



*Technical University of Crete*  
*School of Production Engineering and Management*  
*Turbomachines and Fluid Dynamics Laboratory*

---

Σύγκριση των μεθόδων Ελεύθερης Παραμόρφωσης και Αρμονικών  
Συντεταγμένων για την ταυτόχρονη παραμόρφωση γεωμετρίας και  
υπολογιστικού πλέγματος σε εφαρμογές βέλτιστου αεροδυναμικού  
σχεδιασμού

**Comparing Free-Form Deformation and Harmonic Coordinates for the  
concurrent geometry and computation grid deformation in aerodynamic  
shape optimization**

---

by

**Sofia Tavla**

Diploma Thesis

Supervisor: Prof. Dr. Ioannis K. Nikolos

Chania, June 2021





*Technical University of Crete*

*School of Production Engineering and Management*

*Turbomachines and Fluid Dynamics Laboratory*

---

Σύγκριση των μεθόδων Ελεύθερης Παραμόρφωσης και Αρμονικών  
Συντεταγμένων για την ταυτόχρονη παραμόρφωση γεωμετρίας και  
υπολογιστικού πλέγματος σε εφαρμογές βέλτιστου αεροδυναμικού  
σχεδιασμού

**Comparing Free-Form Deformation and Harmonic Coordinates for the  
concurrent geometry and computation grid deformation in aerodynamic  
shape optimization**

---

by

**Sofia Tavla**

Diploma Thesis

**Approved by:**

Dr. Ioannis K. Nikolos

---

Professor, Supervisor

Dr. Argiris Delis

---

Associate Professor

Dr. Georgios Arampatzis

---

Assistant Professor

Chania, June 2021



## **Abstract**

In the field of aerodynamics, shape optimization aims to obtain high performance aerodynamic configurations by the optimization of an objective function, subject to specific geometrical constraints. Such problems include the maximization and minimization of the lift and drag forces, which act on an airfoil, respectively.

Initially, the proper selection of the deformation technique, which later on will produce the candidate geometries, is of paramount importance in the optimization process. Specifically, during shape optimization, it is crucial for the computational grid – on which nodes the flow equations (Euler & Navier-Stokes) are solved – to continuously adapt to the new geometrical entities. To this end, in recent decades several grid and shape parameterization techniques have been developed, with the common goal of minimizing both the computational cost and time required for the deformation, and at the same time, handle intricate geometries.

In the present dissertation, two of the most prevalent methods of deformation are examined; the Free Form Deformation (FFD) and the Harmonic Function-based deformation techniques. Initially, an extensive literature review of Free Form Deformation and Harmonic Functions-based deformation methodologies, used for shape parameterization and grid adaptation, is conducted. Furthermore, a modified Harmonic functions-based methodology – developed in the Turbomachinery and Fluid Dynamics Laboratory of the Technical University of Crete (TUC) – is presented in detail and tested.

Finally, a grid interpolation algorithm is developed and presented in the context of the present work. The purpose of the aforementioned algorithm is to

enable data interpolation between two computational grids with different densities, during the aerodynamic shape optimization procedure. The mesh interpolation algorithm was implemented in FORTRAN 90 programming language.

## Περίληψη

Στον τομέα της αεροδυναμικής, στόχος της βελτιστοποίησης σχήματος αποτελεί η εύρεση υψηλής απόδοσης αεροδυναμικών μορφών, μέσω της βελτιστοποίησης της τιμής μίας αντικειμενικής συνάρτησης υπό συγκεκριμένους γεωμετρικούς περιορισμούς. Τέτοια προβλήματα περιλαμβάνουν για παράδειγμα τη μεγιστοποίηση και ελαχιστοποίηση των δυνάμεων άνωσης και οπισθέλκουσας, οι οποίες δρουν σε μία αεροτομή, αντίστοιχα.

Αρχικά, σε μία εφαρμογή αεροδυναμικής βελτιστοποίησης καθοριστικής σημασίας αποτελεί η επιλογή της συγκεκριμένης τεχνικής παραμόρφωσης, μέσω της οποίας θα παρασταθούν αργότερα οι υποψήφιες γεωμετρίες. Πιο συγκεκριμένα, κατά τη διάρκεια της διαδικασίας, απαραίτητη κρίνεται η συνεχής προσαρμογή του υπολογιστικού πλέγματος - στους κόμβους του οποίου επιλύονται οι διαφορικές εξισώσεις της ροής (Euler ή Navier-Stokes) - γύρω από τις υποψήφιες γεωμετρίες. Για αυτό το σκοπό και στοχεύοντας στην ελαχιστοποίηση του υπολογιστικού κόστους και χρόνου που απαιτείται κατά τη διάρκεια της παραμόρφωσης, αλλά και στην ικανότητα διαχείρισης πολύπλοκων γεωμετριών, ποικίλες τεχνικές παραμόρφωσης πλέγματος και σχήματος αναπτύχθηκαν τις τελευταίες δεκαετίες.

Στην παρούσα διπλωματική εργασία θα παρουσιαστούν αρχικά δύο από τις επικρατέστερες μεθόδους παραμόρφωσης, της Free Form Deformation (Ελεύθερη Παραμόρφωση) και εκείνης που βασίζεται εξ' ολοκλήρου στη χρήση αρμονικών συναρτήσεων (Harmonic Functions). Ειδικότερα, στη συγκεκριμένη εργασία θα γίνουν εκτενείς βιβλιογραφικές επισκοπήσεις των προαναφερόμενων μεθόδων. Στη συνέχεια, θα παρουσιαστεί και θα ελεγχθεί μία τροποποιημένη μέθοδος Αρμονικών Συναρτήσεων, που αναπτύχθηκε στο Εργαστήριο Στροβιλομηχανών και Ρευστοδυναμικής,

της Σχολής Μηχανικών Παραγωγής και Διοίκησης του Πολυτεχνείου Κρήτης.

Τέλος, θα παρουσιαστεί και αναλυθεί ο αλγόριθμος παρεμβολής υπολογιστικών πλεγμάτων, που αναπτύχθηκε στα πλαίσια της παρούσας διπλωματικής εργασίας. Σκοπός του παραπάνω αλγορίθμου είναι η παρεμβολή δεδομένων μεταξύ δύο πλεγμάτων με διαφορετική πυκνότητα, κατά τη διάρκεια της βελτιστοποίησης αεροδυναμικού σχήματος. Ο αλγόριθμος παρεμβολής υπολογιστικών πλεγμάτων υλοποιήθηκε σε γλώσσα προγραμματισμού FORTRAN 90.

# Table of Contents

Abstract.....	v
Περίληψη .....	vii
Table of Contents .....	ix
List of Figures .....	xiii
Chapter 1 .....	1
1.1 Aerodynamic Shape Optimization .....	1
1.2 Need for grid adaptation.....	3
1.3 Grid Adaptation Methods.....	5
1.4 Literature survey on geometrical parameterization techniques in Aerodynamic Optimization.....	7
1.4.1 The Basis Vector approach .....	8
1.4.2 The Domain Element approach .....	8
1.4.3 The Partial Differential Equation (PDE) Approach .....	9
1.4.4 The Discrete Approach.....	9
1.4.5 The Polynomial and Spline Approaches .....	9
1.4.6 The CAD-Based Approach .....	12
1.4.7 The Analytical Approach.....	13
1.4.8 Free-Form Deformation (FFD) in Aerodynamic Shape Optimization.....	13
1.4.9 Multidisciplinary Aerodynamic-Structural Shape Optimization Using Deformation (MASSOUD) .....	15

1.5	The proposed approach.....	17
1.6	Literature survey on the Harmonic Functions-based parameterization techniques .....	18
1.6.1	Introduction .....	18
1.6.2	Harmonic coordinates for character articulation .....	18
1.6.3	Boundary Element Formulation of Harmonic Coordinates .....	21
1.6.4	Holomorphic coordinates .....	23
1.6.5	Biharmonic Coordinates .....	24
1.6.6	Bounded biharmonic coordinates.....	26
1.6.7	Pseudoharmonic coordinates .....	28
1.6.8	Harmonic Guidance for Surface Deformation.....	30
1.6.9	Robust One-to-One Sweeping with Harmonic S-T mapping.....	33
1.6.10	Volumetric T-Spline construction using Boolean operations.....	35
1.6.11	All hexahedral mesh generation via inside-out advancing front based on harmonic fields .....	36
1.6.12	Direct Product Volumetric parameterization of handlebodies via harmonic fields.....	39
1.6.13	Dynamic Harmonic fields for surface processing .....	43
1.6.14	Skeleton based cage guided by harmonic fields.....	45
1.6.15	Variational Harmonic Maps.....	47
1.6.16	Bounded distortion Harmonic Shape Interpolation .....	50
1.6.17	Feature-aligned harmonic volumetric mapping using MFS. ....	54
1.6.18	Shape Parameterization and Grid Adaptation using Harmonic Coordinates and their application to the Aerodynamic Design Optimization.....	57

1.6.19	Summary .....	58
Chapter 2	.....	60
2.1	Introduction to Free Form Deformation (FFD) .....	60
2.1.1	Free Form Deformation Variations .....	60
2.1.2	Free Form Deformation applications on Aerodynamic Shape Optimization.....	71
2.1.3	Free Form Deformation process.....	78
2.2	2D B-Spline FFD .....	81
2.2.1	Implementation of the Procedure.....	82
Chapter 3	.....	86
3.1	Introduction to Harmonic Functions.....	86
3.2	Harmonic Functions properties .....	88
3.2.1	The Mean-Value principle .....	88
3.2.2	The Maximum/Minimum modulus principle .....	88
3.2.3	The Uniqueness property .....	88
3.3	Dirichlet problem .....	89
3.4	B-Spline Curves .....	90
3.5	Parameterization in 2D.....	92
3.6	Deformation in 2D.....	95
3.7	Conclusion.....	98
Chapter 4	.....	99
4.1	Introduction to Mesh Interpolation .....	99
4.2	Algorithm Analysis.....	101
4.2.1	Division of the source mesh .....	101

4.2.2	Point-in-Element check and Data Interpolation.....	103
4.3	Flow chart of the proposed interpolation algorithm.....	116
4.4	Results of the algorithm .....	119
4.4.1	Triangular mesh interpolation .....	119
4.4.2	Quadrilateral mesh interpolation .....	122
4.4.3	Hybrid mesh Interpolation.....	124
4.5	Conclusion.....	128
Chapter 5	.....	129
5.1	Test Case 1: Single internal B-Spline boundary. ....	129
5.2	Test Case 2: Two internal B-Spline boundaries. ....	140
Chapter 6	.....	151
References	.....	153

## List of Figures

<b>Figure 1.1.1</b> The aerodynamic shape optimization (ASO) process.....	1
<b>Figure 1.4.1</b> The initial shape of a domain element (left) and its deformed geometry (right) (J. Samareh, 1999).....	8
<b>Figure 1.4.2</b> Airfoil representation by a 3 <sup>rd</sup> degree B – Spline curve and 7 control points (Leloudas, 2014).....	11
<b>Figure 1.4.3</b> Original and optimized blade profiles constructed by NURBS curves and their control points (Li et al., 2016).....	12
<b>Figure 1.4.4</b> Original and Deformed Foil and Corresponding FFD Control Points (Bai and Chen, 2013).....	15
<b>Figure 1.4.5</b> A deformation based on MASSOUD approach (Samareh, 1999).	16
<b>Figure 1.6.1</b> Image of Mickey Mouse computational mesh and the cage surrounded by (Mavronikola, 2017).....	19
<b>Figure 1.6.2</b> Deformation based on harmonic coordinates (Joshi et al., 2007).	20
<b>Figure 1.6.3</b> Left: Interior control cage (green). Right: Deformed facial region except the internal cage area (Joshi et al., 2007).....	21
<b>Figure 1.6.4</b> Left: Original Model. Middle and Right: Model deformations using Boundary Element Formulation of Harmonic Coordinates (Rustamov, 2008). .....	22
<b>Figure 1.6.5</b> Left: Deformation of a sphere into an ellipse in Amardillo’s stomach. Right: Exterior deformation caused by the deformation of the sphere (Rustamov, 2008).....	22
<b>Figure 1.6.6</b> A deformation based on Holomorphic coordinates parameterization. In (a) the original goldfish shape is shown, while in (b) the grid covering its geometry is presented. In (c) the deformed grid and the handle	

points used (black spots) are shown. In (d) the final geometry is displayed (Li and Liu, 2012). .....24

**Figure 1.6.7** Comparison of Biharmonic and Variational Harmonic Maps deformation (Weber et al., 2012). .....26

**Figure 1.6.8** Crocodile deformation occurred by making the use of skeleton, cage and points (Jacobson et al., 2011). .....28

**Figure 1.6.9** A comparison between barycentric coordinate function of a non – convex polygon. Left: Harmonic coordinate function. Middle: Moving Least Squares coordinate function. Right: Maximum Entropy coordinate function (Chen and Gotsman, 2016). .....30

**Figure 1.6.10** Left: A visualization of the harmonic field. Red and blue colored areas are formed due to the boundary conditions (1.32) and (1.31) set at two vertices on the tip of the left arm and bellow the middle of the trunk, respectively. Middle: The deformed object. Right: A difference in deformation propagation occurred following the increase of the vertices at both arms where the boundary condition (1.32) is set (Zayer et al., 2005). .....31

**Figure 1.6.11** The cat surface triangles are mapped on the lion in order to succeed deformation transfer. Colored areas visualize the surface parts origin. White tringles show the abscess of correspondence in the particular areas (Zayer et al., 2005). .....32

**Figure 1.6.12** (1) M1 mapping on H1 unit disk. (2) M2 mapping on H2 unit disk. (3) H1 mapping on H2 (4) H2 mapping on M2 (Cai and Tautges, 2014). .....34

**Figure 1.6.13** Left: A physical model. Right: Meshed bounded surfaces (Cai and Tautges, 2014). .....34

**Figure 1.6.14** Left: Harmonic field calculation given maximum and minimum temperature values at the highest and lowest points, respectively. Middle: Harmonic field calculation given maximum and minimum temperature values at the highest and lowest cross regions, respectively. Right: Decomposition of the torus (Liu et al., 2014). .....35

**Figure 1.6.15** (a) The boundary conditions on an elliptic cylinder. Red-colored area indicates the maximal valued vertices, while blue-colored vertices are constrained with the minimal value. (b) The distribution of the principal harmonic field over the elliptic cylindrical volume. Red-colored and blue colored areas indicate high and low scalar values, respectively. The short blue lines represent the gradient vector at the corresponding points (Li and Tong, 2012). .....38

**Figure 1.6.16** The distribution of the Radial Harmonic field. Red-colored and blue-colored areas mark high and low scalar field vectors. The streamlines of the Radial Vector Field are indicated with light blue color (Li and Tong, 2012). .....38

**Figure 1.6.17** Left: Harmonic field construction in Venus model. Right: All – hexahedral mesh generation (Li and Tong, 2012). .....39

**Figure 1.6.18** (a) The surface of the 3D model  $M$  mapped on the surface of the polycube  $P$ . (b) The partition of the polycube  $P$ . (c) The surface of the 3D model  $M$ . (d) The partition of the 3D model  $M$  resulted by the partition of the polycube  $P$  (Xia *et al.*, 2010). .....40

**Figure 1.6.19** Figures from (a) to (d) show the polycube (up) and 3D model (down) computed harmonic fields. Figure (e) shows the several iso – surfaces of harmonic fields (Xia *et al.*, 2010). .....41

**Figure 1.6.20** The generated hexahedral mesh using the Direct Product Volumetric Parameterization method (Xia *et al.*, 2010). .....42

**Figure 1.6.21** Left: Harmonic vector field computation according to the user - specified constraint vectors (red arrows). Right: Calculation of the Harmonic field after the insertion of additional constraint vectors (Xu *et al.*). .....44

**Figure 1.6.22** Deformation transfer between a source (cat) and a target (dog) surface. Handles applied on cat are indicated with red color and change throughout the procedure (Xu *et al.*, 2009). .....45

**Figure 1.6.23** Left: The harmonic field on a 3D model. Bottom Right: The intersection points on  $M$  obtain through following the integral lines (black dotted lines) of the harmonic field beginning from  $p$ . Top Right: The quadrilateral section formed by four sampled intersection points (Casti et al., 2019). .....46

**Figure 1.6.24** (a) The input of the particular technique consists of the volume and its skeleton around which the cage will be formed. (b) Bending points are selected by the user. (c) Cross – sections at bending nodes are constructed through tracing the harmonic field integral lines beginning from individual bending points. (d) The welding of the quadrilateral sections around the bending nodes forms the topology of the cage. (e) Presentation of the volume inside its created cage (Casti et al., 2019). .....47

**Figure 1.6.25** Left: Initial model’s geometry enclosed in a cage. Right: Model’s deformation using variational harmonic maps deformation method (Casti et al., 2019). .....49

**Figure 1.6.26** Left: Position and Jacobian constraints imposed in the circular area by the user. Right: The detail – preserving deformation of the marked area occurred (Ben-Chen et al., 2009). .....50

**Figure 1.6.27** Considering the input mappings for  $t = 0$  and  $t = 1$  and the source image (domain  $\Omega$ ) of the dragon, the output mappings produced by the interpolation of the metric tensor and  $\eta$  variant for  $t = 0.5$  are presented (Chien, Chen and Weber, 2016). .....54

**Figure 1.6.28** Polycubes (a) and (e) are mapped onto double torus (b) and kitten (f), respectively. Color – encoded distance field of (f), (g) are transferred under the mapping to (d), (h), respectively (Li et al., 2010). .....56

**Figure 1.6.29** (a) The surface mesh of David’s head. (b) A generated hexahedral mesh on the polycube domain. (c) The generated volumetric hexahedral mesh on David’s head. (d) The generated mesh in the interior of the volume (Li et al., 2010). .....56

**Figure 1.6.30** Above: The Mach number distribution on the initial airfoil shape. Red colored areas indicate the presence of high Mach numbers compared to the blue colored areas on which low Mach numbers were obtained. Below: The deformed airfoil along and the Mach distribution around it (Mavronikola, 2017). .....58

**Figure 2.1.1** Left: Twisting of a tapered primitive. Right: A bent, twisted and tapered primitive (Barr, 1984). .....61

**Figure 2.1.2** Disc transformation into a hand (Coquillart and Jancéne, 1991). 62

**Figure 2.1.3** (a) A single Bezier curve, a control polygon and user specified axes, (b) the object to be deformed, (c) cube mapping on each control polygon segment and (d) the initial geometry’s wrapping along the Bezier curve (Chang and Rockwood, 1994). .....63

**Figure 2.1.4** Left: A star – shaped lattice located on a disk. Middle: The deformed star – shaped lattice containing the modified disk area. Right: The final geometry of the disk (MacCracken and Joy, 1996)......64

**Figure 2.1.5** Above: The original 3D shape of a teapot. Below: The deformation of the two surfaces (denoted with green and yellow colors) induce the modification of the model’s geometry (Feng et al., 1996). .....64

**Figure 2.1.6** Local and global deformation based on DFFD technique (Faloutsos et al., 1997)......65

**Figure 2.1.7** Left: The initial shape. Right: The deformed geometry (Feng et al., 2012). .....66

**Figure 2.1.8** Left: The initial geometry of a water pitcher. Middle and Right: Deformations resulted by water pitcher’s bottom compression (Hirota et al., 2000). .....67

**Figure 2.1.9** Left: The input geometry and six control points placed on it. Middle and Right: A sequence of deformations produced by the alteration of control points’ position (Yoshizawa et al., 2002). .....67

<b>Figure 2.1.10</b> (a) The initial bounding box in which the 3D model is embedded. (b)-(e) The hierarchical process of the geometry approximation. (f) The generated parametric lattice (Ono et al., 2002). .....	68
<b>Figure 2.1.11</b> Control triangulations for head (left), profile (middle) and ear (right) (Ilic and Fua, 2002).....	69
<b>Figure 2.1.12</b> (a) The initial shape to be deformed, (b) specification of the area to be deformed, (c) construction of the initial control mesh, (d) the deformed control mesh occurred by the relocation of the blue point, (e) the deformed geometry (Kobayashi and Ootsubo, 2003). .....	69
<b>Figure 2.1.13</b> Left: The initial shape of a teapot. Middle: The teapot is embedded in the scalar field. Right: The deformed shape of the teapot (Hua and Qin, 2003).....	70
<b>Figure 2.1.14</b> Left: The Stanford bunny model cut with a cylinder tool from above. Right: The DFFD's stitching algorithm was applied in order to close the opening (Schein and Elber, 2004).....	70
<b>Figure 2.1.15</b> Left: The initial geometry. Middle left: The initial control lattice. Middle right: The deformed control lattice. Right: The resulted deformed geometry (Song and Yang, 2005). .....	71
<b>Figure 2.1.16</b> Left: The initial shape of an airfoil constructed by a surface grid is embedded in the FFD control lattice. Right: The deformation of the FFD lattice is transferred to the embedded airfoil geometry (Ronzheimer, 2002). .....	72
<b>Figure 2.1.17</b> The control lattice (black box) indicates the part of the aircraft volume to be deformed (Andreoli et al., 2003). .....	73
<b>Figure 2.1.18</b> Left: The airfoil is embedded in FFD control lattice (red colored). Right: The deformation of the control lattice is not only passed to the embedded airfoil but to the surrounding computational grid (blue colored), also (Désidéri et al., 2004).....	74
<b>Figure 2.1.19</b> Above: A trivariate volume deformation. Below: The deformed grid causes the deformation of the embedded airfoil (Samareh, 2004). .....	75

**Figure 2.1.20** The deformed wing (red colored) occurred by the application of FFD, the updated wing root junction area (green colored) following the RBF interpolation and the aircraft body surface (blue colored) which remains fixed throughout the aforementioned procedure (Liu et al., 2017). .....76

**Figure 2.1.21** Above: The initial FFD lattice (red color) and the deformed one (green color) following the application of the AP FFD. Below: The airfoil shape produced by the classical FFD (red color) and the one after the application of the AP FFD (green color) (Leloudas et al., 2018). .....77

**Figure 2.1.22** The construction of the parametric lattice. The lattice consists of an ordered set of control points indicated with white color. Red colored bars indicate the neighboring control points (Sederberg and Parry, 1986). .....78

**Figure 2.1.23** The construction of the local (parametric) coordinate system (Sederberg and Parry, 1986). .....79

**Figure 2.1.24** The modified lattice containing the deformed object (Sederberg and Parry, 1986).....80

**Figure 2.1.25** A deformed object (Sederberg and Parry, 1986).....80

**Figure 2.2.1** A piecewise polynomial curve consisted of three segments ( $C_i$ ) (Piegl and Tiller, 1995).....81

**Figure 2.2.2** Initial geometry of the embedded airfoil in the lattice (Amoiralis, 2005). .....84

**Figure 2.2.3** Deformed geometry of the embedded airfoil in the lattice (Amoiralis, 2005). .....85

**Figure 3.4.1** (a) Cubic B – Splines  $U = 0, 0, 0, 0, 14, 12, 34, 1, 1, 1, 1$ , (b) Cubic curve using the Basis Functions of (a) (Piegl and Tiller, 1995).....91

**Figure 3.5.1** The computational domain and the embedded shape of interest (airfoil) constructed by a B – Spline curve.....92

**Figure 3.5.2** The discrete solution  $h_3$  resulted for the Basis Function  $N_{3,3}$ , applied as a Dirichlet boundary condition on the B-spline boundary of the computational domain. ....94

**Figure 3.6.1** The resulted deformation (blue color) of B-Spline boundary curve and the subsequent modification of the surrounding computational grid occurred through the movement of the three control points. ....96

**Figure 3.6.2** A comparison between the deformations occurred, by the alteration of three control points, through the B – Spline and Harmonic Functions procedures. ....97

**Figure 4.1.1** In mesh interpolation problems, data from the source mesh (blue color) must be mapped to the destination mesh (red color).....99

**Figure 4.2.1** Format of input .txt file containing source mesh information....101

**Figure 4.2.2** The source and destination mesh are indicated with blue and red colors, accordingly. The destination node D is located into the source element .with A, B and C nodes.....104

**Figure 4.2.3** FunctionValues.txt format. ....104

**Figure 4.2.4** Format of the file containing all the necessary info of each element neighbors.....105

**Figure 4.2.5** Format of the file which contains destination mesh information. ....106

**Figure 4.2.6** Graphic representation of inequality (4.6). Point *C* has the maximum *Y* Cartesian coordinate ( $Y_c = Y_{s,max}$ ), while point *A* has the minimum one ( $Y_A = Y_{s,min}$ ). Additionally, point *B* has the maximum *X* Cartesian coordinate ( $X_B = X_{s,max}$ ), and point *C* has the minimum one ( $Y_C = Y_{s,min}$ ). ....107

**Figure 4.2.7** Triangular barycentric coordinates of point D inside the triangle ABC.....109

**Figure 4.2.8** Line segment AC divides the tetrahedron ABCD (blue colored) into two separate triangular areas, ABC and ACD, accordingly. The query point E (red colored) is located into ACD triangle. ....111

**Figure 4.2.9** Destination node A is located outside the source mesh indicated with blue color.....112

**Figure 4.2.10** The calculation of the individual distances between the query point and the surrounding destination nodes. Red circular points indicate the destination mesh nodes, while points marked with blue squares represent the source mesh nodes. Each black colored arrow depicts the calculated distance between the query point and the detected destination nodes.....113

**Figure 4.3.1** Flow chart of the proposed algorithm, which combines the Barycentric Coordinates (BCs) and Inverse Distance Weighting (IDW) interpolation schemes.....118

**Figure 4.4.1** The geometry of an S – Shape pipe.....119

**Figure 4.4.2** The source mesh of the S-Shape pipe is constructed by 16082 nodes and 31102 elements and is indicated with blue color. ....120

**Figure 4.4.3** The destination mesh of the examined geometry consists of 4055 nodes and 7580 elements and is colored with red. ....120

**Figure 4.4.4** The function values on source mesh nodes.....121

**Figure 4.4.5** The destination values obtained following the application of the proposed interpolation algorithm. ....121

**Figure 4.4.6** The topology of the source mesh in the middle of the S – shape pipe.....122

**Figure 4.4.7** The topology of the destination mesh in the middle of the S – shape pipe.....123

**Figure 4.4.8** The function values on the nodes of the source mesh. ....123

**Figure 4.4.9** The function values on the nodes of the destination mesh.....124

**Figure 4.4.10** The airfoil shape. ....125

**Figure 4.4.11** The topology of the source mesh. ....125

**Figure 4.4.12** The topology of the destination mesh.....126

**Figure 4.4.13** Each node of the source grid is assigned a flow velocity value. Color variation indicates the magnitude of the velocity around the examined geometry. For example, low flow velocity values are marked with blue color contrary to high flow velocity areas indicted with red color. ....127

<b>Figure 4.4.14</b> The flow velocity contour resulted from the interpolation of the values on the source grid nodes to the destination mesh. ....	127
<b>Figure 5.1.1</b> The initial shape of a smooth closed periodic curve of 2 <sup>nd</sup> degree embedded in an unstructured grid consisted of triangular elements.....	134
<b>Figure 5.1.2</b> A closer examination of the initial shape of the B-Spline curve (shown in Figure 5.1.1) as well as of the unstructured grid surrounded by..	135
<b>Figure 5.1.3</b> The solution of the Laplace equation calculated by the application of the first basis function ( $N1(i)$ ) as a Dirichlet boundary condition.....	135
<b>Figure 5.1.4</b> The solution of the Laplace equation using the third basis function ( $N3(i)$ ) as a boundary condition. ....	136
<b>Figure 5.1.5</b> The solution of the Laplace equation using the second basis function ( $N2(i)$ ) as a boundary condition. ....	136
<b>Figure 5.1.6</b> The solution of the Laplace equation for the fourth basis function ( $N4(i)$ ).....	137
<b>Figure 5.1.7</b> The initial (blue-colored) and the deformed (red-colored) unstructured grids. ....	138
<b>Figure 5.1.8</b> A more detailed examination of the initial and deformed grids, respectively. ....	138
<b>Figure 5.1.9</b> A visualization of the initial (blue-color) and deformed (red-color) grids, respectively. ....	139
<b>Figure 5.2.1</b> The initial unstructured grid containing both B-Spline curve boundaries.....	141
<b>Figure 5.2.2</b> A closer examination of the unstructured grid depicted in Figure 5.2.1.....	142
<b>Figure 5.2.3</b> The solution of the Laplace equation following by the application of the first B-Spline basis function for the first B-Spline curve. ....	143
<b>Figure 5.2.4</b> The solution of the Laplace equation for the second basis function for the fifth B-Spline curve.....	143

**Figure 5.2.5** The solution of the Laplace equation for the first B-Spline basis function around the second B-Spline curve. ....144

**Figure 5.2.6** The solution of the Laplace equation for the fifth B-Spline basis function for the second B-Spline curve. ....144

**Figure 5.2.7** A visual comparison between the initial (blue-colored) and the deformed (red-colored) grids. ....146

**Figure 5.2.8** A more detailed examination of the initial (blue-colored) and deformed (red-colored) unstructured grids of the first B-Spline curve. ....147

**Figure 5.2.9** A further closer examination of the initial (blue-colored) and deformed (red-colored) unstructured grids consisted of triangular elements, around the first B-Spline curve. ....148

**Figure 5.2.10** A detailed examination of the initial (blue-colored) and deformed (red-colored) unstructured grids around the second B-Spline boundary. ....149

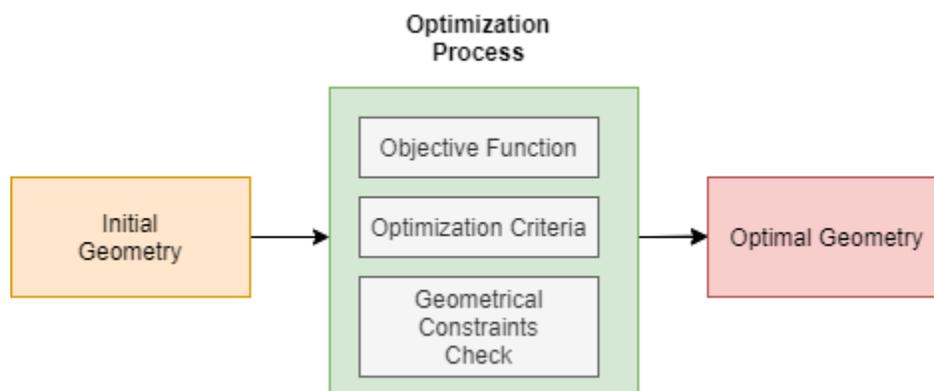
**Figure 5.2.11** A closer examination of the initial (blue-colored) and deformed (red-colored) unstructured grids around the second B-Spline boundary. ....150

# Chapter 1

## Introduction

### 1.1 Aerodynamic Shape Optimization

Aerodynamic Shape Optimization (ASO) is one of the key components of the aerodynamic design process. During the particular stage, the objective is to obtain aerodynamic configurations of high-performance, subject to several operational and geometrical constraints, which minimize or maximize the value of the objective function established by the designer. However, in many real-world applications, the design process of an aerodynamic shape is characterized by more than one potential objectives. As a result, a multi-objective optimization is performed, where the optimal solution represents the one with the least net trade-offs, for example, the lift-to-drag ratio and aircraft fuel consumption (Ren, 2016). A schematic representation of a general aerodynamic design process is provided in **Figure 1.1.1**.



**Figure 1.1.1** The aerodynamic shape optimization (ASO) process.

Furthermore, an aerodynamic design process can be characterized as either a direct or an inverse one. The aim of a direct design approach is to obtain a solution that maximizes or minimizes the value of one or many objective functions via the alteration of the geometrical parameters defining the aerodynamic configuration at hand.

On the other hand, the inverse design approach deals with the design of an aerodynamic geometry that results in a pre-defined aerodynamic characteristic, such as the target pressure distribution (Zhang et al., 2015).

Although shape optimization aims at producing the most suitable solution to a specific aerodynamic design problem, it may become a laborious and computationally expensive process. In fact, even the most recent optimization approaches depend on the user's experience and capabilities of handling the appropriate optimization tools and defining the most suitable geometry parameterization method. Moreover, by considering the required computational time to perform - in most cases - the aerodynamic analysis on the model of interest, the proper integration of required software within the general optimization scheme should be considered. More specifically, according to Wakayama and Kroo (2012), the employed methodologies must be simple enough in order to be executed as many times as required and sufficiently capture and analyze, at the same time, the local geometrical features of the entity.

## 1.2 Need for grid adaptation

In general, the aerodynamic shape optimization consists of an iterative procedure, wherein the fluid motion around each candidate solution has to be accurately calculated. Additionally, by taking into consideration that in most cases the particular fluid motion is governed by non-linear partial differential equations (Euler equations for inviscid flows and Navier-Stokes for viscous ones) that do not possess analytic solutions, their numerical approximations with appropriate flow solvers are required.

Thus, aiming to obtain the approximate solutions of the governing equations, it is necessary to apply the proper domain discretization on which the PDEs will be locally solved. The aforementioned space discretization can be either consisted of triangular or/and quadrilateral elements, or hexahedral, tetrahedral and prismatic elements (cells) on 2D or 3D, respectively. Each cell is connected with its neighboring ones regarding their topology. The overall connected domain is called a *grid*.

An important parameter of the aforementioned procedure consists of the computational time needed in order to achieve the optimal geometrical solution. The computational time involved depends on multiple factors, most important of which is the optimization method chosen. An additional key factor contributing to the total computational time reduction is the use of a grid adaptation technique throughout the shape optimization process. More specifically, in the present case, the computational grid is created once at the beginning of the procedure and is constantly adapting to the shape deformation produced at each iteration. Furthermore, the flow equation solution calculated on previous steps can be

applied as the initial flow field on those following. Thus, the convergence of the optimization problem can be greatly accelerated.

### 1.3 Grid Adaptation Methods

Given the importance of grid adaptation (mesh deformation or mesh morphing) techniques, over the latest years, several methodologies have been proposed, aiming to reduce the required computational time, while enhancing the quality of the resulting meshes. Such methodologies, which are presented in Section 1.4, are based on the RBF (Radial Basis Functions) interpolation, algebraic methods, physical analogies and partial differential equation (PDEs) methods (Gagliardi and Giannakoglou, 2019).

As long as the spring analogy methodologies are concerned, the node coordinates of the deformed mesh can be computed as a solution to the static equilibrium equations. Specifically, the deformation propagation is simulated by the motion of linear or torsional springs, which are placed on or between the cells' nodes. Therefore, for the application of the particular method, the mesh topology should be available. Nevertheless, in cases of high mesh density and large node displacements, the spring analogy method is characterized by poor performance, since negative volume production has been observed (Selim and Koomullil, 2016).

Another grid adaptation method relies on the solution of the linear elasticity equations so as to calculate the displacement of the mesh nodes that occurred. In particular, linear elasticity equations are solved in order to obtain the displacement of a given mesh. Despite the increased method applicability, high computational cost is involved (Mavronikola, 2017).

In addition, through the solution of Laplace's partial differential equation, the deformation propagation within the internal grid is succeeded. More specifically, due to the satisfaction of the minimum/maximum principle of the Laplace

equations, the displacement of interior mesh nodes are bounded by the values on the boundary of the domain (Selim and Koomullil, 2016).

In the algebraic dumping method, the updated node location is derived from algebraic equations, which are dependent on the displacement of the closest to the moving boundary nodes. Through the aforementioned methodology, a dynamic grid movement is achieved, following large-scale deformations. In spite of the robustness of the aforementioned method, the generated deformation may be rigid close to the boundary areas (Zhao and Forhad, 2003).

Finally, grid deformation propagation can be succeeded by the Radial Basis Functions (RBF) interpolation. In particular, the RBF interpolation distributes the displacement of boundary nodes to the interior nodes in relation to their distance between a set of user-specified nodes (centers). As a result, the aforementioned method is capable of producing enriched quality deformed meshes by preserving the - closest to the boundary - cells orthogonality. However, RBF interpolation is characterized by increased computational cost (Gagliardi and Giannakoglou, 2019).

## 1.4 Literature survey on geometrical parameterization techniques in Aerodynamic Optimization

Shape parameterization represents a key part of the ASO process. A parameterization technique should be flexible enough in order to describe a wide range of complicated shapes by making the use of the minimum possible number of geometrical parameters (Amoiralis and Nikolos, 2008).

According to Samareh (1999), a successful geometry parameterization method should possess a number of specific characteristics, which are provided below:

- Despite the shape changes, it should maintain the smoothness of the geometry.
- It should provide local control during the shape deformation.
- It should be able to perform properly the sensitivity analysis.
- It should be connected to the CAD system used.
- It should be intuitive enough and able to be established in a quick way.

According to Samareh (1999), several parameterization approaches have been proposed, such as: (a) the basis vector approach, (b) the domain element approach, (c) the Partial Differential Equation (PDE) approach, (d) the discrete one, (e) the polynomial and spline approach, (f) the CAD-based approach, (g) the analytical approach and (h) the Free-Form Deformation (FFD) approach. A brief introduction of the aforementioned parameterization techniques, as presented in the study of Samareh (1999), is provided in the following paragraphs.

### 1.4.1 The Basis Vector approach

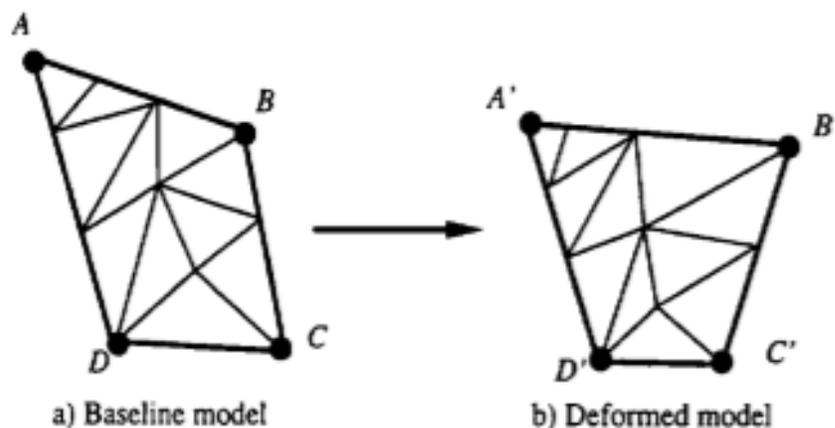
Introduced by Pickett et al. (1973), the basis vector approach is a parameterization technique through which the changes of the shape of interest are expressed by means of the following equation:

$$\mathbf{R} = \mathbf{r} + \sum_i \mathbf{u}_i \mathbf{U}_i, \quad (1.1)$$

where  $\mathbf{R}$  represents the deformed shape,  $\mathbf{r}$  is the baseline shape (initial shape),  $\mathbf{u}_i$  is the design variable and  $\mathbf{U}_i$  is the design perturbation.

### 1.4.2 The Domain Element approach

This parameterization technique is based on a linked set of grid nodes (vertices), which form a “domain element”. Initially, an inverse mapping between the mesh’s nodes and the domain element is calculated, resulting to a fixed set of parameterization coordinates (for each mesh node). Then, every movement of domain element’s vertices provoke the deformation of the entire grid, which is formed by. An application of the basis vector approach is shown in **Figure 1.4.1**.



**Figure 1.4.1** The initial shape of a domain element (left) and its deformed geometry (right) (J. Samareh, 1999).

### 1.4.3 The Partial Differential Equation (PDE) Approach

Bloor and Wilson (1995) introduced a shape parameterization methodology aimed at the construction of surfaces by making the use of partial differential equations. More specifically, posed as a boundary-value problem, parametric surfaces were generated as the solutions to elliptic partial differential equations.

### 1.4.4 The Discrete Approach

In the aforementioned parameterization approach the discrete curve boundary points act as the design variables. According to Samareh (1999), this approach is easy to be implemented; however the deformations are restricted by the number of the design variables. Additionally, the discrete approach is prone to producing non-smooth geometries and may be a non-viable alternative for shape optimization applications.

### 1.4.5 The Polynomial and Spline Approaches

Aiming at the reduction of the total number of the design variables, Braibant and Fleury (1984) proved that Bezier and B-Spline curves consist of two feasible parameterization schemes for the shape optimization problem.

An  $n^{\text{th}}$  degree Bezier curve is defined by:

$$\mathbf{C}(u) = \sum_{i=0}^n B_{i,n}(u) \mathbf{P}_i, \quad 0 \leq u \leq 1 \quad (1.2)$$

where  $B_{i,n}(u)$  are the basis functions (Bernstein polynomials) and  $\mathbf{P}_i$ ,  $i = 0, \dots, n$  are the position vectors of the control points, which in the particular case are the design variables.

Basis functions  $B_{i,n}(u)$  are the  $n^{\text{th}}$  degree Bernstein polynomials:

$$B_{i,n}(u) = \frac{n!}{i!(n-i)!} u^i (1-u)^{n-i}. \quad (1.3)$$

Although Bezier splines can reduce the number of the geometric parameters needed, they do not possess the local deformation property, due to the single segment curves they consist of. Furthermore, in order to satisfy a large number of constraints and construct a complex shape at the same time, a high-degree polynomial curve is required. A feasible solution to avoid the shortcomings of Bezier splines is the construction of multiple piecewise polynomial curves, such as B-Splines.

Constructed by several low-degree Bezier segments,  $p$  degree B-splines are defined as:

$$\mathbf{C}(\mathbf{u}) = \sum_{i=0}^n \mathbf{P}_i N_{i,p}(u), \quad u_0 \leq u \leq u_m \quad (1.4)$$

where the basis functions  $N_{i,p}(u)$ , considering that  $n = m - p$ , are given by

$$N_{i,0}(u) = \begin{cases} 1, & u_i \leq u \leq u_{i+1} \\ 0, & \text{otherwise} \end{cases} \quad (1.5)$$

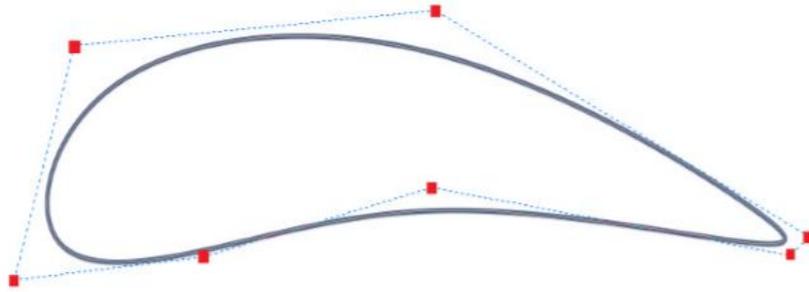
$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \quad (1.6)$$

The non-decreasing sequence  $U = [u_0, \dots, u_m]$  of  $u_i$  knots is called the *knot vector*.

In particular, the knot vector usually takes the following form

$$\mathbf{U} = \left[ \underbrace{u_0, \dots, u_0}_{p+1}, u_{p+1}, \dots, u_{m-p-1}, \underbrace{u_m, \dots, u_m}_{p+1} \right]. \quad (1.7)$$

For equally distributed knots the constructed curve is called uniform (Masters et al., 2015). An airfoil constructed by a 3<sup>rd</sup> degree B-spline curve and seven (7) control points (in red color) is represented in **Figure 1.4.2**.

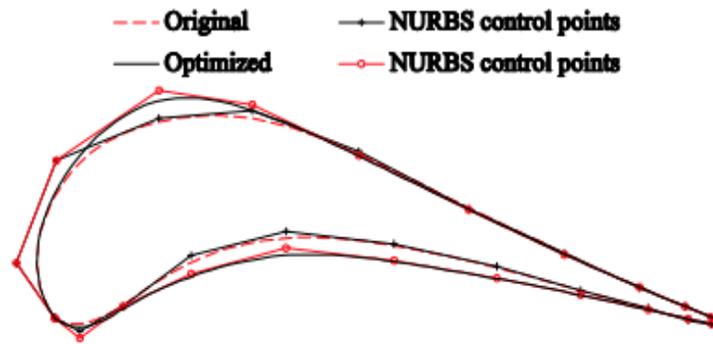


**Figure 1.4.2** Airfoil representation by a 3<sup>rd</sup> degree B – Spline curve and 7 control points (Leloudas, 2014).

Nevertheless, due to the inability of B-spline curves to represent implicit conic sections, Non-Uniform Rational B-splines (NURBS) were introduced. Specifically, a NURBS curve is described as:

$$\mathbf{C}(u) = \frac{\sum_{i=0}^n N_{i,p}(u) \mathbf{w}_i \mathbf{P}_i}{\sum_{i=0}^n N_{i,p}(u) \mathbf{w}_i}, \quad a \leq u \leq b. \quad (1.8)$$

Similarly to B-splines notation,  $\mathbf{P}_i$  represents the  $i$  –  $th$  control point,  $N_{i,p}(u)$  are the  $p$  degree B-spline basis functions and  $\mathbf{w}_i$  is the  $i$  –  $th$  non-negative control points' weight (Piegl and Tiller, 1995). An efficient NURBS based parameterization approach was introduced for the purpose of structural shape optimization. The example of the initial (red dashed line) and optimized (continuous black line) geometries of nozzle blade profiles constructed by NURBS curves and their control points are shown in **Figure 1.4.3**.



**Figure 1.4.3** Original and optimized blade profiles constructed by NURBS curves and their control points (Li et al., 2016).

### 1.4.6 The CAD-Based Approach

The majority of most recent CAD systems combine dimension-driven modelling with feature-based design. As a result, through the assignment of object dimensions as geometrical parameters, direct manipulation of the geometry of interest is offered (Hardee et al., 1999).

