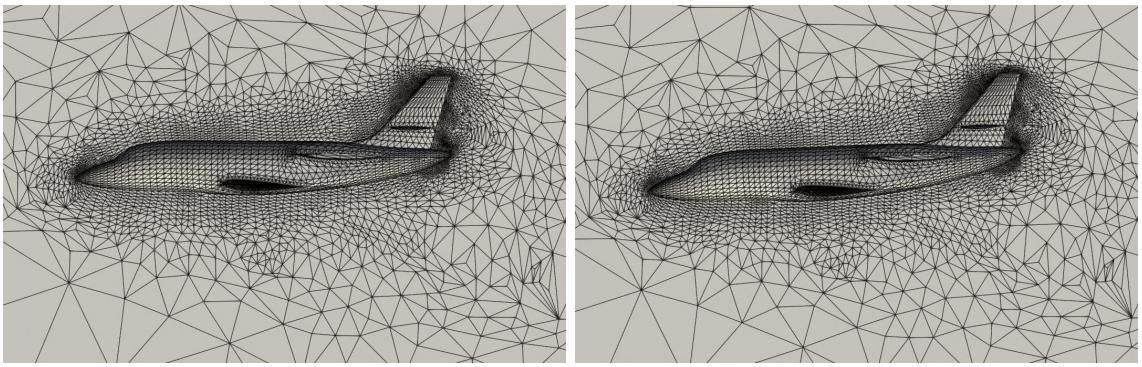
Σχήμα 9: *Στρέβλωση κύβου. (α): Αρχικό πλέγμα. (β): Τελικό πλέγμα. (γ): Εσωτερική διαμόρφωση.*



Σχήμα 10: *Αρχικό πλέγμα γύρω από αεροσκάφος.*

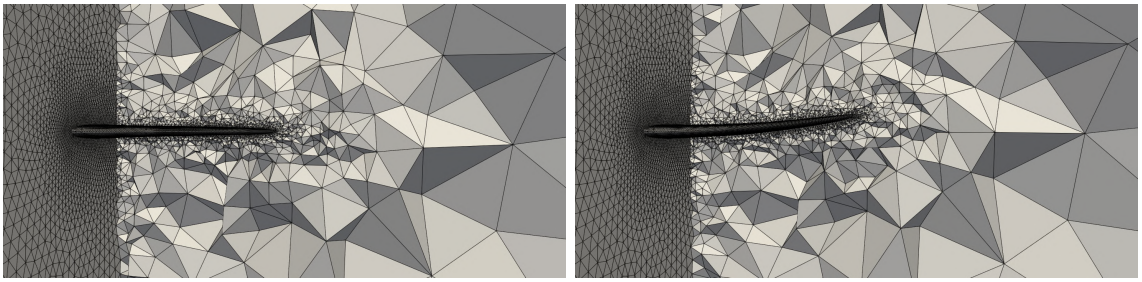
Κάμψη Πτέρυγας

Με στόχο την προσομοίωση μιας διαδικασίας παραμόρφωσης σχήματος στις τρεις διαστάσεις, χρησιμοποιείται ένα υπολογιστικό πλέγμα γύρω από την πτέρυγα ONERA M6. Η σημαντική αύξηση του αριθμού των κόμβων παρέχει μια ευκαιρία αξιολόγησης της απόδοσης της μεθόδου σε μεγάλα 3Δ πλέγματα. Για αυτήν τη δοκιμή, η πτέρυγα υποβάλλεται σε κάμψη. Η μέθοδος ΚΑΣ επιδεικνύει εξαιρετική απόδοση στη διαχείριση της μετατόπισης αυτού του πλέγματος, αποτρέποντας αποτελεσματικά το σχηματισμό ανεστραμμένων κελιών και διατηρώντας την ποιότητα και την πυκνότητα του αρχικού πλέγματος.



(α) *(β)*

Σχήμα 11: *Κλίση αεροσκάφους. (α): Αρχικό πλέγμα. (β): Τελικό πλέγμα.*



(α) *(β)*

Σχήμα 12: *Κάμψη πτέρυγας. (α): Αρχικό πλέγμα. (β): Τελικό πλέγμα.*

Ανακεφαλαίωση - Συμπεράσματα

Η ανάλυση των αποτελεσμάτων αποκαλύπτει ότι η χρήση των γραμμικοποιημένων εξισώσεων της συζευγμένης μεθόδου αποδεικνύεται ως η ταχύτερη για την επίτευξη σύγκλισης. Αξίζει να σημειωθεί ότι και η συζευγμένη και η αποσυζευγμένη μέθοδος, συμπεριλαμβανομένων των μη-γραμμικών παραλλαγών τους, ενδέχεται να αντιμετωπίσουν πρόβλημα εμφάνισης ανεστραμμένων κελιών όταν τα όρια του πλέγματος υπόκεινται σε σημαντικές παραμορφώσεις. Για να αντιμετωπιστεί αυτό, κρίνεται απαραίτητη η ενσωμάτωση της μεθόδου τμηματικής μετατόπισης. Συνοψίζοντας, η νέα προσέγγιση της τεχνικής ΚΑΣ για την πεπλεγμένη επίλυση του συστήματος των συζευγμένων εξισώσεων που καθορίζουν τη μετατόπιση κάθε κόμβου αναδεικνύεται ως ένα αξιόπιστο και αποτελεσματικό εργαλείο για την προσαρμογή του πλέγματος.

