

# AUTOMATIC FREE-FORM DEFORMATION AS GEOMETRIC PARAMETERIZATION FOR AERODYNAMIC DESIGN

Marco Aurélio Leonel Matunaga<sup>1</sup> & João Luiz F. Azevedo<sup>2</sup>

<sup>1</sup>Instituto Tecnológico de Aeronáutica, DCTA/ITA, 12228-900, São José dos Campos, SP, Brazil

<sup>2</sup>Instituto de Aeronáutica e Espaço, DCTA/IAE/ALA, 12228-904, São José dos Campos, SP, Brazil

## Abstract

Aerodynamic optimization requires the use of adequate methodologies to describe the geometry and its alterations. In order to achieve this goal, a CAD program can be used, but the automation, performance and implementation make this approach unfeasible for industrial applications. Currently, one of the most employed geometry parameterization methods is known as free-form deformation or FFD. This methodology is a simple way to describe the geometry and its deformation. The parameterization process can be achieved by creating a box that involves the geometry followed by a matrix-vector multiplication. One issue is the need to create one box for each geometry, which can be a challenging task for complex geometries. In the present study, in order to accomplish this task, we show an automatic FFD algorithm to parameterize the geometry and the matrix-vector multiplication processes used to alter the shape. Therefore, we create an user-friendly FFD process. We show results for two and three dimensional problems for different aerodynamic parameters such as dihedral, span, sweep, twist and wing shape. Moreover, the outcomes here presented can help engineers interested in understanding how the FFD method can be employed in an aerodynamic shape optimization study.

**Keywords:** Free-Form Deformation, Geometric Parameterization, Shape Alterations, Aerodynamic Design

## 1. Introduction

Due to recent improvements in computational components and efficient algorithms, engineers are able to study more complex problems through computational fluid mechanics. Currently, there are several optimization studies of airliner wings and wing-body configurations with high-fidelity models, such as the Reynolds-Averaged Navier-Stokes equations [1, 2, 3]. Besides that, there are efforts intended at solving aerostructural optimization problems of three dimensional wings [4]. The adjoint method is one approach that makes it possible to find solutions of such complex optimization problems in an efficient manner. The computational time required for calculating derivatives, evaluated by finite differences and complex-step method, is proportional to the number of design variables. Therefore, those methods are very expensive to be applied to complex or real problems. On the other hand, the computational time to compute derivatives by the adjoint method is proportional to the number of variables of interest. As a result, this approach is more efficient than finite differences and complex-step methods.

An aerodynamic optimization process sometimes employs different computational tools, one of these tools is used to describe the geometry and its alterations. According to Ref. [5], a successful geometric parameterization process has to be automated, it has to be capable to provide geometric alterations and it has to supply derivatives. Besides that, it is necessary to do all the tasks mentioned before in an efficient way, *i.e.*, with a compact number of design variables.

One of the geometric parameterizations that fulfills the requirements of Ref. [5] is the free-form deformation method, sometimes referred as FFD. Initially proposed to deform objects in the context of soft animation [6], free-form Deformation employed as a geometric parameterization has many

advantages such as local and global shape control, easy derivative computation and a compact set of design variables, even for complex geometries such as wings and wing-body configurations [7]. Moreover, the alterations of the geometry are very intuitive and easy to implement. In the field of aerodynamic optimization, one of the first studies to use FFD as geometric parameterization was presented in Ref. [8]. Nowadays, there are different computational fluid dynamics programs that use free-form deformation [9, 10, 7]. One interesting approach is the axial-FFD deformation shown in Ref. [10]. In this method some wing parameters are changed by altering the control points of the box, as stated before, but other wing geometry features, such as span, sweep and dihedral, are changed by the axial deformation and other features, twist, chord and sectional shape, are modified by a sequence of transformation matrices [10].

For the present study, one of the objectives is to create a geometric parameterization capable of adequately modifying with many wing aerodynamic parameters. In other words, the goal of the present paper is to present a FFD capable to modify the wing geometry. Moreover, the authors aim to study the most important geometric features of an airplane wing, *i.e.*, airfoil shape, taper ratio, aspect ratio, twist, dihedral and sweep angle. Hence, we would be perform a study of the influence of all these parameters on the wing aerodynamics. Another goal of the present work is to automate the process of free-form deformation. Compared with other programs, SU2 [9] and pyGEO from the MACH-Framework [7], the user inputs will be reduced and will be ease to perform the geometric parameterization. In the proposed FFD method, the user does not have to provide the coordinates of the FFD, because the box is automatically created.

## 2. Free-Form Deformation Algorithm

The principle behind the free-form deformation algorithm is simple. It works like a jello with an object inside it. If you change the jello the object inside will suffer a similar modification [11]. The steps of a general free-form deformation algorithm are illustrated in Fig. 1. The first step is to create the volume that involves the geometry. The exterior object can be described by any trivariate tensor product, such as, Bernstein, B-Splines or NURBS [7, 6]. The second step is to embed the geometry of interest, in other words, one has to perform a mapping process between the Cartesian coordinates of the geometry and the parametric coordinates of the volume. The mapping can be described as a search problem. Therefore, this process can be done by a Newton search algorithm. The third operation is to move the volume control points in order to achieve the desired modifications. In the present work, many of the wing alterations are made through matrix-vector multiplications. References [12, 13] present different formulations of transformations that can be used. Other changes are made by simply adding values in the coordinates of the FFD control points. In the next sections we explain the details of the FFD algorithm and how we can perform the alterations in the geometry of interest.