In particular, solid modelling CAD systems implement either a boundary representation or apply a constructive solid geometry method to represent a physical and solid object (LaCourse, 1995). However, due to possible abnormalities on the surface of the models resulted by the CAD-based approach, the application of automatic grid generation tools and the parameterization of a shape may become a demanding and laborious process. In addition, in the case of a non-recovered incompleteness of a CAD model, several challenges during the mesh construction procedure may arise (Amoiralis, 2005).

### 1.4.7 The Analytical Approach

Hicks and Henne (1978) introduced a geometry parameterization approach, based on the addition of weighted shape functions. More specifically, the airfoil shape can be represented by two analytical equations. In particular, each equation is constituted by the coordinates of the upper and lower surface baseline sections ( $y_{us_{basic}}, y_{ls_{basic}}$ ) and the corresponded weighted sum of a set of shape functions ( $f_i$ ).

$$y_{us} = y_{us_{basic}} + \sum_{i=1}^5 a_i f_i \quad (1.9)$$

$$y_{ls} = y_{ls_{basic}} + \sum_{i=1}^5 b_i f_i \quad (1.10)$$

Thus, the contribution of each shape function to the airfoil design is determined through the participation coefficients ( $a_i, b_i$ ) assignment. In the aforementioned parameterization technique the selected weighted shape functions are smooth functions based on earlier successful airfoil designs. However, despite the effectiveness of the analytical approach in wing parameterization applications, complex geometries may not be able to be represented in an accurate way (Hicks and Henne, 1978).

### 1.4.8 Free-Form Deformation (FFD) in Aerodynamic Shape Optimization

Originated by the computer graphics field, the FFD technique is based on the indirect deformation of a shape in space, through the manipulation of the control points of trivariate Bezier volumes. Later, a modified version of FFD was presented, by Lamousin and Waggenpack (1994) integrating the Non-Uniform B-splines as basis functions (NFFD). Additionally, Samareh (1999) applied the NFFD method in order to perform aerodynamic Computational Structural Mechanics

(CSM) parameterization, while Amoignon et al. (2014) investigated the influence of NURBS degree on the regularity of the optimized shapes produced by the NFFD technique.

Furthermore, Liu et al. (2017) presented a parameterization technique based on the combination of Radial Basis Function (RBF) and FFD to handle the junction area between the wing and the body of the aircraft. Specifically, following the FFD, the aim was to ensure the geometrical continuity preservation, through the RBF deformation interpolation on the junction area. As a result, according to the authors the introduced hybrid parameterization technique demonstrated effectiveness and proved to be feasible.

In addition, Bai and Chen (2013a) proposed an aerodynamic optimization scheme based on the direct manipulation of the aerodynamic shape. Through the introduction of a user-specified pilot points' location on the object geometry and their displacements, the FFD lattice associated with was accordingly modified. Thus, a direct shape manipulation is achieved. Furthermore, concerning the drag reduction optimization, the DFFD method showed improved capability compared to the original FFD technique. **Figure 1.4.4** shows a deformation according to the DFFD method.

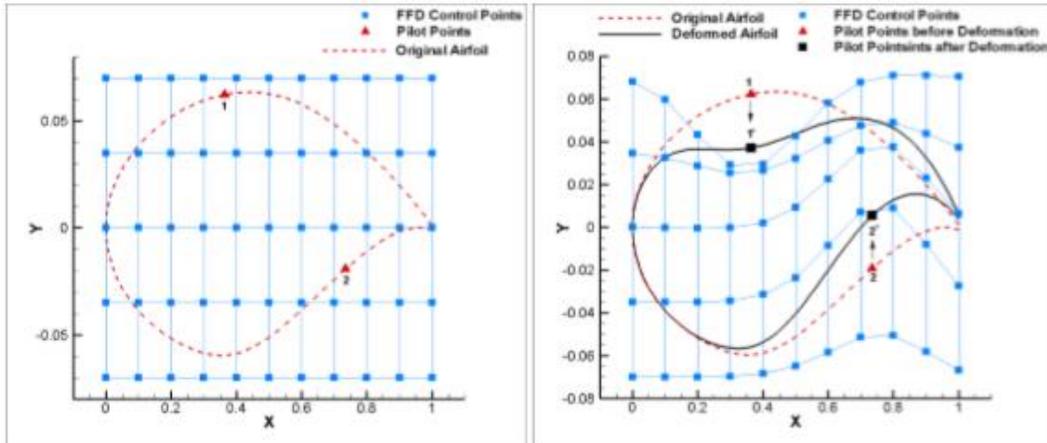


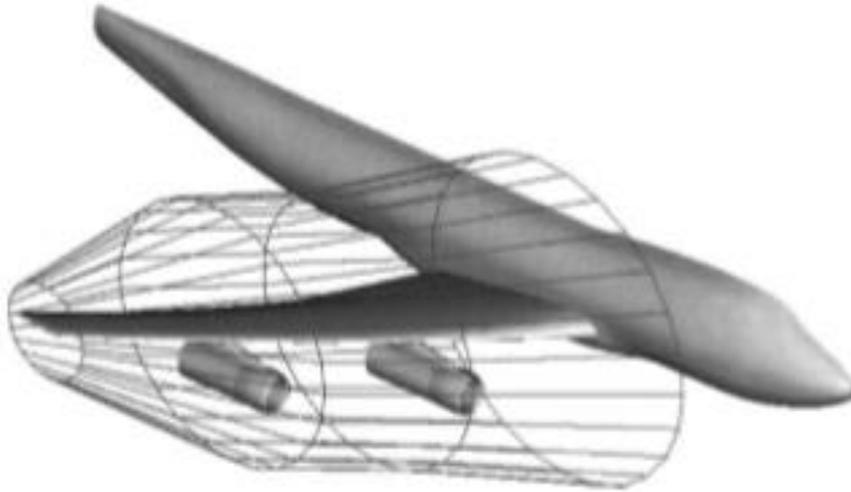
Figure 1.4.4 Original and Deformed Foil and Corresponding FFD Control Points (Bai and Chen, 2013).

### 1.4.9 Multidisciplinary Aerodynamic-Structural Shape Optimization Using Deformation (MASSOUD)

In spite of the efficient parameterization capabilities FFD method processes, the established design variables may have no significance for the design engineers. In order for the particular shortcoming to be addressed and reduce the number of the problem's geometrical parameters, Samareh (1999) introduced the Multidisciplinary Aerodynamic-Structural Shape Optimization Deformation (MASSOUD) approach. The key feature of the particular method is the parameterization of the shape perturbations (changes in thickness, camber, twist, shear and planform) occurred during the optimization, rather than the geometry itself. Furthermore, borrowed by the computer graphics field, a modified set of Soft Objects Algorithms (SOA) is applied for the purpose of subsequent model deformation through image morphing techniques. Therefore, during the optimization process, the examined surface grid is updated as follows:

$$\mathbf{R} = \mathbf{r} + \mathbf{U}(\mathbf{u}), \quad (1.11)$$

where  $\mathbf{R}$  represents the deformed grid,  $\mathbf{r}$  is the baseline grid,  $\mathbf{u}$  is the design variable vector and  $\mathbf{U}$  the design perturbation. In addition, according to Samareh (1999), by making the use of NURBS representation, strong local control and smoothness of the deformed geometry are succeeded. A deformation based on MASSOUD approach is shown in **Figure 1.4.5**.



**Figure 1.4.5** A deformation based on MASSOUD approach (Samareh, 1999).

## 1.5 The proposed approach

The aim of the present dissertation is to perform an extensive literature review of Free Form Deformation and Harmonic Functions-based deformation methodologies, used for shape parameterization, shape morphing, and grid adaptation. Then, a 2D shape deformation and computational grid adaptation technique based on the Harmonic Functions is presented for the purpose of Aerodynamic Shape Optimization.

The diploma thesis is organized as follows:

- In Chapter 1, an introduction to the Aerodynamic Shape Optimization and to the existing geometry and grid parameterization & deformation techniques is provided.
- In Chapter 2, an in-depth presentation of the Free-Form Deformation technique is introduced.
- In Chapter 3 the proposed scheme, based on the application of the B-Spline basis functions as Harmonic Functions for the concurrent and conformal deformation of the B-Spline boundary and the surrounding computation grid, is presented.
- In Chapter 4, a mesh interpolation methodology is presented and tested, as an auxiliary tool for the Harmonic Function-based parameterization.
- In Chapter 5, indicative results of the Harmonic Functions-based deformation methodology are presented and commented.
- In Chapter 6, the conclusions resulted from the application of the proposed shape deformation scheme, are outlined and discussed.

## **1.6 Literature survey on the Harmonic Functions-based parameterization techniques**

### **1.6.1 Introduction**

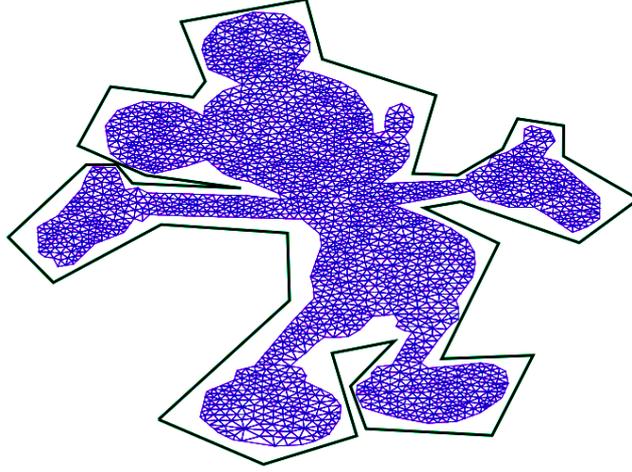
It should be emphasized that a key point in the shape optimization process is the definition of the initial geometry, by making the use of the minimum possible number of parameters (independent design variables) combined with a robust and accurate optimization algorithm.

In order to succeed the aforementioned characteristics and construct, at the same time, a parameterization technique applicable in diverse fields of studies, researchers have drawn their attention on harmonic functions and their desired properties. The parameterization techniques, analyzed in the present section were originated from the computer graphics field for the purpose of character articulation and mesh generation.

### **1.6.2 Harmonic coordinates for character articulation**

Harmonic coordinates were firstly introduced in 2005 by Pixar Animation Studios. Based on mean-value coordinates by Floater (2003), the harmonic functions applied on Closed Triangular Meshes, as presented in the work of Ju et al. (2005), included seven key parameterization properties for character articulation.

According to Joshi et al. (2007), let a polyhedron in 2D called cage  $C$ , where the volume to be deformed is embedded as shown in **Figure 1.6.1**.



**Figure 1.6.1** Image of Mickey Mouse computational mesh and the cage surrounded by (Mavronikola, 2017).

Then, for every cage vertex  $\mathbf{C}_i$ , a function  $h_i(p)$  subject to the following conditions is sought:

1. Interpolation:  $h_i(\mathbf{C}_j) = \delta_{i,j}$ , where  $\delta_{i,j} = 1$  for  $i = j$  and  $\delta_{i,j} = 0$  for  $i \neq j$
2. Smoothness: The functions  $h_i(p)$  are at least  $C^1$  smooth in the interior of the cage.
3. Non-negativity:  $h_i(p) \geq 0$  for all  $p \in C$
4. Interior locality: Minimum and maximum value of harmonic coordinates are strictly located at the cage vertices.
5. Linear reproduction: Every object point ( $p$ ) is described as a linear representation of its harmonic function and harmonic coordinates of cage vertex  $\mathbf{C}_i$ . Thus,  $\mathbf{p} = \sum_i h_i(p)\mathbf{C}_i$
6. Affine - invariance:  $\sum_i h_i(p) = 1$  for all  $p$  in  $C$
7. Strict generalization of barycentric coordinates: When  $C$  is a simplex,  $h_i(p)$  is the barycentric coordinate of  $p$  with respect to  $\mathbf{C}_i$ .

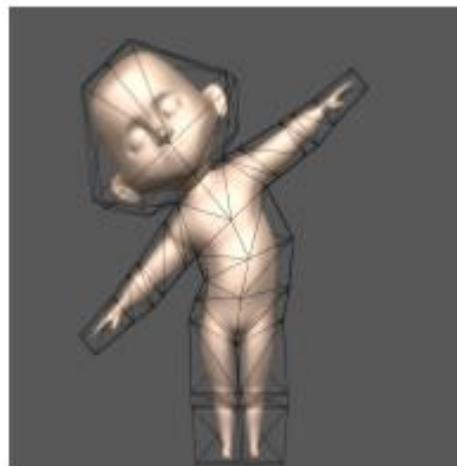
However, the described method is extended from two to 3D objects with the addition of the final property:

8. Dimension reduction:  $d$ -dimensional coordinates when restricted to a  $k < d$  dimensional facet, reduce to  $k$  - dimensional harmonic coordinates.

Coordinate functions, which satisfy the mentioned properties, are called harmonic coordinates and are obtained as solutions to the Laplace's equation:

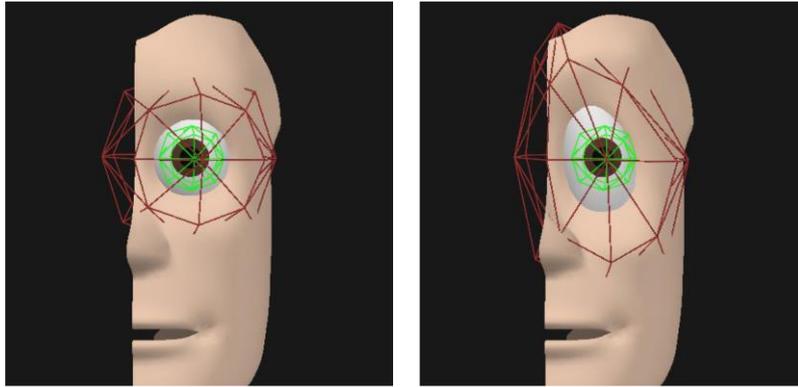
$$\nabla^2 h_i(p) = 0. \quad (1.12)$$

Accordingly, by the displacement of cage (boundary) vertices, a deformation of the initial geometry is achieved (Joshi et al., 2007). A deformation based on harmonic coordinates is shown in **Figure 1.6.2**.



**Figure 1.6.2** Deformation based on harmonic coordinates (Joshi et al., 2007).

Moreover, in an attempt of Joshi et al. (2007) to enhance the interior controllability and broaden the method's applicability, the addition of a supplementary cage encircling the areas of interest was proposed. Thus, the object parts bounded to the additional cage, maintain their initial geometrical characteristics followed by any deformation. An example of the particular method's extension is shown in **Figure 1.6.3**.

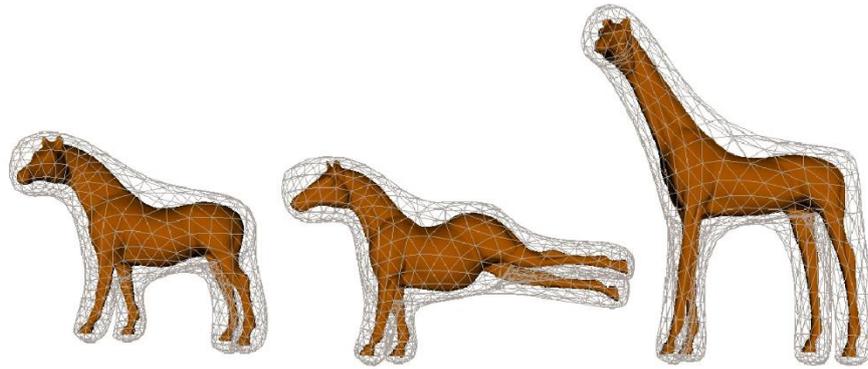


**Figure 1.6.3** Left: Interior control cage (green). Right: Deformed facial region except the internal cage area (Joshi et al., 2007).

Nevertheless, the main limitation of the discussed process is the excessive amount of computer memory required, both for pre-computing processing and storing the entire solution grid for the purpose of simultaneous geometry and cage deformation.

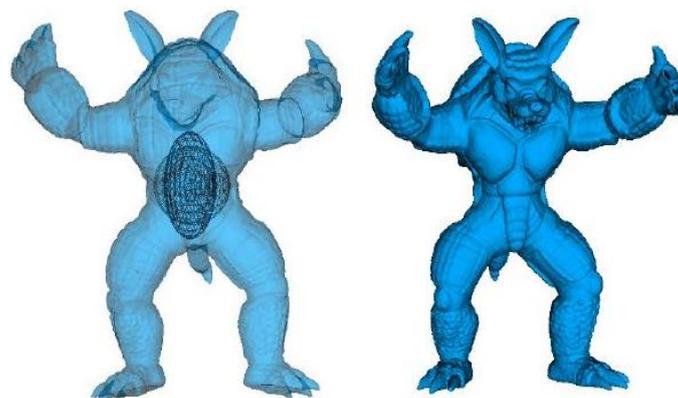
### 1.6.3 Boundary Element Formulation of Harmonic Coordinates

In a major advance in geometry parameterization based in harmonic coordinates later in 2007, the Boundary Element Formulation of Harmonic Coordinates was proposed. Specifically, Rustamov (2008) applied the Boundary Element Method (BEM) in order to calculate the Harmonic Coordinates of an arbitrary point  $p$  at the interior of the control cage. Furthermore, based on the usage of nested cages for local control enhancement by Joshi et al. (2007), Rustamov (2008) applied the BEM in order to extrapolate the Harmonic Functions first at the exterior cage and then, at the entire Euclidean space. In addition, derived by the comparison between the BEM formulations of transfinite harmonic coordinates (Belyaev, 2006), transfinite Shepard's interpolation (Shepard, 1968) and Mean Value coordinates (Hormann and Floater, 2006; Ju et al., 2005), a generalization of the harmonic coordinates was proposed, called the *weakly singular interpolates*. **Figure 1.6.4** presents a deformation based on the aforementioned methodology.



**Figure 1.6.4** Left: Original Model. Middle and Right: Model deformations using Boundary Element Formulation of Harmonic Coordinates (Rustamov, 2008).

Although the BEM proved capable of adequately addressing the problems of the excessive amount of required storage space and the increased computational time associated with the computation of the harmonic coordinates, the proposed extrapolation scheme lacked of two major properties; those of non-negativity and locality. Moreover, Rustamov (2008) proved that the weakly singular interpolates possess the interpolation, smoothness, linear precision and affine invariance properties. **Figure 1.6.5** shows a deformation resulted by exterior harmonic coordinates.



**Figure 1.6.5** Left: Deformation of a sphere into an ellipse in Amardillo's stomach. Right: Exterior deformation caused by the deformation of the sphere (Rustamov, 2008).

#### 1.6.4 Holomorphic coordinates

Li and Liu (2012) presented a cage-free deformation method by taking advantage of the holomorphic coordinates. The particular technique relies on the application of a set of vertices over the shape of interest in order to succeed deformation. By definition, holomorphic coordinates are constructed by conjugate pairs of harmonic functions  $\psi_i$  and  $\varphi_i$ . Thus, for a set of handle points  $\mathbf{u}_i$ :

$$h_i = \varphi_i + i\psi_i . \quad (1.13)$$

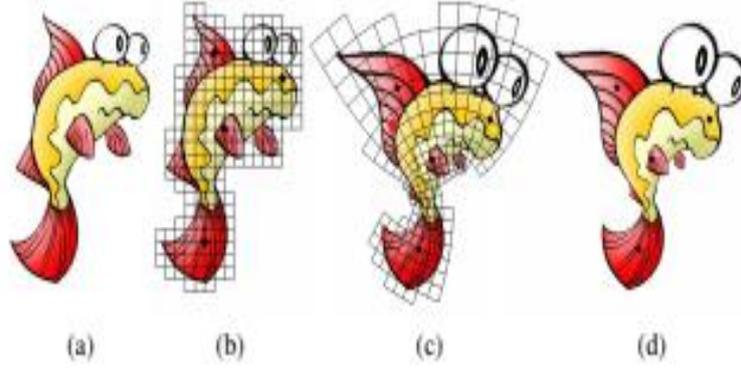
Therefore, the shape of interest can be deformed by direct point handling without the construction of an external cage.

The new position of a shape point is obtained as follows:

$$\mathbf{v}' = \sum_i h_i \mathbf{u}'_i , \quad (1.14)$$

where  $\mathbf{u}'_i$  is the displaced position of the  $i - th$  handle point.

Holomorphic coordinates satisfy interpolation, conformity, smoothness, linear reproduction and similarity properties. However, in contrast to Harmonic Coordinates, the aforementioned coordinates do not possess the non-negativity property (Li and Liu, 2012). A deformation based on Holomorphic coordinates parameterization is shown in **Figure 1.6.6**.



**Figure 1.6.6** A deformation based on Holomorphic coordinates parameterization. In (a) the original goldfish shape is shown, while in (b) the grid covering its geometry is presented. In (c) the deformed grid and the handle points used (black spots) are shown. In (d) the final geometry is displayed (Li and Liu, 2012).

### 1.6.5 Biharmonic Coordinates

Weber et al. (2012) proposed a natural generalization of the Harmonic Coordinates, which is enhanced by the additional property of boundary derivative data interpolation. The key to the proposed methodology is the solution to the *Biharmonic Dirichlet Problem*:

$$\begin{aligned}\Delta^2 f(x) &= 0, \quad x \in \Omega \\ f(x') &= g_1, \quad x' \in \partial\Omega \\ \frac{\partial f}{\partial n}(x') &= g_2, \quad x' \in \partial\Omega.\end{aligned}\tag{1.15}$$

where  $f$  is a biharmonic function at any point  $x$  in the computational domain  $\Omega$ ,  $\Delta$  is the Laplace operator and  $g_1$  and  $g_2$  known values of the biharmonic function and its derivative on the boundary  $\partial\Omega$ , respectively. In order to solve the aforementioned problem, Weber et al. (2012) applied the Boundary Element Method over polygonal domains. The resulted coordinates were called *Biharmonic*.

Therefore, in order to obtain the value of a function  $f$  inside the domain  $\Omega$ :

$$f(x) = \sum_{j=1}^m \mathbf{a}_j(x)f_j + \boldsymbol{\beta}_j(x)d_j, \quad (1.16)$$

where  $\mathbf{a}_j, \boldsymbol{\beta}_j$  are the biharmonic coordinates,  $f_j$  are the prescribed values at  $j^{\text{th}}$  control polygon vertex and  $d_j$  the known derivative values along the corresponding control polygon edges. The biharmonic coordinates satisfy the following properties:

$$\sum_{j=1}^m \mathbf{a}_j(x)v_j + \boldsymbol{\beta}_j(x)n_j = \mathbf{x} \quad (1.17)$$

$$\sum_{j=1}^m \mathbf{a}_j(x) = 1 \quad (1.18)$$

$$\mathbf{a}_j(v_j) = \delta_{ij}$$

$$\frac{\partial \mathbf{a}_j}{\partial \mathbf{n}}(x') = 0, x' \in \partial\Omega \quad (1.19)$$

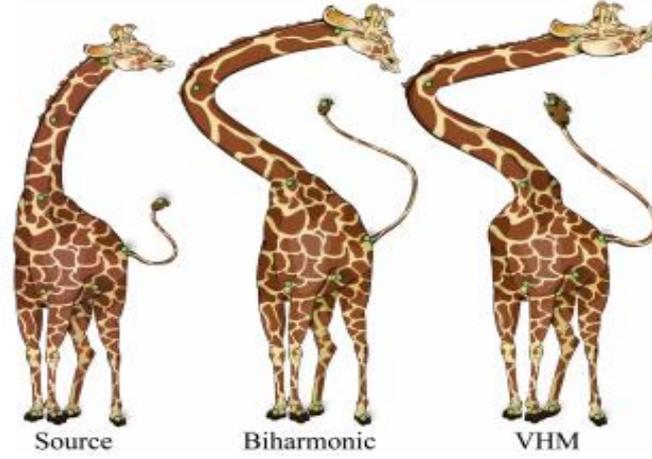
$$\boldsymbol{\beta}_j(x') = 0, x' \in \partial\Omega$$

$$\frac{\partial \boldsymbol{\beta}_j}{\partial \mathbf{n}}(x') = \delta_{ij}, x' \in e_i,$$

where  $v_j$  is the  $j^{\text{th}}$  control cage vertex,  $e_i$  the  $i^{\text{th}}$  edge of the control cage,  $n$  the unit normal to the boundary,  $x'$  an arbitrary point on the boundary,  $x$  an internal point and  $\delta_{ij}$  the Kronecker delta. By moving the control polygon cage vertices, the shape of interest is deformed.

According to Weber et al. (2012), the biharmonic coordinates offer complete control over the boundary of the shape and its derivatives. Additionally, through the application of the BEM, the computation of the aforementioned coordinates is efficient and the discretization of the domain is not required. However, compared to the Harmonic Coordinates, the Biharmonic do not possess the minimum/maximum principle (the capability of attaining both the highest and the lowest value inside the bounded domain), while the sensitivity to user input and incorrect boundary derivative data may lead to unsatisfactory

results. A visualization of biharmonic coordinates deformation compared to Variational Harmonic Maps method is presented in **Figure 1.6.7**.



**Figure 1.6.7** Comparison of Biharmonic and Variational Harmonic Maps deformation (Weber et al., 2012).

### 1.6.6 Bounded biharmonic coordinates

In order to simplify the design and control of the deformation process, Jacobson et al. (2011) proposed a methodology based on the development of linear blending weights that provide smooth and intuitive deformations on objects. In particular, in order to compute the transformation of a domain  $\Omega$ , a weighted blend of handles  $H_j$  (control cage vertices, single points, skeleton bones or a region) transformations are computed. More specifically, a weighted function  $w_j : \Omega \rightarrow \mathbb{R}$  is associated with every handle point the choice of whom determines whether intuitive and smooth deformations will be developed or not. Provided the affine transformation (position, rotation, scaling)  $T_j$  for each handle point  $H_j, j = 1, \dots, m$  by the user, the new location of a point  $p \in \Omega$  is defined by Equation (1.20) as follows:

$$p' = \sum_{j=1}^m w_j(p) T_j p. \quad (1.20)$$

In order to compute the weights  $w_j$ , the Laplacian Energy is minimized:

$$\arg \min_{\mathbf{w}_j, j=1,\dots,m} \sum_{j=1}^m \frac{1}{2} \int_{\Omega} \|\Delta \mathbf{w}_j\|^2 dV, \quad (1.21)$$

$$\text{subject to: } \mathbf{w}_j|_{H_k} = \delta_{jk} \quad (1.22)$$

$$\mathbf{w}_j|_F \text{ is linear } \quad \forall F \in F_C \quad (1.23)$$

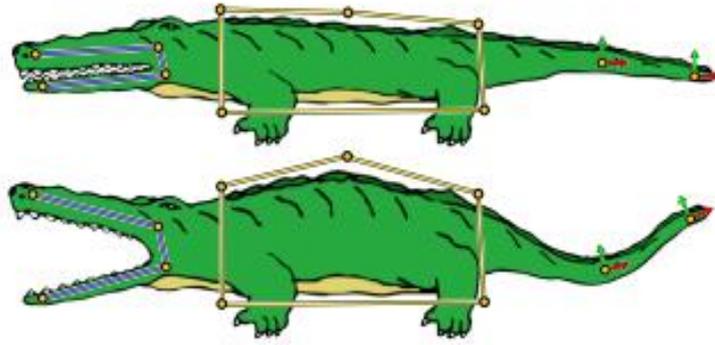
$$\sum_{j=1}^m \mathbf{w}_j(p) = 1 \quad \forall p \in \Omega \quad (1.24)$$

$$0 \leq \mathbf{w}_j \leq 1, \quad j = 1, \dots, m \quad \forall p \in \Omega, \quad (1.25)$$

where  $F_C$  is the set of all cage faces,  $\delta_{jk}$  the Kronecker's delta and  $\Delta$  the Laplace operator (Jacobson et al., 2011).

According to Jacobson et al. (2011), the bounded biharmonic weights produce smooth and shape-aware deformations even for complex geometries. Furthermore, non-negativity and partition-of-unity properties are satisfied. Additionally, it was experimentally observed that bounded biharmonic weights possess the locality, sparsity and non-local maxima properties as well. Moreover, due to the generality of the proposed methodology, additional control over the minimization of the Laplacian energy is possible to be established.

However, in order to compute the weights  $w_j$  the discretization of the domain is required combined with the increased amount of optimization time is needed. Furthermore, the bounded biharmonic weights do not satisfy the linear reproduction property, thus they do not, necessarily reproduce linear functions. A shape deformation based on the aforementioned method is presented in **Figure 1.6.8**.



**Figure 1.6.8** Crocodile deformation occurred by making the use of skeleton, cage and points (Jacobson et al., 2011).

### 1.6.7 Pseudoharmonic coordinates

Due to the absence of their closed form, in order to approximate Harmonic coordinates, a discrete Laplace equation in the triangulated domain must be solved numerically. The resulted linear system is large and excessive computational time is required, in order to obtain the final solution. As a result, several coordinate schemes have been proposed which have closed-form solutions, less computational time is needed for their computation and act as approximations to the Harmonic Coordinates. Chen and Gotsman (2016) investigated the quality of the aforementioned approximations produced by popular barycentric coordinate schemes and studied the pseudo-harmonicity of the *Moving Least Squares (MLS) Coordinates* as presented by Manson and Schaefer (2010).

According to Chen and Gotsman (2016), one of the basic tests to specify the accuracy of the approximation is to check whether the tested coordinates coincide with the harmonic coordinates in the special case of the contour to be a unit circle. More specifically, consider the *transfinite barycentric coordinates* of a planar curve  $S$ , where a function  $f$  in a 2D point  $t$  in the interior of a closed planar curve is given by:

$$f(t) = \oint_S K(s, t) f(s) ds, \quad (1.26)$$

given a value  $f(s)$  of function  $f$  on a 2D point  $s$  on  $S$  and  $K(s, t)$  a real-valued Kernel function. If given a transfinite barycentric coordinate scheme, reduces to the (harmonic) *Poisson kernel*:

$$K_p(w, z) ds = \frac{ds}{|w - z|^2}, \quad (1.27)$$

for the special case that the contour is a *unit cycle*, the particular coordinates are called *pseudoharmonic*. Note that,  $w$  represent a point on the contour of the circle,  $z$  is an interior to the unit circle point and  $ds$  is the arc-length differential (Chen and Gotsman, 2016).

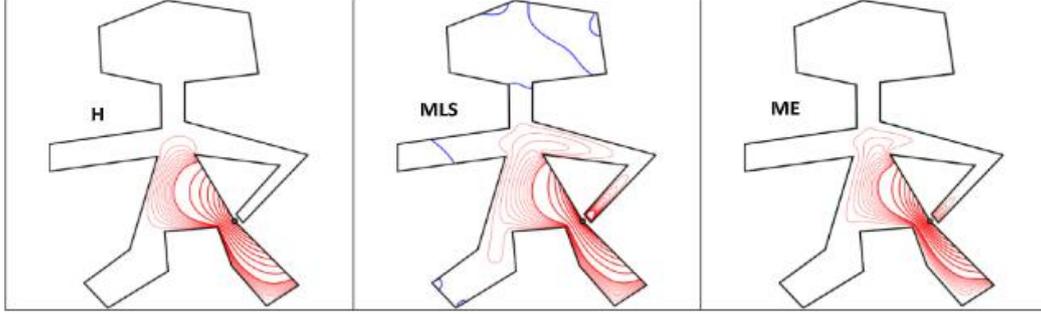
The kernel  $K$  of Equation (1.26) satisfies the properties:

$$\text{Non-negativity: } K(s, t) \geq 0, \quad \forall s \in S, t \in \text{int}(S) \quad (1.28)$$

$$\text{Constant precision: } \oint_S K(s, t) ds = 1, \quad \forall t \in \text{int}(S) \quad (1.29)$$

$$\text{Linear precision: } \oint_S sK(s, t) ds = t, \quad \forall t \in \text{int}(S). \quad (1.30)$$

Chen and Gotsman (2016) proved that the affine-based MLS and similarity-based MLS coordinates are pseudoharmonic. In addition, it was concluded that among coordinate-based transfinite interpolation schemes such as Wachspress Coordinates (Wachspress, 1975), Laplace Coordinates (Pinkall and Polthier, 1993), Gordon-Wixom Coordinates (Gordon and Wixom, 1974), Mean-Value Coordinates (Floater, 2003), Maximum Entropy Coordinates (Hormann and Sukumar, 2008), Poisson Coordinates (Li and Hu, 2013) and MLS Coordinates, the latter offer higher quality approximations on irregular convex or non-convex polygons. In **Figure 1.6.9** a comparison between Harmonic, Moving Least Squares and Maximum Entropy coordinate functions is shown.



**Figure 1.6.9** A comparison between barycentric coordinate function of a non – convex polygon. Left: Harmonic coordinate function. Middle: Moving Least Squares coordinate function. Right: Maximum Entropy coordinate function (Chen and Gotsman, 2016).

### 1.6.8 Harmonic Guidance for Surface Deformation

Zayer et al. (2005) introduced two alternative methods for smooth deformation interpolation aiming at preserving the global shape of the object of interest as well. According to the first editing scenario, deformation is succeeded through repositioning a small number of a triangulated domain vertices by the user, while all the other are placed in an automatic manner by the system. According to Zayer et al. (2005), three types of vertices are considered; free, fixed and edited. The first category includes all vertices which are to be displaced, while the second one consists of points which maintain their initial position during the whole process. Edited vertices are defined by the user and a change in their position causes the deformation the object is desired to be subject to. Local deformation interpolation is achieved through the solution of the Laplace's equation (1.12) over the whole region of interest, given the following boundary conditions:

$$\nabla^2 h(\text{fixed vertices}) = 0 \quad (1.31)$$

and

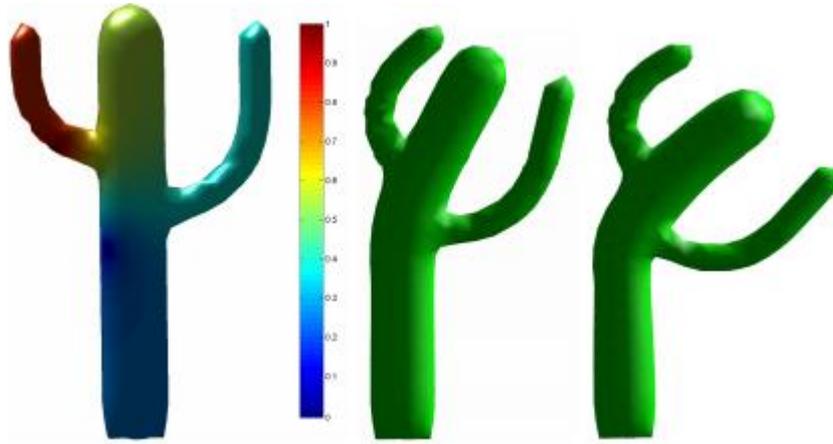
$$\nabla^2 h(\text{edited vertices}) = 1, \quad (1.32)$$

where  $h$  is the harmonic function.

Then, the fragmented mesh resulted by the deformation is reconstructed through the solution of the sparse linear system:

$$Lx = b, \quad (1.33)$$

where  $\mathbf{b}$  denotes the vector containing the divergence values at all vertices,  $L$  is the Laplace-Beltrami operator and  $\mathbf{x}$  are the new positions of the vertices. A deformation based on the particular editing scenario is shown in **Figure 1.6.10**.



**Figure 1.6.10** Left: A visualization of the harmonic field. Red and blue colored areas are formed due to the boundary conditions (1.32) and (1.31) set at two vertices on the tip of the left arm and below the middle of the trunk, respectively. Middle: The deformed object. Right: A difference in deformation propagation occurred following the increase of the vertices at both arms where the boundary condition (1.32) is set (Zayer et al., 2005).

On the other hand, the second surface editing scenario is based on the establishment of correspondence between source and target objects of comparable geometries. Then, harmonic fields are used in order to “transfer” deformation from source to the target surface.

Specifically, according to Zayer et al. (2005), the user selects few corresponding pairs of points denoted as markers  $m_i$ ,  $i = 1, \dots, k$ , to “establish a semantic correlation” between the two surfaces. Then, for each marker, a harmonic field  $h_i$  is defined with the boundary condition:

$$\mathbf{h}_i(m_j) = 1, \quad i = j \quad (1.34)$$

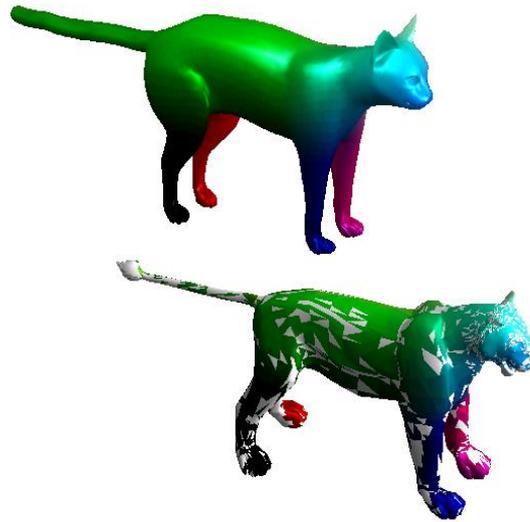
and

$$\mathbf{h}_i(m_j) = 0, \quad i \neq j. \quad (1.35)$$

Each vertex of both source and target meshes will be assigned a  $k$  – *dimensional* vector containing the harmonic functions resulted, one for each marker. The harmonic functions on an arbitrary mesh point  $p$  satisfy the partition-of-unity property, thus:

$$\sum_{l=1}^k \mathbf{h}_l(p) = 1. \quad (1.36)$$

Source and target surfaces with comparable geometries have similar harmonic fields and the corresponding vectors assigned on every triangular element of both meshes (source and target) have close field values resulting, subsequently, to the deformation transfer. Finally, Equation (1.33) is solved in order to reconstruct the target mesh. In **Figure 1.6.11** a source-target surface correspondence is presented.



**Figure 1.6.11** The cat surface triangles are mapped on the lion in order to succeed deformation transfer. Colored areas visualize the surface parts origin. White triangles show the absence of correspondence in the particular areas (Zayer et al., 2005).

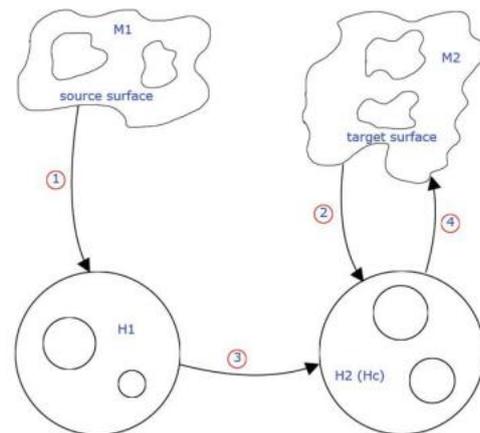
---

Additionally, Zayer et al. (2005) proposed an interpolation technique for deformation propagation depending solely on establishing correspondence on markers. Thus, local deformations applied on markers of the source mesh are used as constraints for harmonic interpolation on the target surface. Once again, the Equation (1.33) is solved in order to obtain the deformed mesh of the target geometry.

Nonetheless, for the purpose of preserved geometrical details between the related surfaces, additional markers may be added. Furthermore, according to Zayer et al. (2005) local deformations, obtained by harmonic interpolation, are within the convex hull of the given deformation at the markers. As a result, local deformations cannot be reproduced between the specific areas. Moreover, the proposed deformation transfer techniques between surfaces with semantic differences are not capable of producing pleasing results.

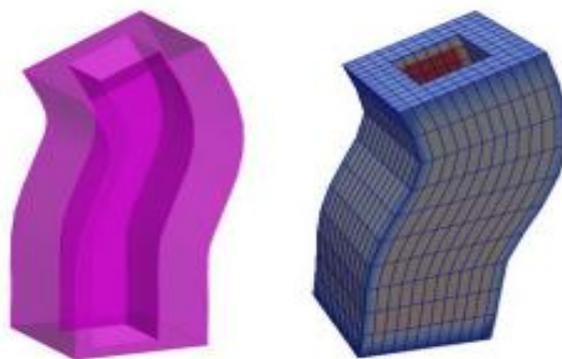
### 1.6.9 Robust One-to-One Sweeping with Harmonic S-T mapping

Cai and Tautges (2014) proposed a robust mesh generation technique based on Harmonic mapping and the established correspondence between the source (S) and the target (T) surface for interior nodes location in volumes with concavities. According to the authors, harmonic mapping  $M \rightarrow H$  is the mapping between two Riemannian manifolds, where the Dirichlet energy is minimized. During the harmonic mapping process, source ( $M1$ ) and target ( $M2$ ) surface triangulations are harmonically mapped on two 2-D unit disks,  $H1$  and  $H2$ , respectively. Therefore, by making the use of the linear approximation method, the outmost boundary source and target vertices are allocated on the 2D unit disks ( $H1$  and  $H2$ ) boundaries in order to correspond. Thereupon, barycentric coordinates from each mesh node of  $M1$  are computed and their 2D position at  $H1$  is obtained. Then, the corresponding 3D position on the target surface  $M2$  of every mesh node at  $H2$  is calculated. **Figure 1.6.12** presents the Harmonic mapping process.



**Figure 1.6.12** (1) M1 mapping on H1 unit disk. (2) M2 mapping on H2 unit disk. (3) H1 mapping on H2 (4) H2 mapping on M2 (Cai and Tautges, 2014).

To locate internal nodes in volumes with concavities, Cai and Tautges (2014) proposed the Harmonic Interpolation technique. More specifically, the target quadrilateral mesh is translated in the inverse sweeping direction and the source surface is reached. Then, all-quad meshes in both source and target surface are converted into triangular. Since the location of the interior nodes in the source object is known, the bounding surface of the target (deformed) geometry may act as a cage. Thus, the final position of the interior nodes in the deformed object may be calculated using the harmonic interpolation as presented in [Section 1.5.1](#), where for all vertices  $C_i$  on the target bounding surface, the same boundary conditions hold. A deformation process is shown in **Figure 1.6.13**.

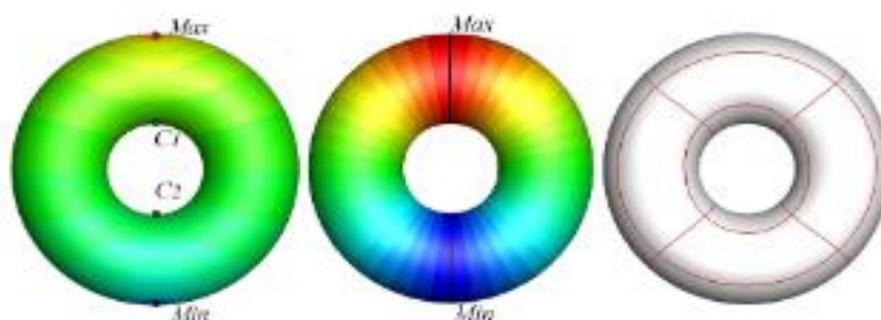


**Figure 1.6.13** Left: A physical model. Right: Meshed bounded surfaces (Cai and Tautges, 2014).

### 1.6.10 Volumetric T-Spline construction using Boolean operations

Liu et al. (2014) proposed a volumetric T-Spline construction algorithm in order to overcome the limitations related to the application of NURBS in the isometric analysis. Such drawbacks include the absence of local refinement and the resulted gaps between two neighboring surface patches. The main idea behind constructing a volumetric T-spline model on an object with complex geometry is the decomposition of the initial shape into hexahedral components and the development of Boolean operations for volumetric T-Spline control mesh creation.

In order to divide the geometry into coherent surfaces, the harmonic field is calculated. According to Liu et al. (2014), such a field consist of that of the steady-state temperature distribution. Therefore, the key to the entire process is to assign the maximum and minimum temperature values on two separate points belonging to the model, which will act as boundary conditions. Then, Laplace's equation is solved over the surface mesh and the critical points of the field (min, max and saddle) are obtained. On the sections formed by the aforementioned points, the minimum and maximum temperature values are assigned and the harmonic field is recalculated. Finally, following the tracing of the gradient lines, the model is divided into parts. The aforementioned technique applied to a torus is presented in **Figure 1.6.14**.



**Figure 1.6.14** Left: Harmonic field calculation given maximum and minimum temperature values at the highest and lowest points, respectively. Middle: Harmonic field calculation given maximum and minimum temperature values at the highest and lowest cross regions, respectively. Right: Decomposition of the torus (Liu et al., 2014).

### 1.6.11 All hexahedral mesh generation via inside-out advancing front based on harmonic fields

Li and Tong (2012) proposed an inside out Advancing Front method to generate an unstructured all-hexahedral mesh for a given volume constructed, initially, by an isotropic tetrahedral mesh. The key of the aforementioned method consisted of the calculations of two harmonic fields which determined the orientation of the produced elements and guided the whole hex-mesh advancement process. Initially, in order to construct the harmonic field over the whole volume, Li and Tong (2012) proposed the solution of the discrete Laplace's equation of the form:

$$\sum_{j \in N(i)} w_{ij}(u_i - u_j) = 0, \quad (1.37)$$

over the tetrahedral mesh  $M$ , where  $N(i)$  is the one-ring neighborhood and  $u_i$  the harmonic scalar value of the function  $f: M \rightarrow R$  of the vertex  $v_i$ , respectively. The term  $w_{ij}$  is a real-valued weight assigned to the edge  $e_{ij}$  and can be calculated by the Equation (1.38) as follows:

$$w_{ij} = \sum_{k=1}^n \|e_{pq}\| \cos(\theta_k), \quad (1.38)$$

where  $e_{pq}$  is the opposite edge of  $e_{ij}$  and  $\theta_k, k = 1, \dots, n$  is the dihedral angle.