Bibliography

[1] Lange, K.: *Optimization*. Springer Science and Business Media, 2013.

[2] Pierre, D. A.: *Optimization Theory with Applications*. Dover Publications, INC., New York, 1986.

[3] Chong, E. K. P., Lu, W. S., and Zak, S. H.: *An Introduction to Optimization: With Applications to Machine Learning*. John Wiley & Sons, 2023.

[4] Fang, S. and Wang, H.: *Basics for Optimization Problem*. Springer Science and Business Media, 2021.

[5] Cavazzuti, M.: *Deterministic Optimization*. Springer Science and Business Media, 2012.

[6] Papoutsis-Kiachagias, E. M., Giannakoglou, K. C., and Papadimitriou, D. I.: *Discrete and continuous adjoint methods in aerodynamic robust design problems*. CFD and Optimization, ECCOMAS Thematic Conference, Turkey, 2011.

[7] Giannakoglou, K. C.: *Deterministic and Stochastic Optimization Methods and Applications*. National Technical University of Athens (NTUA), 2012.

[8] *Evolutionary Algorithms SYstem (EASY)*. http://147.102.55.162/EASY/.

[9] Giannakoglou, K. C., Papadimitriou, D. I., and Kampolis, I. C.: *Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels*. Computer Methods in Applied Mechanics and Engineering, 195(44-47):6312-6329, 2006.

[10] Giannakoglou, K. C., Asouti, V. G., Kampolis, I. C., and Zymaris, A. S.: *Multilevel optimization strategies based on metamodel-assisted evolutionary algorithms, for computationally expensive problems*. IEEE Congress on Evolutionary Computation, Singapore, p. 4116-4123, doi: 10.1109/CEC.2007.4425008., 2007.

[11] Habashi, F.: *Computational Fluid Dynamics Techniques*. Gordon and Breach Publishers, 1995.

[12] Giannakoglou, K. C.: *Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence.* Progress in Aerospace Sciences, 38(1):43-76, 2002.

[13] Giannakoglou, K. C.: *Optimization Methods in Aerodynamics.* National Technical University of Athens (NTUA), 2006.

[14] Tsiakas, K., Roge, G., Amoignon, O., Trompoukis, X., Asouti, V., Quach, B., Kleinveld, S., and Giannakoglou, K. C.: *Coupled adjoint fluid-structure interaction technique for flexible wing shape optimization.* ECCOMAS Virtual Congress, 2021.

[15] Kapsoulis, D. H., Asouti, V. G., Giannakoglou, K. C., Porziani, S., Costa, E., Groth, C., Cella, U., and Biancolini, M. E.: *Evolutionary aerodynamic shape optimization through the RBF4AERO platfrom.* ECCOMAS Congress, VII European Congress on Computational Methods in Applied Sciences and Engineering, 2016.

[16] Selim, M. M. and Koomullil, R. P.: *Mesh deformation approaches - a survey.* Journal of Phsical Mathematics 7:2, 2016.

[17] Burg, C.: *Analytic study of 2D and 3D grid motion using modified Laplacian.* International Journal for Numerical Methods in Fluids, 52(2):163-197, 2006.

[18] Boer, A. De, Schoot, M. S. Van der, and Bijl, H.: *Mesh deformation based on radial basis function interpolation.* Computers Structures, 85(11-14):784-795, 2007.

[19] Biancolini, M. E., Viola, I. M., and Riotte, M.: *Sails trim optimization using CFD and RBF mesh morphing.* Computers & Fluids, 93:46-60, 2014.

[20] Sederberg, T. W. and Parry, S. R.: *Free-form deformation of solid geometric models.* Proceedings of the 13th annual conference on Computer graphics and interactive techniques, p. 151-160, 1986.

[21] Mavronikola, M.: *Shape parameterization and grid adaptation using harmonic coordinates. Application to the aerodynamic optimization of 2D cascades and ducts.*, Diploma Thesis, School of Mechanical Engineering, NTUA, 2017.

[22] Blom, F. J.: *Considerations on the spring analogy.* International Journal for Numerical Methods in Fluids, 32(6):647-668, 2000.