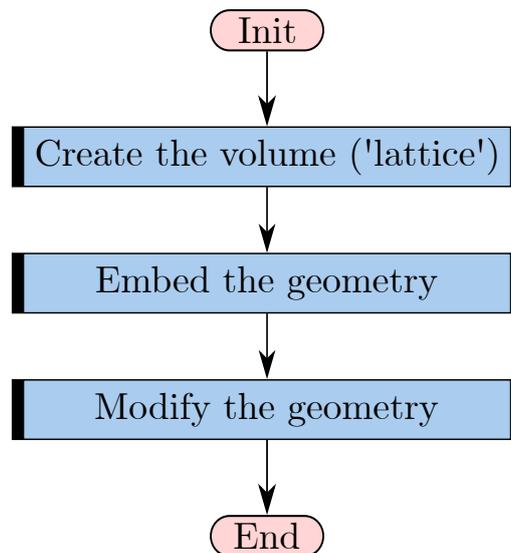


Figure 1 – Steps of a Free-Form Deformation program.

### 3. Automatic Free-Form Deformation

#### 3.1 Overall Optimization Process

In many FFD approaches, the user has to provide the coordinates of the FFD box or a file with the coordinates [9, 7]. The algorithm, that will be described herein, only requires the mesh file with boundary conditions correctly identified, the number of control points of the box, which plane the wing section leans on and the wing parameters to be altered. Besides, the algorithm can be employed to different geometries without any code modification. Therefore, in view of these aspects, we consider our approach automatic and user-friendly.

In the proposed approach, the FFD box can be created as a preprocessing step to the optimization workflow, or in the geometry parameterization, as illustrated in Fig. 2. The second approach does not significantly affect the computational time, because this process is done just once. Actually, you can perform one optimization and get the information about the FFD to perform other optimization studies for the same geometry. One can also create the FFD box as a preprocessing step before the entire aerodynamic optimization flow.

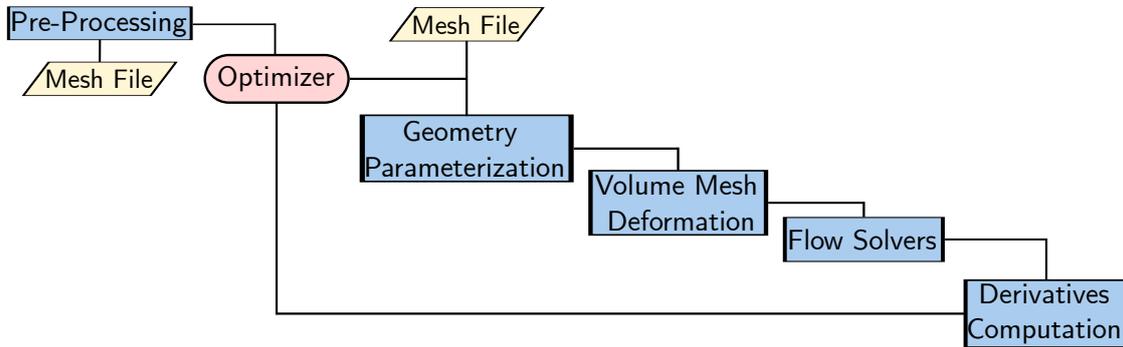


Figure 2 – Flowchart illustrating the aerodynamic optimization process.

#### 3.2 Create the Box

The first step of any FFD algorithm is to create a volume or “lattice” around the geometry that the user wants to change. There are no requirements for the FFD box, besides to encompass the entire zone that you want to change. However, there are good practices, such as, one should create the “lattice” points not so far from the geometry. There are several possibilities to create a box around of the geometry. It would be interesting if we can create an algorithm that can be used to any type of geometry. However, herein we focus in the development of a general FFD approach to describe any typical airliner wing.

The first step is to compute the control points used to describe the FFD box. To illustrate how the FFD box is created, consider a wing with streamwise sections that lie on the “XZ” plane, the user inputs  $n$ ,  $m$  and  $l$  control points in the  $x$ ,  $y$ , and  $z$ -direction. The algorithm used to compute the box control points is formed by the following steps:

1. Distribute the  $m$  points along the wing span, near to the leading and trailing edges.
2. Divide the wing into  $m$  sections.
3. Collect all the trailing edge points of all  $m$  sections created in the previous step. These points will be used as reference points to alter the geometry.
4. Compute the points between leading edge and trailing edge for each section created on step 2.