Then, the linear system:

$$\mathbf{L}\mathbf{u} = \mathbf{0}, \quad (1.39)$$

is obtained, where  $\mathbf{u} = \{u_0, u_1, \dots, u_n\}$  and  $\mathbf{L}$  is the Laplacian operator equal to:

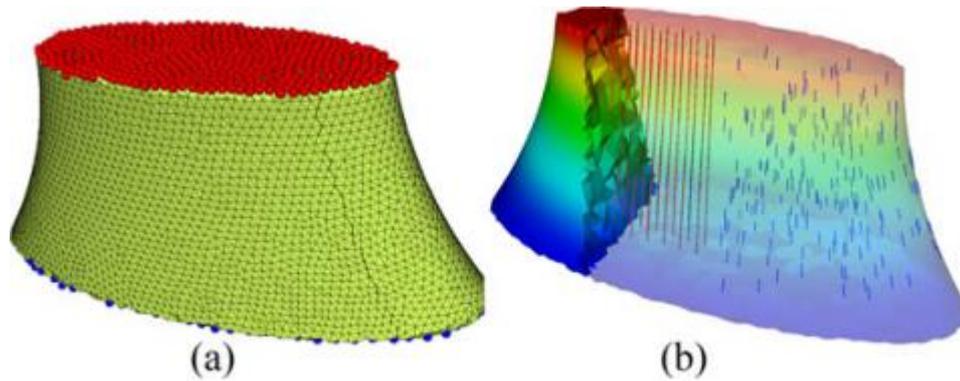
$$\mathbf{L}_{ij} = \left\{ \begin{array}{ll} \sum_{v_k \in N(i)} w_{ik} & \text{if } i = j \\ -w_{ij} & \text{if } j \in N(i) \\ 0 & \text{otherwise} \end{array} \right\}. \quad (1.40)$$

The solution of the Equation (1.39) consists of the harmonic discrete scalar values on the corresponding mesh vertices. Subsequently, every point  $v$  of the volume its harmonic function equals to:

$$u(v) = \sum_{i=1}^4 u_i \varphi_i(v), \quad (1.41)$$

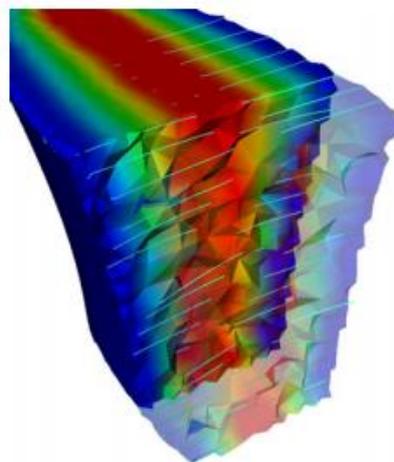
where  $\varphi_i(v)$  is the linear basis function on each vertex of the tetrahedron in which the point lies.

Therefore, in order to solve the Equation (1.39) the following boundary condition was set: the selected boundary vertices of the one side of the volume had the minimal scalar value equal to 0, while the boundary vertices belonged to the other volume side were constrained with the maximal value equal to 1. In addition, according to Li and Tong (2012), the discrete gradient vectors of the harmonic field should be perpendicular to the corresponding normal of the boundary surface. The constructed harmonic field, under the aforementioned boundary conditions, was referred to as "*Principal*" and determined the orientation of the constructed elements. **Figure 1.6.15** shows the boundary conditions set on an elliptical cylinder and the corresponding principal harmonic field.



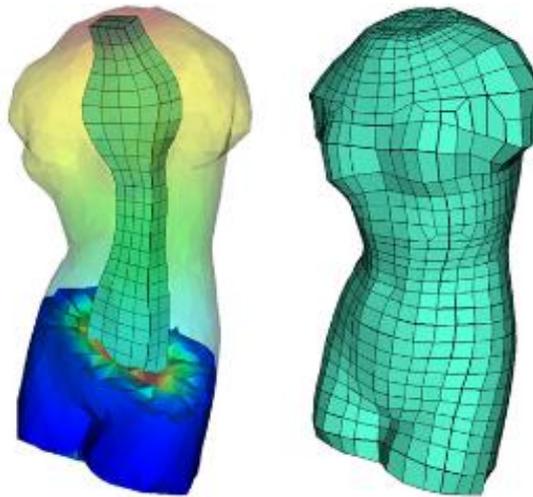
**Figure 1.6.15** (a) The boundary conditions on an elliptic cylinder. Red-colored area indicates the maximal valued vertices, while blue-colored vertices are constrained with the minimal value. (b) The distribution of the principal harmonic field over the elliptic cylindrical volume. Red-colored and blue colored areas indicate high and low scalar values, respectively. The short blue lines represent the gradient vector at the corresponding points (Li and Tong, 2012).

On the contrary, the harmonic field obtained under the boundary condition where the vertices of the initial hexahedral elements are constrained with the minimal value, while the ones belonged on the tetrahedral mesh boundary were equal to the maximal value, was referenced to as “*Radial Harmonic Field*”. The gradient vector field was calculated in order to trace the streamlines of Radial Harmonic field and indicates the direction of the advancement process on each individual vertex. A visualization of the produced Radial Harmonic field on an arbitrary volume is shown in **Figure 1.6.16**.



**Figure 1.6.16** The distribution of the Radial Harmonic field. Red-colored and blue-colored areas mark high and low scalar field vectors. The streamlines of the Radial Vector Field are indicated with light blue color (Li and Tong, 2012).

In addition, Li and Tong (2012) proposed a complex volume segmentation technique. The main idea was to generate hexahedral-meshed subvolumes with smooth interfaces between the segmented areas. To achieve this, they proposed the solution of the Linear System (1.39) with the following boundary condition: the scalar value of each boundary vertex equals to the random walk distance to the specific seed face. The constructed harmonic fields on the Venus model and the produced hex mesh are shown in **Figure 1.6.17**.

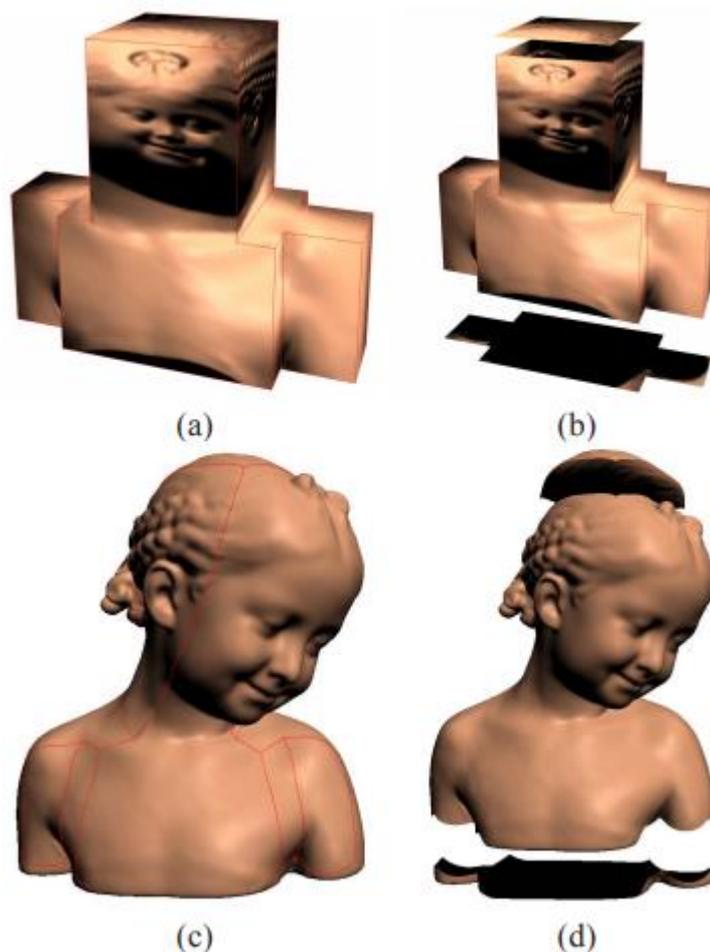


**Figure 1.6.17** Left: Harmonic field construction in Venus model. Right: All – hexahedral mesh generation (Li and Tong, 2012).

### 1.6.12 Direct Product Volumetric parameterization of handlebodies via harmonic fields

Xia et al. (2010) proposed a volume parameterization method by exploiting the harmonic field properties. Given a manually designed polycube  $P$  and a 3D geometry represented by a tetrahedral mesh  $M$ , a polycube map between the boundary  $\partial P$  and  $\partial M$  is constructed by the Divide and Conquer method introduced by He et al. (2009). Then, due to the bijectivity of the aforementioned polycube mapping, the user-specified partition of the boundary surface  $\partial P$  into floor  $B_0$ , ceiling  $B_1$  and walls  $D$ , automatically, results

into the identical partition of the surface  $\partial M$ . A partition between the polycube and a 3D model is shown in **Figure 1.6.18**.



**Figure 1.6.18** (a) The surface of the 3D model  $M$  mapped on the surface of the polycube  $P$ . (b) The partition of the polycube  $P$ . (c) The surface of the 3D model  $M$ . (d) The partition of the 3D model  $M$  resulted by the partition of the polycube  $P$  (Xia *et al.*, 2010).

Now, consider two harmonic functions on  $M$  and  $P$ ,  $f_M: M \rightarrow \mathbb{R}$  and  $f_P: P \rightarrow \mathbb{R}$ , accordingly. Then, through the solution of the Laplace's equation:

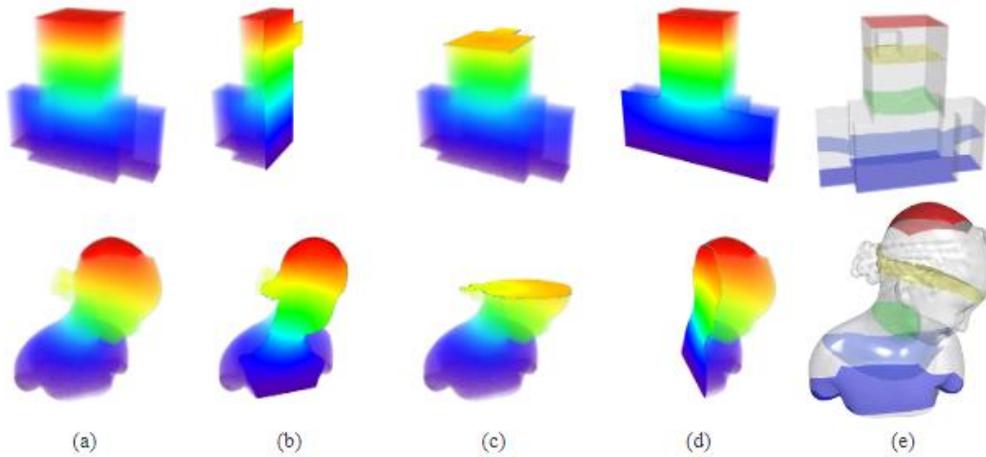
$$\Delta f(p) = 0 \quad \forall p \notin B_0 \cup B_1, \quad (1.42)$$

under the boundary conditions:

$$f(p) = 0 \quad \forall p \in B_0 \quad (1.43)$$

$$f(p) = 1 \quad \forall p \in B_1, \quad (1.44)$$

the harmonic functions  $f_M$  and  $f_P$ , are obtained. Afterwards, the gradient field vector of  $f_M$ ,  $\nabla f_M$ , is calculated and the integral curve (whose tangent vector equals to the gradient field) is traced. The points where the integral curve intersects the floor and ceiling of  $M$  are denoted with  $p$  and  $q$ , respectively. Following the creation of the polycube map, each point on the ceiling and floor of  $M$  is mapped to the polycube  $P$ . The computation of the harmonic fields on the polycube  $P$  and the 3D model  $M$  is presented in **Figure 1.6.19**.



**Figure 1.6.19** Figures from (a) to (d) show the polycube (up) and 3D model (down) computed harmonic fields. Figure (e) shows the several iso – surfaces of harmonic fields (Xia et al., 2010).

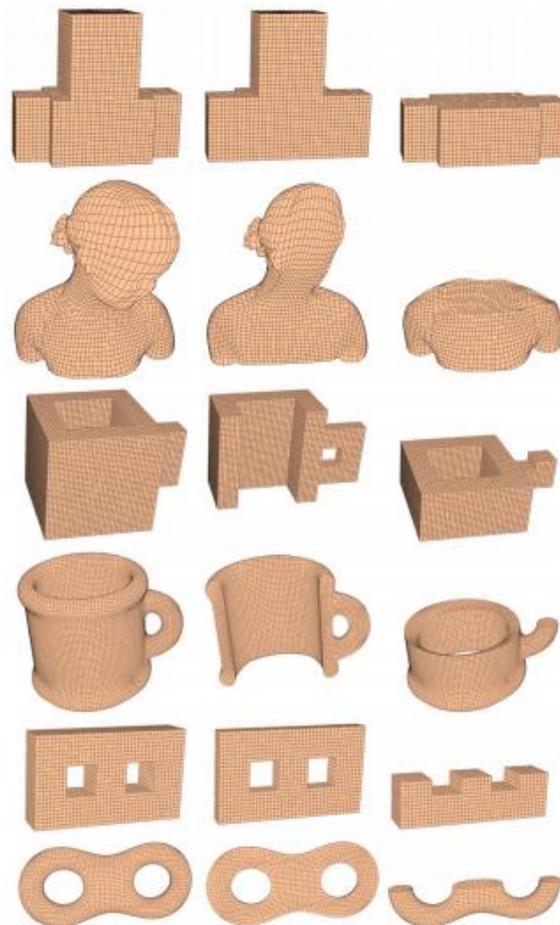
Then, the integral curve of the gradient vector  $\nabla f_P$  is computed and the intersection points on the floor and ceiling of  $P$ ,  $p'$  and  $q'$  are obtained, respectively. Subsequently, given an interior point of  $M$ , denoted as  $s$ , the corresponding unique point in  $P$ ,  $s'$ , is found through the Equation (1.45):

$$f_M(s) = f_P(s'). \quad (1.45)$$

Thus, by locating each interior point of  $P$  given the corresponding point's position on  $M$ , the volumetric mapping for the interior is constructed.

Similarly, to construct a mapping for the walls, the gradient of every triangular element of the boundary of the floor on  $M$ ,  $\partial B_0^M$ , is calculated and the integral curve beginning from the aforementioned boundary is traced. Then, through the floor map  $h: B_0^M \rightarrow B_0^P$ , each integral curve on  $M$  is mapped to a corresponding curve on  $P$ , thus, every point on  $M$  is mapped to a unique point on  $P$ .

Furthermore, resulted from the regular structure of the polycube used in the aforementioned volumetric parameterization scheme, the proposed methodology was applied for the generation of hexahedral meshes on 3D domains. A generated hexahedral mesh resulted from the application of the aforementioned technique, is shown in **Figure 1.6.20**.



**Figure 1.6.20** The generated hexahedral mesh using the Direct Product Volumetric Parameterization method (Xia et al., 2010).

### 1.6.13 Dynamic Harmonic fields for surface processing

Xu et al. (2009) introduced a fast updating harmonic field method for dynamic conditions handling applied for surface processing applications. The aforementioned methodology is based on the solution of the linear system (1.46) on a closed manifold triangular mesh:

$$(\mathbf{L} + \mathbf{P} + \mathbf{R}\mathbf{R}^T - \mathbf{B}\mathbf{B}^T)\mathbf{u} = \mathbf{P}\mathbf{b}, \quad (1.46)$$

where  $\mathbf{u}$  is the vector containing the unknown approximate harmonic function values on the mesh vertices. Term  $\mathbf{L}$  denotes the unconstrained Laplacian matrix and equals to:

$$\mathbf{L} = \mathbf{D} - \mathbf{W}, \quad (1.47)$$

where  $\mathbf{W}_{ij} = \frac{1}{2}(\cot\alpha_{ij} + \cot\beta_{ij})$ , while  $\alpha_{ij}$  and  $\beta_{ij}$  are the opposite angles of  $(i, j)$  edge in the triangular mesh, respectively. In case of edge  $(i, j)$  not existing then  $\mathbf{W}_{ij} = 0$ .

Matrix  $\mathbf{D}$  is diagonal and consists of the row sums of  $\mathbf{W}$ . Furthermore, the diagonal penalty matrix  $\mathbf{P}$  in Equation (1.46) equals to:

$$\mathbf{P}_{ij} = \begin{cases} a, & \text{if } i \in S \text{ and } i = j \\ 0, & \text{otherwise} \end{cases}, \quad (1.48)$$

where the penalty factor  $a = 1.0 \cdot 10^8$  and  $S$  the set of indices of the vertices on which the boundary conditions are forced. In addition, matrices  $\mathbf{R}$  and  $\mathbf{B}$  are equal to:

$$\mathbf{R}_{ij} = \begin{cases} \sqrt{a}, & i = j \in S_{ins} \\ 0, & \text{otherwise} \end{cases}, \quad (1.49)$$

and

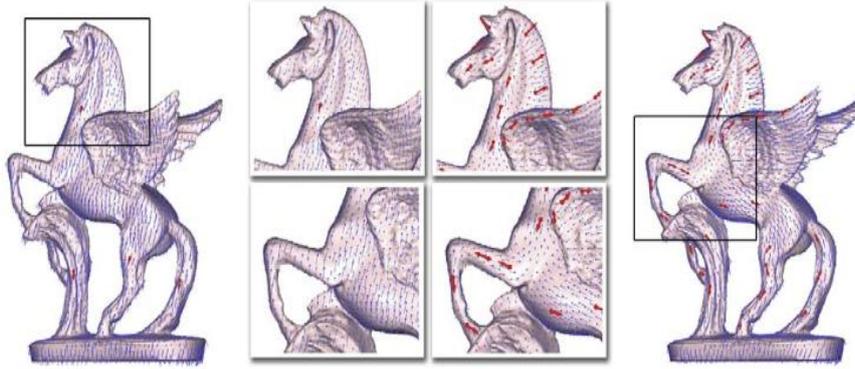
$$\mathbf{B}_{ij} = \begin{cases} \sqrt{a}, & i = j \in S_{del} \\ 0, & otherwise \end{cases}. \quad (1.50)$$

Terms  $S_{ins}$  and  $S_{del}$  denote the sets of indices referred to locations of added and deleted constraints. Moreover, vector  $\mathbf{b}$  in Equation (1.46) equals to:

$$\mathbf{b}_i = \begin{cases} 0 & \text{if } i \notin S \\ s_i & \text{if } i \in S \end{cases}, \quad (1.51)$$

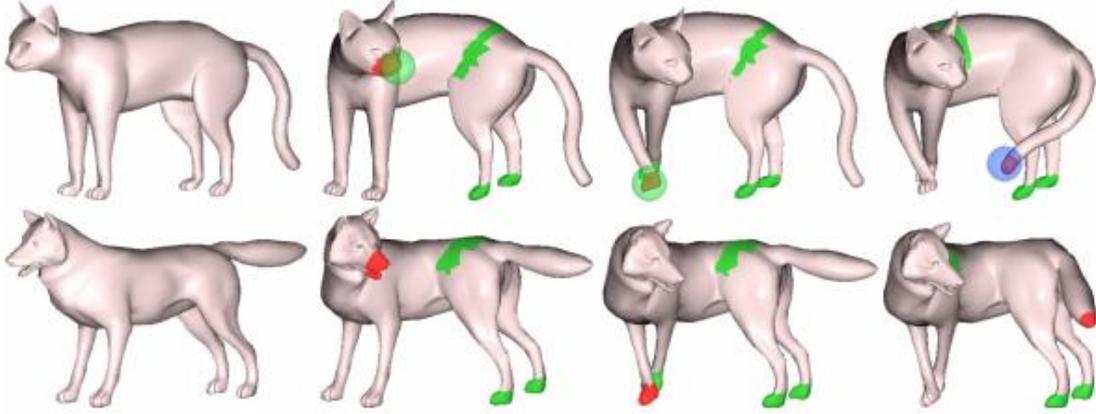
where the term  $s_i$  consist of the given harmonic field value at site  $i = 1, \dots, n$ .

The harmonic field computed following the addition of new constraint vectors is shown in **Figure 1.6.21**.



**Figure 1.6.21** Left: Harmonic vector field computation according to the user - specified constraint vectors (red arrows). Right: Calculation of the Harmonic field after the insertion of additional constraint vectors (Xu et al.).

Xu et al. (2009) demonstrated the effectiveness of the interactive harmonic field updating methodology on various surface processing applications, such as the shape deformation and deformation transfer between a source and a target surface by making the use of dynamic handles. In all presented applications the proposed methodology offered real-time performance and interactive user experience on surface processing. **Figure 1.6.22** presents a deformation transfer between a source (cat) and a target (wolf) shape by making the use of dynamic handles.



**Figure 1.6.22** Deformation transfer between a source (cat) and a target (dog) surface. Handles applied on cat are indicated with red color and change throughout the procedure (Xu et al., 2009).

#### 1.6.14 Skeleton based cage guided by harmonic fields

Casti et al. (2019) introduced a user-assisted skeleton-based cage generation tool. The input of the method consists of a triangular mesh  $M$  representing the surface of a 3D model of choice and its skeleton  $S$ . Provided a number of bending nodes on the skeleton by the user, a corresponding segmentation of the volumetric model is resulted and an initial control cage is constructed for each piece of the object. In order to create the surfaces which will later contribute to the formation of the final cage, a tetrahedral mesh is generated and a harmonic field on the latter mesh's vertices is calculated. More specifically, given a vertex point  $p$  of the tetrahedral mesh, the solution of the Laplace's equation:

$$\Delta f(p) = 0, \quad (1.52)$$

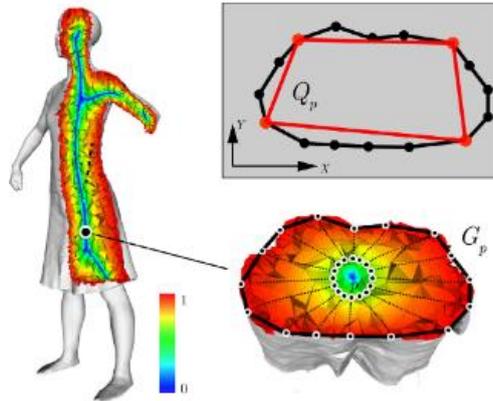
is sought, under the Dirichlet boundary conditions of Equation (1.53) and (1.54), respectively:

$$f(p) = 0, \quad \forall p \in S \quad (1.53)$$

$$f(p) = 1, \quad \forall p \in M. \quad (1.54)$$

Then, the harmonic function is linearly interpolated at the interior of each element belonging to the tetrahedral mesh.

Induced by the boundary conditions (1.53) and (1.54), the integral lines of the produced harmonic field begin from a bending point  $p$  of  $S$  and terminate at the surface mesh  $M$ . The aforementioned intersection points form a ring centered at  $p$  and consist of the closest points of  $p$ . The area spanned transversely by the integral lines of the harmonic field is called a *cutting surface*. For each ring centered at a bending point  $p$  a quadrilateral is formed with its vertices consisted of four sampled intersection points. An example of a calculated harmonic field together with the ring area and the quadrilateral formed by, are shown in **Figure 1.6.23**.

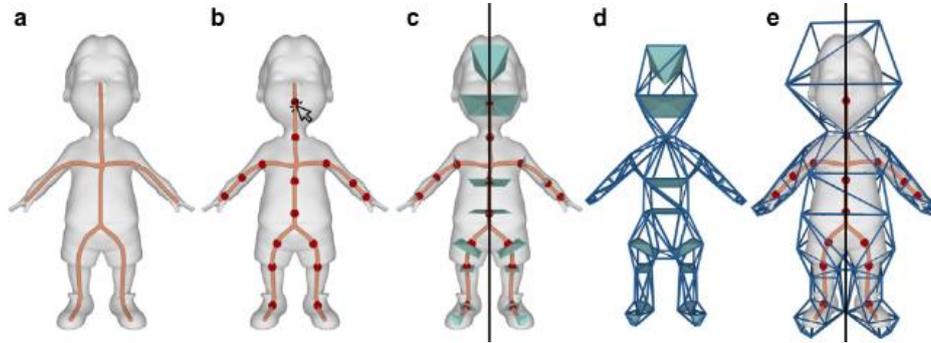


**Figure 1.6.23** Left: The harmonic field on a 3D model. Bottom Right: The intersection points on  $M$  obtain through following the integral lines (black dotted lines) of the harmonic field beginning from  $p$ . Top Right: The quadrilateral section formed by four sampled intersection points (Casti et al., 2019).

Therefore, the quadrilateral sections are connected pairwise and their convex hull is computed. Finally, by welding all the convex hulls for each segment of the 3D model, an initial control cage is constructed. In case of the control cage intersects the surface mesh  $M$ , the corresponding face is projected onto  $M$  following the integral lines of the previously calculated harmonic field.

Therefore, through the welding of the aforementioned quadrilateral sections, which better align to the cutting surfaces, the final topology of the cage is

created. A cage constructed using the presented technique is shown in **Figure 1.6.24**.



**Figure 1.6.24** (a) The input of the particular technique consists of the volume and its skeleton around which the cage will be formed. (b) Bending points are selected by the user. (c) Cross – sections at bending nodes are constructed through tracing the harmonic field integral lines beginning from individual bending points. (d) The welding of the quadrilateral sections around the bending nodes forms the topology of the cage. (e) Presentation of the volume inside its created cage (Casti et al., 2019).

### 1.6.15 Variational Harmonic Maps

Ben-Chen et al. (2009) proposed a space deformation technique based on a set of harmonic functions that have a closed-form expression. The concept underlying the aforementioned methodology consists of the computation of a harmonic mapping where the user places a number of position and orientation constraints, given a cage embedding the volume of interest. Then, an optimization procedure is followed in order to produce a smooth deformation coupled with detail preservation.

More specifically, consider the fundamental solution of the Laplace's equation (which is the *Green's function*) in a 3D domain  $\Omega$  bounded by a triangular mesh  $S(V, F)$  with  $V$  vertices and  $F$  faces:

$$G(q, p) = \frac{1}{4\pi|q - p|} = \hat{\varphi}(q, p), \quad (1.55)$$

and its gradient:

$$\nabla G(q, p) \cdot \hat{n}(q) = \hat{\psi}(q, p), \quad (1.56)$$

where  $p(x, y, z) \in \Omega$ ,  $q(x, y, z) \in \partial\Omega$  and  $\hat{n}(q)$  is the unit normal direction to the surface  $S = \partial\Omega$ . Note that Green's function (1.55) has a closed-form expression for any dimension and considering that consists of a solution to Laplace's equation is also harmonic. Thus, according to Ben-Chen et al. (2009), all its partial and higher derivatives are also harmonic.

Then, a deformation mapping  $f: \Omega \rightarrow \mathbb{R}^3$  can be defined:

$$f_{a,b}(p) = \sum_{t \in F} \int_{q \in t} a_v(q) \hat{\phi}(q, p) - \sum_{t \in F} \int_{q \in t} b_t(p) \hat{\psi}(q, p), \quad (1.57)$$

where  $a$  is a piecewise linear map on  $S$  defined by values  $a_v$  at vertices of the triangular mesh and  $b$  a constant piecewise map defined by values  $b_t$  at the faces of the aforementioned grid.

Given the matrix notation of Equation (1.57):

$$\mathbf{f}_{a,b}(p)_{1 \times 3} = (\boldsymbol{\varphi}_{1 \times n} \quad \boldsymbol{\psi}_{1 \times m}) \begin{pmatrix} \boldsymbol{\alpha}_{n \times 3} \\ \mathbf{b}_{m \times 3} \end{pmatrix}, \quad (1.58)$$

the Jacobian of the deformation  $\mathbf{f}$  at a point  $p$ ,  $\mathbf{J}_f(p)$ , as well as the Hessian  $\mathbf{H}_f(p)$  define the orientation of a point and the smoothness of the deformation around it, accordingly. Terms  $n, m$  in Equation (1.58) denote the numbers of vertices and faces, respectively.

Then, provided by the user a set of  $r$  points  $q_i \in \Omega$  and their target position  $f(q_i) = f_i$ , in addition to a set of  $s$  points  $t_i$  together with their orientation  $\mathbf{J}_F(t_i) = g_i$ ,  $k$  points  $w_i \in \partial\Omega$  are sampled in order to approximate the smoothness of the deformation. Furthermore, the rigidity of the deformation is estimated through the sampling of  $d$  points on  $l$  rigidity lines

which can be obtained by e.g. a given skeleton of the deformed shape by the user.

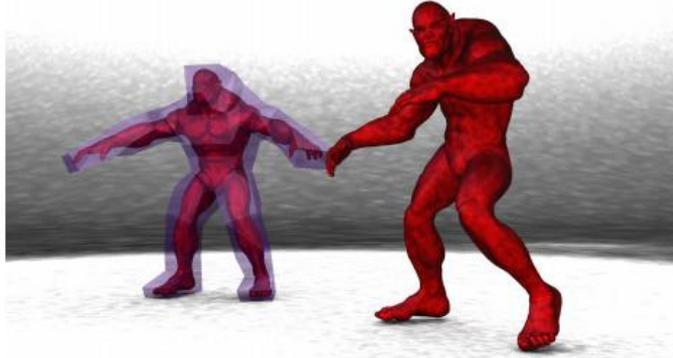
Therefore, the solution of the optimization problem:

$$\min_{a,b,R_i} E(f_{a,b}) = \sum_{i=1}^d \|J_F(m_i) - R_i\|_F^2 + \lambda^2 \sum_{i=1}^k \|H_F(w_i)\|_F^2 \quad (1.59)$$

$$s. t. \quad \forall i = 1, \dots, r \quad f_{a,b}(q_i) = f_i, \quad \forall i = 1, \dots, s \quad J_F(r_i) = g_i \quad (1.60)$$

$$\forall i = 1, \dots, d \quad R_i^T R_i = I, \quad (1.61)$$

defines a smooth and detail-preserving deformation which satisfies the user imposed constraints (1.60) and (1.61). Note that the unknowns of the optimization problem consist of  $a, b, R_i$  where  $R_i$  are the rotations a point should undergo. A deformation resulted from the solution of the aforementioned optimization problem is shown in **Figure 1.6.25**.



**Figure 1.6.25** Left: Initial model's geometry enclosed in a cage. Right: Model's deformation using variational harmonic maps deformation method (Casti et al., 2019).

A detail-preserving deformation produced following the prescription of the target position and Jacobian constraints is presented in **Figure 1.6.26**.



**Figure 1.6.26** Left: Position and Jacobian constraints imposed in the circular area by the user. Right: The detail – preserving deformation of the marked area occurred (Ben-Chen et al., 2009).

### 1.6.16 Bounded distortion Harmonic Shape Interpolation

Chien et al. (2016), presented a shape interpolation method aiming at producing smooth and one-to-one (injective) mappings that have bounded conformal and isometric distortion. The input of the aforementioned methodology consists of two smooth, harmonic mappings with bounded isometric distortion which are, afterward, linearly blended in order to produce the output mapping. Moreover, the proposed interpolation technique is composed of three diverse variants; the second complex dilatation  $v$ , the unscaled  $\eta$  and the metric tensor  $M_h$ .

More specifically, consider a connected domain  $\Omega$  and an arbitrary point  $z \in \Omega$ . Then, any harmonic mapping  $f: \Omega \rightarrow \mathbb{R}^2$  can be written as a sum of a holomorphic  $\Phi$  and an antiholomorphic function  $\bar{\Psi}$ :

$$f(z) = \Phi(z) + \bar{\Psi}(z). \quad (1.62)$$

Furthermore, let a function  $f: \Omega \rightarrow \mathbb{R}^2$  and its Jacobian matrix  $J_f$  at any point in the domain. Then, the matrix  $J_f$  can be written as a sum of a similarity  $S_2 = \begin{pmatrix} a & -b \\ b & a \end{pmatrix}$  and an anti-similarity matrix  $A_2 = \begin{pmatrix} c & d \\ d & -c \end{pmatrix}$ , where  $a, b, c, d \in \mathbb{R}^2$ . Here, the similarity is considered in terms of the Frobenius norm.

Considering the above, the functions  $\Phi(z)$  and  $\bar{\Psi}(z)$  of Equation (1.62) can be obtained through the integration of the parts acquired from the decomposition of the Jacobian matrix of the harmonic mapping  $f: \Omega \rightarrow \mathbb{R}^2$ . Moreover, it should be mentioned that the derivatives  $\Phi' = f_z$  and  $\bar{\Psi}' = f_{\bar{z}}$  are also holomorphic and anti-holomorphic, respectively. Additionally, let the two input holomorphic - harmonic mappings  $f_z^0, f_z^1: \Omega \rightarrow \mathbb{R}$ . Then, the interpolation mapping can be obtained through the linear interpolation of the logarithms of the input mappings, as presented in Equation (1.63):

$$f_z^t = (f_z^0)^{1-t} (f_z^1)^t, \quad (1.63)$$

where the term  $t$  denotes a specific point in the time interval  $[0, 1]$ .

In order to generate mappings with bounded conformal distortion, Chien et al. (2016) introduced the *second complex dilatation*  $v$  variant. Considering a planar mapping  $g$ , the variant  $v$  equals to:

$$v = \frac{\overline{g_{\bar{z}}}}{g_z}. \quad (1.64)$$

Then, the second complex dilation  $v^t$  for a time  $t \in [0, 1]$  can be obtained through Equation (1.65):

$$v^t = (1-t)v^0 + tv^1, \quad (1.65)$$

where  $v^0, v^1$  are also holomorphic. In addition, in order to interpolate the quantity  $v^t$  in the interior of the domain  $\Omega$ , the Equation (1.66) holds:

$$f_{\bar{z}}^t = \overline{v^t f_z^t}. \quad (1.66)$$

Equation (1.66) appears to be anti-holomorphic and it is able, thus, to produce harmonic mappings.

In addition to the first variant of the aforementioned technique, Chien et al. (2016) proposed the *scaled variant*  $\eta$ , aiming at producing local injective mappings with bounded geometric distortion.

Consider a planar mapping  $g$  and the quantity  $\eta = g_{\bar{z}}\overline{g_z}$ , then the linear interpolation of  $\eta$  is obtained:

$$\boldsymbol{\eta}^t = (1 - t)\boldsymbol{\eta}^0 + t\boldsymbol{\eta}^1. \quad (1.67)$$

Additionally, let the *scaled variant*  $\tilde{\boldsymbol{\eta}}^t = \rho(t)\boldsymbol{\eta}$  with  $\rho: [0, 1] \rightarrow (0, 1)$  and  $\rho(0) = \rho(1) = 1$ . Then, the interpolation of the scaled variant  $\tilde{\boldsymbol{\eta}}^t$  in the interior of the domain  $\Omega$ , equals to:

$$\boldsymbol{f}_{\bar{z}}^t = \frac{\tilde{\boldsymbol{\eta}}^t}{\boldsymbol{f}_z^t}. \quad (1.68)$$

Similar to Equation (1.66), Equation (1.68) is anti-holomorphic and it can, thus, produce harmonic mappings.

Additionally, in order to produce mappings that preserve the conformal and isometric distortion, Chien et al. (2016) introduced the *metric tensor variant*. Consider a planar mapping  $h: \Omega \rightarrow \mathbb{R}^2$  where its metric is defined:

$$\boldsymbol{M}_h = \boldsymbol{J}_h^T \boldsymbol{J}_h, \quad (1.69)$$

where  $\boldsymbol{J}_h$  is the Jacobian matrix of the mapping  $h$ . Then, following the decomposition of the Jacobian  $\boldsymbol{J}_h$ , the quantity  $\boldsymbol{A} = |h_z|^2 + |h_{\bar{z}}|^2$  is produced.

Now, let two input mappings  $\boldsymbol{f}^0$  and  $\boldsymbol{f}^1$  and their metrics  $\boldsymbol{M}_f^0$  and  $\boldsymbol{M}_f^1$ , respectively. In order to interpolate the metric tensor  $\boldsymbol{M}_f^t$ , the metrics  $\boldsymbol{M}_f^0$  and  $\boldsymbol{M}_f^1$  are blended. Thus,

$$\boldsymbol{M}_f^t = (1 - t)\boldsymbol{M}_f^0 + t\boldsymbol{M}_f^1. \quad (1.70)$$

Similarly, consider the scaled variant interpolation of Equation (1.67) and the linear interpolation of the quantity  $A$ :

$$\mathbf{A}^t = (1 - t)\mathbf{A}^0 + t\mathbf{A}^1. \quad (1.71)$$

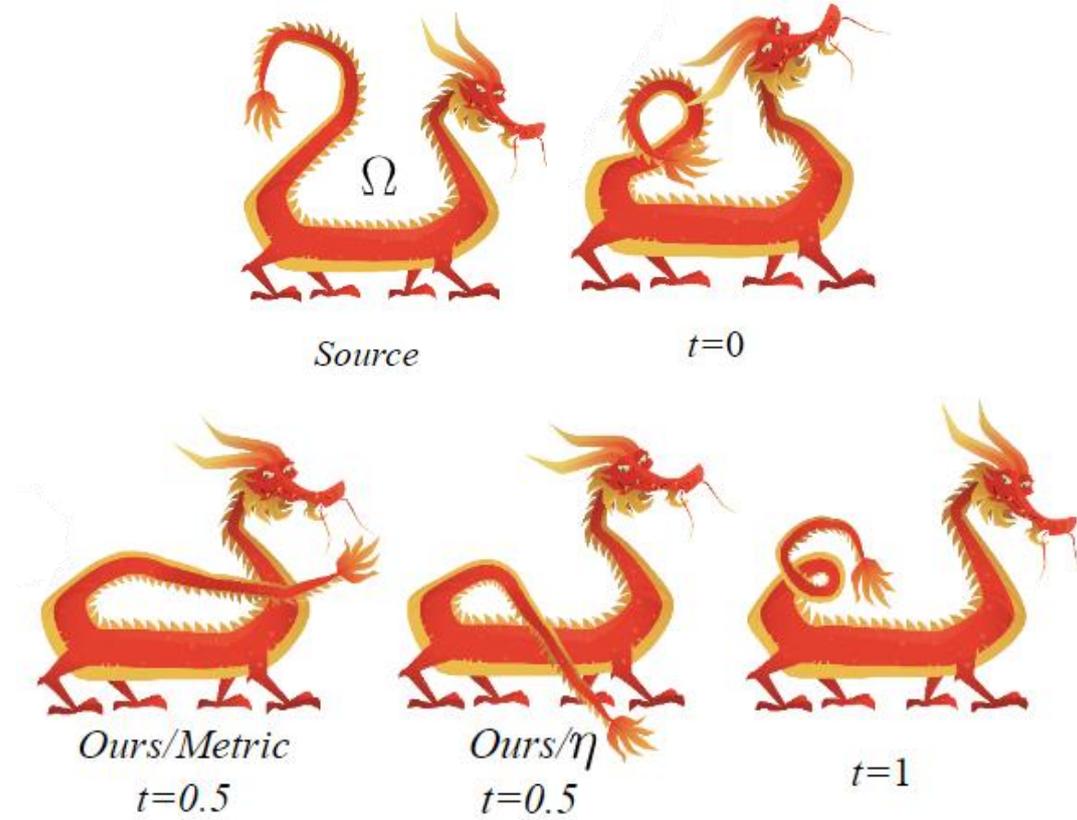
Therefore, according to Chien et al. (2016) in order to preserve the geometric distortion, the metric  $\mathbf{M}_f^t$  is linearly blended on the boundary  $\partial\Omega$  of the domain  $\Omega$ . Thus, the positive root of the Equation (1.72):

$$|\mathbf{f}_z^t|^2 = \frac{\mathbf{A}^t \pm \sqrt{(\mathbf{A}^t)^2 - 4|\boldsymbol{\eta}^t|^2}}{2}, \quad (1.72)$$

is considered the linear interpolation formula for the metric tensor on  $\partial\Omega$ . In order to interpolate the quantity  $|\mathbf{f}_z^t|$  in the interior of  $\Omega$ , the Dirichlet boundary problem is solved through applying the value  $\ln|\mathbf{f}_z^t|$  on the boundary  $\partial\Omega$ . The resulted quantity consists of a holomorphic function.

On the other hand, in order to obtain the anti-holomorphic function  $\mathbf{f}_{\bar{z}}^t$ , the scaled variant  $\boldsymbol{\eta}^t$  is directly interpolated in the interior of the domain  $\Omega$  (and not just on the boundary).

Eventually, each variant of the proposed interpolation technique possesses the ability to produce smooth harmonic mappings with bounded geometric distortion. A demonstration of the produced harmonic mappings given the input mapping for  $t = 0$  and  $t = 1$  is shown in **Figure 1.6.27**.



**Figure 1.6.27** Considering the input mappings for  $t = 0$  and  $t = 1$  and the source image (domain  $\Omega$ ) of the dragon, the output mappings produced by the interpolation of the metric tensor and  $\eta$  variant for  $t = 0.5$  are presented (Chien, Chen and Weber, 2016).

### 1.6.17 Feature-aligned harmonic volumetric mapping using MFS.

Li et al. (2010) proposed a harmonic volumetric mapping methodology that establishes a correspondence between two solid models of similar topology. The input of the method consists of the volumetric geometries of the 3D models,  $M_1 \in \mathbb{R}^3$  and  $M_2 \in \mathbb{R}^3$ , respectively. In addition to prior, the boundary mapping  $\mathbf{f} = (f'^1, f'^2, f'^3)$  between the boundary surfaces  $\partial M_1$  and  $\partial M_2$  is given.

Then, considering a real harmonic function  $f^i (i = 1, 2, 3)$  on the three axes  $x, y, z$ , respectively, the main objective of the particular method is to compute the harmonic volumetric mapping:

$$\mathbf{f}: M_1 \rightarrow M_2 = \mathbf{f} = \sum_{i=1}^{i=n_s} \mathbf{f}_i = \sum_{i=1}^{i=n_s} (f_i^1, f_i^2, f_i^3), \quad (1.73)$$

where for a given collocation point  $p \in \mathbf{P} = \{P_1, P_2, \dots, P_{n_c}\}$ , the boundary conditions (1.74) or (1.75) hold:

$$\mathbf{f}(p) = \mathbf{f}'(p), \quad i = 1 \quad (1.74)$$

$$\delta \mathbf{f}'(p) = \mathbf{f}'(p) - \sum_{i=1}^{i=n_s-1} \mathbf{f}_i(p), \quad i > 1. \quad (1.75)$$

On the other hand, for an internal point  $p \in M_1$ , Laplace's equation in 3D is solved:

$$\Delta f^i = \frac{\partial^2 f^i}{\partial x^2} + \frac{\partial^2 f^i}{\partial y^2} + \frac{\partial^2 f^i}{\partial z^2} = 0. \quad (1.76)$$

Thus, in order to obtain the harmonic functions  $f_i^j$ ,  $j = 1, 2, 3$ ,  $i = 1, \dots, n_s$  the linear system of Equation (1.77) is formulated:

$$\mathbf{A}_i^j \cdot \mathbf{w}_i^j = \mathbf{b}_i^j, \quad (1.77)$$

where given a set of source points  $\mathbf{Q} = \{Q_1, Q_2, \dots, Q_{n_s}\}$  in the exterior of  $M_1$ , the  $uv_{th}$  element of the coefficient matrix  $A$  equals to the Green's function:

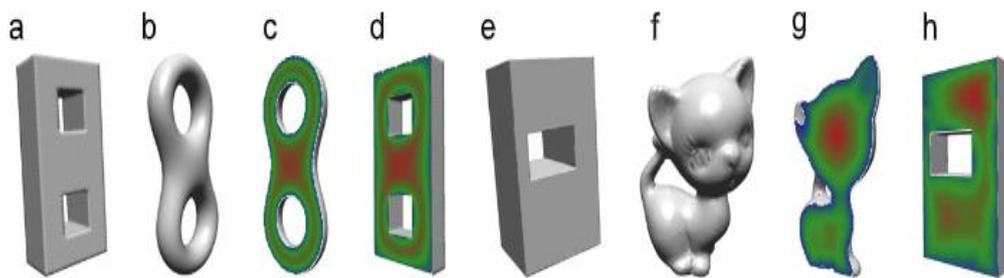
$$\mathbf{A}_{uv} = K(\mathbf{P}_u, \mathbf{Q}_v) = \frac{1}{4\pi|\mathbf{P}_u - \mathbf{Q}_v|}, \quad (1.78)$$

and the term  $\mathbf{b}_i^j$  denotes the values of the boundary conditions on the collocation points as presented in Equations (1.74) and (1.75).