[23] Farhat, C., Degand, C., Koobus, B., and Lesoinne, M.: *Torsional springs for two-dimensional dynamic unstructured fluid meshes.* Computer Methods in Applied Mechanics and Engineering, 163(1-4):231-245, 1998.

[24] Farhat, C. and Degand, C.: *A three-dimensional torsional spring analogy method for unstructured dynamic meshes.* Computer Methods in Applied Mechanics and Engineering, 80(3-4):305-316, 2002.

[25] Lynch, D.: *Elastic grid deformation for moving boundary problems in two space dimensions.* 3rd International Conference, Mississipi University, Oxford, 1980.

[26] Johnson, A. A. and Tezduyar, T. E.: *Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces.* Computer Methods in Applied Mechanics and Engineering, 119(1-2):73-94, 1994.

[27] Hahn, H.: *Rigid Body Dynamic of Mechanisms: 1 Theoretical Basis.* Springer, 2002.

[28] *ParaView*. https://www.paraview.org/.

[29] Kontos, A.: *Adaptive deformation of 2D/3D structured and unstructured grids using the motion pattern of the rigid body.*, Diploma Thesis, School of Mechanical Engineering, NTUA, 2018.

[30] Giannakoglou, K. C., Bergeles, G., and Anagnostopoulos, J.: *Numerical Analysis for Engineers.* National Technical University of Athens (NTUA), 2003.

[31] Liatsikouras, A.: *Rigid Body Motion Techniques in Grid Displacement and CAD-Free Shape Parameterization, in Aerodynamic Optimization.* PhD thesis, School of Mechanical Engineering, NTUA and ESI Group, 2023.

[32] Georgopoulos, V.: *Programming of a mesh displacement method using radial basis functions and quaternions.*, Diploma Thesis, School of Mechanical Engineering, NTUA, 2021.

[33] Frey, P. J. and Borouchaki, H.: *Surface mesh quality evaluation.* INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING, Int. J. Numer. Meth. Engng. 45, 101–118, 1999.

[34] Knupp, P. M.: *Remarks on mesh quality.* 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 2007.

[35] Tsolovikos, A.: *Deformation of computational meshes using delaunay graph parameterization - applications in the adjoint-based aerodynamic shape optimization.*, Diploma Thesis, School of Mechanical Engineering, NTUA, 2018.

[36] Liatsikouras, A.: *Programming of a mesh displacement method using radial basis functions and a preconditioner for use in aerodynamic optimization.*, Diploma Thesis, School of Mechanical Engineering, NTUA, 2015.

[37] Apostolou, P.: *2D and 3D unstructured mesh displacement-adaptation using torsional springs.*, Diploma Thesis, School of Mechanical Engineering, NTUA, 2015.

[38] Zervas, C.: *Shape parameterization and adaptive deformation of 3D computational grids using harmonic coordinates. applications in aerodynamic optimization.*, Diploma Thesis, School of Mechanical Engineering, NTUA, 2018.

[39] Knupp, P. M.: *Algebraic mesh quality metrics.* SIAM J. Sci. Comput., 23:193–218, 2001.

[40] Knupp, P. M.: *Algebraic mesh quality metrics for unstructured initial meshes.* Finite Elements in Analysis and Design, 39(3):217-241, 2013.

[41] Knupp, P. M.: *Hexahedral mesh untangling algebraic mesh quality metrics.* Parallel Computing Sciences Department, Sandia National Laboratories, M/S 0847, P.O. Box 5800, Albuquerque, NM 87185-0847, pknupp@sandia.gov, 2000.

[42] Liu, A. and Joe, B.: *Relationship betwen tetrahedron shape measures.* BIT Numerical Mathematics, 34(2):268-287, 1994.

[43] Munson, T.: *Mesh shape-quality optimization using the inverse mean-ratio metric.* Mathematical Programming, 110(3):561-590, 2007.

[44] Dompierre, J., Guibault, P., Labbe, F., and Camerero, R.: *Proposal of benchmarks for 3D unstructured tetrahedral mesh optimization.* Proceedings of the 7th International Meshing Roundtable, Dearborn, Mischigan, USA 26-28, Sandia National Lab, pp.459-478, 1998.

[45] *OpenFOAM*. https://openfoam.org/.

[46] Afzal, A., Ansari, Z., Faizabadi, A. R., and Ramis, M. K.: *Parallelization strategies for computational fluid dynamics software: State of the art review.* Archives of Computational Methods in Engineering, 24:337-363, 2017.

[47] *Portable Extensible Toolkit for Scientific Computation (PETSc).* https://petsc.org/release/manual/.

[48] *Portable Extensible Toolkit for Scientific Computation (PETSc).* https://petsc4py.readthedocs.io/en/stable/manual/.