Once we have determined the box control points, we can describe the FFD volume using B-Splines by

$$V(u, v, w) = \sum_{i=0}^n \sum_{j=0}^m \sum_{k=0}^l N_{i,p}(u) N_{j,q}(v) N_{k,r}(w) P_{i,j,k}, \quad (1)$$

as stated before,  $n$ ,  $m$  and  $l$  are the control points in the  $x$ ,  $y$ , and  $z$ -directions, respectively. The  $N_{i,p}$ ,  $N_{j,q}$ ,  $N_{k,r}$  are the basis functions and  $P_{i,j,k}$  are the control points of the FFD volume. The variables  $p$ ,  $q$ , and  $r$  are the polynomial degrees of the basis functions. The B-Spline basis functions are joined by strictly increasing vectors, known as knot vectors, such that the continuity of the curves is ensured at each knot boundary [7]. For the present study, the knot vectors are defined as

$$\begin{aligned}\mathbf{U} &= \{ \underbrace{0, \dots, 0}_{p+1}, u_{p+1}, \dots, u_n, \underbrace{1, \dots, 1}_{p+1} \} \\ \mathbf{V} &= \{ \underbrace{0, \dots, 0}_{q+1}, v_{q+1}, \dots, v_m, \underbrace{1, \dots, 1}_{q+1} \} \\ \mathbf{W} &= \{ \underbrace{0, \dots, 0}_{r+1}, w_{r+1}, \dots, w_o, \underbrace{1, \dots, 1}_{r+1} \}.\end{aligned}\quad (2)$$

### 3.3 Embed the Geometry

Once the FFD volume is described by a B-Spline, the next process is to embed the object inside the volume. It is a mapping process between the parametric coordinates and the geometric coordinates of the object that will be altered. This process can be done by numerical search methods [14]. Mathematically, we need to find the optimum solution of the following problem

$$\begin{aligned}\text{Minimize } & \|V(\hat{u}, \hat{v}, \hat{w}) - \hat{V}\| \\ \text{Subject to } & 0 \leq \hat{u}, \hat{v}, \hat{w} \leq 1\end{aligned}\quad (3)$$

When the insertion process is finished, the alterations of the FFD box will be mapped to the object inside the volume.

### 3.4 Geometric Manipulation

Many alterations of interest here can be made by linear transformations such as

$$\begin{aligned}x' &= ax + by \\ y' &= cx + dy,\end{aligned}\quad (4)$$

or in matrix form

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \underbrace{\begin{bmatrix} a & b \\ c & d \end{bmatrix}}_{\mathbf{M}} \begin{bmatrix} x \\ y \end{bmatrix}, \quad (5)$$

where  $M$  is the transformation matrix. For example, to rotate a two dimensional object, we can use the following rotation matrix:

$$\mathbf{R}_\theta = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \quad (6)$$

However, transformations written in the linear form are not the best approach. If we need to combine one transformation and translation of the object, the use of the linear form can be a difficult task [12]. To address this issue, we employed affine transformations, which can be defined as a composition of a linear transformation and a translation. This approach allows us to combine different transformations and objects translations in an easy manner. However, to perform affine transformations, it is necessary to write the vectors in homogeneous notation. Where a three-dimensional vector is represented as  $\mathbf{v} = (v_x, v_y, v_z, 0)^T$  and a point become  $\mathbf{v} = (v_x, v_y, v_z, 1)^T$ . Therefore, we can translate a geometry using point coordinates and Eq. (6) becomes the following  $3 \times 3$  matrix

$$\mathbf{R}_\theta = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (7)$$

In the next subsections, we detail how the alterations in the geometry are made. Moreover the mathematical expressions used to perform the changes in the geometries are presented.

### 3.5 Twist

The wing twist is a rotation in relation to an axis. In our case, we rotate the wing around an axis that passes through the trailing edge. In an airliner wing, cannot be described simply in Cartesian coordinates and, consequently, we can not simply use a rotation matrix as in Eq. (7). The solution to alter the wing twist is to use three matrices, one to translate the FFD box to the reference axis, another to rotate around the axis and one to translate the “lattice” back to its position. The mathematical expression that alters the wing twist is given by

$$\mathbf{X} = \mathbf{T}(\mathbf{p}) \mathbf{R}_x(\theta) \mathbf{T}(-\mathbf{p}), \quad (8)$$

where the  $\mathbf{T}(-\mathbf{p})$  is the matrix used to translate the geometry to one of the Cartesian axis,  $\mathbf{T}(\mathbf{p})$  is the matrix used to translate back to the original position and  $\mathbf{R}_x(\theta)$  is the matrix used to rotate  $\theta$  degrees around the  $x$ -axis. The translation matrix is given by

$$\mathbf{T}(\mathbf{p}) = \mathbf{T}(p_x, p_y, p_z) = \begin{bmatrix} 1 & 0 & 0 & p_x \\ 0 & 1 & 0 & p_y \\ 0 & 0 & 1 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (9)$$

and the rotation matrix  $\mathbf{R}(\theta)$  is given by

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (10)$$

The geometrical interpretation of Eq. (8) is given in Fig. 3. First, we translate the wing to the origin axis, then we rotate the wing and finally we translate it back to the original position. Our computational implementation does not change the wing root.