Subsequently, the linear system of Equation (1.77) is solved in order to compute the unknown weight function  $\mathbf{w}_i^j$  on each source point  $\mathbf{Q} = \{Q_1, Q_2, \dots, Q_{n_s}\}$ . Then, in order to compute a harmonic function on a point  $p$  on the boundary or in the interior of  $M_1$  the Equation (1.79) is applied:

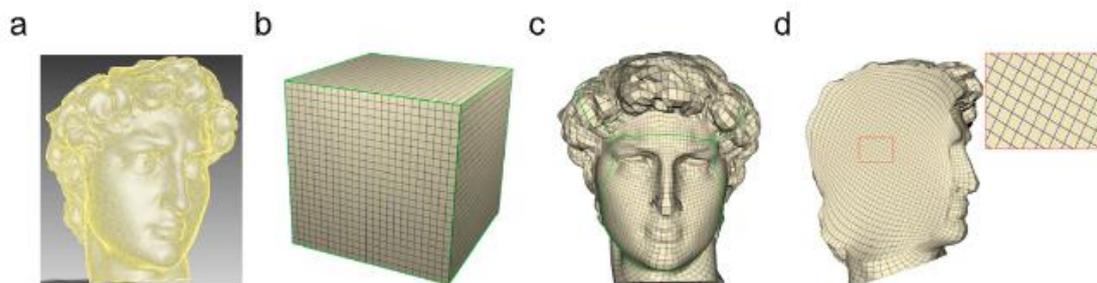
$$f^i(\mathbf{w}^i, \mathbf{Q}; p) = \sum_{n=1}^{n_s} w_n^i K(p, \mathbf{Q}_n). \quad (1.79)$$

Moreover, Li et al. (2010) proposed the addition of feature constraints such as handle points, skeleton, curves and surfaces ideal for matching heterogeneous volumetric data. The establishments of two smooth correspondences between a polycube and a torus as well as a polycube and a kitten are depicted in **Figure 1.6.28**.



**Figure 1.6.28** Polycubes (a) and (e) are mapped onto double torus (b) and kitten (f), respectively. Color – encoded distance field of (f), (g) are transferred under the mapping to (d), (h), respectively (Li et al., 2010).

Additionally, the application of the proposed technique for hexahedral mesh generation is presented in **Figure 1.6.29**.



**Figure 1.6.29** (a) The surface mesh of David's head. (b) A generated hexahedral mesh on the polycube domain. (c) The generated volumetric hexahedral mesh on David's head. (d) The generated mesh in the interior of the volume (Li et al., 2010).

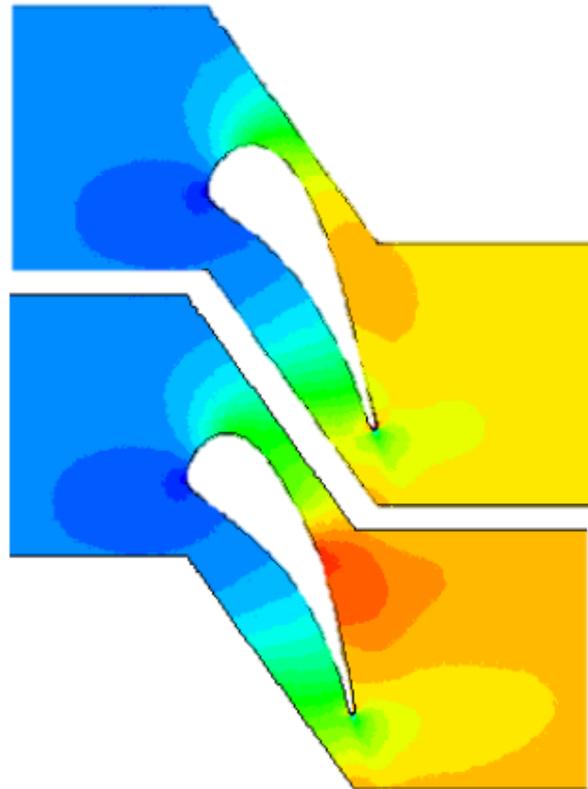
---

### 1.6.18 Shape Parameterization and Grid Adaptation using Harmonic Coordinates and their application to the Aerodynamic Design Optimization

Mavronikola (2017) presented a shape parameterization and grid adaptation methodology based on the harmonic coordinates introduced by Joshi et al. (2007). At first, given a 2D aerodynamic shape, represented by a triangular mesh (e.g. an airfoil), its control cage was constructed. Then, following the solution of Laplace's equation (1.12) under the same Dirichlet boundary conditions as applied in Joshi et al. (2007), every mesh node was assigned a number of harmonic coordinates depending on the number of control points on the cage. Afterwards, the values of the harmonic coordinates obtained on the triangular mesh nodes were interpolated to a grid for Computational Fluid Dynamics (CFD) applications.

Therefore, by the displacement of the control points, the shape of the aerodynamic figure, as well as the computational mesh, were concurrently modified. Moreover, a system of two nested control cages was proposed in order to ensure both the periodicity and smoothness of the computational mesh close to the boundaries of the internal cage. Eventually, the proposed methodology was integrated into an aerodynamic design optimization scheme.

Based on the work of Mavronikola (2017), Zervas (2018) introduced the aforementioned shape parameterization and grid adaptation technique for the case of 3D unstructured grids. The initial shape of an airfoil, as well as its deformation, occurred following the shape parameterization and grid adaptation technique introduced by Mavronikola (2017) are presented in **Figure 1.6.30**.



**Figure 1.6.30** Above: The Mach number distribution on the initial airfoil shape. Red colored areas indicate the presence of high Mach numbers compared to the blue colored areas on which low Mach numbers were obtained. Below: The deformed airfoil along and the Mach distribution around it (Mavronikola, 2017).

### 1.6.19 Summary

The purpose of the present literature review was to provide the latest scientific works concerning geometry parameterization and handling techniques based on harmonic functions. Due to the necessity for smooth and plausible shape deformation results, several methods have been proposed, either by making the use of harmonic coordinates and their approximations or harmonic fields and maps. For the purpose of the aforementioned methods to succeed, high parameterization accuracy and stability come at the price of an increased demand on computational time and algorithm complexity. In addition, further research in developing a powerful parameterization method, suitable for general geometries, is required. Furthermore, in order to construct a less

complex algorithm, the smallest possible number of parameters may be used, so as to easily define and effectively handle them throughout the entire process.

## Chapter 2

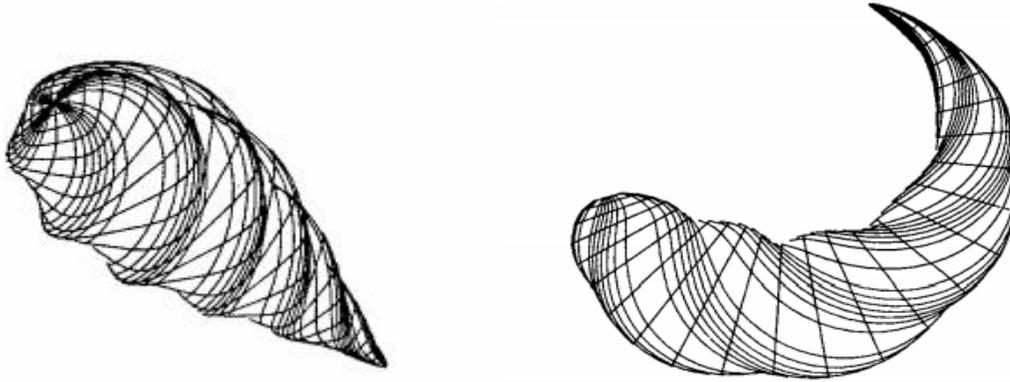
# Free Form Deformation

### 2.1 Introduction to Free Form Deformation (FFD)

Due to the ability of the indirect shape manipulation and the reduced number of design parameters handled, the Free Form Deformation represents one of the most popular and diverse parameterization methods, not only in computer graphics but also in the aerodynamic optimization field. The fundamental idea underlying the FFD algorithm is to achieve an indirect deformation of the shape at hand by embedding it into a parametric control grid (lattice); then by transforming the geometry of the particular lattice, every object enclosed to it undergoes the same deformation (Leloudas et al., 2020). Over the latest years, several FFD-based parameterization methods have been proposed. A brief introduction to the developed techniques for both computer graphics and aerodynamic shape optimization purposes is implemented in Section 2.1.1 and Sections 2.1.2, respectively.

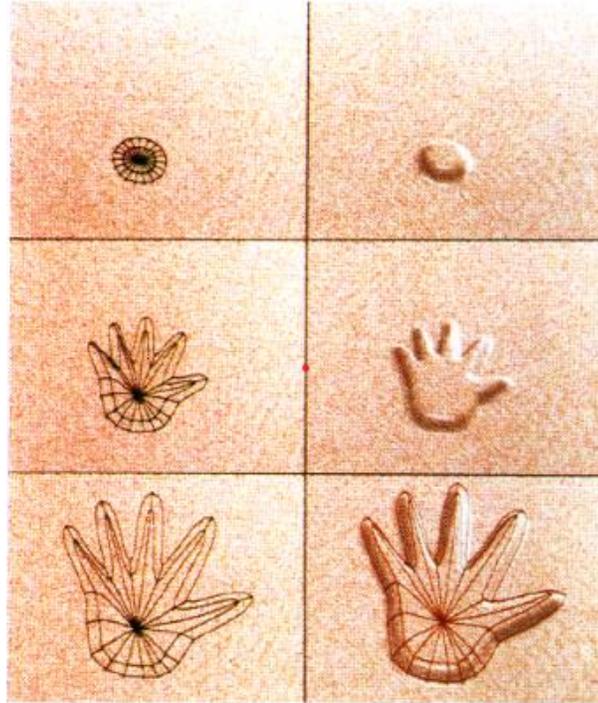
#### 2.1.1 Free Form Deformation Variations

Barr (1984) was the first to introduce a hierarchical solid modeling technique in order to achieve deformation of complex geometries. In particular, through the hierarchical combination of transformations (twisting, bending, tapering or similar) of simpler geometries, a complex shape modification is succeeded. A deformation based on the aforementioned approach is presented in **Figure 2.1.1**.



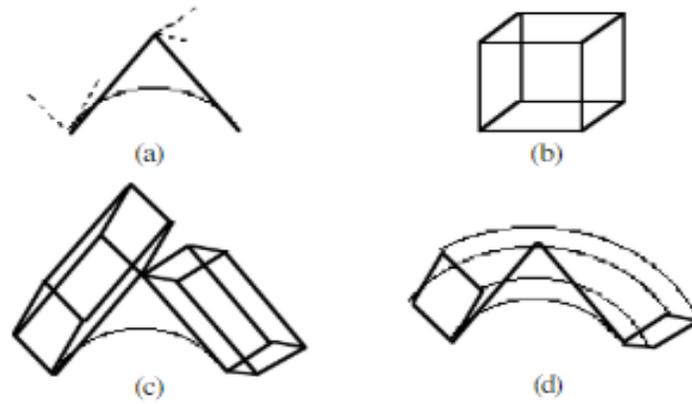
**Figure 2.1.1** Left: Twisting of a tapered primitive. Right: A bent, twisted and tapered primitive (Barr, 1984).

Later on, Sederberg and Parry (1986) proposed a general geometry modification technique for solid objects, based on Bernstein polynomials and the displacement of a set of handle points (i.e. control points). The aforementioned study was the origin of several Free Form Deformation-based techniques. Following Sederberg & Parry's methodology, Griessmair and Purgathofer (1989) implemented Free Form Deformation by making the use of trivariate B-Spline basis functions for the purpose of parametric lattice construction. Later, Coquillart (1990) introduced an efficient method of modeling and modification of cloth-like surfaces. The aim of the Extended Free Form Deformation (EFFD) was the geometry deformation, through the bending and insertion of shaped bumps on the object's surface of interest. Thus, the aforementioned deformation technique is independent of handled body geometry. In addition to EFFD, Coquillart and Jancéne (1991) introduced the Animated Free Form Deformation (AFFD) approach. In particular, the abovementioned method is based on the shape modification and subsequent motion by a sequence of independent shape transformations. In **Figure 2.1.2** an AFFD deformation is presented



**Figure 2.1.2** Disc transformation into a hand (Coquillart and Jancéne, 1991).

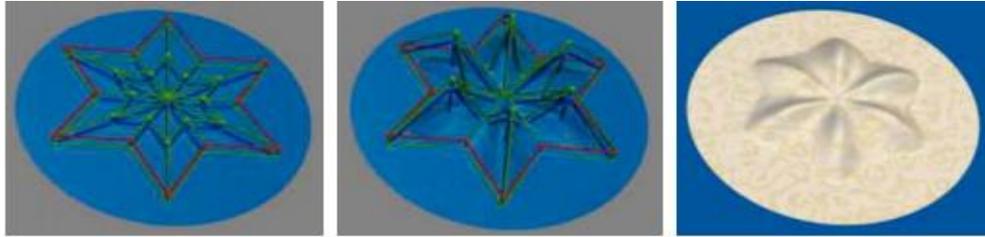
Chang and Rockwood (1994) suggested a 3D FFD method based on a generalized form of the de Casteljau algorithm. Specifically, for the purpose of succeeding deformation, by making the use of a single Bézier curve, a control polygon and a user-specified axis, several affine transformations are repeatedly applied in space. A demonstration of the Generalized de Casteljau approach of FFD is shown in **Figure 2.1.3**.



**Figure 2.1.3** (a) A single Bezier curve, a control polygon and user specified axes, (b) the object to be deformed, (c) cube mapping on each control polygon segment and (d) the initial geometry's wrapping along the Bezier curve (Chang and Rockwood, 1994).

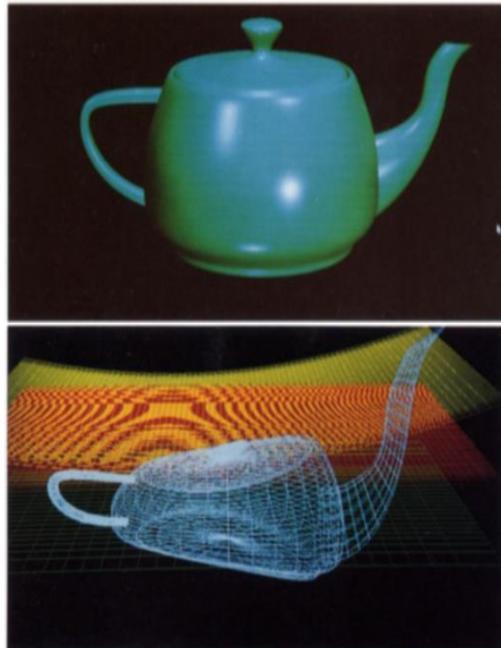
Lamousin and Waggenpack (1994), presented a FFD methodology based on the construction of B-Spline rational basis functions over a non-uniform knot vector (NURBS). Later, Samareh (1999) applied the NFFD approach in order to perform aerodynamic Computational Structural Mechanics (CMS) parameterization.

MacCracken and Joy (1996) proposed a variation of the original FFD technique based on the 3D lattice refinement into a sequence of lattices that converge uniformly to a region of 3D space, throughout the application of the Catmull-Clark subdivision methodology. As a result, any modification that occurred in the repeatedly refined lattices is transferred to the 3D space. A deformation based on MacCracken's and Joy's approach is shown in **Figure 2.1.4**.



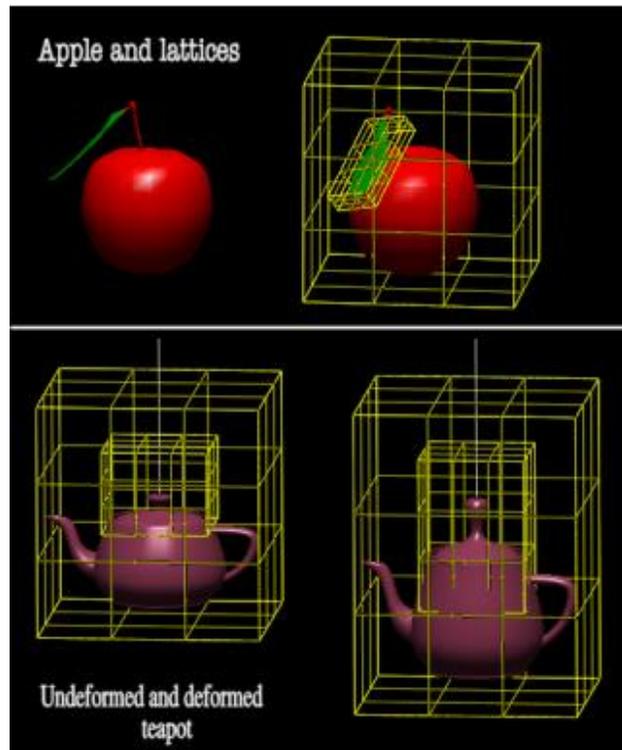
**Figure 2.1.4** Left: A star – shaped lattice located on a disk. Middle: The deformed star – shaped lattice containing the modified disk area. Right: The final geometry of the disk (MacCracken and Joy, 1996).

Feng et al. (1996) introduced a modification of the Free Form Deformation based on the combination of two parametric surfaces in order to define a 3D parametric domain. More specifically, the object of interest is mapped onto the newly-constructed 3D space. Then, following the deformation of the parametric domain, the object's geometry is, automatically, modified. **Figure 2.1.5** shows the original 3D shape of a teapot and its deformation occurred as a consequence of the individual parametric surfaces' shape modification.



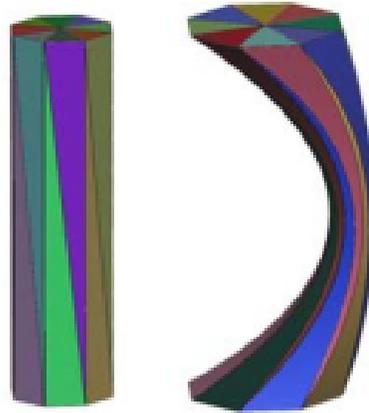
**Figure 2.1.5** Above: The original 3D shape of a teapot. Below: The deformation of the two surfaces (denoted with green and yellow colors) induce the modification of the model's geometry (Feng et al., 1996).

An additional approach based on the classical FFD is the Dynamic Free Form Deformation presented by Faloutsos et al. (1997). The aforementioned technique was capable of transforming a wide class of objects into dynamic characters. As a result, a natural deformation of the examined objects was achieved. A deformation based on the DFFD technique is presented in **Figure 2.1.6**.



**Figure 2.1.6** Local and global deformation based on DFFD technique (Faloutsos et al., 1997).

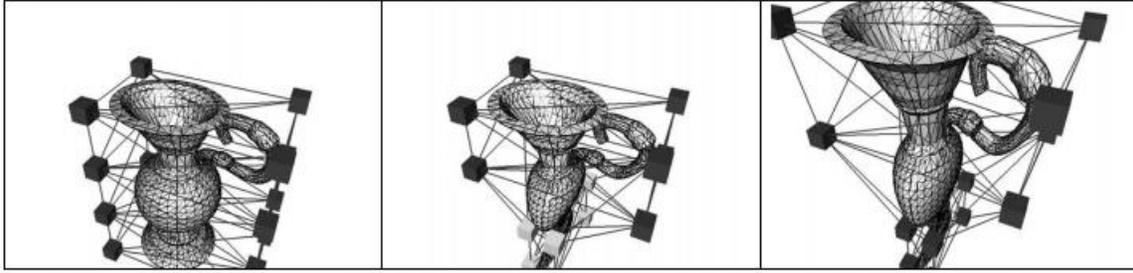
Feng et al. (2012) proposed the Accurate FFD methodology which is based on the polyhedral model deformation achieved through the sampling of a small number of control points. In particular, through the Accurate FFD technique, the polyhedral object is described by triangular Bezier patches which, subsequently, yield a B-Spline volume. As a result, despite the few representative points sampled, the deformation result appears to be similar to theoretical. An AFFD deformation is shown in **Figure 2.1.7**.



**Figure 2.1.7** Left: The initial shape. Right: The deformed geometry (Feng et al., 2012).

Based on the work of Griessmair and Purgathofer (1989), Gain and Dodgson (1999) proposed an adaptive refinement and decimation technique for the interactive Free Form Deformation. Specifically, the aim of the study was to prevent the decrease of approximation quality of polygon-mesh objects caused by the distortion which the FFD typically provokes.

Hirota et al. (2000) presented a variation of the FFD methodology by integrating the minimization of the elastic energy subject to the volume-preserving criterion. The aim of the study was compute the position of the deformed control lattice's nodes, given the boundary representation of an object and the desired deformation. Improved performance compared to the traditional FFD techniques is achieved. A deformation based on Hirota et al. (2000) approach is presented in **Figure 2.1.8**.



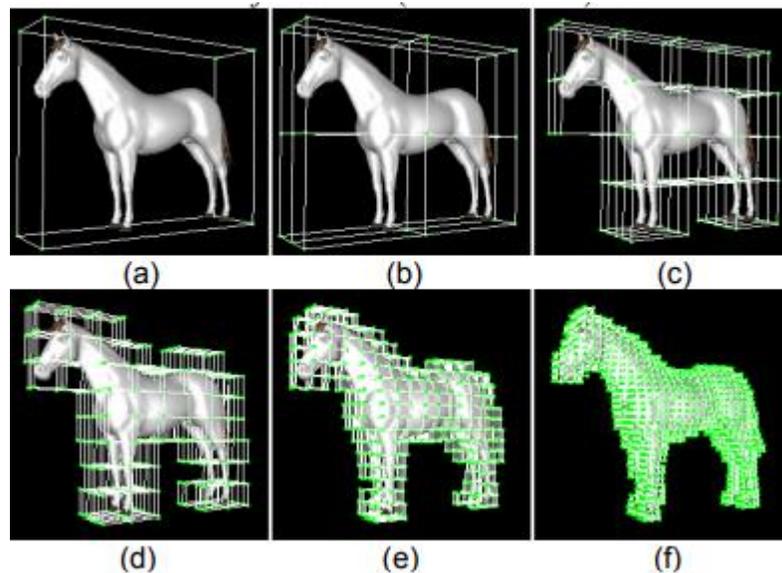
**Figure 2.1.8** Left: The initial geometry of a water pitcher. Middle and Right: Deformations resulted by water pitcher's bottom compression (Hirota et al., 2000).

Yoshizawa et al. (2002) presented a variation of Free Form Deformation based on the input of a mesh-represented geometry and a number of control points. The rationale of the methodology consists of the generation of a concatenation of deformations caused by the alteration of individual control points' location in order to, eventually, reach the desired shape. It worth noting that the present technique is not making any use of the input mesh connectivity. Thus, direct manipulation of the shape represented by multiple point datasets is feasible. A deformation that occurred through the modification of the control points' location is depicted in **Figure 2.1.9**.



**Figure 2.1.9** Left: The input geometry and six control points placed on it. Middle and Right: A sequence of deformations produced by the alteration of control points' position (Yoshizawa et al., 2002).

In order for the Free Form Deformation methodology to be effective and produce the desired both global and local deformations, the parallelepiped parametric lattice should adequately conform to the shape of the embedded object. Ono et al. (2002) presented a variant of the FFD based on the automatic generation of the parametric lattice which approximates the geometry of the object to be deformed. At first, a bounding box is automatically generated embedded in the 3D model. Afterwards, the box is hierarchically refined and a series of “*multiresolution lattices*” are generated. Then, the user is able to select the desired parametric lattice either consisted of few control points for global deformation or the one which offers the best approximation of the 3D model’s geometry for local deformations. The process of parametric lattice construction which approximates the geometry of the 3D model is shown in **Figure 2.1.10**.



**Figure 2.1.10** (a) The initial bounding box in which the 3D model is embedded. (b)-(e) The hierarchical process of the geometry approximation. (f) The generated parametric lattice (Ono et al., 2002).

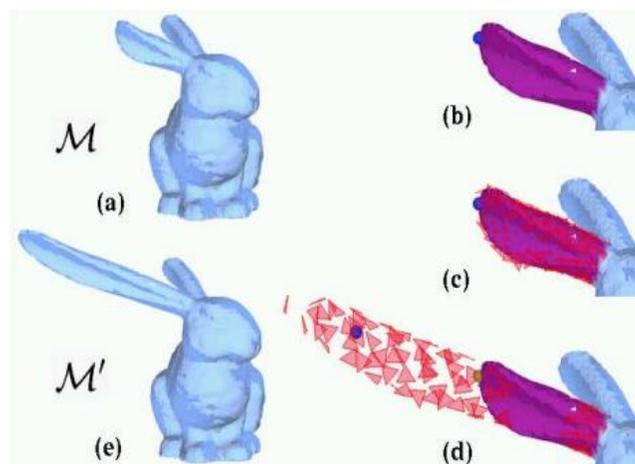
Ilic and Fua (2002) introduced the Dirichlet Free Form Deformation (DFFD) technique by making the use of arbitrary placed control points. The main idea

behind the novel Free Form Deformation (DFFD) method is the geometry deformation through the modification of a triangular cage encircling the shape of interest. The triangular control cages of a human head and ear created, through the application of DFFD, are presented in **Figure 2.1.11**.



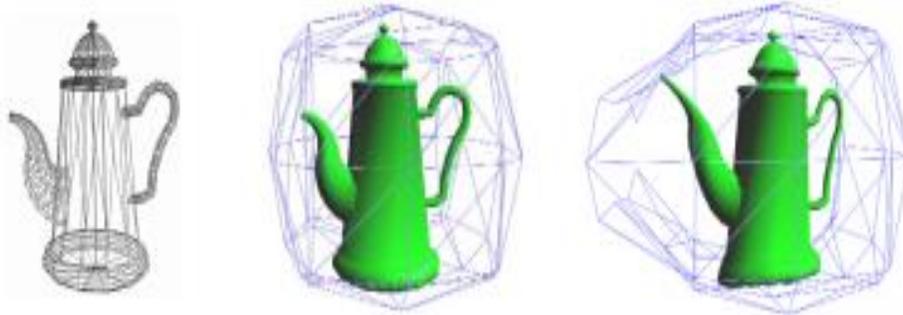
**Figure 2.1.11** Control triangulations for head (left), profile (middle) and ear (right) (Ilic and Fua, 2002).

Kobayashi and Ootsubo (2003) presented a novel FFD technique called t-FFD. According to the authors, a polygonal mesh or a point cloud is deformed through an arbitrary topology triangular control mesh. **Figure 2.1.12** demonstrates the t-FFD process.



**Figure 2.1.12** (a) The initial shape to be deformed, (b) specification of the area to be deformed, (c) construction of the initial control mesh, (d) the deformed control mesh occurred by the relocation of the blue point, (e) the deformed geometry (Kobayashi and Ootsubo, 2003).

Compared to the lattice-based traditional FFD technique, Hua and Qin (2003) made the use of the scalar field as the parametric space. Thus, through the motion of the scalar field vertices, the Free Form Deformation of the object of interest occurs. A scalar-field Free Form Deformation is presented in **Figure 2.1.13**.



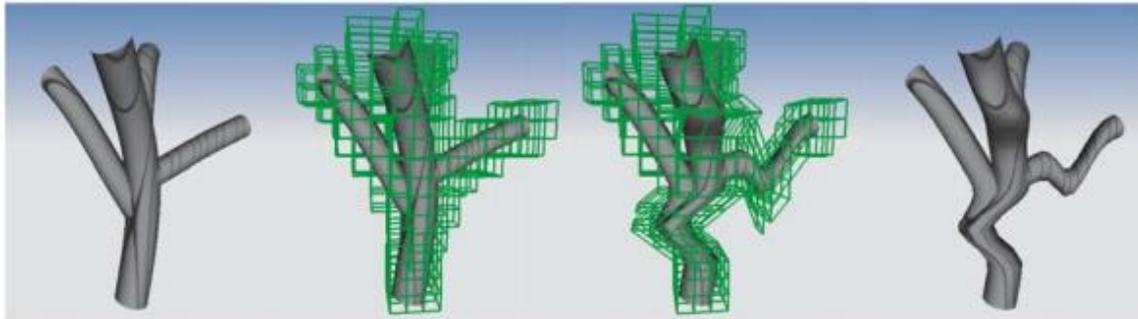
**Figure 2.1.13** Left: The initial shape of a teapot. Middle: The teapot is embedded in the scalar field. Right: The deformed shape of the teapot (Hua and Qin, 2003).

Schein and Elber (2004) introduced a variant of FFD, called Discontinuous Free Form Deformation. As stated by the authors, the particular technique offers the designer the ability to model and incorporate geometric discontinuities such as gaps and holes to the deformation process. A deformation based on Discontinuous FFD is presented in **Figure 2.1.14**.



**Figure 2.1.14** Left: The Stanford bunny model cut with a cylinder tool from above. Right: The DFFD's stitching algorithm was applied in order to close the opening (Schein and Elber, 2004).

Later on, Song and Yang (2005) proposed a novel FFD technique named w-TFFD. The aforementioned method is based on the application of - a generalization of NUBRS - the weighted T-Spline volumes and T - Junctions. According to the authors, T-Spline volumes produce flexible control lattices and yield to the adaptive deformation of objects with arbitrary topology. In addition, to approximate the object's geometry, the OCTREE algorithm is executed and the control lattice is automatically constructed. A w-TFFD process is presented in **Figure 2.1.15**.



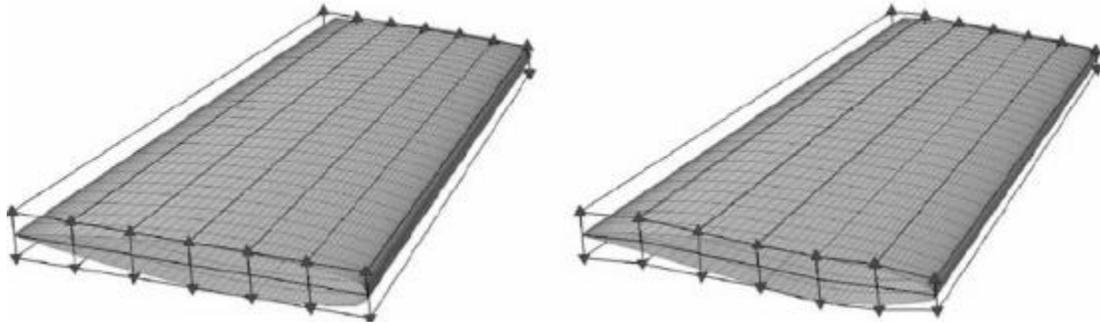
**Figure 2.1.15** Left: The initial geometry. Middle left: The initial control lattice. Middle right: The deformed control lattice. Right: The resulted deformed geometry (Song and Yang, 2005).

### 2.1.2 Free Form Deformation applications on Aerodynamic Shape Optimization

Due to the simplicity of its nature and the effectiveness on deforming either locally or globally intricate geometries, the Free Form Deformation found a plethora of applications on aerodynamic shape optimization.

Ronzheimer (2002) introduced a post-parameterization tool tailored for the existing CAD models which was based upon the FFD technique. More specifically, the present methodology is implemented on the “MegaCads” software through which a NURBS-based FFD lattice is generated and controlled. The post-parameterization tool of Ronzheimer (2002) was tested on an elbow joint, a 2D

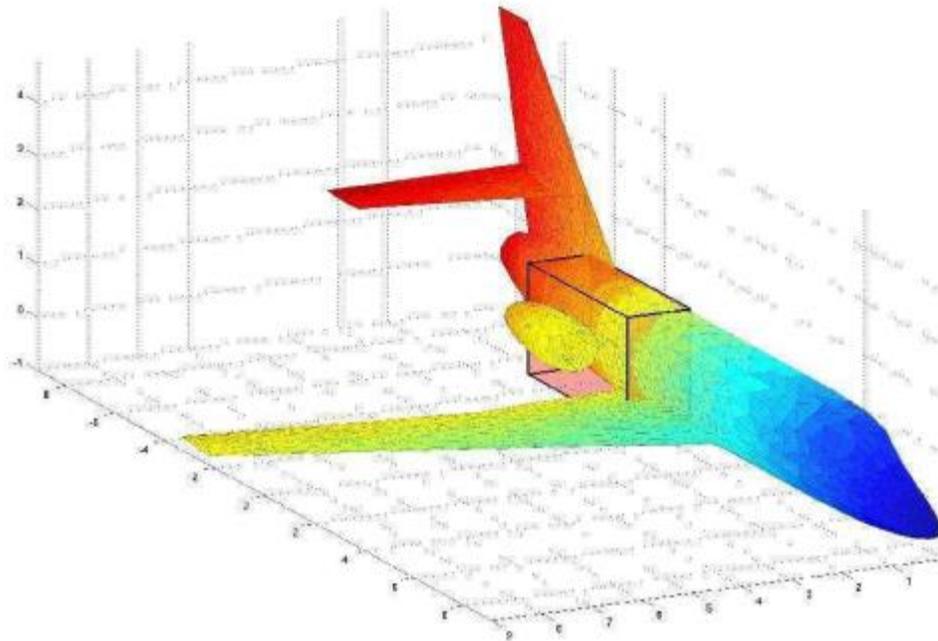
airfoil section and on the design of various wing - tip planform geometries. **Figure 2.1.16** shows the initial and deformed shapes of a 3D airfoil constructed by a surface grid embedded in the control lattice.



**Figure 2.1.16** Left: The initial shape of an airfoil constructed by a surface grid is embedded in the FFD control lattice. Right: The deformation of the FFD lattice is transferred to the embedded airfoil geometry (Ronzheimer, 2002).

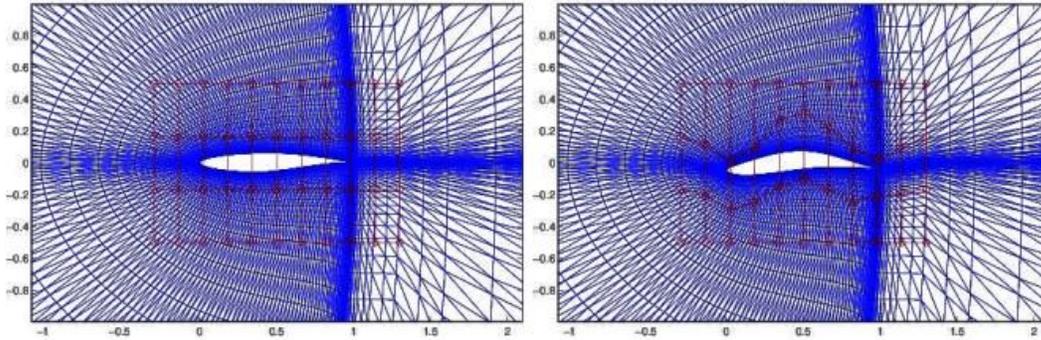
Later on, Ronzheimer (2005) applied the Free Form Deformation technique as a parameterization tool for the aerodynamic design optimization process. The introduced methodology was implemented on an airfoil aiming at acquiring its optimum aerodynamic shape. Moreover, in order to demonstrate the effectiveness of the aforementioned technique, Ronzheimer (2005) performed the inverse design of the airfoil.

Andreoli et al. (2003) applied the Free Form Deformation technique as a parameterization tool on the optimization of complex 3D aerodynamic shapes. The proposed methodology was based on the employment of 3D Bezier curves in combination with genetic algorithms and the simplex optimization methodology. Andreoli et al. (2003) implemented the aforementioned technique on airfoils, wings and general 3D aircraft models. A Free Form Deformation for an engine pylon body is shown in **Figure 2.1.17**.



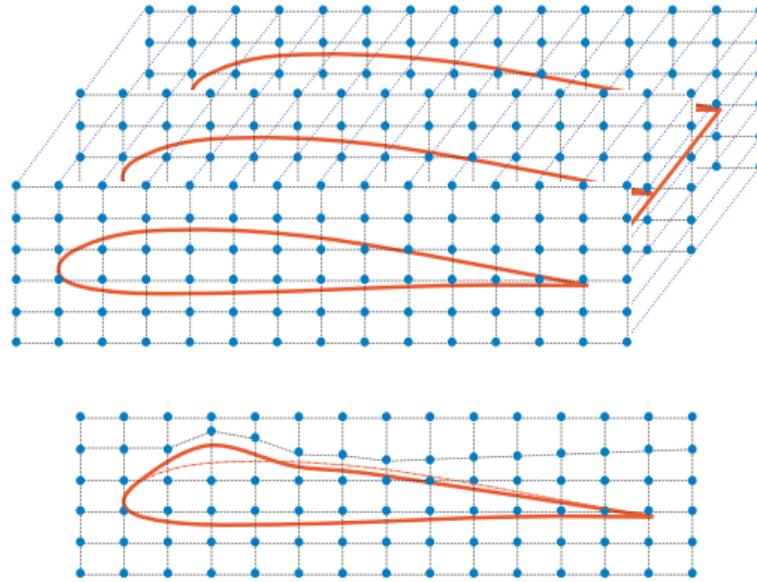
**Figure 2.1.17** The control lattice (black box) indicates the part of the aircraft volume to be deformed (Andreoli et al., 2003).

Désidéri et al. (2004) presented a shape optimization technique based on Bezier polynomials for the aerodynamic design of transonic aircrafts. More specifically, the aforementioned methodology employs a 3D unstructured grid by making the use of the Finite Elements Method (FEM) in order to obtain the solution of compressible flow. Moreover, through the particular variant of Free Form Deformation technique, the modification of the computational grid was succeeded. Additionally, during the optimization procedure, a genetic algorithm was used along with the simplex method. A deformation generated by the proposed Free Form Deformation technique is shown in **Figure 2.1.18**.



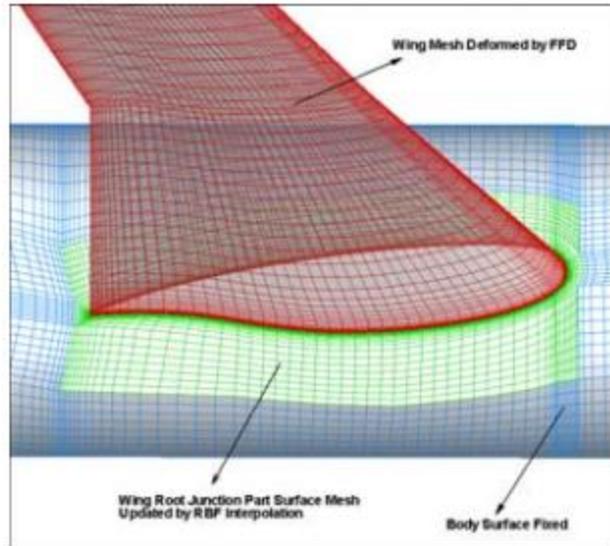
**Figure 2.1.18** Left: The airfoil is embedded in FFD control lattice (red colored). Right: The deformation of the control lattice is not only passed to the embedded airfoil but to the surrounding computational grid (blue colored), also (Désidéri et al., 2004).

Samareh (2004) introduced a variation of the classical Free Form Deformation technique based upon the Non-Uniform Rational B-Splines (NURBS) for the aerodynamic shape optimization. More specifically, the aforementioned methodology disregards the initial grid topology; therefore, it is capable of handling both structured and unstructured grids. Additionally, the grid is created once and its topology remains fixed throughout the optimization, thus any modification desired is directly applied to it. Moreover, through the proposed methodology a trivariate volume can be effectively represented by a bivariate surface resulting in the significant reduction of the design variables. A trivariate volume and its deformed shape are shown in **Figure 2.1.19**.



**Figure 2.1.19** Above: A trivariate volume deformation. Below: The deformed grid causes the deformation of the embedded airfoil (Samareh, 2004).

Liu et al. (2017) proposed a hybrid geometrical parameterization based on the combination of Radial Basis Function (RBF) and Free Form Deformation. The present study aimed to preserve the original connectivity of the junction area between the wing and the body of the aircraft despite the deformations that occurred throughout the optimization procedure. More specifically, following the application of the FFD methodology for wing parameterization and deformation, the RBF interpolation was employed in order to calculate the displacements that occurred on the wing root junction mesh nodes caused by the deformation. Then, an aerodynamic shape optimization scheme was established based upon the Cuckoo search algorithm and the Kriging surrogate model. Liu et al. (2017) demonstrated the effectiveness of the proposed technique on a DLR F4 wing body configuration. **Figure 2.1.20** shows the updated junction area between the wing and the body following the RBF interpolation.

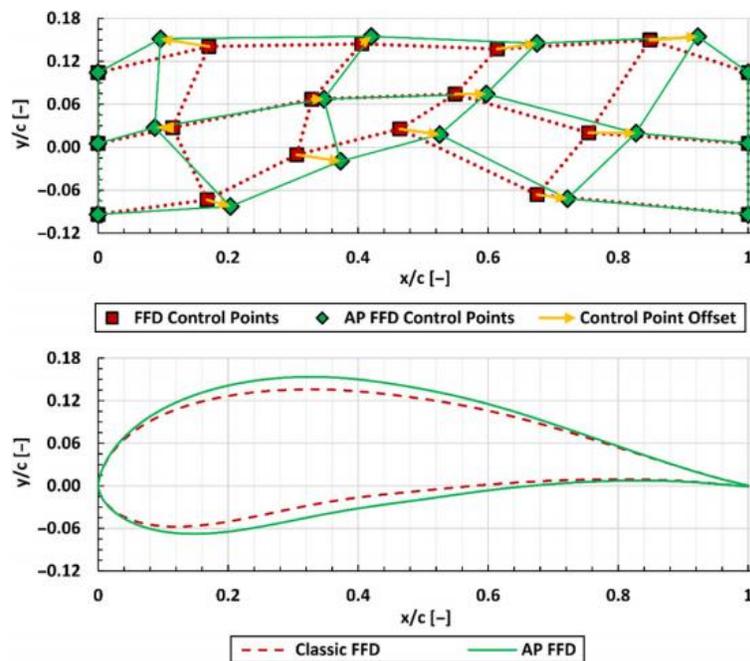


**Figure 2.1.20** The deformed wing (red colored) occurred by the application of FFD, the updated wing root junction area (green colored) following the RBF interpolation and the aircraft body surface (blue colored) which remains fixed throughout the aforementioned procedure (Liu et al., 2017).

Additionally, Bai and Chen (2013b) presented an aerodynamic optimization scheme based on the direct manipulation of the aerodynamic shape. In particular, following the establishment of both the locations and the displacements of few user-specified pilot points on the object of interest, the FFD lattice associated with, is accordingly modified resulting in direct manipulation of the object's geometry. Moreover, aiming at the reduction of the drag force, Bai and Chen (2013b) established an aerodynamic shape optimization scheme that demonstrated good feasibility.

Despite the numerous advantages that the Free Form Deformation technique offers in aerodynamic shape optimization, it should be mentioned that it is characterized by the inability to preserve the cross-sectional area of a reference shape throughout the optimization procedure. Based on the aforementioned observation, Leloudas et al. (2018) proposed the Area Preserving Free Form Deformation (AP FFD). The aim of the study was to conserve the cross-sectional

area of a reference airfoil following the consecutive deformations that occurred throughout the optimization procedure. More specifically, given the area of an initial airfoil, the minimum displacements of the FFD control points were calculated in order for the deformed airfoil shapes to recover the reference cross-sectional area. A comparison between the airfoil geometries and the control lattices formed before and following the Area-Preserving FFD is demonstrated in **Figure 2.1.21**.



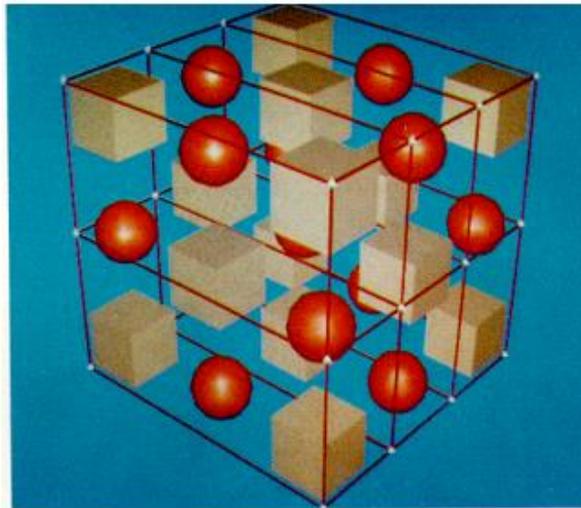
**Figure 2.1.21** Above: The initial FFD lattice (red color) and the deformed one (green color) following the application of the AP FFD. Below: The airfoil shape produced by the classical FFD (red color) and the one after the application of the AP FFD (green color) (Leloudas et al., 2018).

### 2.1.3 Free Form Deformation process

Given the numerous variations of the classical Free Form Deformation technique, the main core of the process followed remains the same. The four major steps throughout the FFD process, according to Sederberg and Parry (1986) are:

#### First Step: Construction of the parametric lattice

Initially, the object to be deformed is placed in a 2D or 3D parametric space. The constructed lattice is defined by a set of control points and parametric basis functions, establishing at the same time a local coordinate system. A visualization of the first step of the FFD technique is shown in **Figure 2.1.22**.

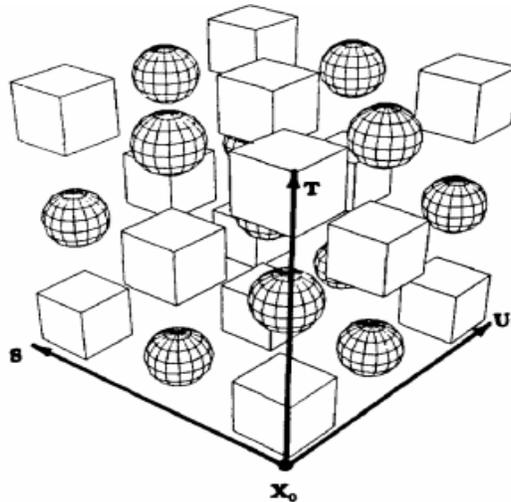


**Figure 2.1.22** The construction of the parametric lattice. The lattice consists of an ordered set of control points indicated with white color. Red colored bars indicate the neighboring control points (Sederberg and Parry, 1986).

#### Second Step: Embedding the object within the lattice

Thus, each point with  $(x, y)$  cartesian coordinates or  $(x, y, z)$  on the embedded 2D or 3D geometry, respectively, is described by  $(u, v)$  (for 2D) or  $(u, v, w)$  (for 3D) parametric coordinates. Considered that the computation of the set of parametric coordinates consists of an iterative procedure, Quadtree or Octree algorithms are

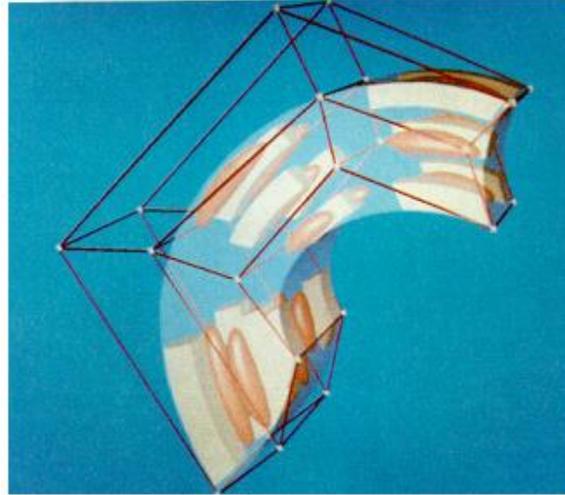
applied. The construction of the local coordinate system is depicted in **Figure 2.1.23**.



**Figure 2.1.23** The construction of the local (parametric) coordinate system (Sederberg and Parry, 1986).

### Third Step: Deformation of the Parametric Space

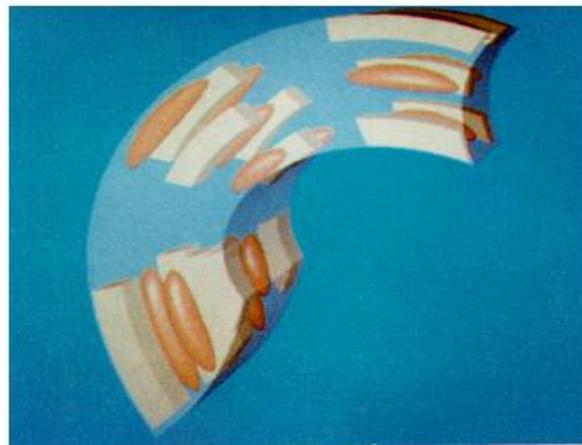
The deformation of the parametric space is specified by the adjustment of the FFD lattice control points. Given the numerous variations of the original FFD technique, the deformation process followed varies. Considered the paradigm of NURBS-based FFD (NFFD) technique (Lamousin and Waggenpack, 1994), the deformation is applied through the alteration of the weight assigned for each control point. The deformed object embedded in the lattice is presented in **Figure 2.1.24**.



**Figure 2.1.24** The modified lattice containing the deformed object (Sederberg and Parry, 1986).

**Fourth Step: Evaluating the Results of the Deformation**

The evaluation process consists of the calculation of the Cartesian Coordinates of the deformed object, given its modified geometry and its parametric coordinates. Note that the parametric coordinates of each point on the embedded shape do not change throughout the entire process compared with its Cartesian Coordinates which are affected by the modification of the control lattice. The deformed object is shown in **Figure 2.1.25**.

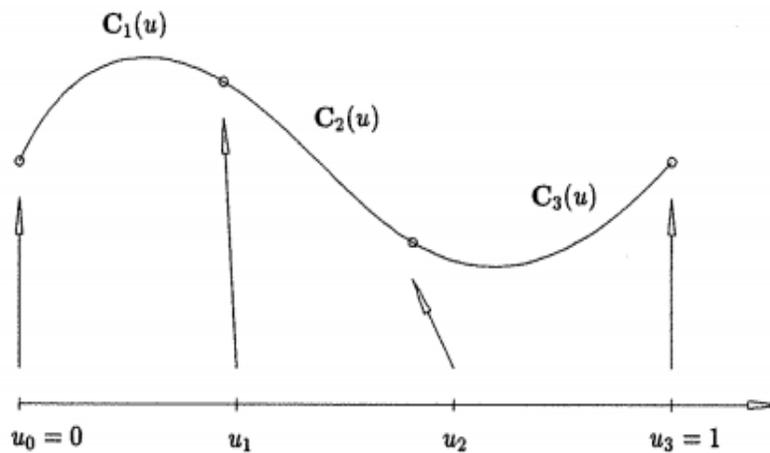


**Figure 2.1.25** A deformed object (Sederberg and Parry, 1986).

## 2.2 2D B-Spline FFD

Over the past decades, in most design optimization problems curved involved shapes were consisted of a single polynomial or rational segment. As a result, the specific process was susceptible to great inadequacies and showed numerous drawbacks. According to Piegl and Tiller (1995), the three main shortcomings included the need for higher degree curves in order to satisfy a large number of constraints and represent in an accurate way a complex geometry. In addition to the prior drawbacks, curves created by just one segment appeared to be not suitable for interactive shape design, yielding to insufficient local control.

Hence, the solution to the aforementioned problems was the usage of piecewise polynomial curves as shown in **Figure 2.2.1**.



**Figure 2.2.1** A piecewise polynomial curve consisted of three segments ( $C_i$ ) (Piegl and Tiller, 1995).

As it can be observed from **Figure 2.2.1**, a piecewise polynomial curve consists of several  $n - th$  degree polynomial segments ( $C_i(u)$ ) where  $u_i$  maps to the endpoint of each section.

Considering the necessity of a methodology that provides local controlled deformations, B-Spline surfaces appear to be a sufficient solution to the

subsequent modification a shape undergoes throughout the Free Form Deformation technique. In the particular diploma thesis, a 2D B-Spline based FFD version is utilized.

### 2.2.1 Implementation of the Procedure

Through the integration of B-Splines into the FFD methodology, the valuable properties of complex shape handling and local control are exploited. In the present section, the main theoretical background of the B-Spline surfaces and Free Form Deformation are presented.

#### *First Step: Construction of the parametric lattice*

Let  $\mathbf{S}(u, v)$  be a planar B-Spline surface defined by a bidirectional net of control points  $\mathbf{P}_{i,j} = (x_{i,j}, y_{i,j})$ , two knot vectors  $\mathbf{U}, \mathbf{V}$  and the products of the univariate B-Spline functions  $N_{i,p}, N_{j,q}$  (Piegl and Tiller, 1995) as:

$$\mathbf{S}(u, v) = \sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{i,j}, \quad (2.1)$$

where  $p \in [0, n]$  and  $q \in [0, m]$  refer to the degree of basis functions.

Furthermore, let  $N_{i,p}(u)$  be a B-Spline basis function of  $p^{\text{th}}$  degree in  $u$  direction and  $N_{j,q}(v)$  be a B-Spline basis function of  $q^{\text{th}}$  degree in  $v$  direction defined over two non-periodic and non-uniform knot vectors  $\mathbf{U}, \mathbf{V}$ , respectively.

The value of each knot vector can be calculated (provided in the particular segment for the  $u$  direction) as:

$$u_i = \begin{cases} 0, & 0 \leq i \leq p + 1 \\ i - p, & p + 1 \leq i \leq n + 1 \\ n - p - 1, & n + 1 \leq i \leq n + p + 1 \end{cases}. \quad (2.2)$$

Knot vector  $\mathbf{V}$  is likewise computed.

Therefore, through the application of deBoor, Cox and Mansfield recurrence formula, the  $i^{\text{th}}$  B-Spline basis functions of  $p^{\text{th}}$  degree in  $u$  direction are obtained:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (2.3)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u). \quad (2.4)$$

Basis functions for  $v$  direction are similarly calculated.

Eventually, the Cartesian Coordinates  $(x_t, y_t)$  of an arbitrary point inside the parametric space are obtained, by making the use of its Parametric Coordinates  $(u_t, v_t)$  as follows:

$$\mathbf{R}(u, v) = \frac{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v) \mathbf{P}_{i,j}}{\sum_{i=0}^n \sum_{j=0}^m N_{i,p}(u) N_{j,q}(v)}. \quad (2.5)$$

**Second Step: Embedding the object within the lattice**

Considering that the object to be deformed is initially defined by its Cartesian Coordinates, an iterative procedure is followed, in order to obtain its parametric coordinates, necessary for the implementation of FFD. The calculation of  $(u_t, v_t)$  parametric coordinates for each point of the shape of interest is performed by the application of the QUADTREE algorithm in 2D.

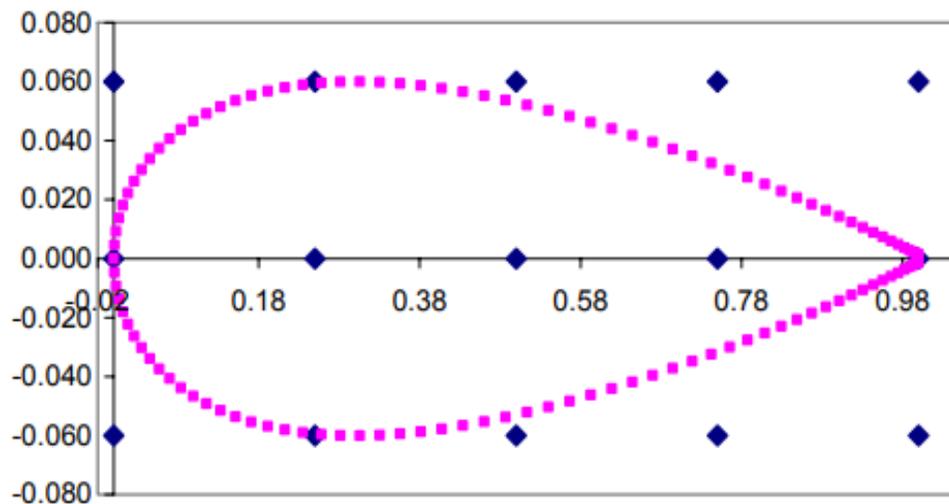
For each point on the object the following procedure is recurrently applied:

- i. The parametric surface is divided in four equal subareas.
- ii. The Cartesian Coordinates of each subarea vertex are calculated by making the use of Equation (2.5).

- iii. Obtained Cartesian Coordinates resulted from step (ii), are compared to the Cartesian Coordinates of each point of the object of interest, for the purpose of locating the subarea to which it belongs.
- iv. Once the latter subarea is found, is now divided in four equally created subareas and steps ii-iv are repeated until a certain number of subdivisions is reached or a desirable accuracy is achieved. Therefore, the Parametric Coordinates of the object's point under study are defined as the Parametric Coordinates of the center of the last calculated subarea in which it lies.

### Third Step: Deformation of the Parametric Space

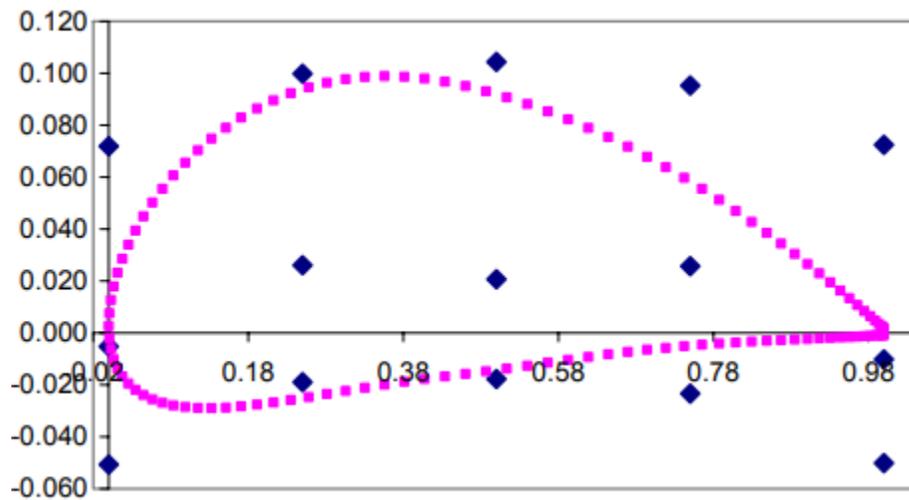
Having calculated the Parametric Coordinates of each point of the object under consideration, the Free Form Deformation technique is implemented. Thus, in order to deform the embedded shape in the FFD lattice, the positions of B-Splines control points are altered. The initial airfoil geometry embedded in the parametric lattice is shown in **Figure 2.2.2**.



**Figure 2.2.2** Initial geometry of the embedded airfoil in the lattice (Amoiralis, 2005).

***Fourth Step: Evaluating the Results of the Deformation***

Through the relocation of the control points, the new Cartesian Coordinates of an arbitrary point in the parametric space can be calculated by the application of equation (2.5). It should be noted that throughout the deformation procedure, the parametric coordinates  $(u_t, v_t)$  of each point on the object, do not change in comparison with its Cartesian Coordinates  $(x_t, y_t)$ . The deformed shape of the airfoil is presented in **Figure 2.2.3**.



**Figure 2.2.3** Deformed geometry of the embedded airfoil in the lattice (Amoiralis, 2005).

## Chapter 3

# Mesh morphing and shape deformation through the application of B-Spline Basis Functions as Harmonic Functions

### 3.1 Introduction to Harmonic Functions

Nowadays, a great deal of engineering and physical phenomena, such as heat conduction, the diffusion process, or even the way the sound, the light and the waves are propagating through space, can be effectively formulated by Partial Differential Equations (PDEs). Laplace's equation:

$$\nabla h_i^2 = \frac{\partial^2 h_i}{\partial x^2} + \frac{\partial^2 h_i}{\partial y^2} = 0,$$

constitutes a typical example of PDEs which, for instance, models the steady-state temperature distribution, the flow of a fluid and electrostatic potentials. As the name betrays, the Laplacian was named in honor of the French scholar and polymath Pierre-Simon Laplace (1749 - 1827), who was the first to study its properties. By its nature, according to Asmar and Grafakos (2018), the Laplacian of a function measures the difference between the value of the function on a certain point and the average value of the same in a neighborhood of it. Therefore, those functions, whose values do not considerably fluctuate, have a lower Laplacian. As a result, the real-valued functions which are twice differentiable  $h: \Omega \rightarrow \mathbb{R}$  and satisfy Laplace's equation in the open subset  $\Omega$  of  $\mathbb{R}^n$ , are referred as "*Harmonic Functions*" and vary in a quite regular way. Thus, every point's location in the open

subset, may be written as a linear combination of the resulted harmonic functions  $h_i$  corresponding to a control point  $P_i$ , according to Equation (3.1):

$$p = \sum_{i=0}^n h_i(p) \mathbf{P}_i . \quad (3.1)$$

## 3.2 Harmonic Functions properties

The three major properties of Harmonic Functions are:

- the Mean-Value principle (cf. Section 3.2.1),
- the Maximum/Minimum principle (cf. Section 3.2.2),
- the Uniqueness property (cf. Section 3.2.3).

### 3.2.1 The Mean-Value principle

Theorem: Let  $h$  be a harmonic function in a region  $\Omega$ . Then  $h$  satisfies the Mean-Value property in the following sense: If  $z$  is in  $\Omega$  and the close disk  $B_r(z)$  ( $r > 0$ ) is contained in  $\Omega$ , then

$$h(z) = \frac{1}{2\pi} \int_0^{2\pi} h(z + re^{it}) dt. \quad (3.2)$$

Alternatively stated, the abovementioned expression refers to the fact that the value of a harmonic function  $h$  at the center of the close disk  $B_r$  is equal to the average of  $h$  values on the surface of it (Asmar and Grafakos, 2018).

### 3.2.2 The Maximum/Minimum modulus principle

Theorem: Suppose that  $h$  is a real-valued harmonic function on a region  $\Omega$ . If  $h$  attains a maximum and minimum in  $\Omega$ , then  $h$  is constant in  $\Omega$  (Asmar and Grafakos, 2018).

### 3.2.3 The Uniqueness property

Corollary: Resulted from the Maximum/Minimum principle, the aforementioned property states that, given the function values on the boundary of a region, there is a single harmonic function defined on the particular domain with the specific boundary values (Axler et al., 1992).

### 3.3 Dirichlet problem

In order to obtain the function which acts as a solution of a Partial Differential Equation in a given region, a set of known values of the aforementioned function must perform as a condition on the boundary of the domain. The aforementioned condition is called a “*boundary condition*” and the problem involving the PDE along with the specified boundary conditions is called the “*boundary value problem*” (Asmar and Grafakos, 2018). Therefore, the boundary value problem of acquiring the solution of Laplace’s equation given a set of boundary function values is called the “*Dirichlet problem*”.

### 3.4 B-Spline Curves

Considering the main shortcomings of producing and handling a single polynomial or rational segment curves presented in Section 2.2, polynomial curves that offer local support had been the main focus of interest over the past decades. The present study is focused on the B-Spline polynomial curves implementation.

According to Piegl and Tiller (1995), a  $p - th$  degree B-Spline curve is defined by

$$\mathbf{C}(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i, \quad a \leq u \leq b, \quad i = 0, \dots, n \quad (3.3)$$

where  $P_i$  are the control points and the  $N_{i,p}$  are the  $p - th$  degree B-Spline basis functions. In addition, the polygon formed by the control points  $P_i$  is called *control polygon*.

As mentioned in Section 2.2.1 (Equations 2.3 and 2.4), B-Splines basis functions are defined as follows:

$$N_{i,0}(u) = \begin{cases} 1 & \text{if } u_i \leq u \leq u_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad (3.4)$$

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{i+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u) \quad (3.5)$$

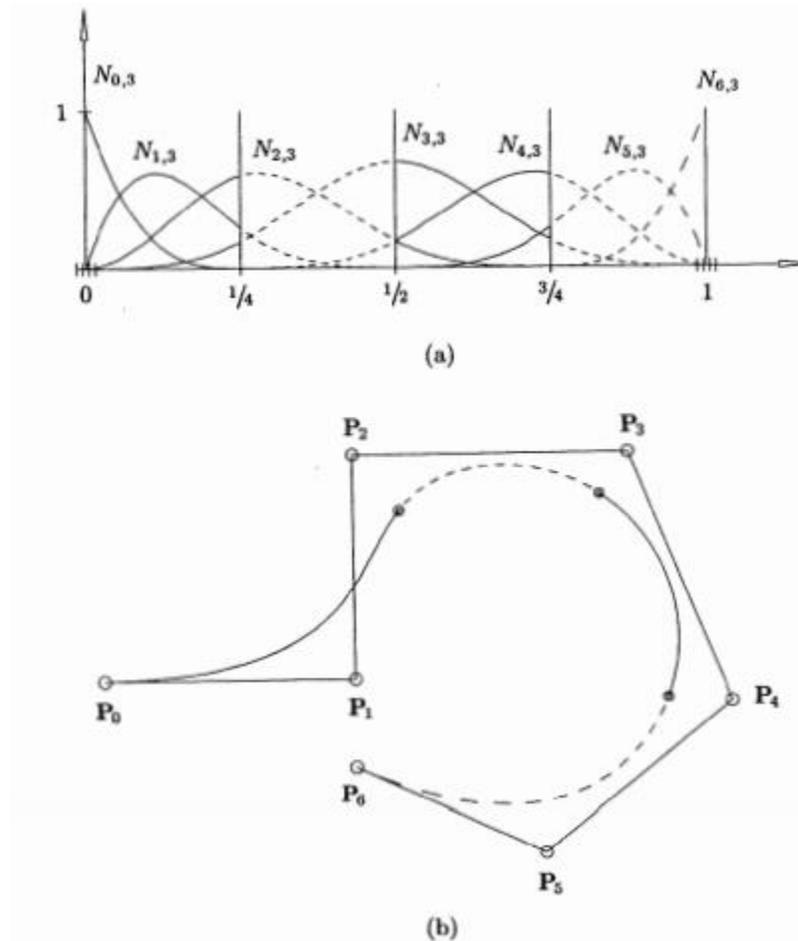
$$N_{i,p} \geq 0 \quad \text{for all } i, p \text{ and } u, \quad (3.6)$$

over a non-periodic knot vector

$$\mathbf{U} = \{a, \dots, a, u_{p+1}, \dots, u_{m-p-1}, b, \dots, b\}, \quad (3.7)$$

where  $m + 1$  is the total number of knots. It should be mentioned that, in several cases, the first and the last knots have a  $k > 1$  multiplicity. In the instance of the

knot vector presented in Expression (3.7),  $a$  and  $b$  knots have a  $p + 1$  multiplicity. An example of cubic B-Spline basis functions on  $U = \{0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1\}$ , as well as the generated cubic curve, are shown in **Figure 3.4.1**.



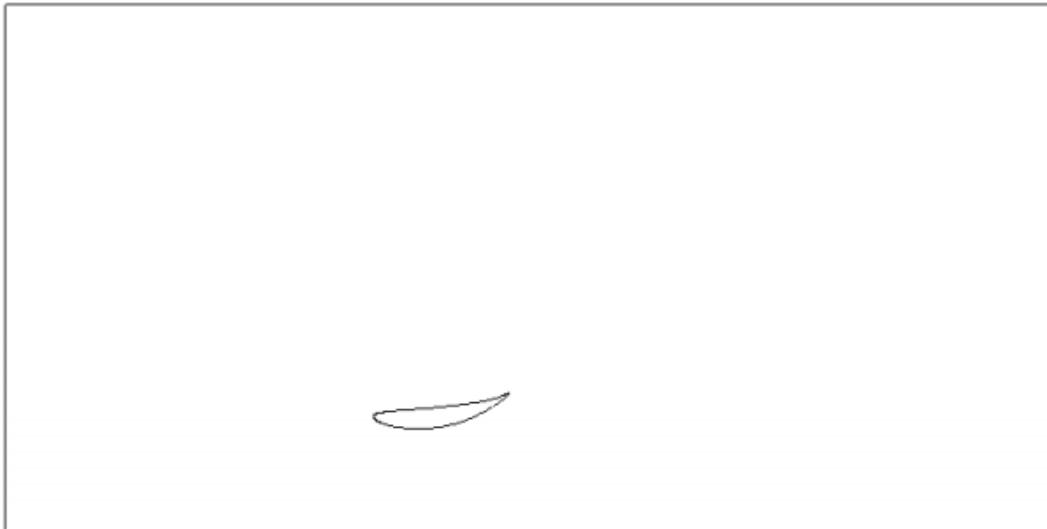
**Figure 3.4.1** (a) Cubic B – Splines  $U = \{0, 0, 0, 0, 1/4, 1/2, 3/4, 1, 1, 1, 1\}$ , (b) Cubic curve using the Basis Functions of (a) (Piegl and Tiller, 1995).

### 3.5 Parameterization in 2D

In the following, the deformation method, developed in the Turbomachines & Fluid Dynamics Laboratory of the Technical University of Crete, will be presented.

In order to perform deformation, the object of interest must be properly parameterized. In other words, the shape to be deformed may be described and, subsequently, modified by making the use of a set of geometrical parameters.

To begin with, consider a computational domain  $\Omega$  and a 2D geometry defined by a B-Spline curve, as shown in **Figure 3.5.1**.



**Figure 3.5.1** The computational domain and the embedded shape of interest (airfoil) constructed by a B – Spline curve.

Furthermore, let the B-Spline curve define one of the boundaries  $\partial\Omega_j$  of  $\Omega$  computational domain. It worth noting that, the constructed boundaries can be either internal (in most cases) or external, while the rest can be defined by different curve types and remain fixed during the parameterization and optimization procedures.

Moreover, consider the initial position  $\mathbf{C}^0(u)$  of a discrete point of a B-Spline curve corresponding to a parameter  $u$ . From B-Spline curve definition is:

$$\mathbf{C}^0(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i^0,$$

where  $\mathbf{P}_i^0$  is the initial position of the the  $i - th$  control point.

In order to properly parameterize the shape of interest, consider the Dirichlet problem. Given the Laplace's equation, a unique continuous harmonic function  $h$  twice continuously differentiable in the interior of the domain  $\Omega$  and continuous on the boundary  $\partial\Omega$  is sought, under a specific boundary condition.

Therefore, for the calculation of the  $i^{th}$  Harmonic Function  $h_i$  of a point  $p(x, y)$  inside the domain, the solution of the Laplace's equation:

$$\Delta h_i(p) = \nabla h_i^2 = \frac{\partial^2 h_i}{\partial x^2} + \frac{\partial^2 h_i}{\partial y^2} = 0, \quad (3.8)$$

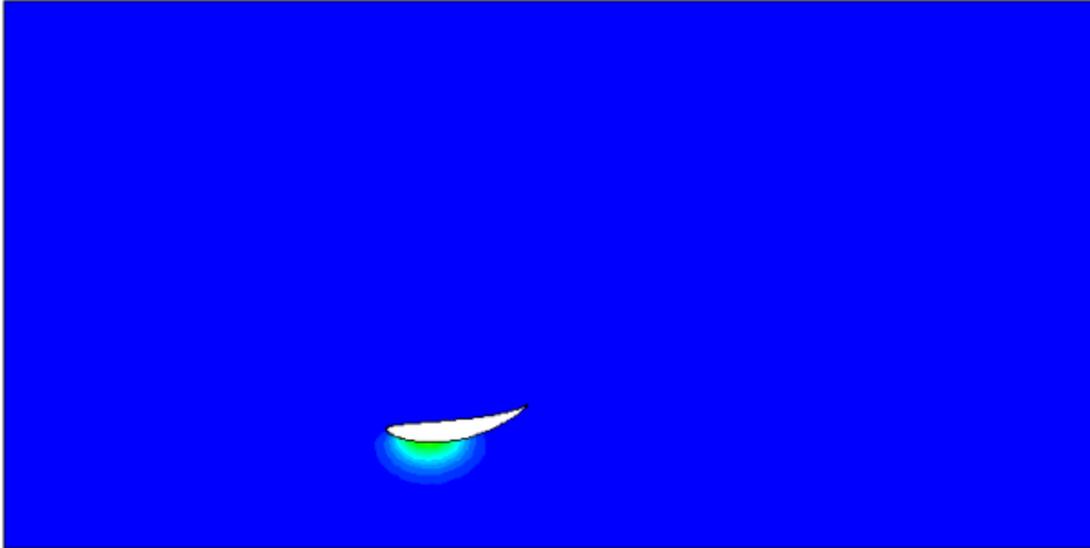
is examined, under the boundary condition:

$$h_i(\mathbf{C}^0(u_l)) = N_{i,p}, \quad a \leq u \leq b, \quad i = 0, \dots, n \quad (3.9)$$

where  $u_l$  is the boundary grid node on which Equation (3.9) is applied.

Alternatively stated, for the purpose of computing the Harmonic Functions on every domain point  $p$ , the Laplace's equation is  $n + 1$  times solved, one for each boundary curve  $N_{i,p}$  on a specific boundary grid node at a time. It should be mentioned that, during the calculation of Laplace's equation solution, if  $u_l$  lies outside the interval  $[u_i, u_{i+p+1}]$ , then  $N_{i,p} = 0$  and, subsequently, the Harmonic Function  $h_i(\mathbf{C}^0(u_l)) = 0$ . For instance, consider a 3<sup>rd</sup> degree B-Spline curve. The 3<sup>rd</sup>

harmonic function values calculated across the computational domain, with the Dirichlet boundary condition  $h_3(C^0(u)) = N_{3,3}$ , are shown in **Figure 3.5.2**.



**Figure 3.5.2** The discrete solution  $h_3$  resulted for the Basis Function  $N_{3,3}$ , applied as a Dirichlet boundary condition on the B-spline boundary of the computational domain.

### 3.6 Deformation in 2D

Given the calculated Harmonic Function values on each point inside the computational domain and the B-Spline boundary curve, the propagation of discrete movements applied on the B - Spline control points across the entire space is desired.

If  $\mathbf{C}^1(u)$  denotes the final position of a boundary grid node and  $\Delta C(u) = \mathbf{C}^1(u) - \mathbf{C}^0(u)$  is its displacement, according to the B-Spline curve definition:

$$\begin{aligned} \Delta C(u) &= \mathbf{C}^1(u) - \mathbf{C}^0(u) = \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i^1 - \sum_{i=0}^n N_{i,p}(u) \mathbf{P}_i^0 \\ &= \sum_{i=0}^n N_{i,p}(u) (\mathbf{P}_i^1 - \mathbf{P}_i^0) = \sum_{i=0}^n N_{i,p}(u) \Delta P_i \end{aligned} \quad (3.10)$$

where  $\Delta P_i = (\mathbf{P}_i^1 - \mathbf{P}_i^0)$ .

Let  $\mathbf{r}$  be the position vector of a point  $p$  inside the computational domain. If  $h_i(\mathbf{r})$  is the value of the  $i - th$  harmonic function on point  $p$  with  $r$  position vector, the interpolation of any movement of the  $i - th$  control point across the computational domain, can be applied through the Equation (3.11):

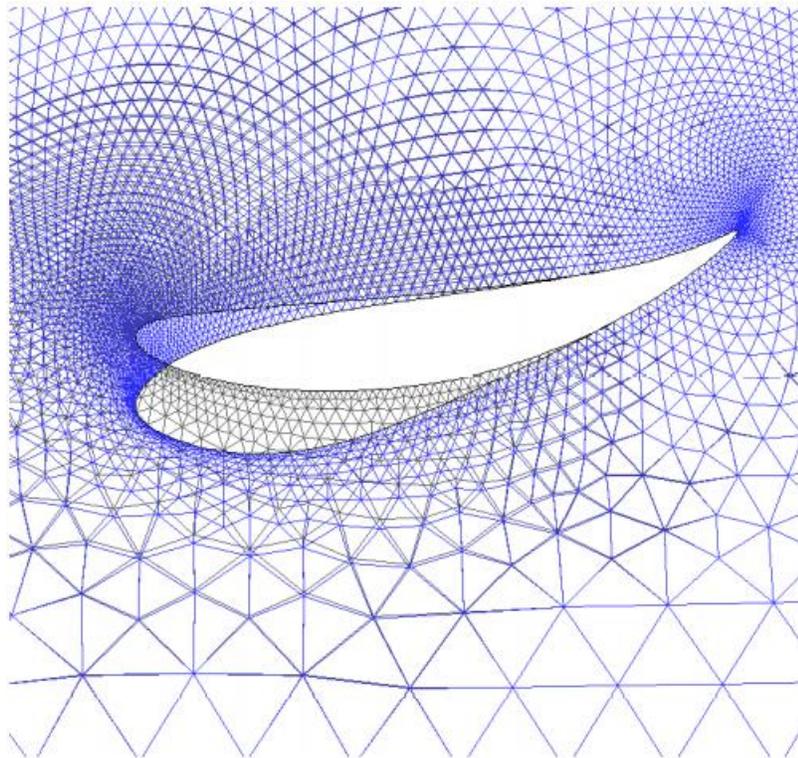
$$\Delta C(\mathbf{r}) = \sum_{i=0}^n h_i(\mathbf{r}) \Delta P_i . \quad (3.11)$$

However, in the special case of a discrete boundary node, Equation (3.11) can be formulated as:

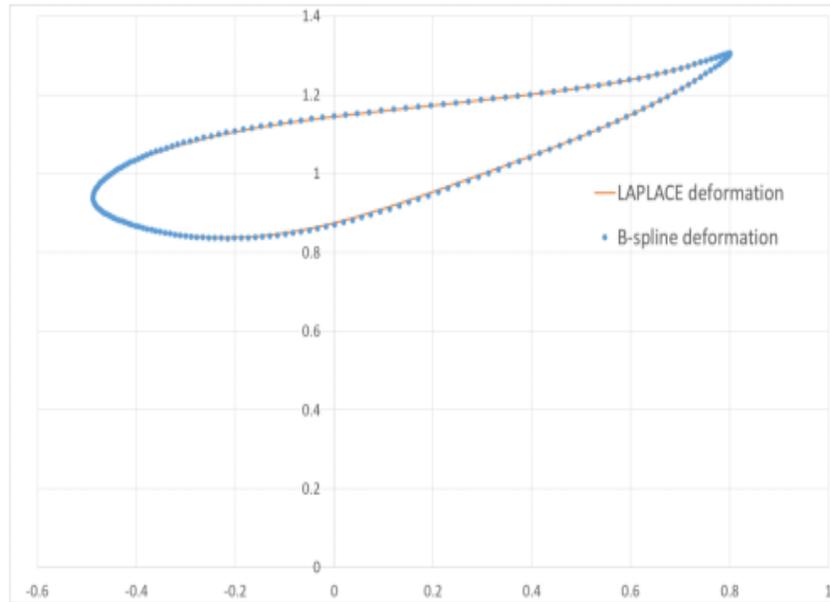
$$\Delta C(\mathbf{r}(u_l)) = \sum_{i=0}^n h_i(\mathbf{r}(u_l)) \Delta P_i = \sum_{i=0}^n N_{i,p}(u_l) \Delta P_i . \quad (3.12)$$

That is to say, any deformation of the B-Spline boundary that occurred through the movement of the curve's control points can be successfully propagated to the interior of the computational domain. Thus, the proposed methodology results in the concurrent and conformable modification of the B-Spline boundary and the entire computational mesh.

A deformation resulted through the proposed methodology and a comparison between the B-Spline and the Harmonic Functions procedures are shown in **Figure 3.6.1** and **Figure 3.6.2**, accordingly.



**Figure 3.6.1** The resulted deformation (blue color) of B-Spline boundary curve and the subsequent modification of the surrounding computational grid occurred through the movement of the three control points.



**Figure 3.6.2** A comparison between the deformations occurred, by the alteration of three control points, through the B – Spline and Harmonic Functions procedures.

### 3.7 Conclusion

In conclusion, it worth noting that the proposed methodology is applicable not only to mesh morphing and design optimization but appears to have a great potential in the graphic design field. Compared to the Harmonic Coordinates deformation technique, the Harmonic Functions calculated in the present study are combined with the B-Spline control points' displacement vectors. As a result, the aforementioned methodology provides a direct manipulation of curved boundaries rather than the necessity of linear boundaries existence in order to be applied (e.g. control polygons).

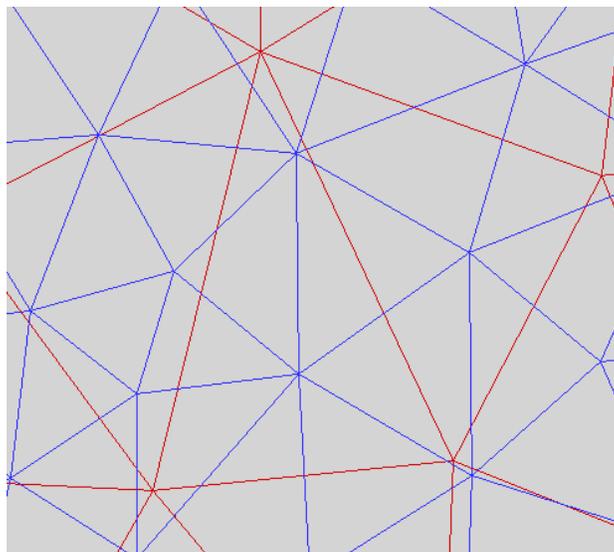
Furthermore, a different and denser computational grid (e.g. for the solution of the flow equations) can be deformed, through the interpolation of the harmonic function values resulted from the coarse to the fine mesh. Alternatively, Laplace's equation may be solved directly to the densest mesh, involving-at the same time-a higher computational cost, spent once at the beginning of the procedure.

## Chapter 4

### Mesh Interpolation

#### 4.1 Introduction to Mesh Interpolation

During the aerodynamic shape optimization, the aim is to obtain accurate and robust solutions of flow equations around the continuously modified geometry. The computational mesh is applied for the purpose of space discretization around the object of interest. Afterwards, numerical flow simulations are implemented in order to evaluate each individual alternative and select, eventually, the optimal geometrical solution. In many cases interpolation of flow quantities between grids of different densities is required. An example of the final (destination) grid superimposed on the initial (source) mesh is shown in **Figure 4.1.1**.



**Figure 4.1.1** In mesh interpolation problems, data from the source mesh (blue color) must be mapped to the destination mesh (red color).

Within the framework of the present dissertation, an interpolation scheme, based on the Barycentric Coordinates proposed by Möbius (1827) and the Inverse Distance Weighting function introduced by Shepard (1968), is implemented. The proposed interpolation methodology consists of three major steps. Initially, the elements of the source mesh are divided into two equally sized subsets. In the second part, for each node of the superimposed destination mesh, the source mesh element, which belongs to, is located and barycentric interpolation is performed. Then, a list of detected and unperceived destination points is created. Finally, for each not located destination node, its distances from every detected node of the destination mesh are computed. Therefore, according to the calculated distances, further Barycentric or Inverse Distance Weighting (IDW) interpolations are performed.

## 4.2 Algorithm Analysis

### 4.2.1 Division of the source mesh

Based on the rationale of the Kd-tree theory mentioned in de Berg et al. (2008), the aim of the first step is the partition of the source mesh elements into two approximately equally-sized groups.

To begin with, for the purpose of performing the necessary operations following, a *.txt* file containing all the source mesh information is read. An example of the file structure is shown in **Figure 4.2.1**.

ELEMENT ID	X1	Y1	X2	Y2	X3	Y3	X4	Y4
16082	31102	0						
1	1.95885487100	0.19760304230	1.96124740200	0.19518767620	1.96322934900	0.19714043070	0	0
2	0.00301769153	0.44331047660	-0.00000000000	0.44000000000	0.00408989372	0.44000000450	0	0
3	1.99687843800	0.19392515590	2.00000000000	0.19000000000	2.00000000000	0.19387096770	0	0
4	1.99697987900	0.30668955350	2.00000000000	0.31000000000	1.99590224500	0.31000000000	0	0
5	0.00313481521	0.55612092740	0.00000000000	0.56000000140	0.00000000000	0.55612903360	0	0
6	1.99647925900	0.19863407010	1.99447172800	0.19591019460	1.99687843800	0.19392515590	0	0

**Figure 4.2.1** Format of input *.txt* file containing source mesh information.

As shown in **Figure 4.2.1**, the first line of the input file contains the total number of source mesh nodes, the number of triangular elements and, finally, the number of quadrilateral elements contained in the examined grid. In the case of **Figure 4.2.1**, the total number of mesh nodes equals to 16082, the number of triangular elements is equal to 31102 and the examined grid is constructed by 0 quadrilaterals. As it can be seen in **Figure 4.2.1**, the particular example consists of a triangular mesh as the input of the algorithm.

Then, considering that the second line indicates the notation for each column of the *.txt* file, each further line contains all the information needed about every element of the source mesh. Therefore, the first column represents the ID of the particular element, the second and the third column indicate the X and Y

Cartesian coordinates of the first element node, the fourth and the fifth column contain the aforementioned coordinates of the second element node and so on. It should be noted that, in the case of a triangular mesh, as presented in **Figure 4.2.1**, the last two columns (eighth and ninth) are equal to zero. However, in the case of a quadrilateral mesh as the input of the algorithm, the latter two columns represent the fourth element node's coordinates, which in that case are non-zero.

Consequently, given the necessary information about the elements which construct the source mesh, a series of operations are performed that will, later on, lead to the division of the space into two equally-sized subareas. At first, X coordinates of each element which is closest to zero (minimum) and most distant from it (maximum) are obtained. Afterward, all the minimum X coordinates, gathered from the previous step, are sorted, by making the use of the Bubble Sort algorithm (Astrachan, 2003). Hence, borrowed by the Kd-tree partitioning rule, for the purpose of dividing the examined mesh, the median of the sorted minimum coordinates is computed. The median of an even number of observations, when the sample is ordered, is calculated through the application of Equation (4.1):

$$median_{even} = \frac{1}{2} (x_{[n/2]} + x_{[n/2+1]}), \quad (4.1)$$

where  $x_{[n/2]}$  indicate the  $n/2$  observation and  $n$  the total number of observations. On the other hand, in the case of an odd length sample list, the median equals to the midmost observation:

$$median_{odd} = x_{[n/2+1/2]}. \quad (4.2)$$

Eventually, by the median computation, the mesh area is partitioned. Thus, each element whose minimum coordinate is less than or equal to the calculated median is relocated to the “left” subarea. On the contrary, the rest of the elements and those with minimum X coordinate greater than the median, are assigned to the “right” subarea. Furthermore, the size of each sublist is returned. Therefore, in the case of an even number ( $n$ ) of elements, the left and right sublist lengths are equal to:

$$length_{left} = length_{right} = n/2. \quad (4.3)$$

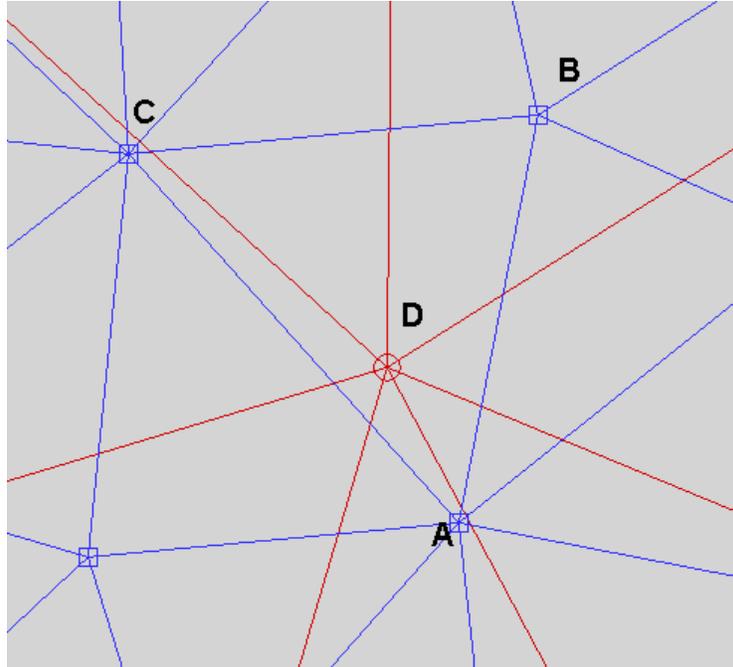
However, in the event of an odd number of elements, the left and right sublist length is computed as follows:

$$length_{right} = n/2 - 1/2 \quad (4.4)$$

$$length_{left} = n - length_{right}. \quad (4.5)$$

#### 4.2.2 Point-in-Element check and Data Interpolation

In Aerodynamic Shape Optimization applications, the examined geometry is subject to multiple deformations. Hence, the surrounding mesh may, successively, adapt to the new geometrical characteristics and carry all the necessary flow information throughout the iterative process. Therefore, to successfully interpolate data contained in certain areas of interest from source (initial) to the destination (final) mesh, each node of the second is necessary to be accurately located in a certain element of the first. **Figure 4.2.2** demonstrates the process of location check.



**Figure 4.2.2** The source and destination mesh are indicated with blue and red colors, accordingly. The destination node D is located into the source element with A, B and C nodes.

Initially, in order to accurately locate a node in an element or its neighbors and perform, eventually, the interpolation of the function of interest, the proper information must be considered. Therefore, two text files containing the function values on source element nodes and the neighbor elements' IDs, are read. For example, in the case of a triangular mesh, as shown in **Figure 4.1.1**, the format of the *.txt* file, containing the function values, is presented in **Figure 4.2.3**.

31102				
1	1.93284231303	1.93164007755	1.93302382420	0.00000000000
2	0.96598922391	0.96602146369	0.96600761869	0.00000000000
3	1.92754862390	1.92485809508	1.92885156295	0.00000000000
4	1.92825963041	1.92869192022	1.92917581570	0.00000000000
5	0.96589055988	0.96587798862	0.96590749942	0.00000000000
6	1.93150776212	1.92978512065	1.92754862390	0.00000000000

**Figure 4.2.3** FunctionValues.txt format.

According to **Figure 4.2.3**, in the first row of the *FunctionValues.txt*, the total

number of source elements is noted. Additionally, from the second row and until the end of the file, the ID of the particular element (first column) and the function values on its nodes (second to fourth column), are listed. It should be mentioned that the last column is equal to zero, as it represents the function value on the fourth node in the case of a quadrilateral element.

Moreover, in the case of a triangular mesh, the format of the *.txt* file containing the neighbor elements' IDs is demonstrated in **Figure 4.2.4**.

31102			
ID	N1	N2	N3
1	28785	3116	27946
2	0	6336	11938
3	0	7	24762
4	0	11927	24842
5	0	24843	11930
6	24761	11891	12271
7	0	11891	3

**Figure 4.2.4** Format of the file containing all the necessary info of each element neighbors.

As shown in **Figure 4.2.4**, the first line of the file enumerates the elements which compose the source mesh, while the second contains the notation of each data column. In addition, from the third line until the end of the present file, the IDs of the examined element (first column), its first neighbor (second column), the second neighbor (third column) and the third neighbor (fourth column) are presented. In the case of a mesh that contains quadrilateral elements, an additional column will be added, in order to contain the fourth neighbor's ID.

It is worth mentioning that concerning a boundary element, the total number of its neighbors is equal to its internal faces. As a result, the neighbors which correspond to the external faces appear to be fictitious, are described as "ghost

*elements*” and their IDs are greater than the total number of elements or equal to zero.