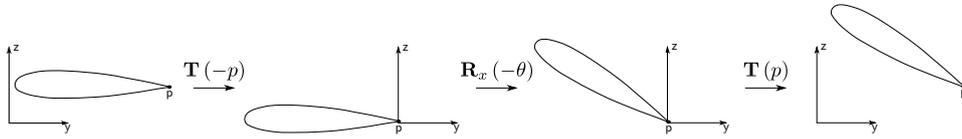


Figure 3 – Transformations used to change the wing twist.

### 3.6 Span

In order to modify the wing span, one can simply add values to the FFD box coordinates in the spanwise direction or use a matrix operation to scale the FFD box. The last option was adopted, because it is more efficient than the former procedure. The wing span is altered by the following matrix

$$S(\mathbf{r}) = S(r_x, r_y, r_z) \begin{bmatrix} r_x & 0 & 0 & 0 \\ 0 & r_y & 0 & 0 \\ 0 & 0 & r_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (11)$$

The vector  $\mathbf{r}$  is formed by the scaling ratio in the  $x$ -direction,  $y$ -direction and in the  $z$ -direction,  $r_x$ ,  $r_y$  and  $r_z$ , respectively. Figure 4 shows the wing before and after the alterations made in the FFD box. The wing root is not altered.

### 3.7 Sweep and Dihedral

The dihedral and sweep angles are altered in a simple manner. In order to change the dihedral angle we move the FFD box in the perpendicular direction of the wing planform. The sweep angle is altered by moving the FFD control points in the chordwise. Both FFD box movements are made by a simple summation in the coordinates of the FFD control points.

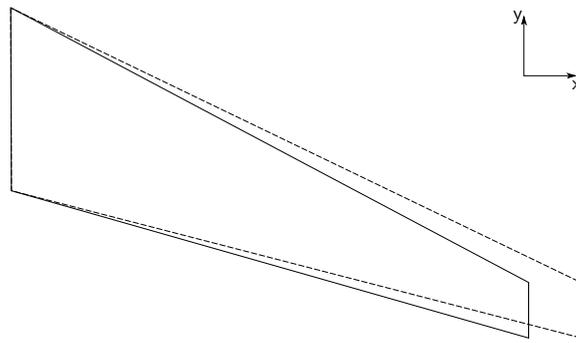


Figure 4 – Example how the wing span is changed using the scale matrix. Dashed lines represent the modified wing and the continuous line the original wing.

### 3.8 Wing Sections

The wing sections are easily altered by a FFD parameterization. Two set of control points are necessary, namely, one for the upper surface and another for the lower surface. By moving these points, the wing section is altered. For this study, we constrain the control points movements. These points can be moved only in the normal direction to wing planform plane. In order to test our implementation, we choose two geometries, one is a NACA0012 rectangular wing and the other geometry is the NASA Common Research Model wing (CRM wing).

## 4. Results

### 4.1 Airfoils

The first test of the algorithm is a two-dimensional case. For this problem, we want to demonstrate how the algorithm works. The idea is to start from a NACA0012 airfoil and to create a new airfoil by changing the FFD surface control points. As mentioned in Section 2, it is necessary to create a “lattice” that wraps the geometry. In this case we create a surface formed by 6 control points in the chordwise direction ( $x$ -direction) and 5 control points in the other direction. However, we just need to points the describe the upper and lower surfaces.

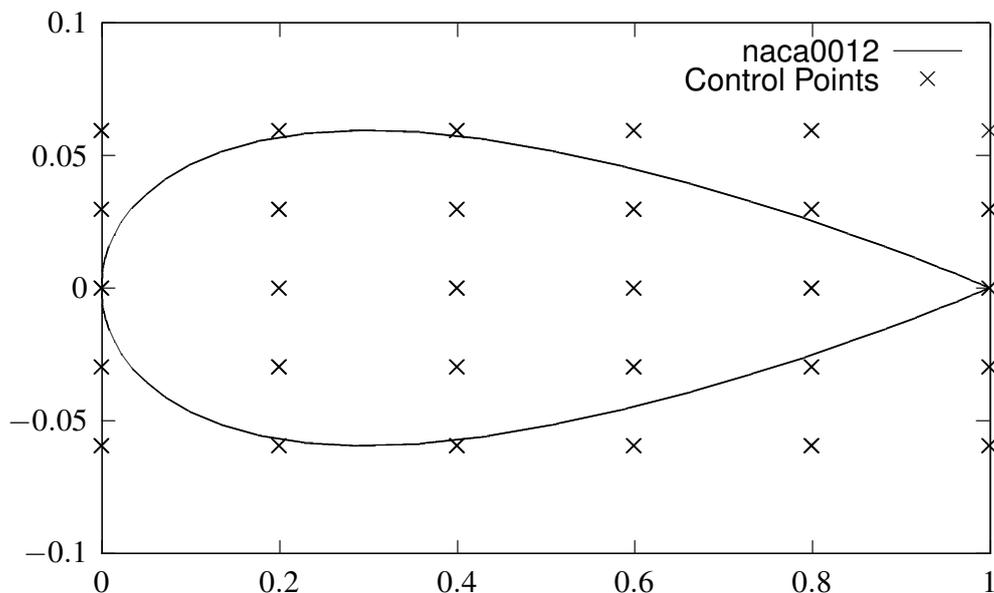


Figure 5 – Original airfoil NACA0012 and the initial position of the control points.