Furthermore, for the purpose of applying the proposed interpolation scheme, it is necessary to integrate all the information regarding the destination mesh topology. **Figure 4.2.5** presents the aforementioned *.txt* file’s format.

```

4000
2.000000000000  0.190000000000
2.000000000000  0.198000000000
2.000000000000  0.206000000000
2.000000000000  0.214000000000
2.000000000000  0.222000000000
2.000000000000  0.230000000000
2.000000000000  0.238000000000
2.000000000000  0.246000000000
2.000000000000  0.254000000000

```

**Figure 4.2.5** Format of the file which contains destination mesh information.

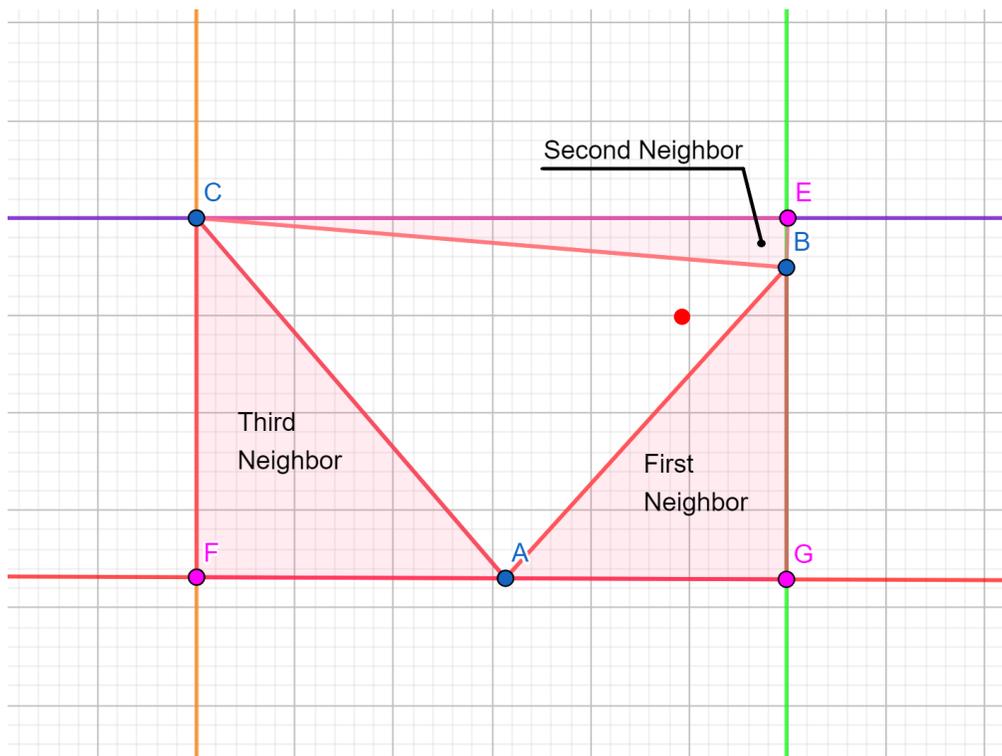
As presented in **Figure 4.2.5**, the first line of the aforementioned file contains the total number of destination mesh nodes, while from the second line until the end of the file, their X coordinates (first column) and Y coordinates (second column) are listed. Finally, given all the essential information regarding the topology of the source and destination meshes, the interpolation scheme is implemented.

Initially, in order to minimize the total number of iterations needed to locate a certain destination node to a source element, the minimum and maximum X coordinates (calculated as explained in Section 4.2.1) of the latter are considered. Furthermore, aiming at the reduction of both computational cost and time, every destination node is directly associated with the left or right “subarea” according to its Cartesian Coordinates.

More specifically, in order to specify whether the destination node is located in the surrounding area of a particular source element or not, the coordinates of the first are compared with the minimum and maximum Cartesian coordinates of the second. Therefore, if inequality (4.6) holds

$$X_q \leq X_{s,min} \text{ and } X_q \geq X_{s,max} \text{ and } Y_q \leq Y_{s,max} \text{ and } Y_q \geq Y_{s,min} , \quad (4.6)$$

then the particular destination node is most likely located in the examined source element or in its surrounding area, where  $X_q$  represents the X coordinate of the examined destination node, while  $X_{s,min}$ ,  $Y_{s,min}$  and  $X_{s,max}$ ,  $Y_{s,max}$  are the minimum and maximum Cartesian coordinates of the tested source element, respectively. A demonstration of the inequality (4.6) is presented in **Figure 4.2.6**.



**Figure 4.2.6** Graphic representation of inequality (4.6). Point C has the maximum Y Cartesian coordinate ( $Y_C = Y_{s,max}$ ), while point A has the minimum one ( $Y_A = Y_{s,min}$ ). Additionally, point B has the maximum X Cartesian coordinate ( $X_B = X_{s,max}$ ), and point C has the minimum one ( $Y_C = Y_{s,min}$ ).

As shown in **Figure 4.2.6**, the blue-colored points illustrate the source element's nodes as well as the area between the red, green, orange and purple-colored lines represents the intersection of the four individual inequalities of inequality (4.6). Furthermore, the red-colored point demonstrates the destination mesh node which appears to be located on the examined source element formed by ABC nodes. Note that the areas between the colored lines and the element's nodes demonstrate the possible location of the three neighbors (in the case of the triangular mesh). In the event of a query point lies outside the element of interest, the three neighboring areas are examined as already mentioned.

Therefore, through the specification of the approximate area where the destination mesh node may lie on, a further and accurate test of location is conducted, by the exploitation of the barycentric coordinates properties.

### *The Point-In-Element check*

In the field of geometry, the barycentric coordinate system consists of a coordinate system that describes the location of a random point on the plane according to a reference simplex (triangle for the 2D plane, tetrahedron for the 3D space, etc.). More specifically, let a triangle in which the query point is located and has masses placed on its vertices. The barycentric coordinates of the query can be interpreted as the individual mass magnitudes on the three vertices which altogether define the query as the center of mass of the simplex. The aforementioned masses can be either zero or negative for points that are located outside the triangular area or positive for queries located inside the simplex.

Hence, according to Floater et al. (2006) let  $\Omega \subset \mathbb{R}^2$  be a convex polygon with vertices  $v_1, v_2, \dots, v_n$ ,  $n \geq 3$ , in a counterclockwise ordering. Any set of

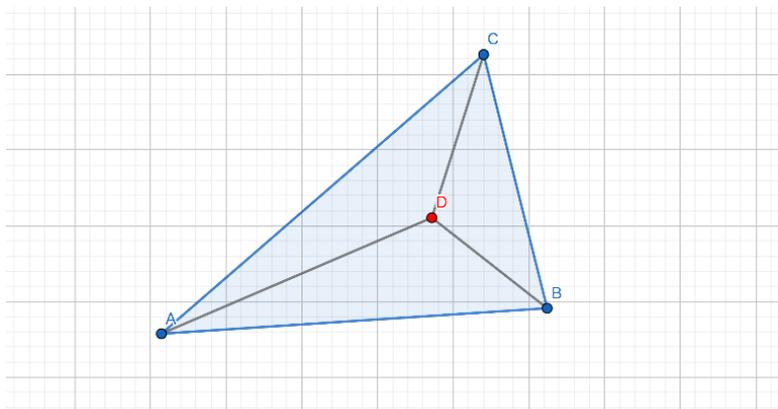
functions  $\lambda_i: \Omega \rightarrow \mathbb{R}, i = 1, \dots, n$  is called “barycentric coordinates” if they satisfy for all  $v \in \Omega$  the three following properties:

$$\lambda_i(v) \geq 0, \quad i = 1, \dots, n \quad (4.7)$$

$$\sum_{i=1}^n \lambda_i(v) = 1 \quad (4.8)$$

$$\sum_{i=1}^n \lambda_i(v) v_i = v. \quad (4.9)$$

The aforementioned presentation of barycentric coordinates consists of a generalization of the triangular barycentric coordinates, where  $n = 3$  and  $\Omega$  is a triangular simplex with vertices  $[v_1, v_2, v_3]$  as presented in **Figure 4.2.7**. From now on, for simplicity reasons, the barycentric coordinates of a query point will be denoted as  $w, v, u$ .



**Figure 4.2.7** Triangular barycentric coordinates of point D inside the triangle ABC.

To begin with, consider the case of a triangular element as shown in **Figure 4.2.7**. In order to obtain the barycentric coordinates  $w, v, u$  of a randomly placed point  $P$  at the interior of a triangular simplex with vertices  $A, B, C$ , property (4.9) is exploited and Equation (4.10) emerges:

$$P = wA + vB + uC . \quad (4.10)$$

From property (4.8) we have:

$$\begin{aligned} w + v + u &= 1 \\ w &= 1 - u - v . \end{aligned} \quad (4.11)$$

Therefore, Equation (4.10) can be formed as follows:

$$\begin{aligned} P &= wA + vB + uC \\ P &= (1 - u - v)A + vB + uC \\ P &= A - uA - vA + vB + uC \\ P - A &= v(B - A) + u(C - A) . \end{aligned} \quad (4.12)$$

If  $c = C - A, b = B - A, p = P - A$  then,

$$p = bv + cu . \quad (4.13)$$

Then, considering that Equation (4.13) has two unknown variables ( $u, v$ ), both sides of it are dotted, initially, with  $c$  and, finally, with  $b$ . The following system with two equations and two unknown variables emerges:

$$u = \frac{(p \cdot c)(b \cdot b) - (b \cdot c)(p \cdot b)}{(c \cdot c)(b \cdot b) - (b \cdot c)(b \cdot c)} \quad (4.14)$$

$$v = \frac{(p \cdot b)(c \cdot c) - (p \cdot c)(c \cdot b)}{(b \cdot b)(c \cdot c) - (b \cdot c)(b \cdot c)} . \quad (4.15)$$

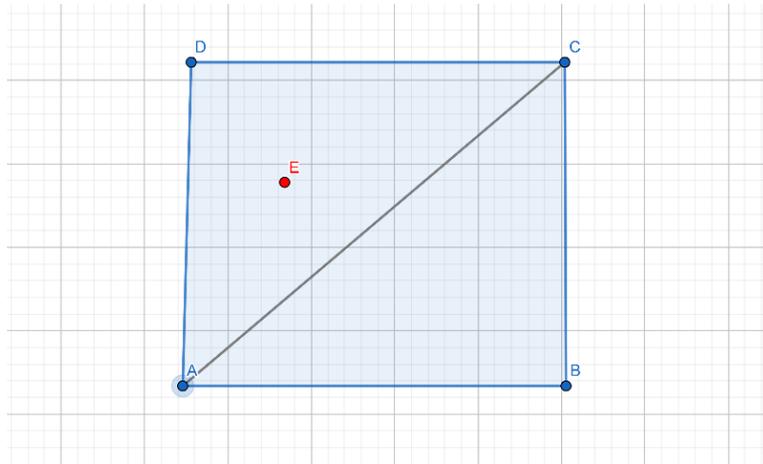
As a result, if a point is located inside a triangular area, according to property (4.7), its barycentric coordinates must be positive as follows:

$$u, v \geq 0 \text{ and} \quad (4.16)$$

$$w = 1 - u - v \geq 0 . \quad (4.17)$$

It should be noticed that in case of a not detected point inside the examined element, its neighbors are, then, tested in the same way.

On the other hand, in the event of testing a quadrilateral element, the barycentric coordinates of a randomly located point are obtained by the division of the simplex into two triangular areas, as presented in **Figure 4.2.8**.



**Figure 4.2.8** Line segment AC divides the tetrahedron ABCD (blue colored) into two separate triangular areas, ABC and ACD, accordingly. The query point E (red colored) is located into ACD triangle.

Thus, through the calculation of BCs of both generated triangles (e.g. ABC and ACD triangles), the query point can be locate accordingly. The case of an embedded point inside a quadrilateral element is shown in **Figure 4.2.8**.

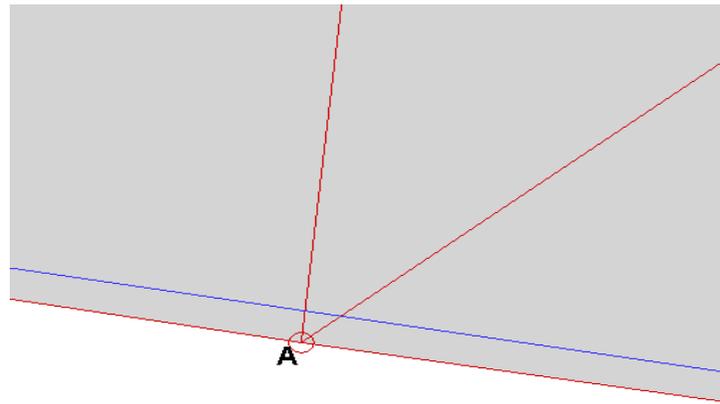
### Barycentric Interpolation

Once the query point is successfully located, the interpolation of the function is performed. Therefore, consider a function  $f: \Omega \rightarrow \mathbb{R}$ , with values  $f(v_1), f(v_2), f(v_3)$  on the triangular element's nodes  $v_1, v_2, v_3$ , respectively, wherein the query point was located. Then, in order to interpolate the function  $f$  on the query point's  $p$  location, Equation (4.18) holds:

$$f(p) = wf(v_1) + vf(v_2) + uf(v_3), \quad (4.18)$$

where  $w, v, u$  are the barycentric coordinates of the query, calculated at the previous step ([Point-In - Element check](#)).

Although a query point undergoes consecutive checks (on the interior of each source element and its neighbors) for the verification of its location, there exist several instances where is not, eventually, detected. The aforementioned case is presented in **Figure 4.2.9**.



**Figure 4.2.9** Destination node A is located outside the source mesh indicated with blue color.

The most common reasons for the particular inexpediency consists of either the difference on the topology of both meshes or the round-off errors resulted from the numerical operations, applied for the specific application. Hence, to successfully interpolate a function on the undetected points, an additional interpolation scheme is applied, called the “*Inverse Distance Weighting (IDW) interpolation*”.

### **Inverse Distance Weighting (IDW) Interpolation**

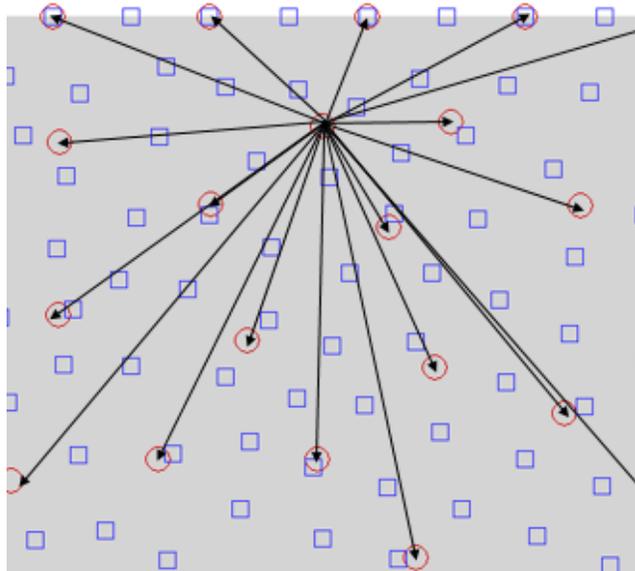
Introduced by Shepard (1968), the Inverse Distance Weighting Interpolation consists of a multivariate interpolation methodology, applied on a given set of scattered points. In order to determine the value of a function on a particular query location, its distance from every point of the given set is measured, and

act as a weight, which is assigned to each one of them. Therefore, the estimation of the function value is obtained through the application of the weighted average of individual values on the provided set of points.

Thus, for the purpose of interpolating a function on the not-located destination nodes, their Euclidean distances from each detected point of the aforementioned mesh, need to be calculated and saved. The Euclidean Distance between two different points is:

$$d_{Euclidean}((x_1, y_1), (x_2, y_2)) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \quad (4.19)$$

where  $(x_1, y_1), (x_2, y_2)$  are their Cartesian coordinates, respectively. Once, the distances between the query point and the located destination nodes are computed, the Bubble Sort algorithm is applied and they, therefore, are sorted in a descending order. A demonstration of the aforementioned process is presented in **Figure 4.2.10**.



**Figure 4.2.10** The calculation of the individual distances between the query point and the surrounding destination nodes. Red circular points indicate the destination mesh nodes, while points marked with blue squares represent the source mesh nodes. Each black colored arrow depicts the calculated distance between the query point and the detected destination nodes.

To begin with, a robust way to locate the destination nodes which were not found initially, is to employ the [Point-In-Element](#) check and, provided the estimation of the location of the element it belongs to, the Barycentric Coordinates interpolation methodology is implemented. However, in the present instance, the only source elements which are tested consist of the three *-closest to the query-* that were observed. Nevertheless, despite the application of the present variation, both [Point-In-Element](#) check and the [BC interpolation](#) processes, remain the same, as described in the previous section.

Till the present section, the proposed algorithm offers a “three-layered” point-in-element check i.e. 1<sup>st</sup> layer according to Inequality (4.6); 2<sup>nd</sup> layer for BCs check according to Inequalities (4.16) & (4.17), 3<sup>rd</sup> for BCs check according to the Euclidean distances between the query and the located destination nodes. However, there is, yet, a possibility of a few queries location to be unspecified. On the account of the aforementioned issue, the Inverse Distance Weighting (IDW) interpolation is applied.

Therefore, according to Witteveen and Bijl (2009), during the IDW interpolation methodology, the interpolation surface  $w(x)$ , through  $n$  data samples  $\mathbf{v} = \{u_1, u_2, \dots, u_n\}$  of the examined function  $u(x) \equiv u(x_i)$  is given by:

$$w(x) = \frac{\sum_{i=1}^n u_i \varphi(r_i)}{\sum_{i=1}^n \varphi(r_i)}, \quad (4.20)$$

with the weighting function:

$$\varphi(r) = r^{-c}, \quad (4.21)$$

where  $r_i$  is the Euclidean distance (Equation 4.19) between the query point  $x$  and the node  $x_i$ , and  $c$  is a positive real number called the “*power parameter*”.

More specifically, in the case of the present study, data points  $x_i$ ,  $i = 1, \dots, 100$  are the hundred - *closest to the query* - destination nodes that were, already, located in a source mesh element, as presented in the previous section. The specific number of data points was selected in order to decrease the total the computational time involved during the interpolation. Furthermore, the power parameter was set equal to five ( $c = 5$ ), following the trial-and-error method.

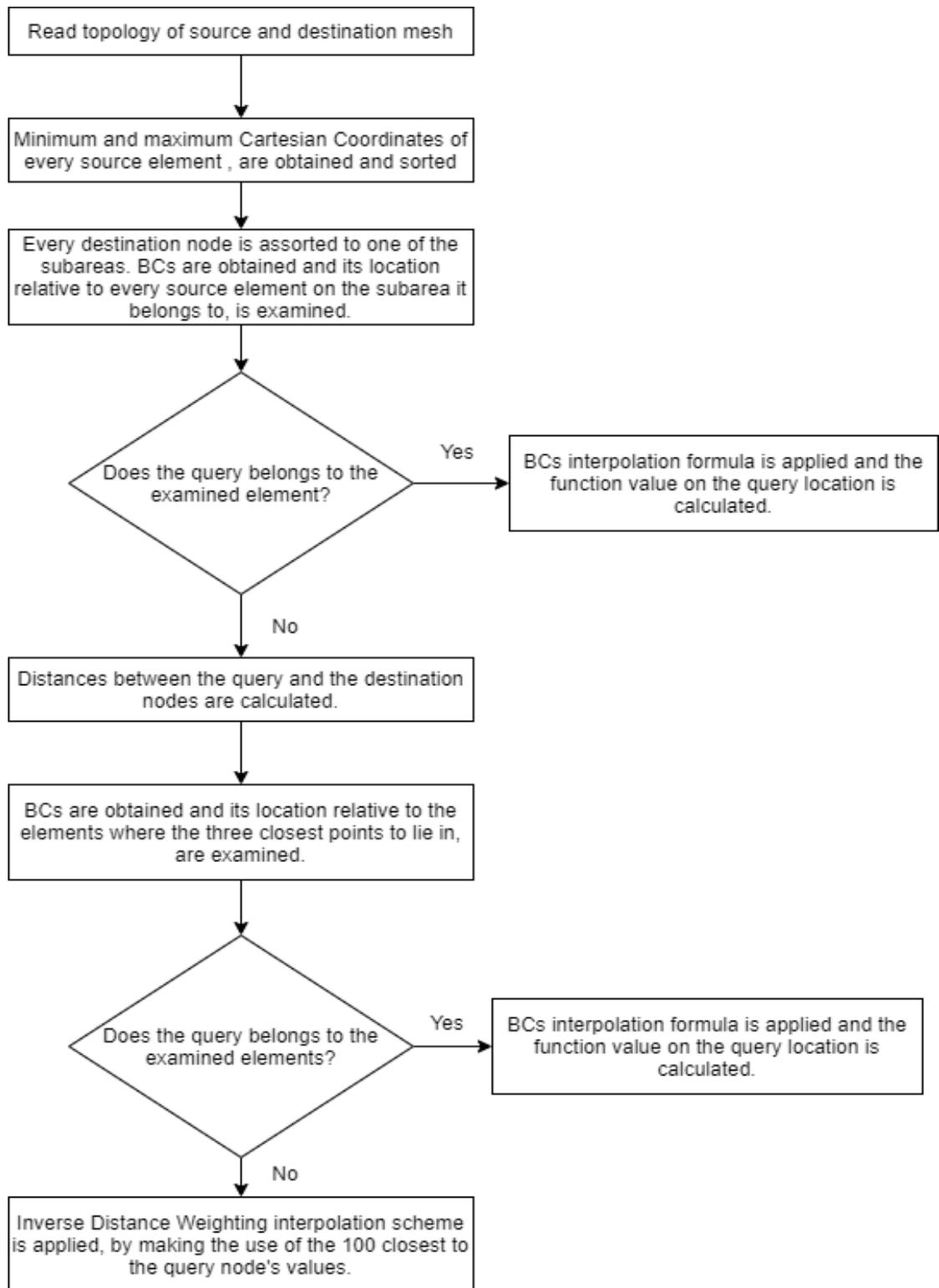
### 4.3 Flow chart of the proposed interpolation algorithm

To summarize, in order to interpolate a destination mesh to a source grid, steps 1 to 7 are followed:

1. Text files containing the topology of the source mesh (elements' Cartesian coordinates and neighboring elements), the function values on the source mesh nodes and the destination mesh nodes' Cartesian Coordinates, are read.
2. Minimum and maximum X and Y coordinates of each source element are obtained. The median of the minimum X coordinates is calculated and two mesh - subareas are created. Every source element is assorted to the proper subarea according to each maximum X coordinate compared with the calculated median.
3. Every destination node is assorted to the left or right subarea according to its X coordinate and the median calculated in step 2. Then, the query is examined and the source element which belongs to is specified, according to its calculated Barycentric Coordinate and their properties.
4. The function of interest is interpolated on the query, through the application of the Barycentric Interpolation formula.
5. In case of a destination node is not located into a source element, its distances between the detected destination nodes, are calculated. Its location compared to the three elements, where the - closest destination nodes - are placed, is checked, according to its Barycentric coordinates and their properties (as in Step 3.).
6. If the examined destination node is, successfully, located into an element (as explained in step 3), the function of interest is interpolated, according to the Barycentric Coordinates formula.

7. If the examined node is not located (following the process described in step 5), the Inverse Distance Weighting Interpolation methodology is applied by making the use of the 100 - closest to the query-destination nodes' values.

The aforementioned interpolation process is demonstrated in **Figure 4.3.1**.



**Figure 4.3.1** Flow chart of the proposed algorithm, which combines the Barycentric Coordinates (BCs) and Inverse Distance Weighting (IDW) interpolation schemes.

## 4.4 Results of the algorithm

In the present study, the cases of triangular, quadrilateral and hybrid mesh interpolation were tested. In all of the three instances, a fine and a coarse mesh were utilized. Furthermore, the flow velocity values on each node of the source mesh were selected to be interpolated to the destination mesh nodes. A detailed presentation of the estimated function, on each occasion, is presented in Sections 4.4.1-4.4.3.

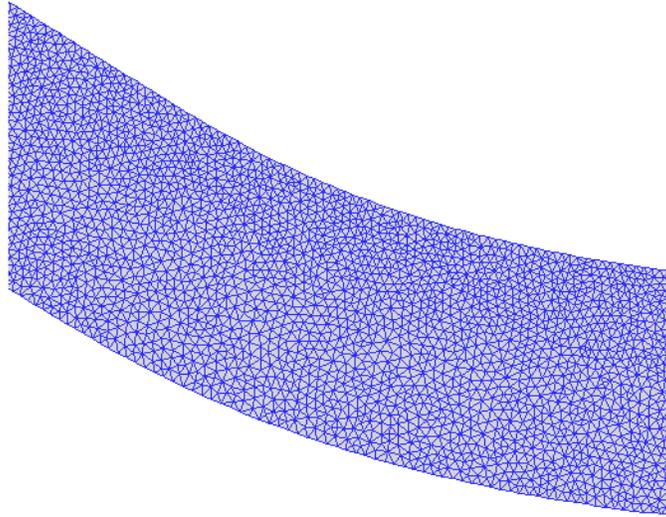
### 4.4.1 Triangular mesh interpolation

In the present instance, a mesh containing 16082 nodes and 31102 triangular elements was chosen as source. Additionally, the selected destination mesh consisted of 4055 nodes and 7580 triangular elements. Both source and destination mesh constituted a fine and coarse discretization of the surface of an S-shape pipe, respectively. The geometry of an S-shape pipe is presented in **Figure 4.4.1**.

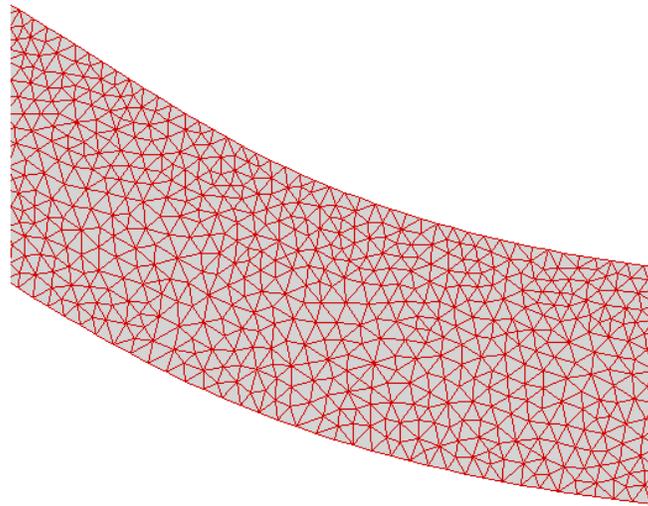


**Figure 4.4.1** The geometry of an S – Shape pipe.

A section of the source (blue-colored) and destination mesh (red-colored) acting as a fine and a coarse discretization of the S-shape pipe's geometry, are demonstrated in **Figure 4.4.2** and **Figure 4.4.3**, respectively.



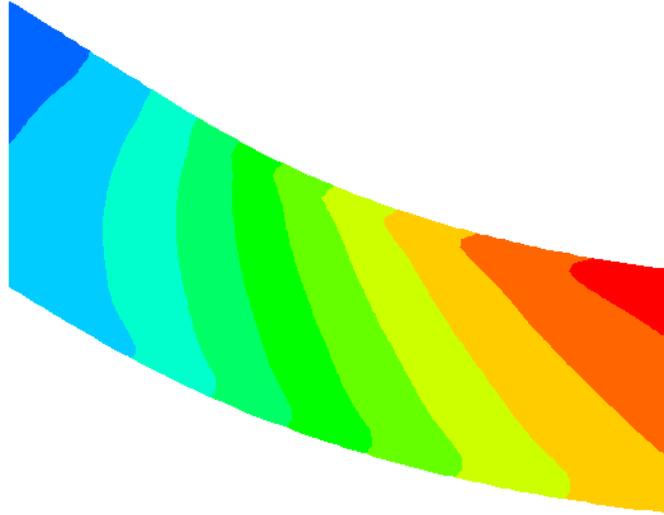
**Figure 4.4.2** The source mesh of the S-Shape pipe is constructed by 16082 nodes and 31102 elements and is indicated with blue color.



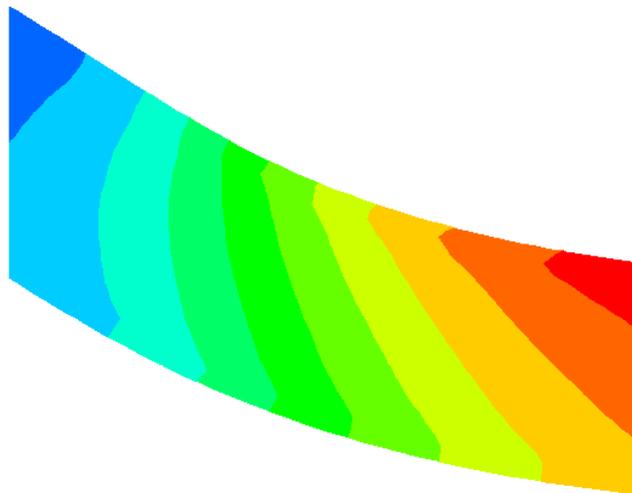
**Figure 4.4.3** The destination mesh of the examined geometry consists of 4055 nodes and 7580 elements and is colored with red.

The function, which was chosen in order to be interpolated from the source to the destination mesh, consisted of the velocity values calculated on the nodes of the former grid.

The function values on the source and destination mesh nodes are presented in **Figure 4.4.4** and **Figure 4.4.5**. In both cases, red colored areas indicate the maximum flow velocity values. On the other hand, blue colored areas demonstrate the minimum velocity values.



**Figure 4.4.4** The function values on source mesh nodes.

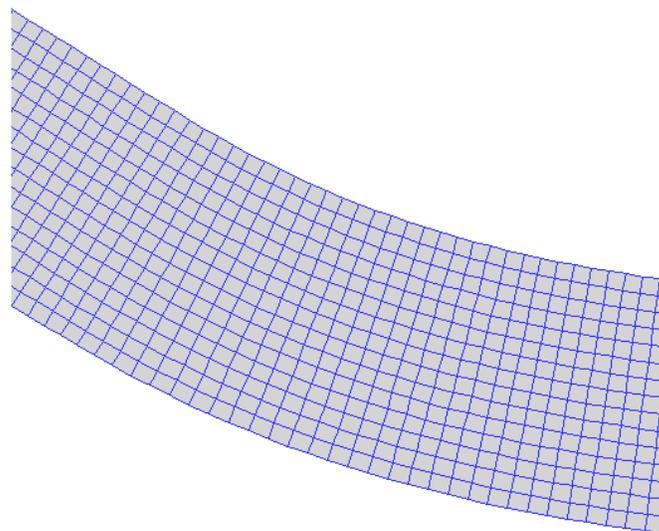


**Figure 4.4.5** The destination values obtained following the application of the proposed interpolation algorithm.

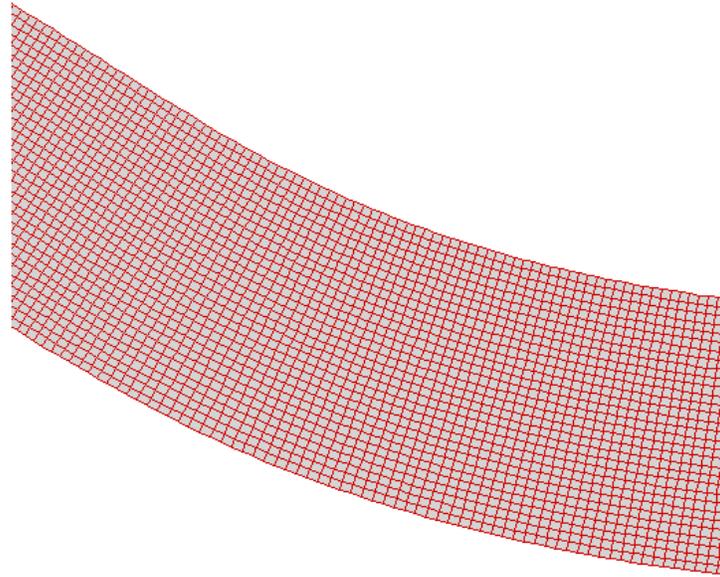
Through the visual comparison of **Figure 4.4.4** and **Figure 4.4.5**, it can be observed that the proposed interpolation scheme resulted to almost identical to the source grid velocity values on the destination mesh. In other words, the velocity contour on both meshes (source & destination) had no substantial differences at all. It worth noting that in the case of interpolating values from coarse to fine triangular meshes, the proposed methodology produced plausible results.

#### 4.4.2 Quadrilateral mesh interpolation

For the quadrilateral mesh interpolation, the source mesh consisted of 4000 nodes and 3735 quadrilateral elements. On the other hand, the destination mesh was constructed by 16000 nodes and 15469 elements. Additionally, the suggested interpolation algorithm was applied for the case of the exact same shape as mentioned in triangular mesh interpolation (Section 4.4.1). The examined shape's geometry is shown in **Figure 4.4.1**. A part of the source (blue colored) and destination grid (red colored) is presented in **Figure 4.4.6** and **Figure 4.4.7**, respectively.

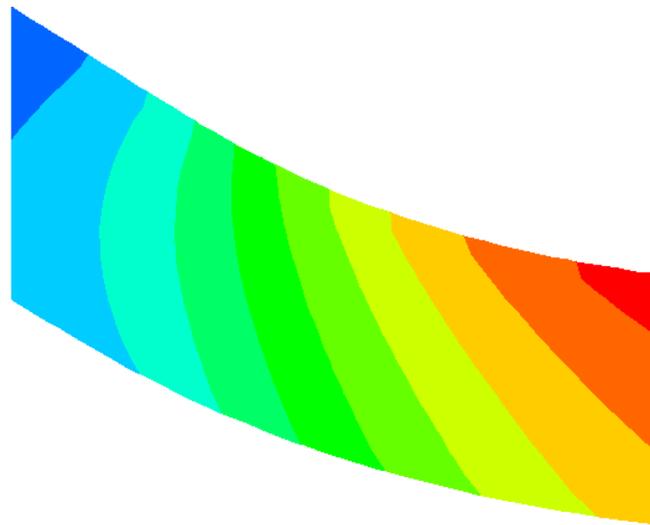


**Figure 4.4.6** The topology of the source mesh in the middle of the S – shape pipe.

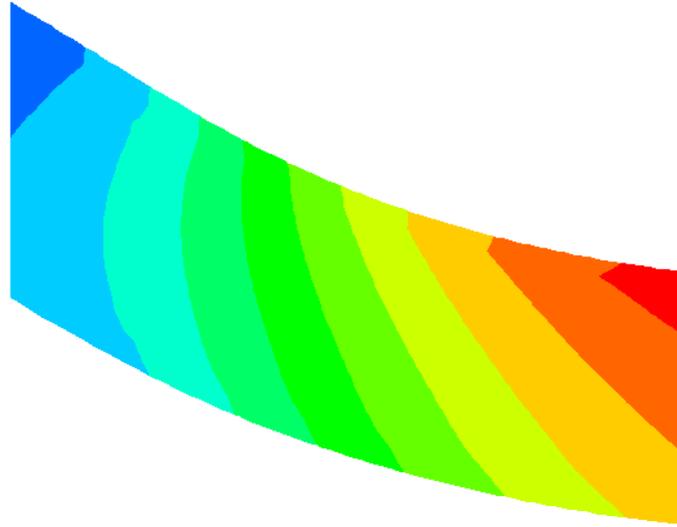


**Figure 4.4.7** The topology of the destination mesh in the middle of the S – shape pipe.

As presented in Section 4.4.1, the selected function for the quadrilateral mesh interpolation was the flow velocity at the interior of the S-shape pipe. The contours of the given velocity values on the source mesh nodes as well as of the interpolated ones on the destination mesh are shown in **Figure 4.4.8** and **Figure 4.4.9**, respectively.



**Figure 4.4.8** The function values on the nodes of the source mesh.

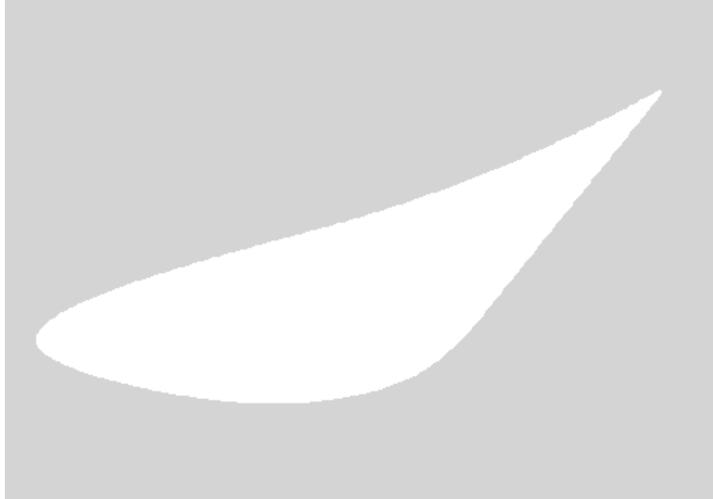


**Figure 4.4.9** The function values on the nodes of the destination mesh.

Following the comparison of **Figure 4.4.8** and **Figure 4.4.9**, it should be noted that the flow velocity contours of source and destination grid appeared to be similar to a considerable extent. Moreover, the proposed scheme produced fair results for the case of interpolating values from the fine to the coarse mesh.

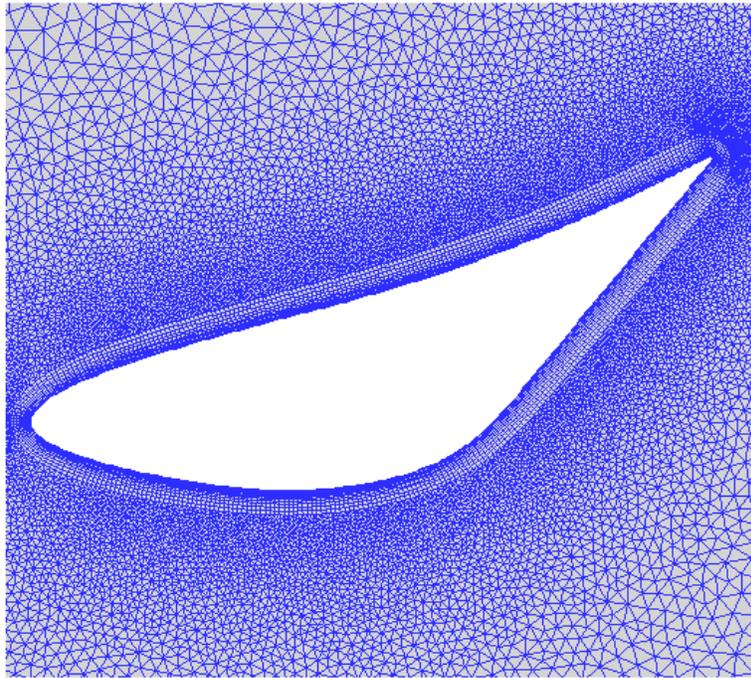
#### 4.4.3 Hybrid mesh Interpolation

In the present section, the interpolation between two hybrid meshes is presented. At first, it should be noted that a hybrid mesh is consisted of both quadrilateral and triangular elements. Therefore, a hybrid mesh consisted of 26671 nodes, 35738 triangles and 8448 quadrilateral elements was utilized as the source one. On the contrary, the chosen destination mesh was composed of 68809 nodes, 101170 triangular elements and 17760 quadrilaterals. Both source and destination grids acted as a fine and a coarse discretization of the exterior of an airfoil shape, respectively. The geometry of the examined airfoil is shown in **Figure 4.4.10**.



**Figure 4.4.10** The airfoil shape.

In **Figure 4.4.11** and **Figure 4.4.12** the topologies of the source (blue-colored) and destination (red-colored) meshes are presented, respectively.

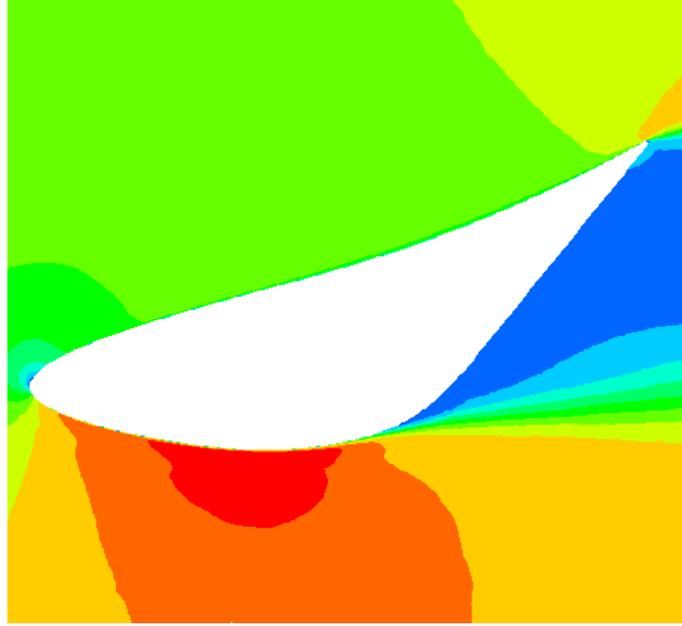


**Figure 4.4.11** The topology of the source mesh.

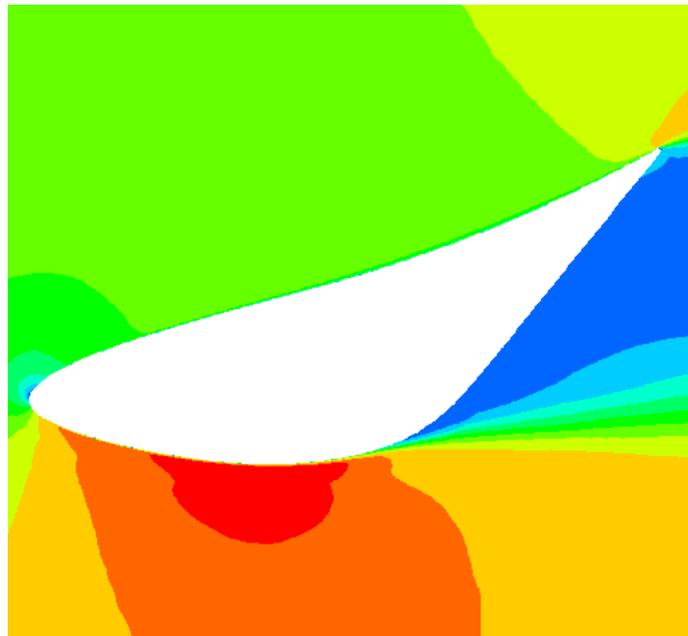


**Figure 4.4.12** The topology of the destination mesh.

As presented in Section 4.4.1 and 4.4.2, the chosen function values to be interpolated, consisted of the fluid velocity given on the source mesh nodes. **Figure 4.4.13** depicts the contour of the flow velocity on the source mesh. On the other hand, **Figure 4.4.14** represents the interpolated on the destination mesh. It worth noting that no deviation was observed between the given values on the source and the interpolated velocity on the destination mesh nodes, respectively. Therefore, the contours of **Figure 4.4.13** and **Figure 4.4.14** did not appear to have considerable differences. Additionally, in the case of mesh interpolation from a fine to a course grid, plausible results were obtained.



**Figure 4.4.13** Each node of the source grid is assigned a flow velocity value. Color variation indicates the magnitude of the velocity around the examined geometry. For example, low flow velocity values are marked with blue color contrary to high flow velocity areas indicated with red color.



**Figure 4.4.14** The flow velocity contour resulted from the interpolation of the values on the source grid nodes to the destination mesh.

## **4.5 Conclusion**

In each one of the three cases tested, the flow velocity values were smoothly interpolated from the initial (source) mesh to the final (destination) one. Consequently, the proposed interpolation scheme appears to be applicable on Triangular, Quadrilateral and Hybrid meshes, given the topology information of the source and destination grids and the function values on the source mesh nodes, as explained in Sections 4.2.1 and 4.2.2. Eventually, it worth noting that the aforementioned interpolation scheme proved to be feasible for the interpolation from the fine to the coarse mesh and inversely.

## Chapter 5

### Application of Harmonic Functions - based deformation

#### 5.1 Test Case 1: Single internal B-Spline boundary.

The first test case consists of a rectangular domain with a single internal boundary, formulated as a smooth closed periodic B-Spline curve of 2<sup>nd</sup> degree, with 4 (different) control points (as being a periodic closed B-Spline curve of 2<sup>nd</sup> degree, the two first control points are repeated at the end of the curve, resulting in actually 6 control points). The coordinates of the control points are listed in **Table 5.1.1**.

x-coordinate	y-coordinate
-3.80327177	9.66863155
-3.41808033	7.30724001
4.16851759	9.73562145
3.76657844	11.71182060

**Table 5.1.1** The Cartesian coordinates of four control points of a 2<sup>nd</sup> degree B – Spline curve.

The initial unstructured grid, consisted of triangular elements, is depicted in **Figure 5.1.1** and **Figure 5.1.2**. In **Table 5.1.2**, the coordinates of the discrete points of the B-Spline boundary are contained, along with the corresponding u-value and the values of the 4 Basis Functions, used as boundary conditions for the consecutive solution of the Laplace equation.

$i$	$x$	$y$	$u(i)$	$N1(i)$	$N2(i)$	$N3(i)$	$N4(i)$
100	-3.61068	8.487941	0	0.5	0.5	0	0
1	-3.61972	8.582142	0.01	0.4608	0.5384	0.0008	0
2	-3.61603	8.675834	0.02	0.4232	0.5736	0.0032	0
3	-3.59962	8.769018	0.03	0.3872	0.6056	0.0072	0
4	-3.57048	8.861691	0.04	0.3528	0.6344	0.0128	0
5	-3.52861	8.953855	0.05	0.32	0.66	0.02	0
6	-3.47401	9.045511	0.06	0.2888	0.6824	0.0288	0
7	-3.40669	9.136656	0.07	0.2592	0.7016	0.0392	0
8	-3.32663	9.227293	0.08	0.2312	0.7176	0.0512	0
9	-3.23385	9.317422	0.09	0.2048	0.7304	0.0648	0
10	-3.12834	9.407041	0.1	0.18	0.74	0.08	0
11	-3.01011	9.49615	0.11	0.1568	0.7464	0.0968	0
12	-2.87914	9.584751	0.12	0.1352	0.7496	0.1152	0
13	-2.73545	9.672842	0.13	0.1152	0.7496	0.1352	0
14	-2.57902	9.760425	0.14	0.0968	0.7464	0.1568	0
15	-2.40987	9.847497	0.15	0.08	0.74	0.18	0
16	-2.228	9.934062	0.16	0.0648	0.7304	0.2048	0
17	-2.03339	10.02012	0.17	0.0512	0.7176	0.2312	0
18	-1.82606	10.10566	0.18	0.0392	0.7016	0.2592	0
19	-1.60599	10.1907	0.19	0.0288	0.6824	0.2888	0
20	-1.3732	10.27523	0.2	0.02	0.66	0.32	0
21	-1.12769	10.35925	0.21	0.0128	0.6344	0.3528	0
22	-0.86944	10.44276	0.22	0.0072	0.6056	0.3872	0
23	-0.59847	10.52576	0.23	0.0032	0.5736	0.4232	0
24	-0.31476	10.60825	0.24	0.0008	0.5384	0.4608	0
25	-0.01833	10.69023	0.25	0	0.5	0.5	0
26	0.278724	10.76874	0.26	0	0.4608	0.5384	0.0008

27	0.564315	10.84082	0.27	0	0.4232	0.5736	0.0032
28	0.838436	10.90647	0.28	0	0.3872	0.6056	0.0072
29	1.101089	10.96569	0.29	0	0.3528	0.6344	0.0128
30	1.352274	11.01848	0.3	0	0.32	0.66	0.02
31	1.59199	11.06483	0.31	0	0.2888	0.6824	0.0288
32	1.820237	11.10476	0.32	0	0.2592	0.7016	0.0392
33	2.037015	11.13825	0.33	0	0.2312	0.7176	0.0512
34	2.242325	11.16532	0.34	0	0.2048	0.7304	0.0648
35	2.436166	11.18595	0.35	0	0.18	0.74	0.08
36	2.618539	11.20015	0.36	0	0.1568	0.7464	0.0968
37	2.789443	11.20792	0.37	0	0.1352	0.7496	0.1152
38	2.948878	11.20926	0.38	0	0.1152	0.7496	0.1352
39	3.096845	11.20417	0.39	0	0.0968	0.7464	0.1568
40	3.233343	11.19265	0.4	0	0.08	0.74	0.18
41	3.358372	11.1747	0.41	0	0.0648	0.7304	0.2048
42	3.471933	11.15031	0.42	0	0.0512	0.7176	0.2312
43	3.574025	11.1195	0.43	0	0.0392	0.7016	0.2592
44	3.664649	11.08225	0.44	0	0.0288	0.6824	0.2888
45	3.743803	11.03857	0.45	0	0.02	0.66	0.319999
46	3.81149	10.98846	0.46	0	0.0128	0.6344	0.352799
47	3.867707	10.93192	0.47	0	0.0072	0.605601	0.387199
48	3.912456	10.86895	0.48	0	0.0032	0.573601	0.423199
49	3.945736	10.79955	0.49	0	0.0008	0.538401	0.460799
50	3.967548	10.72372	0.5	0	0	0.500001	0.499999
51	3.977234	10.64431	0.51	0.0008	0	0.460801	0.538399
52	3.974139	10.56418	0.52	0.0032	0	0.423201	0.573599
53	3.958262	10.48332	0.53	0.0072	0	0.387201	0.605599
54	3.929604	10.40174	0.54	0.0128	0	0.352801	0.634399

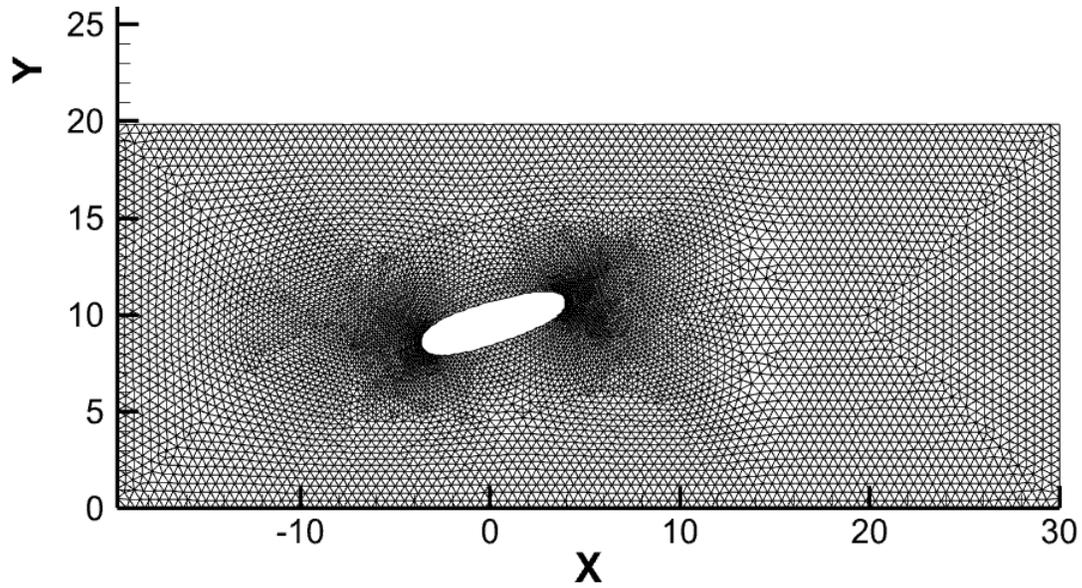
55	3.888164	10.31944	0.55	0.02	0	0.320001	0.659999
56	3.833942	10.23641	0.56	0.0288	0	0.288801	0.6824
57	3.766939	10.15266	0.57	0.0392	0	0.259201	0.7016
58	3.687154	10.06818	0.58	0.0512	0	0.231201	0.7176
59	3.594587	9.982986	0.59	0.0648	0	0.204801	0.7304
60	3.489239	9.897066	0.6	0.08	0	0.180001	0.74
61	3.371109	9.81042	0.61	0.0968	0	0.156801	0.7464
62	3.240198	9.723053	0.62	0.115199	0	0.135201	0.7496
63	3.096504	9.634962	0.63	0.135199	0	0.115201	0.7496
64	2.94003	9.546147	0.64	0.156799	0	0.096801	0.7464
65	2.770773	9.456607	0.65	0.179999	0	0.080001	0.74
66	2.588735	9.366347	0.66	0.204799	0	0.0648	0.7304
67	2.393916	9.275361	0.67	0.231199	0	0.0512	0.7176
68	2.186314	9.183651	0.68	0.259199	0	0.0392	0.701601
69	1.965931	9.09122	0.69	0.288799	0	0.0288	0.682401
70	1.732767	8.998064	0.7	0.319999	0	0.02	0.660001
71	1.48682	8.904183	0.71	0.352799	0	0.0128	0.634401
72	1.228093	8.80958	0.72	0.387199	0	0.0072	0.605601
73	0.956583	8.714254	0.73	0.423199	0	0.0032	0.573601
74	0.672292	8.618205	0.74	0.460798	0	0.0008	0.538401
75	0.375219	8.52143	0.75	0.499998	0	0	0.500002
76	0.077516	8.428127	0.76	0.538398	0.0008	0	0.460802
77	-0.20866	8.342488	0.77	0.573599	0.0032	0	0.423202
78	-0.48332	8.264511	0.78	0.605599	0.0072	0	0.387202
79	-0.74646	8.194199	0.79	0.634399	0.0128	0	0.352802
80	-0.99807	8.131549	0.8	0.659999	0.02	0	0.320001
81	-1.23816	8.076564	0.81	0.682399	0.0288	0	0.288801
82	-1.46673	8.029243	0.82	0.701599	0.039199	0	0.259201

83	-1.68378	7.989584	0.83	0.717599	0.051199	0	0.231201
84	-1.88931	7.957591	0.84	0.730399	0.064799	0	0.204801
85	-2.08331	7.933259	0.849999	0.74	0.079999	0	0.180001
86	-2.26579	7.916592	0.859999	0.7464	0.096799	0	0.156801
87	-2.43675	7.907589	0.869999	0.7496	0.115199	0	0.135201
88	-2.59618	7.90625	0.879999	0.7496	0.135199	0	0.115201
89	-2.7441	7.912573	0.889999	0.7464	0.156799	0	0.096801
90	-2.88049	7.92656	0.899999	0.74	0.179999	0	0.080001
91	-3.00536	7.948212	0.909999	0.730401	0.204799	0	0.064801
92	-3.1187	7.977526	0.919999	0.717601	0.231198	0	0.051201
93	-3.22053	8.014505	0.929999	0.701601	0.259198	0	0.039201
94	-3.31083	8.059146	0.939999	0.682401	0.288798	0	0.028801
95	-3.38961	8.111453	0.949999	0.660001	0.319998	0	0.02
96	-3.45687	8.171422	0.959999	0.634402	0.352798	0	0.0128
97	-3.5126	8.239054	0.969999	0.605602	0.387198	0	0.0072
98	-3.55682	8.314351	0.979999	0.573602	0.423198	0	0.0032
99	-3.58951	8.397312	0.989999	0.538402	0.460798	0	0.0008

**Table 5.1.2** In the first three columns of the table, the IDs and the Cartesian coordinates of B-Spline boundary points are listed. Additionally, the fourth column contains the corresponding u-values on each point. Finally, the fifth, sixth, seventh and eight columns consist of the values of four 2<sup>nd</sup> degree B-Spline basis functions applied as boundary conditions for the solution of the Laplace equation on each point, respectively.

In **Figure 5.1.3** to **Figure 5.1.6**, the solution of the Laplace equation is depicted, for each one of the basis functions, used as boundary conditions upon the internal boundary. Each basis function corresponds to a B-Spline control point. The original unstructured grid was used for the solution of the Laplace equation (4 consecutive times, equal to the number of the (different) control points of the B-Spline curve).

Then, a deformation to the original control points of the internal B-Spline boundary is applied. The deformation to each one of the control points' coordinates is depicted in **Table 5.1.3**.



**Figure 5.1.1** The initial shape of a smooth closed periodic curve of 2<sup>nd</sup> degree embedded in an unstructured grid consisted of triangular elements.

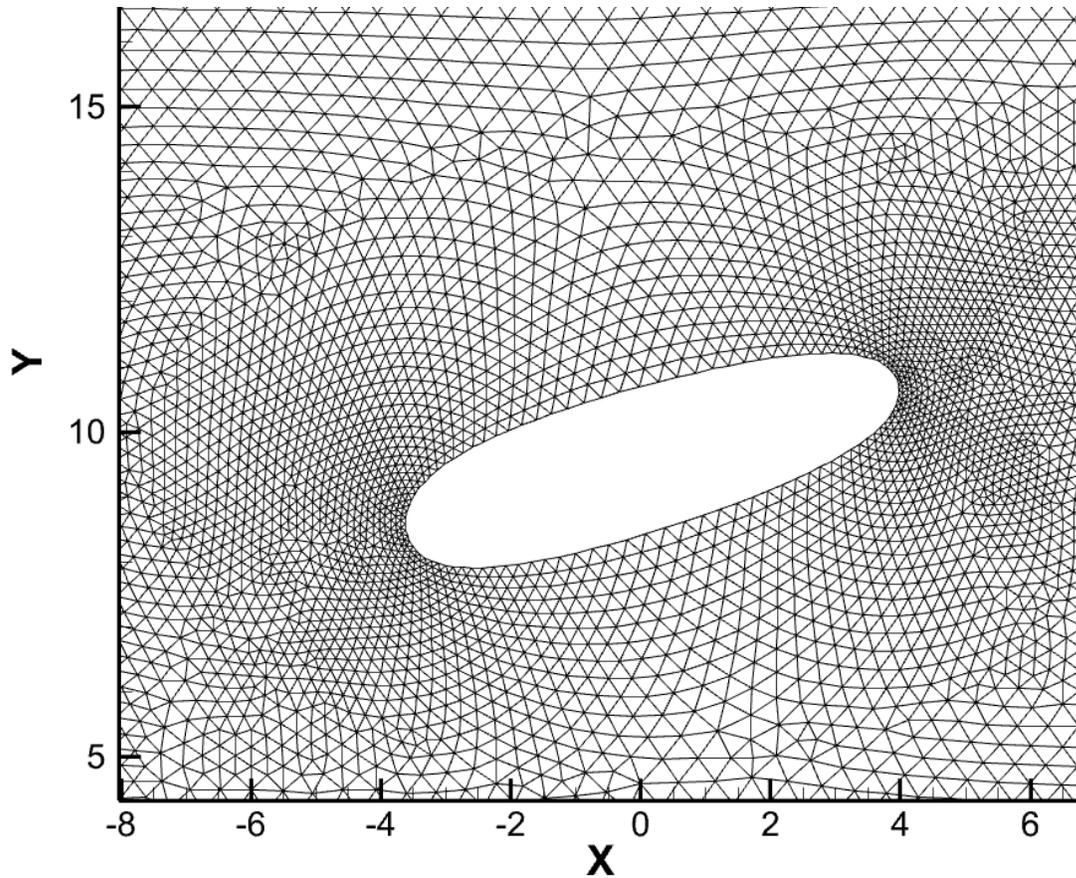


Figure 5.1.2 A closer examination of the initial shape of the B-Spline curve (shown in Figure 5.1.1) as well as of the unstructured grid surrounded by.

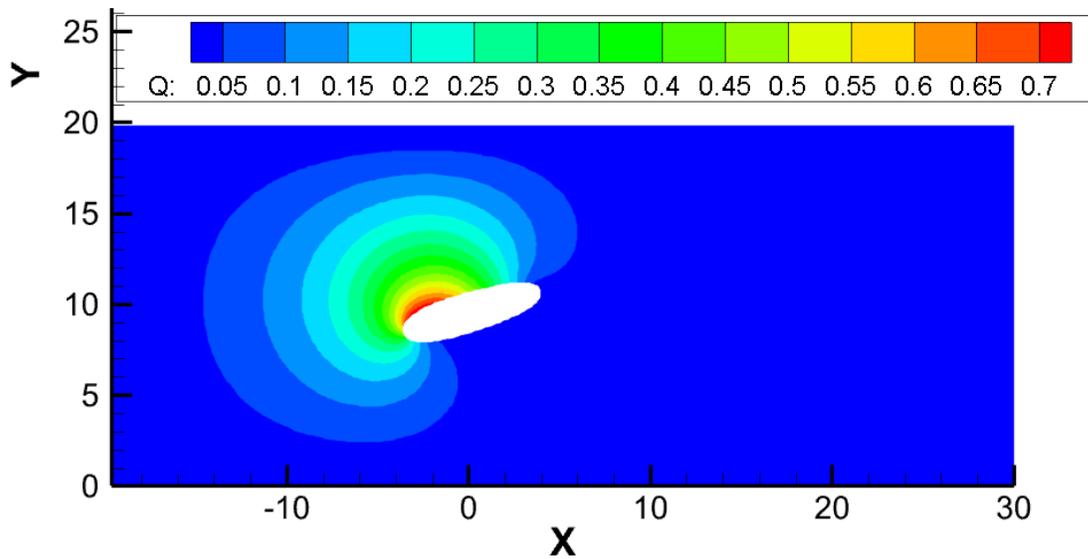


Figure 5.1.3 The solution of the Laplace equation calculated by the application of the first basis function ( $N1(i)$ ) as a Dirichlet boundary condition.

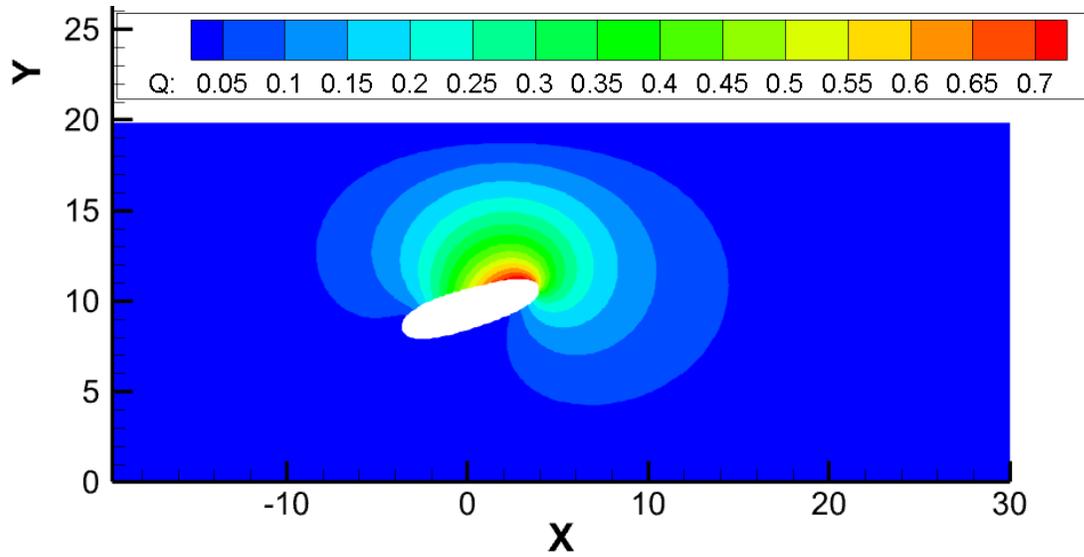


Figure 5.1.5 The solution of the Laplace equation using the second basis function ( $N2(i)$ ) as a boundary condition.

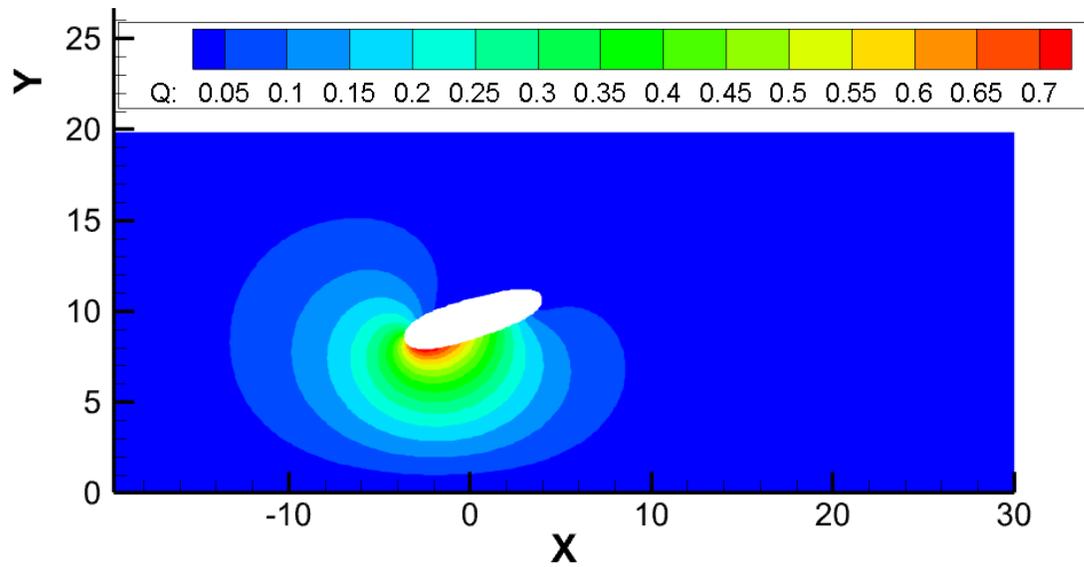


Figure 5.1.4 The solution of the Laplace equation using the third basis function ( $N3(i)$ ) as a boundary condition.

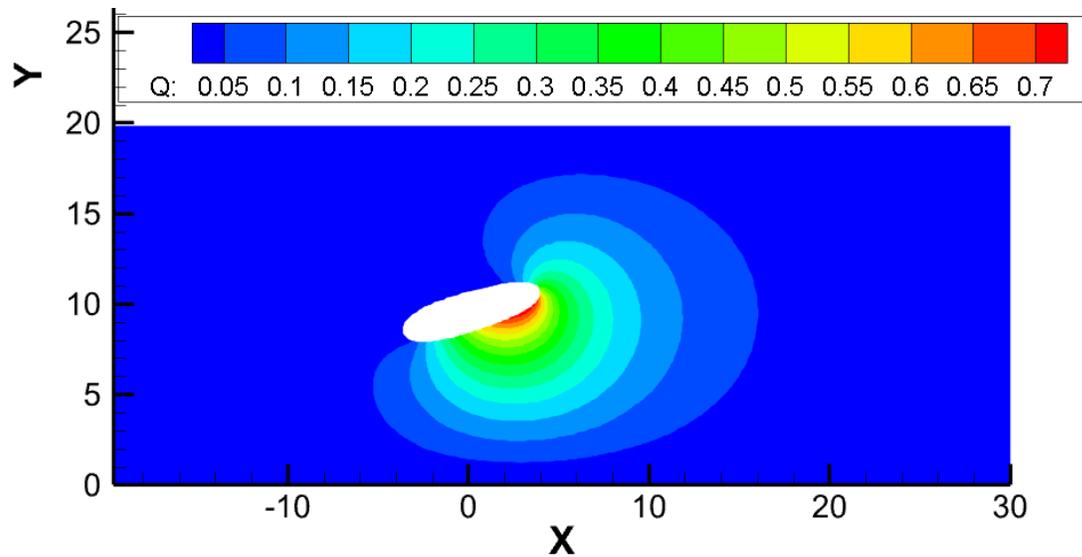


Figure 5.1.6 The solution of the Laplace equation for the fourth basis function ( $N4(i)$ ).

$\Delta x$	$\Delta y$
1.0	-0.5
1.0	-0.5
1.0	-0.5
1.0	-0.5

Table 5.1.3 The displacement of the control points that provokes the deformation of the B-Spline curve.

The applied movement to each one of the control points, using the methodology described in Chapter 3, results in the deformation of the entire unstructured grid, as presented in **Figure 5.1.7**, **Figure 5.1.8** and **Figure 5.1.9**. The original grid is in blue color, while the deformed one is in red color. As it can be seen, there is a smooth deformation of the unstructured grid, which fades-out as we approach the external rectangular boundary of the domain.

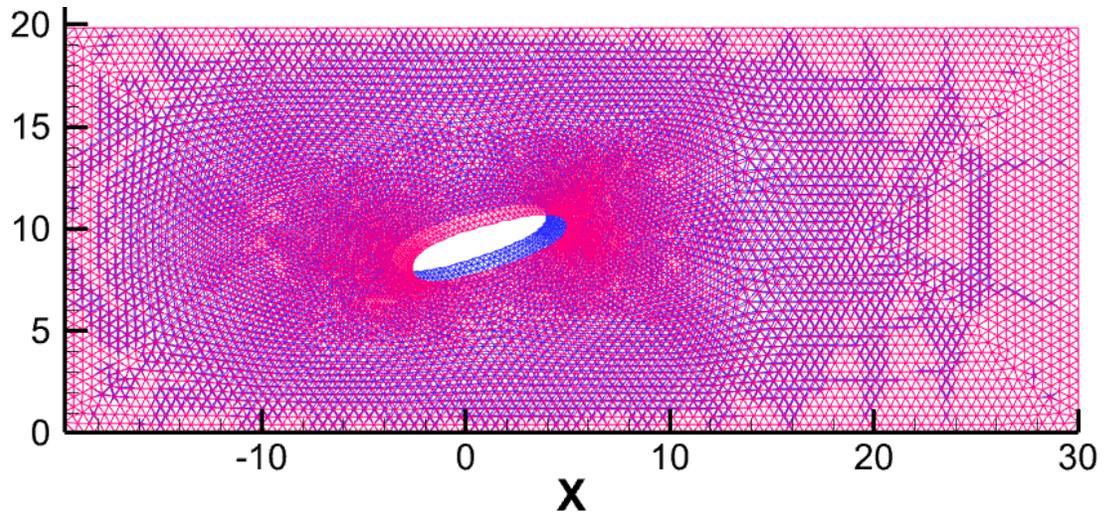


Figure 5.1.7 The initial (blue-colored) and the deformed (red-colored) unstructured grids.

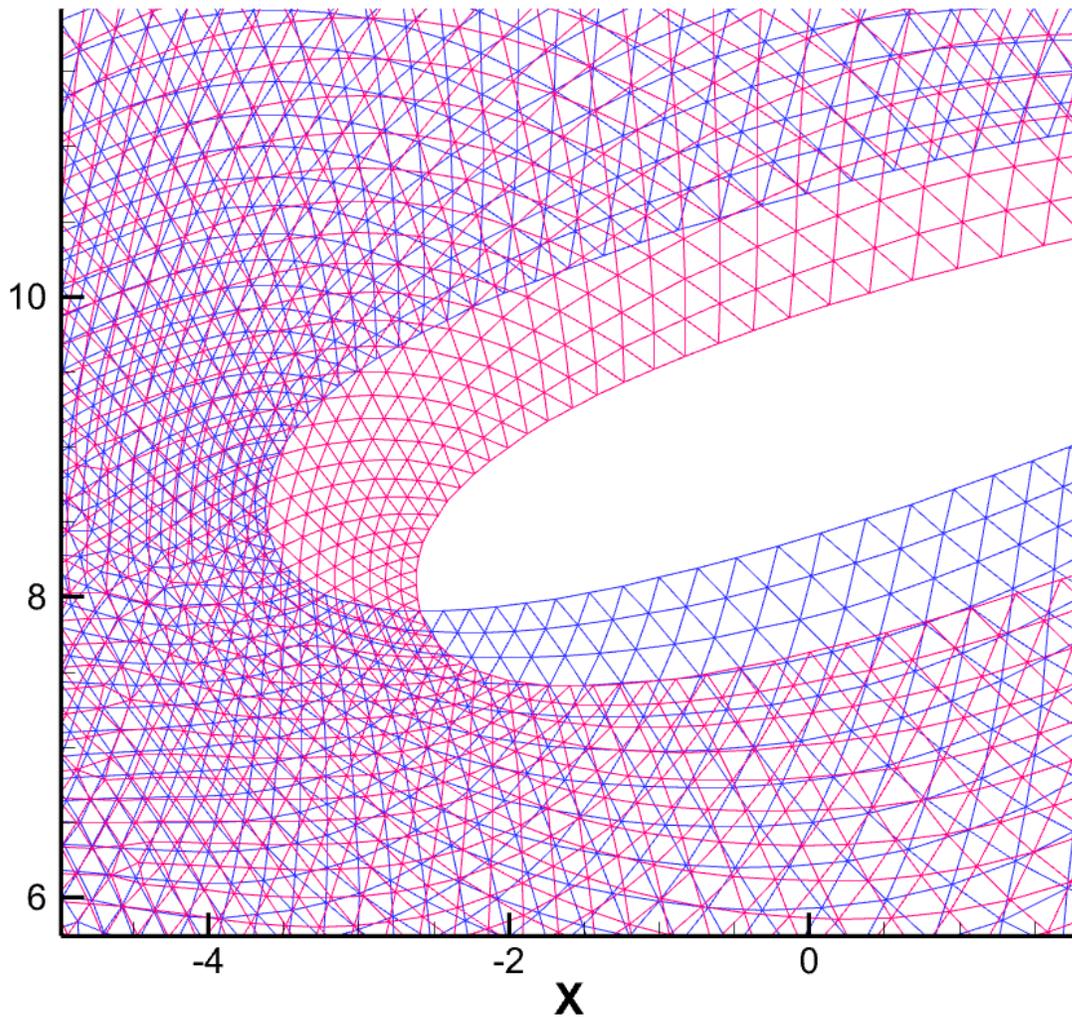


Figure 5.1.8 A more detailed examination of the initial and deformed grids, respectively.

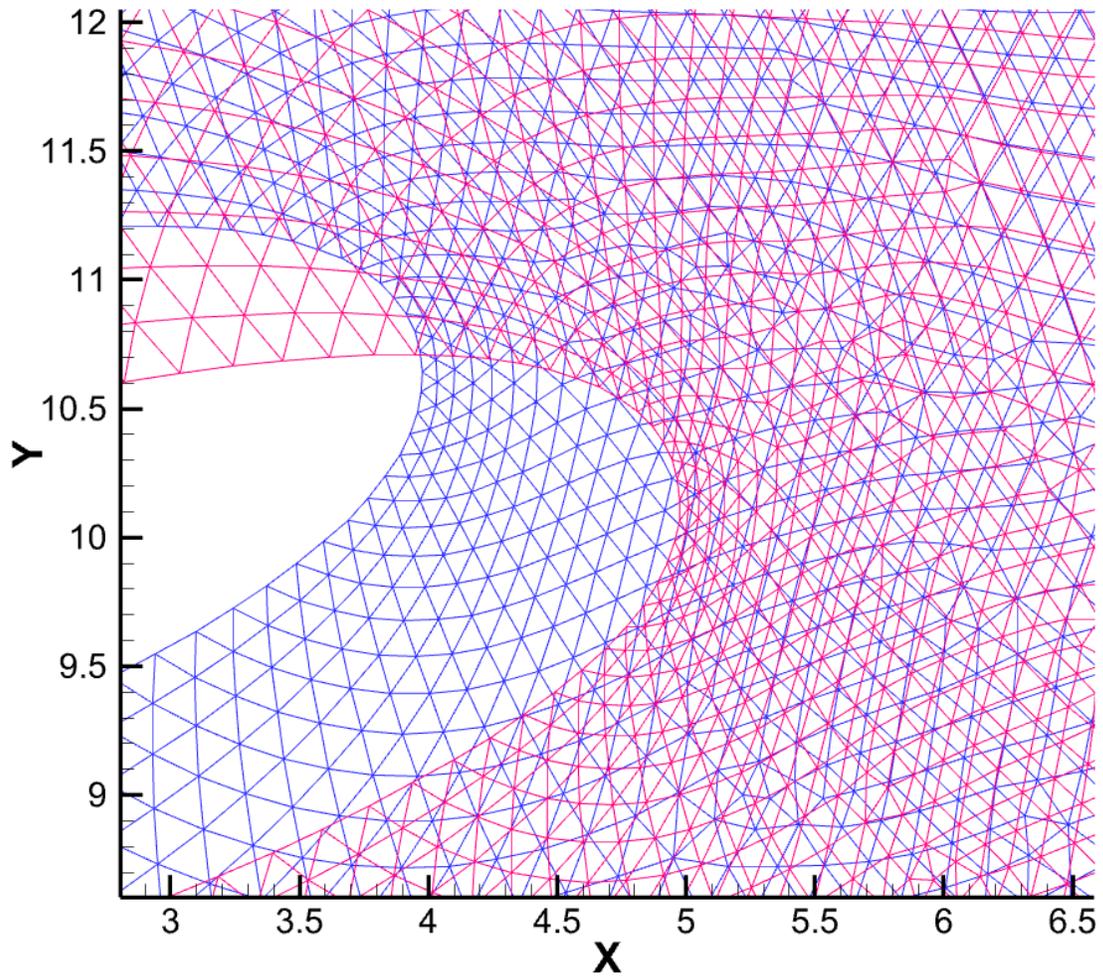


Figure 5.1.9 A visualization of the initial (blue-color) and deformed (red-color) grids, respectively.

## 5.2 Test Case 2: Two internal B-Spline boundaries.

The second test case consists of a rectangular domain with two internal boundaries, formulated as smooth closed periodic B-Spline curves of 3<sup>rd</sup> degree. The first one consists of 11 (different) control points, while the second one consists of 9 (different) control points. As being closed periodic B-Spline curves of 3<sup>rd</sup> degree, the 3 first control points of each curve are repeated at the end of the curve to produce the periodicity, resulting actually in 14 and 12 control points in total, respectively. The coordinates of the control points of the first B-Spline curve are listed in the following **Table 5.2.1**.

x-coordinate	y-coordinate
1.19990253	13.11501503
-0.66915739	13.17731762
-1.90475249	12.69120598
-2.98158932	11.97601223
-2.78379297	11.25177765
-1.73216796	11.19115543
-0.58905482	12.11818314
1.3957088	11.77107239
2.3145895	10.99261284
3.12043738	11.4656105
2.72119808	12.21872234

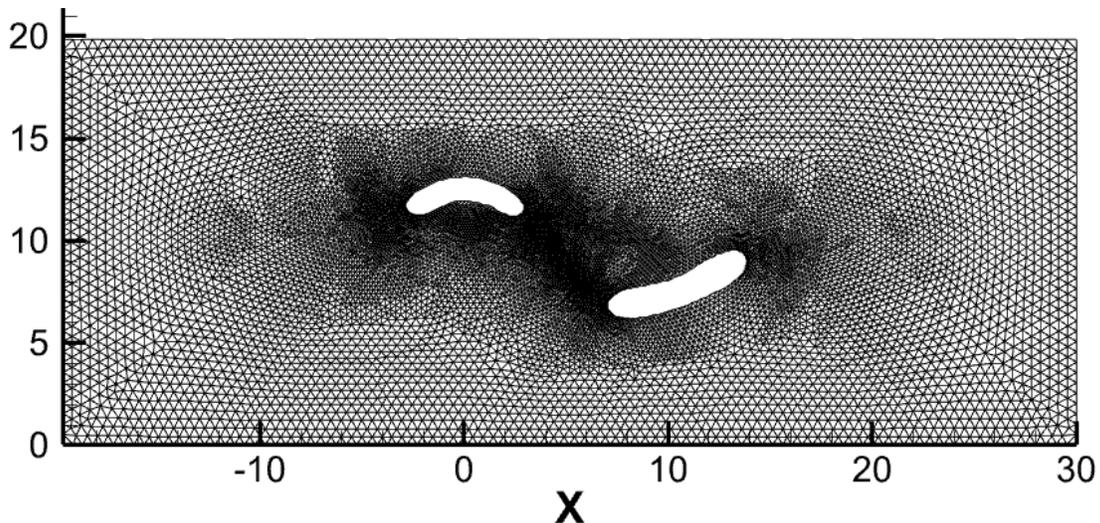
**Table 5.2.1** The control points' Cartesian coordinates belonging to the first B-Spline curve.

The coordinates of the control points of the second B-Spline curve are listed in the following **Table 5.2.2**.

x-coordinate	y-coordinate
10.29767895	7.94940615
7.90498877	7.76820374
6.81564045	7.02900267
7.23188925	6.11807156
9.2277689	6.25089693
11.40646648	6.95119238
14.11855793	8.37674522
13.75686646	9.82143688
11.96165276	9.24493408

**Table 5.2.2** The Cartesian coordinates of the control points belonging to the second B-Spline curve.

The initial unstructured grid, consisted of triangular elements, is depicted in **Figure 5.2.1** and **Figure 5.2.2**.



**Figure 5.2.1** The initial unstructured grid containing both B-Spline curve boundaries.

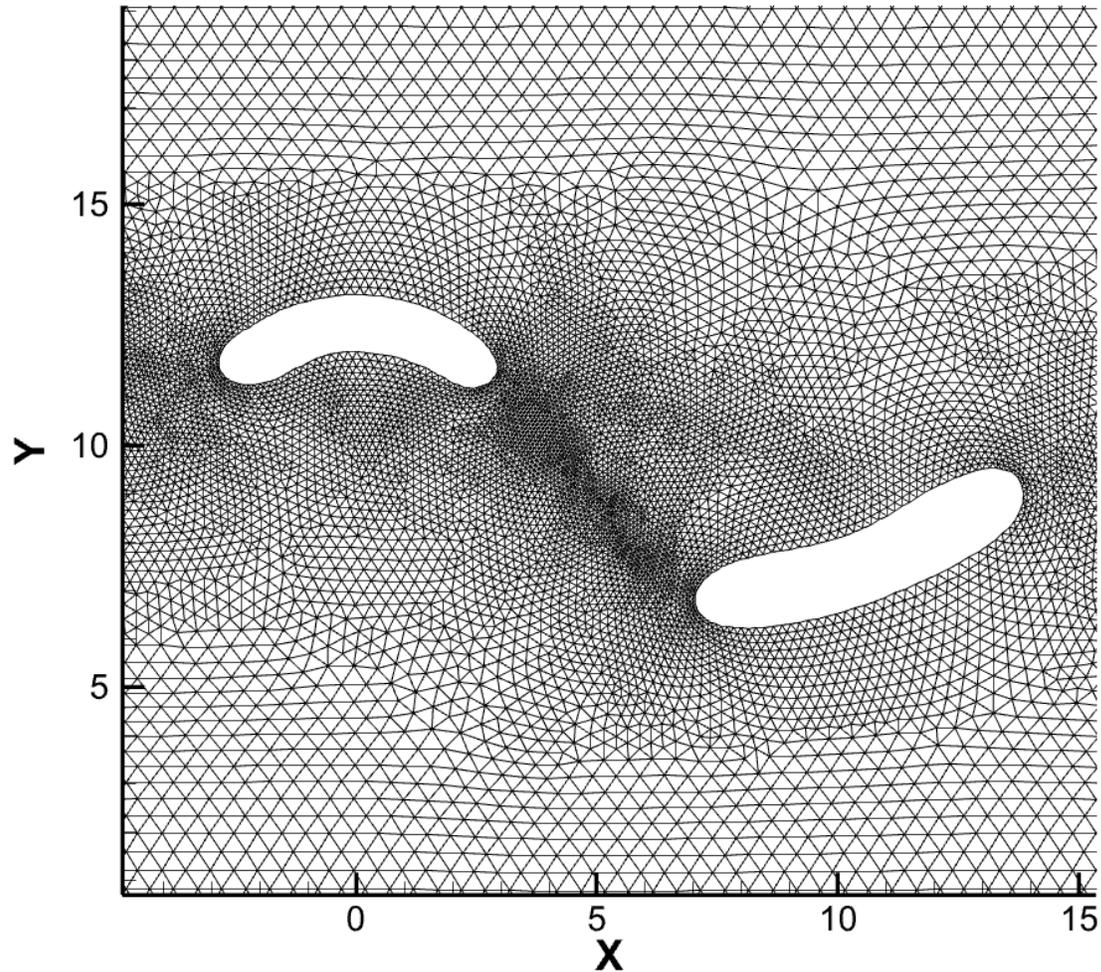


Figure 5.2.2 A closer examination of the unstructured grid depicted in Figure 5.2.1.

In **Figure 5.2.3** to **Figure 5.2.6**, the solution of the Laplace equation is depicted, for only 4 of the basis Functions (for brevity – 2 for each boundary), used as boundary conditions upon the internal boundary (control points 1 & 5 for the 1<sup>st</sup> B-Spline curve and control points 1 & 5 for the second B-Spline curve, respectively). Each basis Function corresponds to a B-Spline control point. The original unstructured grid was used for the solution of the Laplace equation (20=11+9 consecutive times, equal to the number of the (different) control points of the two B-Spline curves).

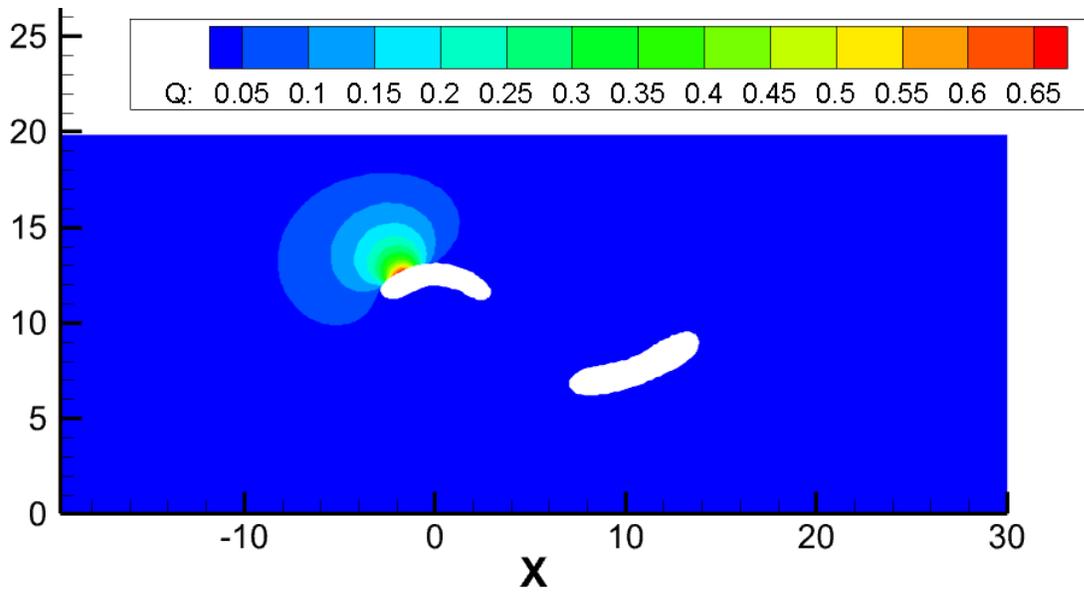


Figure 5.2.3 The solution of the Laplace equation following by the application of the first B-Spline basis function for the first B-Spline curve.

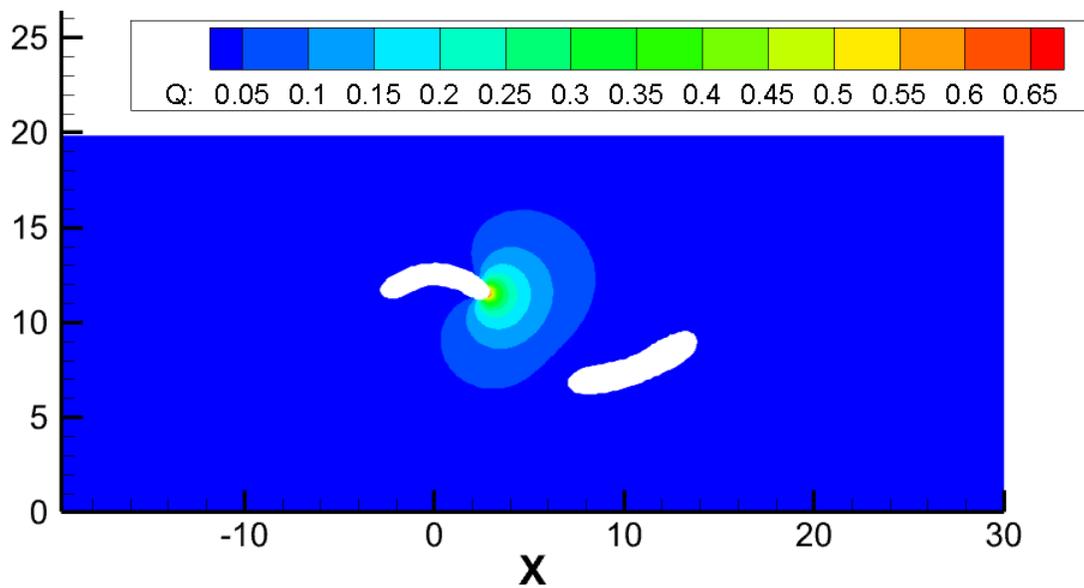


Figure 5.2.4 The solution of the Laplace equation for the second basis function for the fifth B-Spline curve.

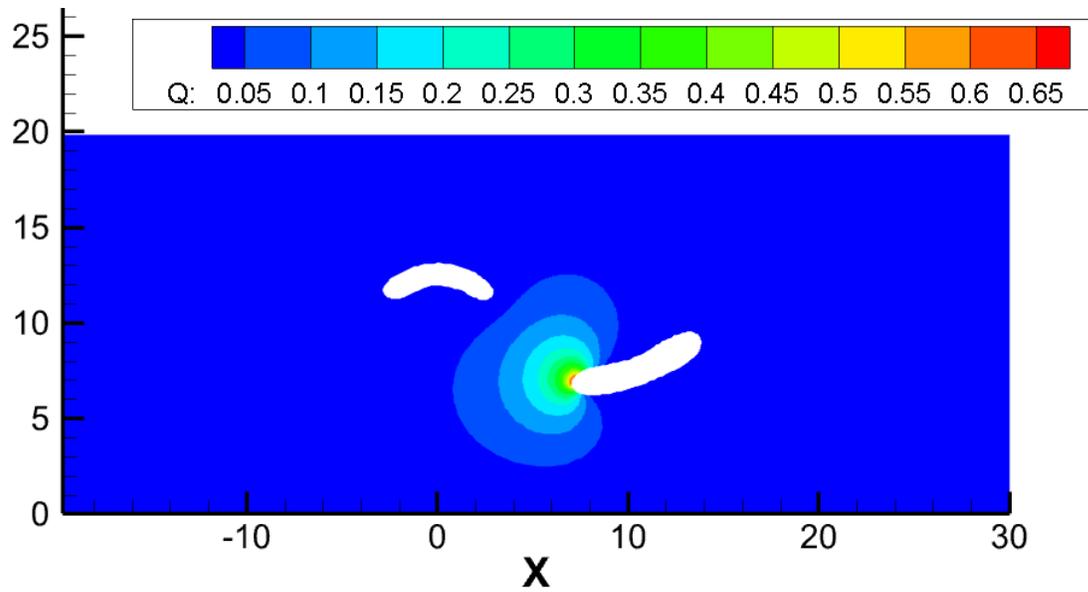


Figure 5.2.5 The solution of the Laplace equation for the first B-Spline basis function around the second B-Spline curve.