The original geometry and the original surface control points are shown in Fig. 5. We modified the original control points that wrap the geometry, and, then we obtain a new airfoil. For this case, we change the control points in order to get the S1223 airfoil. The final geometry and the modified surface control points are shown in Fig. 6.

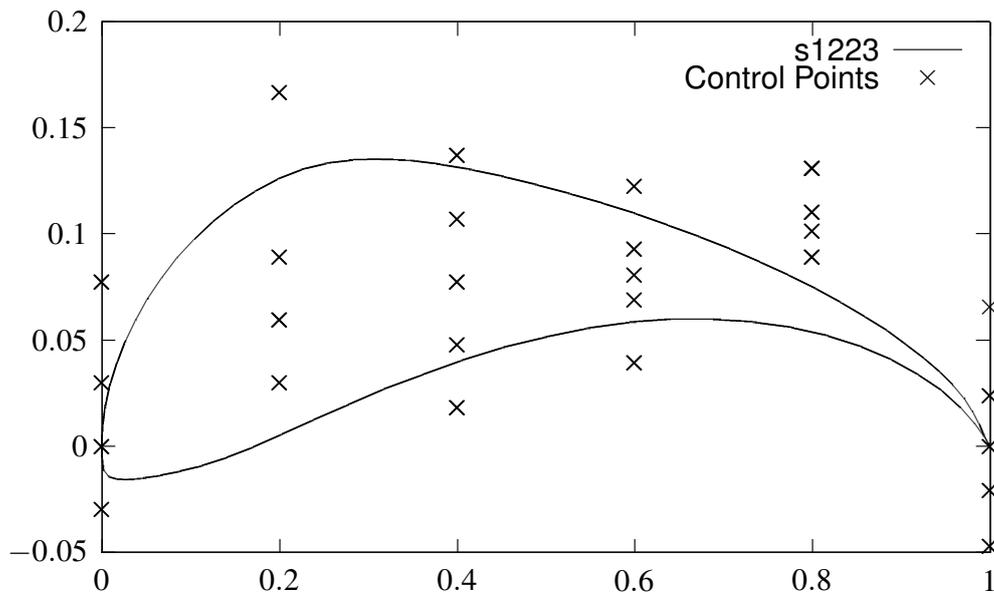


Figure 6 – Modified Airfoil and the final position of the control points.

#### 4.2 Rectangular Wing with NACA0012 Airfoil

For the three-dimensional test, we want to show all the changes that the present algorithm is able to perform. The first geometry that will be altered is a rectangular wing with a constant NACA0012 section. The wing planform lies on plane  $XZ$ , *i.e.*, the chordwise direction is the  $x$ -direction and the spanwise direction corresponds the  $z$ -direction. The box used to parameterize the geometry has 8 control points in the spanwise direction, 6 control points in the chordwise direction and 2 control points in the direction perpendicular of the wing plane. In this section, we present results of twist, span, dihedral, sweep and section alterations.

The first geometric parameter to be analyzed is the wing twist angle. All the results, regarding the wing twist, considered the rotation axis at the wing trailing edge points. Negative angles of twist refer to clockwise rotation (“wash in”). The FFD box has 8 sections, one for each control point in the spanwise direction. In order to create a wash in or wash out in the wing, we rotate these FFD sections.

Figure 7 shows a change in the twist angle in a linear manner. We added positive angles in each FFD section. The angles that create the twist are 0.00, 6.43, 12.86, 19.29, 25.71, 32.14, 38.57 and 45.00 degrees. By default, the program does not change the root incidence angle. It is important to state the reason to change the wing twist parameter. There is another way to change the wing twist. It is by choosing a angle for each section of the FFD. In Fig. 8, the wing twist angles are different for each section. Usually, the wing sections near at the wing root will be twisted down by a negative value in order to control the wing stall. At the wing tip, an increase in the profile wing angle is the usual process in order to compensate the loss on the wing lift due to the reduction in the angle of attack at the wing root. Therefore, the program input can twist the wing sections at a positive value and by a positive value. As said before, the FFD has 8 sections alongside the wing span. The twist angles are 0, 10, 10, 12, 15, 15, 16, -22 degrees.

Figure 9 shows the modification in the wing span. The original wing is the black one and the modified one is the red one. This is a simple modification by using a scaling transformation matrix [13]. Figure 10 shows the change in the wing dihedral angle. The wing in Fig. 10 lies on plane  $XZ$ . Hence, in order to change the wing dihedral, it is necessary to move the FFD control points in the  $y$ -direction. In other to do make alterations in the sweep angle we need to move the FFD box in the  $x$ -direction. Figure 11 shows the volume before and after the alterations.

Figure 12 shows the alterations in all the wing sections. In this case, all the FFD sections are changed in the same manner. The program can also perform different alterations in each section. Figure 13 shows a remodeling in each wing section of the FFD. The FFD control point coordinates were moved

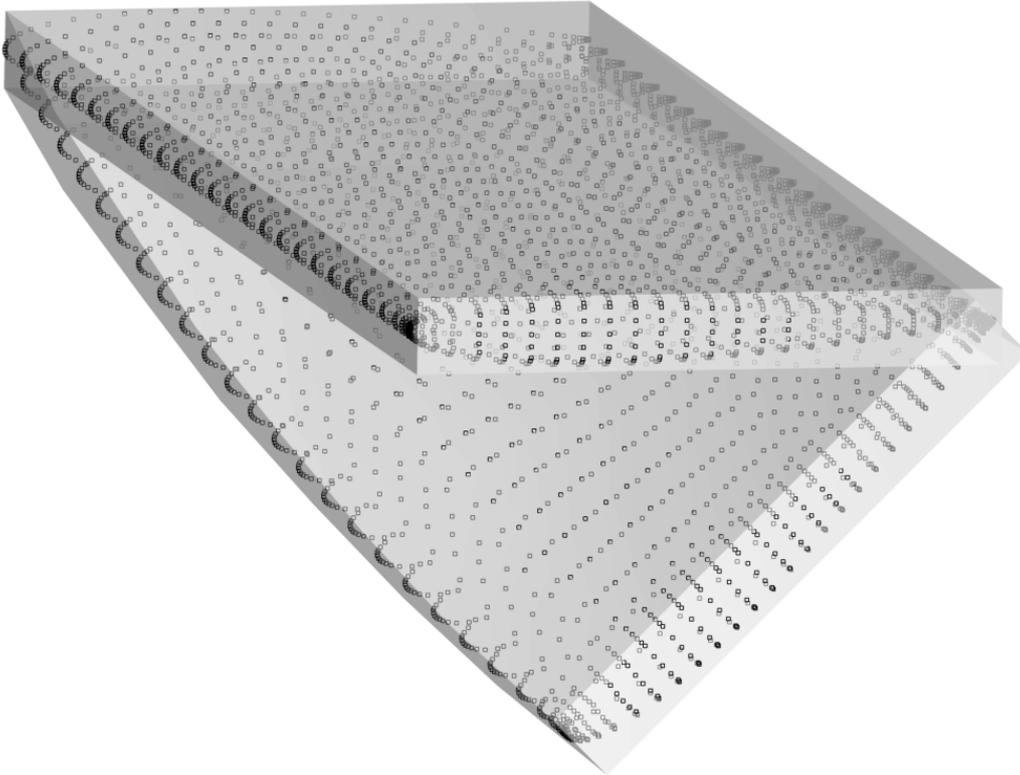


Figure 7 – Wing with a linear twist.

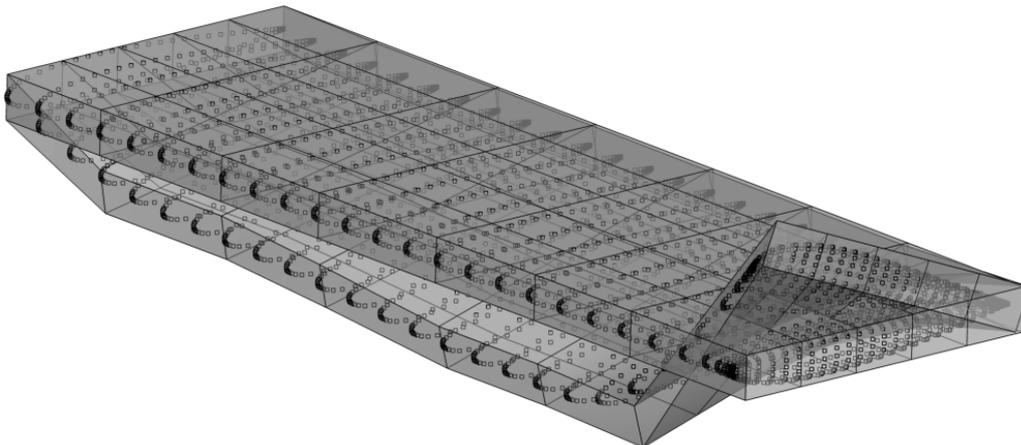


Figure 8 – Wing with different twist angles.

in a range between  $-0.015$  and  $0.015$  from their original position and this movement is restrained in the  $y$ -direction.