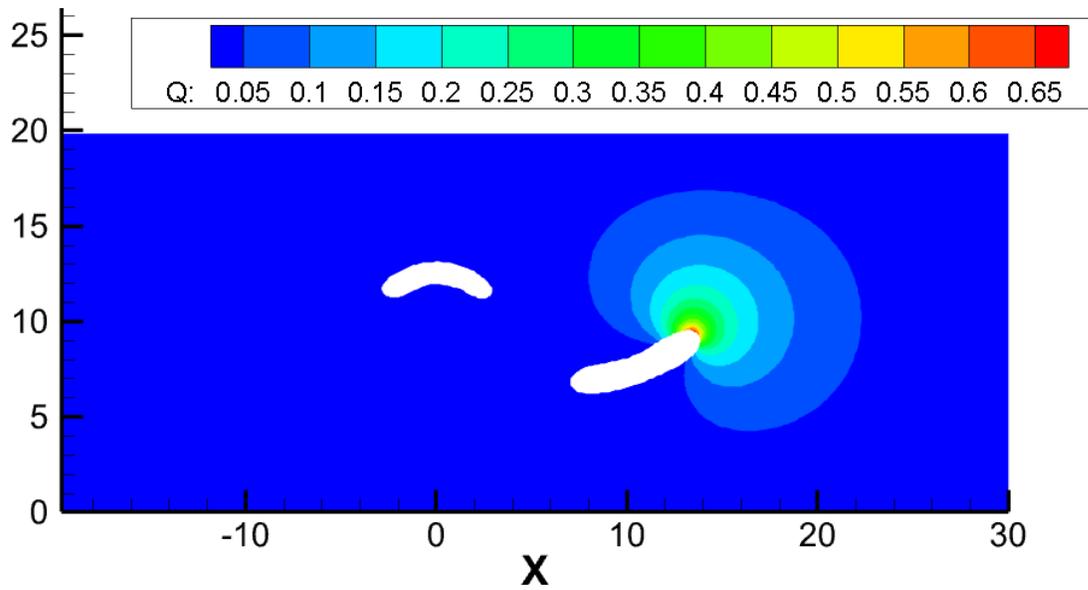


Figure 5.2.6 The solution of the Laplace equation for the fifth B-Spline basis function for the second B-Spline curve.

Then, a deformation to the original control points of the internal B-Spline boundaries is applied. The deformation to each one of the control points' coordinates is depicted in **Table 5.2.3** for the first boundary and in **Table 5.2.4** for the second one.

$\Delta x$	$\Delta y$
0.5	-1
0.5	-1
0.5	-1
0.5	-1
0.5	-1
0.5	-1
0.5	-1
0.5	-1
0.5	-1
0.5	-1
0.5	-1
0.5	-1

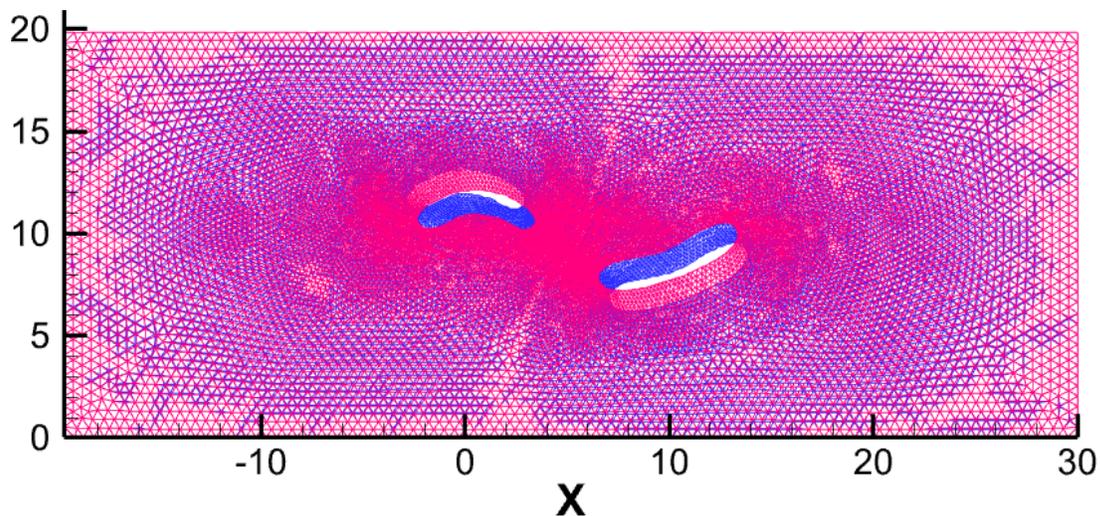
**Table 5.2.3** The displacement of the control points belonging to the first B-Spline curve.

$\Delta x$	$\Delta y$
-0.5	1
-0.5	1
-0.5	1
-0.5	1
-0.5	1
-0.5	1
-0.5	1
-0.5	1

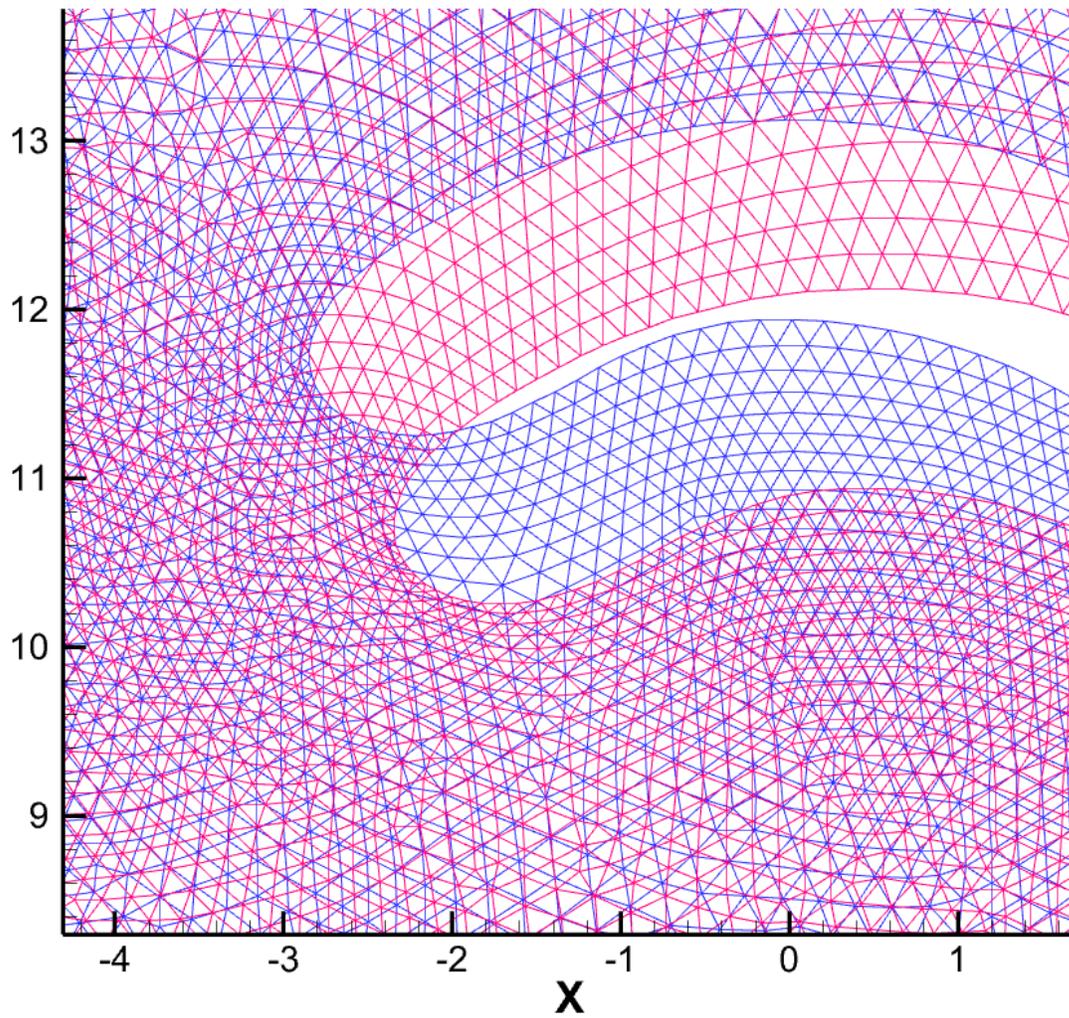
-0.5	1
-0.5	1

**Table 5.2.4** The displacement of the control points for the second B-Spline boundary curve.

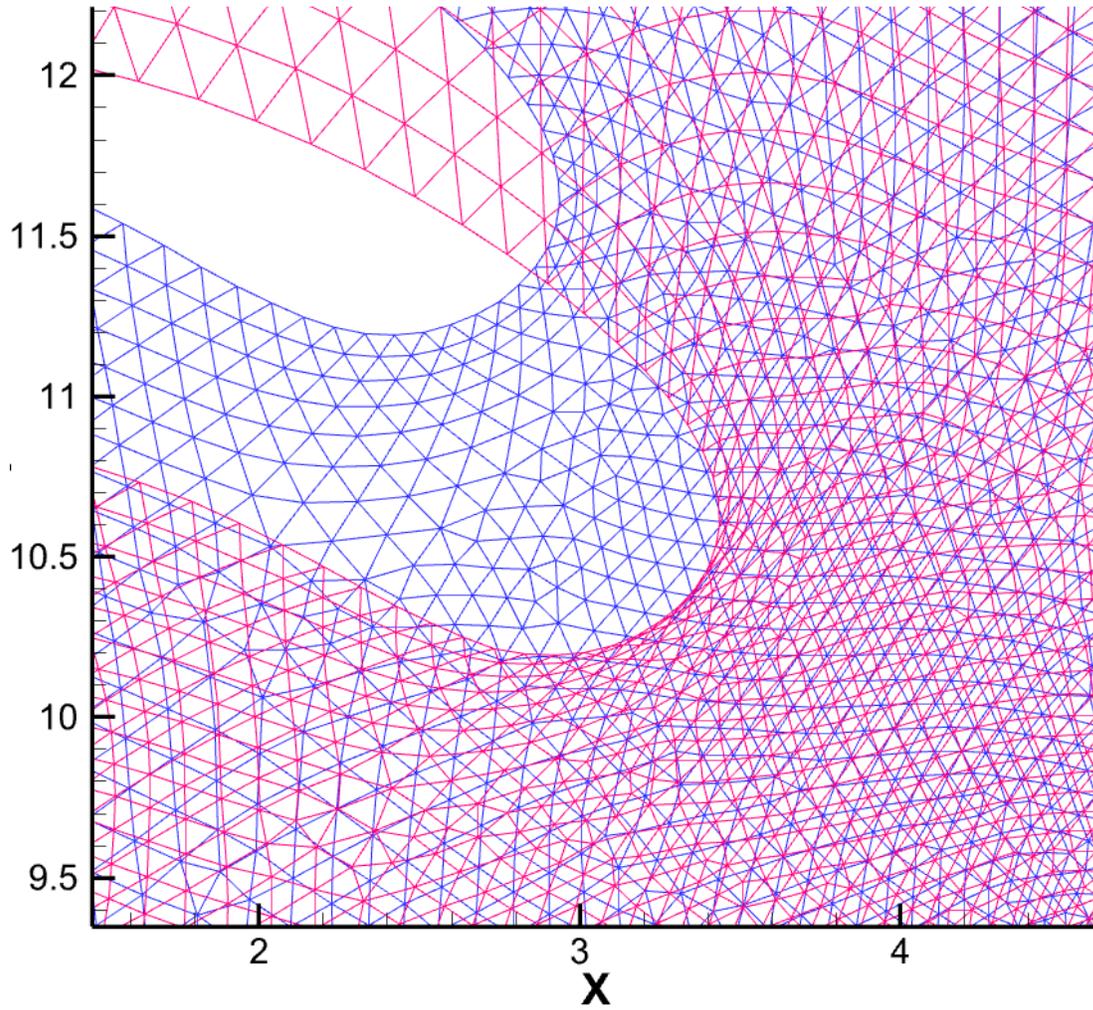
The applied movement to each one of the control points, using the methodology described in Chapter 3, results in the deformation of the entire unstructured grid, as presented in **Figure 5.2.7** to **Figure 5.2.11**. The original grid is in blue color, while the deformed one is in red color. As in the previous test case, there is a smooth deformation of the unstructured grid, which fades-out as we approach the external rectangular boundary of the domain.



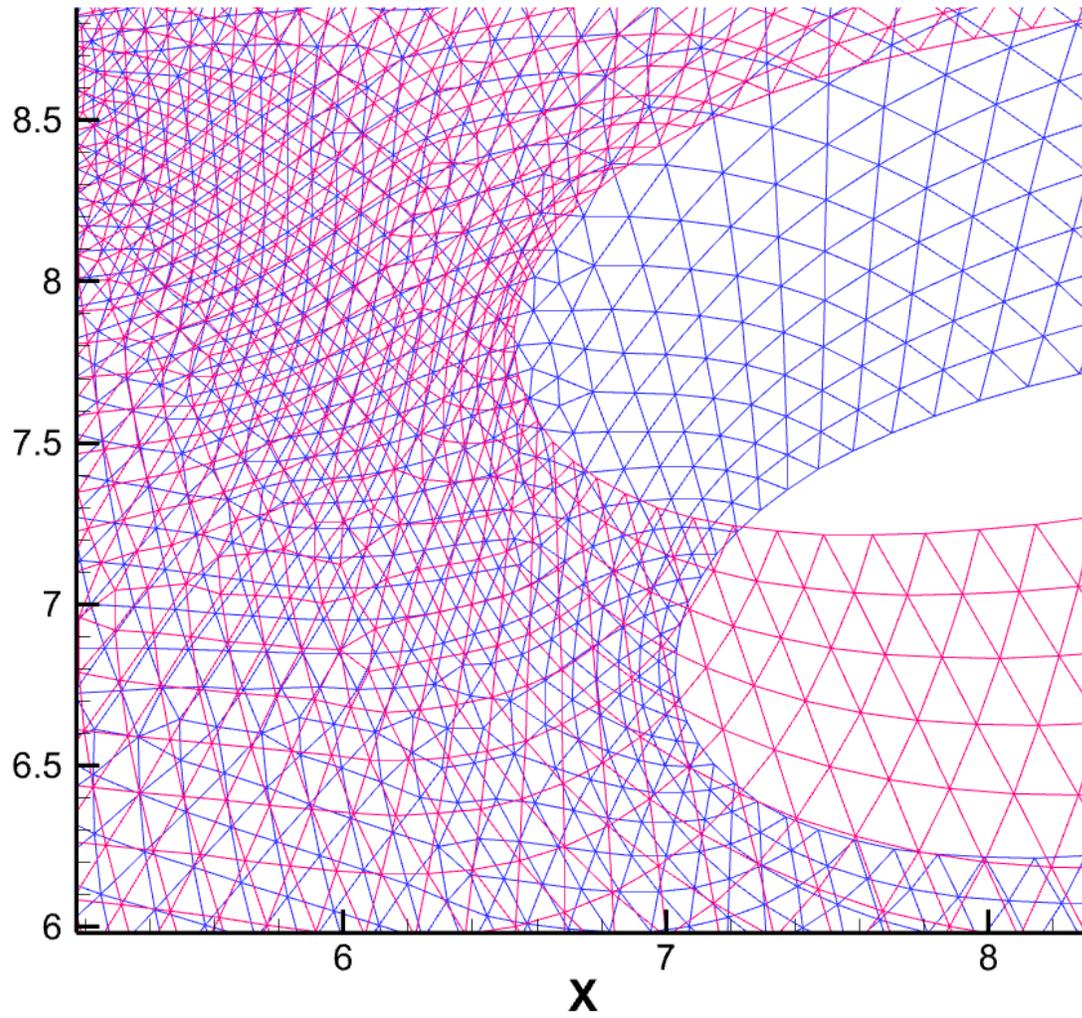
**Figure 5.2.7** A visual comparison between the initial (blue-colored) and the deformed (red-colored) grids.



**Figure 5.2.8** A more detailed examination of the initial (blue-colored) and deformed (red-colored) unstructured grids of the first B-Spline curve.



**Figure 5.2.9** A further closer examination of the initial (blue-colored) and deformed (red-colored) unstructured grids consisted of triangular elements, around the first B-Spline curve.



**Figure 5.2.10** A detailed examination of the initial (blue-colored) and deformed (red-colored) unstructured grids around the second B-Spline boundary.

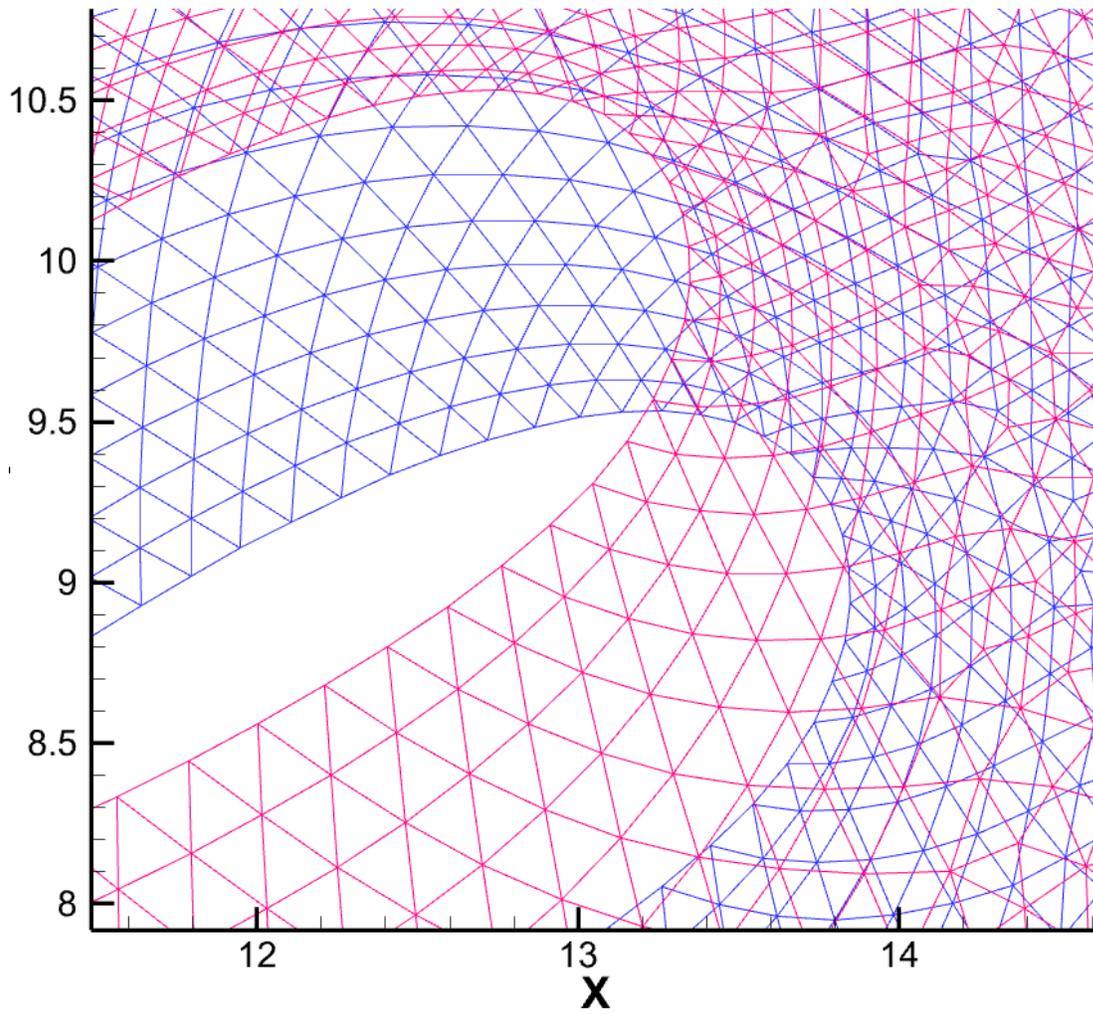


Figure 5.2.11 A closer examination of the initial (blue-colored) and deformed (red-colored) unstructured grids around the second B-Spline boundary.

## Chapter 6

### Conclusions

Three are the main contributions of the current work. The first is an extensive literature review of the existing geometry deformation techniques, based mainly on Free Form Deformation (FFD) and Harmonic functions. The second contribution is the testing of a new methodology (developed in the Turbomachines and Fluid Dynamics Laboratory, Technical University of Crete) for geometry/mesh deformation, combining B-spline theory and Harmonic functions, for the concurrent deformation of the parametrically-defined geometries and the corresponding computational mesh. This methodology is well-suited for aerodynamic design optimization. The third contribution is the development of a mesh-interpolation methodology, for unstructured-hybrid meshes. In this work, the validity of both methodologies was investigated and proved in various tests.

Concerning the geometry-deformation methodology, the following conclusions can be made:

- The proposed novel methodology allows for the concurrent deformation of the curved boundaries and the corresponding computational mesh (structured or unstructured).
- The deformation of the geometry boundaries is applied on the control points of the B-spline curves that describe the corresponding boundaries. Therefore, no reverse-engineering is required at the end of the procedure to compute the parametric definition of the final geometry.

- 
- The computation of the field of the harmonic functions requires significant computational resources. However, this computation is performed only once, at the beginning of the deformation procedure.
  - The resulting quality of the grid deformation is not very good, when large deformations are applied. In some cases negative volumes may result.
  - The proposed methodology is very promising. However, further research is required to expand its potential for aerodynamic shape optimization.
  - Comparing the proposed methodology with FFD, the former provides more smooth mesh deformations, with better local control, as the lattice points (which control the deformation) extend inside the mesh region. On the other hand, at the end of FFD deformation, the deformed geometry is not in parametric form, and a reverse-engineering procedure is required to compute its parametric (B-spline or NURBS) form. In the proposed methodology the control points of the parametric B-spline curves that define the geometry boundaries control both the deformation of the geometry (boundaries) and the computational mesh.
  - The proposed methodology has been developed for periodic B-spline curves, which provide smooth Basis functions.

## References

Amoignon, O., Hradil, J., Navratil, J. (2014), "A numerical study of adaptive FFD in aerodynamic shape optimization", *52nd Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics. National Harbor, Maryland, January 13-17, 2014. Available at: <https://doi.org/10.2514/6.2014-0899>

Amoiralis, E. (2005), "Παραμετροποίηση επιφανειών με χρήση της Τεχνικής Ελεύθερης Παραμόρφωσης (FFD) και εφαρμογή σε προβλήματα βέλτιστης αεροδυναμικής σχεδίασης". Master Thesis, Technical University of Crete, Chania, Greece. Available at: <https://doi.org/10.26233/heallink.tuc.14161>

Amoiralis, E.I., Nikolos, I.K. (2008), "Freeform Deformation Versus B-Spline Representation in Inverse Airfoil Design", *Journal of Computing and Information Science in Engineering*, Vol. 8, No. 2. Available at: <https://doi.org/10.1115/1.2906694>

Andreoli, M., Janka, A., Désidéri, J.-A. (2003), "Free-form-deformation parameterization for multilevel 3D shape optimization in aerodynamics", INRIA. Available at: <https://hal.inria.fr/inria-00071565>

Asmar, N.H., Grafakos, L. (2018), "Harmonic Functions and Applications", *Complex Analysis with Applications*, Springer, Cham, pp. 367–402. Available at: [https://doi.org/10.1007/978-3-319-94063-2\\_6](https://doi.org/10.1007/978-3-319-94063-2_6)

Astrachan, O. (2003), "Bubble sort: an archaeological algorithmic analysis", *ACM SIGCSE Bulletin*, Vol. 35, No. 1, pp. 1-5. Available at: <https://doi.org/10.1145/792548.611918>

Axler, S., Bourdon, P., Ramey, W. (1992), "Basic Properties of Harmonic Functions", *Harmonic Function Theory*, Springer, New York, pp. 1–29. Available at: [https://doi.org/10.1007/0-387-21527-1\\_1](https://doi.org/10.1007/0-387-21527-1_1)

Bai, J.Q., Chen, S. (2013), "Aerodynamic Optimization Design of Airfoil Based on Directly Manipulated FFD Technique", *Applied Mechanics and Materials*, Vol. 390, pp. 121–128. Available at: <https://doi.org/10.4028/www.scientific.net/AMM.390.121>

Barr, A.H. (1984), "Global and local deformations of solid primitives", *ACM SIGGRAPH Computer Graphics*, Vol. 18, No. 3, pp. 21–30. Available at: <https://doi.org/10.1145/964965.808573>

Belyaev, A. (2006), "On Transfinite Barycentric Coordinates", *SGP '06: Proceedings of the fourth Eurographics symposium on Geometry processing*, Eurographics Association, Goslar, Germany, pp. 89-99. Available at: <https://dl.acm.org/doi/10.5555/1281957.1281969>

Ben-Chen, M., Weber, O., Gotsman, C. (2009), "Variational Harmonic Maps for Space Deformation", in Hoppe, H. (Ed.s), *SIGGRAPH '09: ACM SIGGRAPH 2009 papers*, Association for Computing Machinery, New York, United States, pp. 1-11. Available at: <https://dl.acm.org/doi/10.1145/1576246.1531340>

Bloor, M.I.G., Wilson, M.J. (1995), "Efficient parametrization of generic aircraft geometry", *Journal of Aircraft*, Vol. 32, No. 2, pp. 1269–1275. Available at: <https://doi.org/10.2514/3.46874>

Braibant, V., Fleury, C. (1984), "Shape optimal design using B-splines", *Computer Methods in Applied Mechanics and Engineering*, Vol. 44, No. 3, pp. 247–267. Available at: [https://doi.org/10.1016/0045-7825\(84\)90132-4](https://doi.org/10.1016/0045-7825(84)90132-4)

Cai, S., Tautges, T.J. (2014), "Robust One-to-One Sweeping with Harmonic S-T Mappings and Cages", in Sarrate, J. and Staten M. (Ed.s), *Proceedings of the 22nd International Meshing Roundtable*, Springer, Cham, pp. 1–18. Available at: [https://link.springer.com/chapter/10.1007/978-3-319-02335-9\\_1](https://link.springer.com/chapter/10.1007/978-3-319-02335-9_1)

Casti, S., Livesu, M., Mellado, N., Abu Rumman, N., Scateni, R., Barthe, L., Puppo, E. (2019), "Skeleton based cage generation guided by harmonic fields", *Computers & Graphics*, Vol. 81, pp. 140–151. Available at: <https://doi.org/10.1016/j.cag.2019.04.004>

Chang, Y.-K., Rockwood, A.P. (1994), "A generalized de Casteljau approach to 3D free-form deformation", *SIGGRAPH '94: Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, Association for Computing Machinery, New York, United States of America, pp. 257–260. Available at: <https://doi.org/10.1145/192161.192220>

Chen, R., Gotsman, C. (2016), "On pseudo-harmonic barycentric coordinates", *Computer Aided Geometric Design*, Vol. 44, pp. 15–35. Available at: <https://doi.org/10.1016/j.cagd.2016.04.005>

Chien, E., Chen, R., Weber, O. (2016), "Bounded distortion harmonic shape interpolation", *ACM Transaction on Graphics*, Vol. 35, pp. 1–15. Available at: <https://doi.org/10.1145/2897824.2925926>

Coquillart, S. (1990), "Extended free-form deformation: a sculpturing tool for 3D geometric modeling", *SIGGRAPH '90: Proceedings of the 17th Annual Conference on Computer Graphics and Interactive Techniques*, Association for Computing Machinery, Dallas, United States of America, pp. 187–196. Available at: <https://doi.org/10.1145/97879.97900>

Coquillart, S., Jancéne, P. (1991), “Animated free-form deformation: an interactive animation technique”, *ACM SIGGRAPH Computer Graphics*, Vol. 25, No. 4, pp. 23–26. Available at: <https://doi.org/10.1145/127719.122720>

de Berg, M., Cheong, O., van Kreveld, M., Overmars, M. (2008), “Orthogonal Range Searching”, *Computational Geometry: Algorithms and Applications*, Springer, Berlin, Heidelberg, pp. 95–120. Available at: [https://doi.org/10.1007/978-3-540-77974-2\\_5](https://doi.org/10.1007/978-3-540-77974-2_5)

Désidéri, J., Janka, A. (2004), “Multilevel Shape Parameterization for Aerodynamic Optimization – Application to Drag and Noise Reduction of Transonic/Supersonic Business Jet, in Neittaanmaki, P., Rossi, T., Korotov S., Onate, E., Periaux J., and Knorzer D. (Ed.s), *European Congress on Computational Methods in Applied Sciences and Engineering*, Jyväskylä, Finland.

Faloutsos, P., van de Panne, M., Terzopoulos, D. (1997), “Dynamic Free-Form Deformations for Animation Synthesis”, *IEEE Transactions on Visualization and Computer Graphics*, Vol. 3, No. 3, pp. 201–214. Available at: <https://doi.org/10.1109/2945.620488>

Feng, J., Heng, P.-A., Wong, T.-T. (2012), “Accurate B-spline Free-Form Deformation of Polygonal Objects”, *Journal of Graphics Tools*, pp. 11-27. Available at: <https://www.tandfonline.com/doi/abs/10.1080/10867651.1998.10487492>

Feng, J., Ma, L., Peng, Q. (1996), “A new free-form deformation through the control of parametric surfaces”, *Computers & Graphics*, Vol. 20, No. 4, pp. 531–539. Available at: [https://doi.org/10.1016/0097-8493\(96\)00025-8](https://doi.org/10.1016/0097-8493(96)00025-8)

- Floater, M.S. (2003), "Mean value coordinates", *Computer Aided Geometric Design*, Vol. 20, No. 1, pp. 19–27. Available at: [https://doi.org/10.1016/S0167-8396\(03\)00002-5](https://doi.org/10.1016/S0167-8396(03)00002-5)
- Floater, M.S., Hormann, K., Kós, G. (2006), "A general construction of barycentric coordinates over convex polygons", *Advances in Computational Mathematics*, Vol. 24, pp. 311–331. Available at: <https://doi.org/10.1007/s10444-004-7611-6>
- Gagliardi, F., Giannakoglou, K.C. (2019), "A two-step radial basis function-based CFD mesh displacement tool", *Advances in Engineering Software*, Vol. 128, pp. 86–97. Available at: <https://doi.org/10.1016/j.advengsoft.2018.11.011>
- Gain, J.E., Dodgson, N.A. (1999), "Adaptive Refinement and Decimation under Free-Form Deformation", *Eurographics UK '99*, Cambridge.
- Gordon, W.J., Wixom, J.A. (1974), "Pseudo-Harmonic Interpolation on Convex Domains", *SIAM Journal on Numerical Analysis*, Vol. 11, No. 5, pp. 909–933. Available at: <https://doi.org/10.1137/0711072>
- Griessmair, J., Purgathofer, W. (1989), "Deformation of Solids with Trivariate B-Splines", in Hansmann, W., Hopgood F.R.A., Strasser W. (Ed.s), *Eurographics '89*, Eurographics Association. Available at: <https://doi.org/10.2312/egtp.19891010>
- Hardee, E., Chang, K.-H., Tu, J., Choi, K.K., Grindeanu, I., Yu, X. (1999), "A CAD-based design parameterization for shape optimization of elastic solids", *Advances in Engineering Software*, Vol. 30, No. 3, pp. 185–199. Available at: [https://doi.org/10.1016/S0965-9978\(98\)00065-9](https://doi.org/10.1016/S0965-9978(98)00065-9)
- He, Y., Wang, H., Fu, C.-W., Qin, H. (2009), "A divide-and-conquer approach for automatic polycube map construction", *Computers & Graphics*, Vol. 33, No. 3, pp. 369–380. Available at: <https://doi.org/10.1016/j.cag.2009.03.024>

Hicks, R.M., Henne, P.A. (1978), "Wing Design by Numerical Optimization", *Journal of Aircraft*, Vol. 15, No. 7, pp. 407–412. Available at: <https://doi.org/10.2514/3.58379>

Hirota, G., Maheshwari, R., Lin, M.C. (2000), "Fast volume-preserving free-form deformation using multi-level optimization", *Computer-Aided Design*, Vol. 32, No. 8-9, pp. 499–512. Available at: [https://doi.org/10.1016/S0010-4485\(00\)00038-5](https://doi.org/10.1016/S0010-4485(00)00038-5)

Hormann, K., Floater, M.S. (2006), "Mean value coordinates for arbitrary planar polygons", *ACM Transactions on Graphics*, Vol. 25, No. 4, pp. 1424–1441. Available at: <https://doi.org/10.1145/1183287.1183295>

Hormann, K., Sukumar, N. (2008), "Maximum Entropy Coordinates for Arbitrary Polytopes", *Computer Graphics Forum*, Vol. 27, No. 5, pp. 1513–1520. Available at: <https://doi.org/10.1111/j.1467-8659.2008.01292.x>

Hua, J., Qin, H. (2003), "Free-form deformations via sketching and manipulating scalar fields", *SM: 03: Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications*, Association for Computing Machinery, Seattle, Washington, USA, pp. 328–333. Available at: <https://doi.org/10.1145/781606.781660>

Ilic, S., Fua, P. (2002), "Using Dirichlet Free Form Deformation to Fit Deformable Models to Noisy 3-D Data", *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part II*, Springer-Verlag, Berlin, Heidelberg, pp. 704–717. Available at: [https://link.springer.com/chapter/10.1007/3-540-47967-8\\_47](https://link.springer.com/chapter/10.1007/3-540-47967-8_47)

Jacobson, A., Baran, I., Popovic, J., Sorkine, O. (2011), "Bounded Biharmonic Weights for Real-Time Deformation", *ACM Transactions on Graphics*, Vol. 20, No. 4, pp. 1-8. Available at: <https://dl.acm.org/doi/10.1145/2010324.1964973>

Joshi, P., Meyer, M., DeRose, T., Green, B., Sanocki, T. (2007), "Harmonic coordinates for character articulation", *ACM Transactions on Graphics (TOG)*, Vol. 26, No. 3, pp. 71-es. Available at: <https://dl.acm.org/doi/10.1145/1276377.1276466>

Ju, T., Schaefer, S., Warren, J. (2005), "Mean value coordinates for closed triangular meshes", *ACM Transactions on Graphics*, Vol. 24, No. 3, pp. 561-566. Available at: <https://dl.acm.org/doi/10.1145/1073204.1073229>

Kobayashi, K.G., Ootsubo, K. (2003), "t-FFD: free-form deformation by using triangular mesh", *SM '03: Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications*, Association for Computing Machinery, New York, USA, pp. 226–234. Available at: <https://doi.org/10.1145/781606.781641>

LaCourse, D.E. (1995), *Handbook of solid modeling*, McGraw-Hill, Inc., USA.

Lamousin, H.J., Waggenspack, N.N. (1994), "NURBS-based free-form deformations", *IEEE Computer Graphics and Applications*, Vol. 14, No. 6, pp. 59–65. Available at: <https://doi.org/10.1109/38.329096>

Leloudas, S. (2015), *On the optimal design of airfoils*, Diploma Thesis, Technical University of Crete, Chania, Greece. Available at: <https://dias.library.tuc.gr/view/26830>

Leloudas, S.N., Lygidakis, G.N., Eskantar, A.I., Nikolos, I.K. (2020), "A robust methodology for the design optimization of diffuser augmented wind turbine shrouds", *Renewable Energy*, Vol. 150, pp. 722–742. Available at: <https://doi.org/10.1016/j.renene.2019.12.098>

Leloudas, S.N., Strofylas, G.A. and Nikolos, I.K. (2018), "Constrained airfoil optimization using the area-preserving free-form deformation", *Aircraft Engineering*

and Aerospace Technology, Vol. 90, No. 6, pp. 914-926. Available at: <https://doi.org/10.1108/AEAT-10-2016-0184>

Li, M., Tong, R. (2012), "All-hexahedral mesh generation via inside-out advancing front based on harmonic fields", *The Visual Computer*, Vol. 28, pp. 839-847. Available at: <https://doi.org/10.1007/s00371-012-0707-y>

Li, X., Hu, S. (2013), "Poisson Coordinates", *IEEE Transactions on Visualization and Computer Graphics*, Vol. 19, No. 2, pp. 344-352. Available at: <https://doi.org/10.1109/TVCG.2012.109>

Li, X., Xu, H., Wan, S., Yin, Z., Yu, W. (2010), "Feature-aligned harmonic volumetric mapping using MFS", *Computers & Graphics*, Vol. 34, No. 3, pp. 242-251. Available at: <https://doi.org/10.1016/j.cag.2010.03.004>

Li, Y., Li, X., Ren, X. (2016), "Aerodynamic optimization of a high-expansion ratio organic radial-inflow turbine", *Journal of Mechanical Science and Technology*, Vol. 30, pp. 5485-5490. Available at: <https://doi.org/10.1007/s12206-016-1116-3>

Li, Z., Liu, C. (2012), "Holomorphic Coordinates for Conformal Deformations Using Point Handles", *Journal of Information and Computational Science*, Vol. 10, pp. 4827-4834. Available at: <https://doi.org/10.12733/jics20102257>

Liu, L., Zhang, Y., Hughes, T.J.R., Scott, M.A., Sederberg, T.W. (2014), "Volumetric T-spline construction using Boolean operations", *Engineering with Computers*, Vol. 30, pp. 425-439. Available at: <https://doi.org/10.1007/s00366-013-0346-6>

Liu, Y., Duan, Z., Chen, S. (2017), "Aerodynamic Shape Optimization Design of Wing-Body Configuration Using a Hybrid FFD-RBF Parameterization Approach", *Journal of Physics*. Available at: <https://doi.org/10.1088/1742-6596/916/1/012001>

MacCracken, R., Joy, K.I. (1996), "Free-form deformations with lattices of arbitrary topology", *SIGGRAPH '96: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, Association for Computing Machinery, New York, USA, pp. 181–188. Available at: <https://doi.org/10.1145/237170.237247>

Manson, J., Schaefer, S. (2010), "Moving Least Squares Coordinates", *Computer Graphics Forum*, Vol. 29, No. 5, pp. 1517–1524. Available at: <https://doi.org/10.1111/j.1467-8659.2010.01760.x>

Masters, D.A., Taylor, N.J., Rendall, T., Allen, C.B., Poole, D.J. (2015), "Review of Aerofoil Parameterisation Methods for Aerodynamic Shape Optimisation", *53rd AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics. Available at: <https://doi.org/10.2514/6.2015-0761>

Mavronikola, M. (2017), *Shape Parameterization and Grid Adaptation using Harmonic Coordinates*, Diploma Thesis, National Technical University of Athens, Athens. Available at: <https://dspace.lib.ntua.gr/xmlui/handle/123456789/46236>

Möbius, A.F. (1827), *Der barycentrische Calcul*.

Ono, Y., Bing-Yu Chen, Nishita, T., Jieqing Feng, (2002), "Free-form deformation with automatically generated multiresolution lattices", *First International Symposium on Cyber Worlds*, IEEE, Tokyo, Japan. Available at: <https://doi.org/10.1109/CW.2002.1180915>

Pickett, R.M., Rubinstein, M.F., Nelson, R.B. (1973), "Automated Structural Synthesis Using a Reduced Number of Design Coordinates", *AIAA Journal*, Vol. 11, No. 4, pp. 489–494. Available at: <https://doi.org/10.2514/3.50489>

Piegl, L., Tiller, W. (1995), *The NURBS Book*, Springer, Berlin, Heidelberg. Available at: <https://doi.org/10.1007/978-3-642-97385-7>

Pinkall, U., Polthier, K. (1993), "Computing Discrete Minimal Surfaces and Their Conjugates", *Experimental Mathematics*, Vol. 2, No. 1, pp. 15–36. Available at: <https://doi.org/10.1080/10586458.1993.10504266>

Ren, J. (2016), *Aerodynamic shape optimization by multi-fidelity modeling and manifold mapping*, Master Thesis, Iowa State University, Ames. Available at: <https://doi.org/10.31274/etd-180810-4667>

Ronzheimer, A. (2005), "Shape Parametrization Using Freeform Deformation. MEGAFLOW - Numerical Flow Simulation for Aircraft Design", in Kroll N., Fassbender J.K. (Ed.s), *Notes on Numerical Fluid Mechanics and Multidisciplinary Design (NNFM)*, Vol. 89, Springer, Berlin, Heidelberg, pp. 211-222. Available at: [https://doi.org/10.1007/3-540-32382-1\\_15](https://doi.org/10.1007/3-540-32382-1_15)

Ronzheimer, A., (2002), "Post-Parameterization of Complex CAD-Based Aircraft-Shapes Using Freeform Deformation", *8th International Conference on Numerical Grid Generation in Computational Field Simulations*, 1-5 June, Honolulu, USA. Available at: <https://elib.dlr.de/12551/>

Rustamov, R.M., (2008), "Boundary Element Formulation of Harmonic Coordinates".

Samareh, J., (2004), "Aerodynamic Shape Optimization Based on Free-Form Deformation", *10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, American Institute of Aeronautics and Astronautics, 30 August – 1 September, Albany, New York. Available at: <https://doi.org/10.2514/6.2004-4630>

Samareh, J., (1999), "A Survey of Shape Parameterization Techniques", *AIAA Journal*, Vol. 39, No. 5, pp. 877-884. Available at: <https://arc.aiaa.org/doi/10.2514/2.1391>.

Samareh, J.A., (1999), "A Novel Shape Parameterization Approach", *Journal of Aircraft*, Vol. 38, No. 6, pp. 1015-1024. Available at: <https://arc.aiaa.org/doi/10.2514/2.2888>

Schein, S., Elber, G., (2004), "Discontinuous free form deformations", *12th Pacific Conference on Computer Graphics and Applications*, IEEE, 6-8 October, Seoul, South Korea, pp. 227–236. Available at: <https://doi.org/10.1109/PCCGA.2004.1348353>

Sederberg, T.W., Parry, S.R., (1986), "Free-form deformation of solid geometric models", *ACM SIGGRAPH '86 Computer Graphics*, Vol. 20, No. 4, pp. 151–160, New York, NY, USA. Available at: <https://doi.org/10.1145/15922.15903>

Selim, M., Koomullil, R.P. (2016), "Mesh Deformation Approaches – A Survey", *Journal of Physical Mathematics*, Vol. 7, No. 2. Available at: <https://doi.org/10.4172/2090-0902.1000181>

Shepard, D., (1968), "A two-dimensional interpolation function for irregularly-spaced data", *ACM '68: Proceedings of the 1968 23rd ACM National Conference*, Association for Computing Machinery, New York, USA, pp. 517–524. Available at: <https://doi.org/10.1145/800186.810616>

Song, W., Yang, X., (2005), "Free-form deformation with weighted T-spline", *The Visual Computer*, Vol. 21, pp. 139–151. Available at: <https://doi.org/10.1007/s00371-004-0277-8>

Wachspress, E., (1975), *A Rational Finite Element Basis*, Academic Press.

Wakayama, S., Kroo, I., (2012), "Subsonic wing planform design using multidisciplinary optimization", *Journal of Aircraft*, Vol. 32, No. 4, pp. 746-753. Available at: <https://doi.org/10.2514/3.46786>

Weber, O., Poranne, R., Gotsman, C., (2012), "Biharmonic Coordinates", *Computer Graphics Forum*, Vol. 31, No. 8, pp. 2409–2422. Available at: <https://doi.org/10.1111/j.1467-8659.2012.03130.x>

Witteveen, J., Bijl, H., (2009), "Explicit Mesh Deformation Using Inverse Distance Weighting Interpolation", *19th AIAA Computational Fluid Dynamics Conference*, 22-25 June, San Antonio, Texas, USA. Available at: <https://doi.org/10.2514/6.2009-3996>

Xia, J., He, Y., Yin, X., Han, S., Gu, X., (2010), "Direct-Product Volumetric Parameterization of Handlebodies via Harmonic Fields", *2010 Shape Modeling International Conference*, IEEE, Aix-en-Provence, France, pp. 3–12. Available at: <https://doi.org/10.1109/SMI.2010.10>

Xu, K., Zhang, H., Cohen-Or, D., Xiong, Y., (2009), "Dynamic harmonic fields for surface processing", *Computers & Graphics*, Vol. 33, No. 3, pp. 391–398. Available at: <https://doi.org/10.1016/j.cag.2009.03.022>

Yoshizawa, S., Belyaev, A.G., Seidel, H.-P., (2002), "A Simple Approach to Interactive Free-Form Shape Deformations", *10th Pacific Conference on Computer Graphics and Applications*, IEEE, Beijing, China, pp. 471–471. Available at: <https://doi.org/10.1109/PCCGA.2002.1167905>

Zayer, R., Rössl, C., Karni, Z., Seidel, H.-P., (2005), "Harmonic Guidance for Surface Deformation", *Computer Graphics Forum*, Vol. 24, No. 3, pp. 601–609. Available at: <https://doi.org/10.1111/j.1467-8659.2005.00885.x>

Zervas, C., (2018), *Shape Parameterization and Adaptive Deformation of 3D Computational Grids using Harmonic Coordinates, Applications in Aerodynamic*

*Optimization*, Diploma Thesis, National Technical University of Athens, Athens.  
Available at: <http://dx.doi.org/10.26240/heal.ntua.15817>

Zhang, M., Rizzi, A.W., Nangia, R.K., (2015), "Transonic Airfoils and Wings Design Using Inverse and Direct Methods", *53rd AIAA Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, 5-9 January 2015, Kissimmee, Florida. Available at: <https://doi.org/10.2514/6.2015-1943>

Zhao, Y., Forhad, A., (2003), "A general method for simulation of fluid flows with moving and compliant boundaries on unstructured grids", *Computer Methods in Applied Mechanics and Engineering*, Vol. 192, No. 39-40, pp. 4439–4466. Available at: [https://doi.org/10.1016/S0045-7825\(03\)00424-9](https://doi.org/10.1016/S0045-7825(03)00424-9)