#### 4.3 NASA Common Research Model Wing

The NASA Common Research Model (NASA CRM) is a wing-body geometry used in many Computational Fluid Dynamics applications. In the present study, through the NASA CRM wing, the authors evaluate the FFD implementation as a tool to modify an airliner wing. The wing geometry of the NASA CRM model is shown in Fig. 14

The first test case that is shown here concerns an alteration in the geometry by means of a modification in the twist angle. The authors have chosen different angles of twist in order to demonstrate how the wing can be modified in order to evaluate the process of alteration. Figure 15 presents the results when the CRM wing tip was twisted by a angle of 15 degrees. This is not a usual wing twist parameter value, but the authors state that the program can twist the wing geometry at any value,

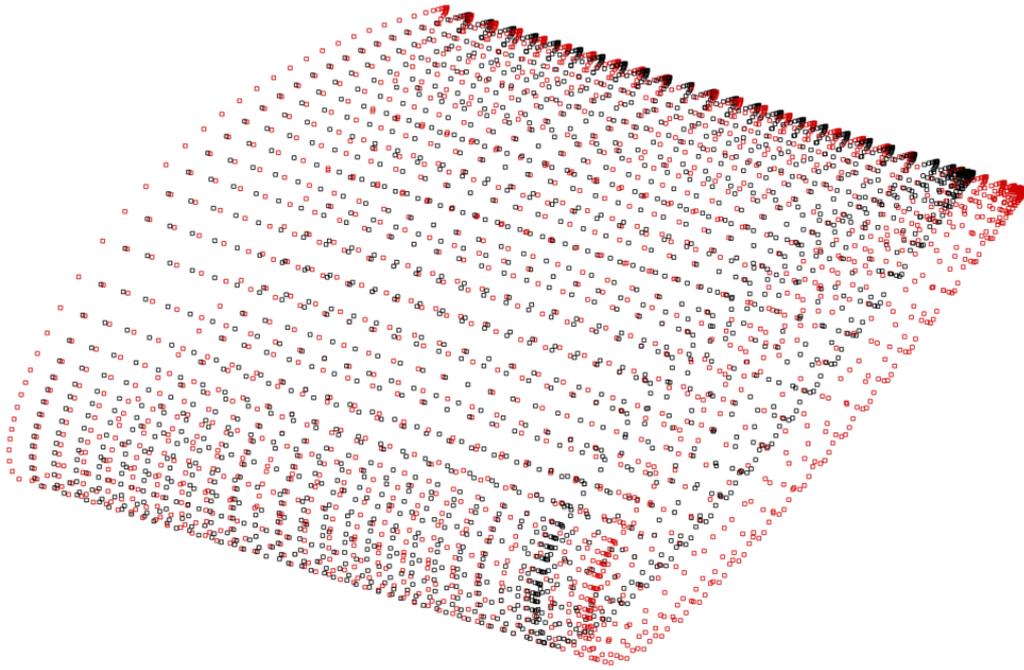


Figure 9 – Change in the wing span. Original wing in black and modified wing in red.

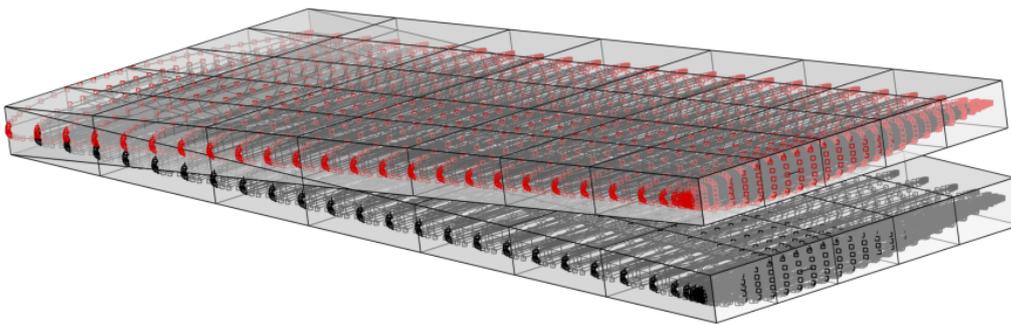


Figure 10 – Changes in the wing dihedral angle.



Figure 11 – Example of change in the sweep angle.

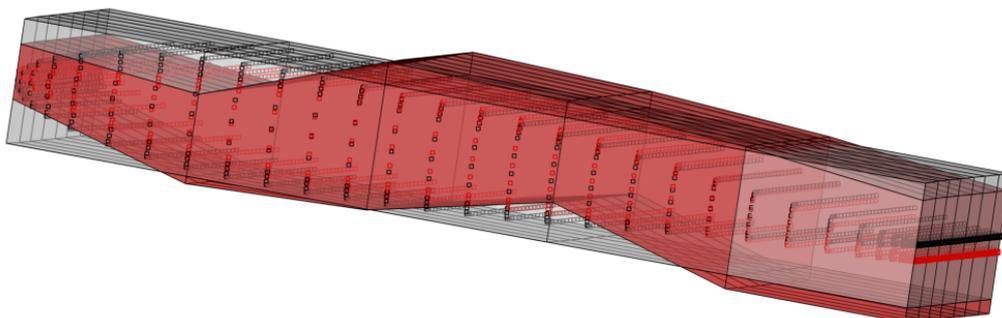


Figure 12 – Alterations in the wing section and in the FFD.

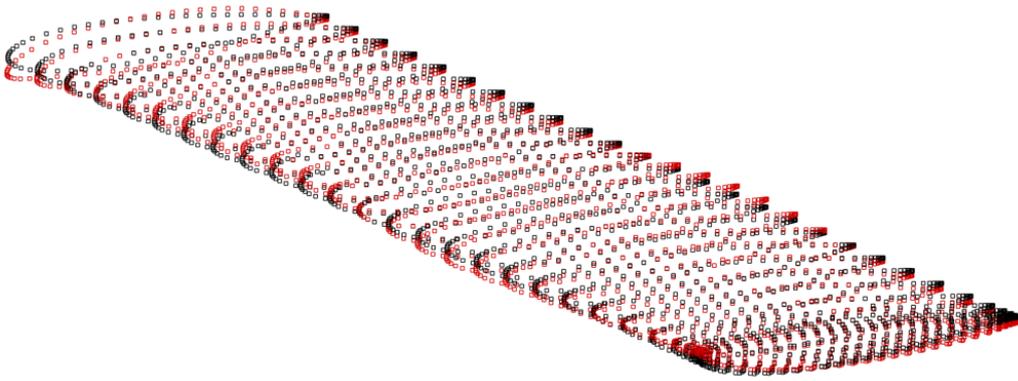


Figure 13 – Alterations in the section made in a random manner.

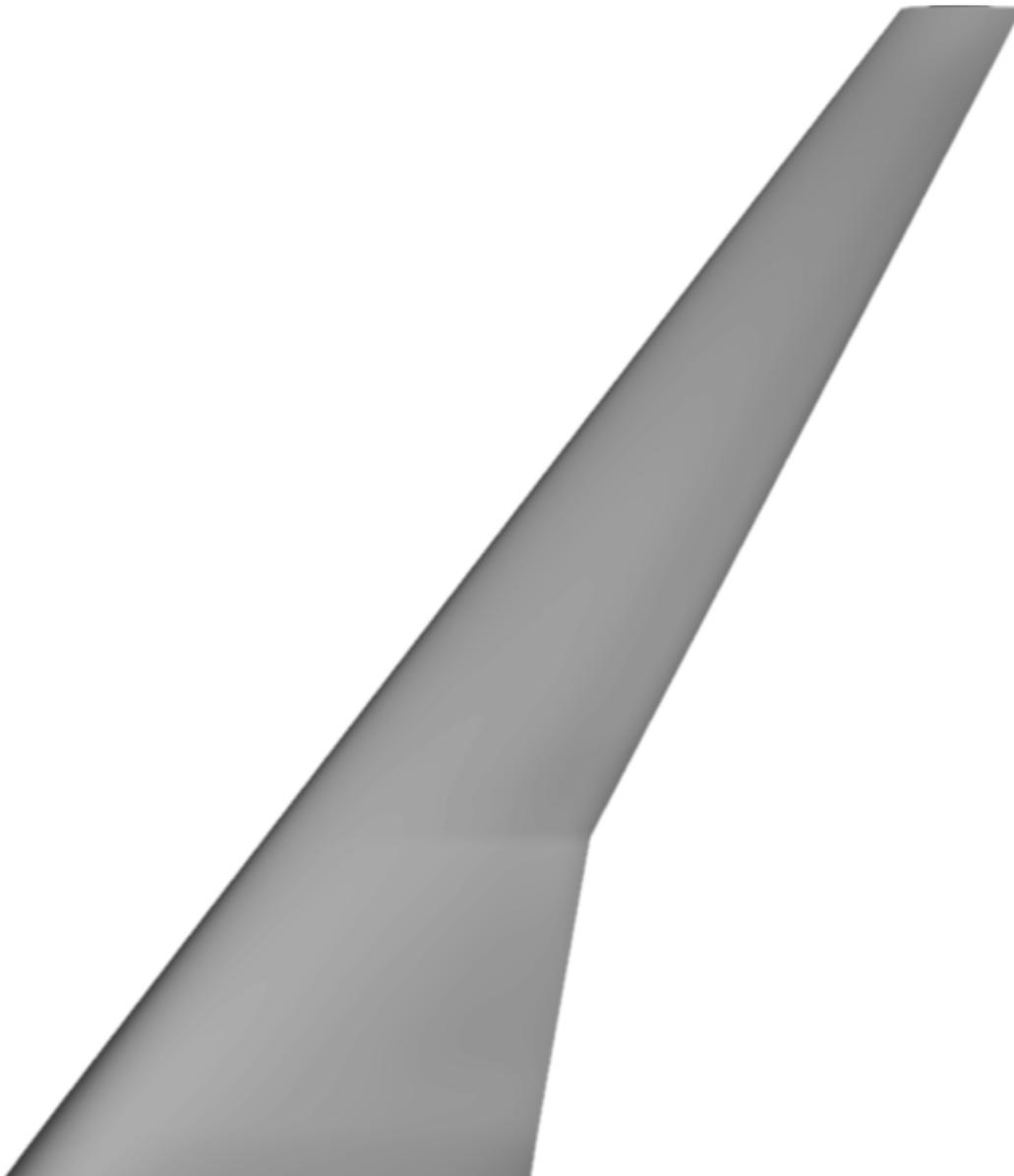


Figure 14 – NASA CRM wing used to evaluate the FFD implementation.

besides we choose value which the change in the twist angle is noticeable. Another test concerns alterations on the wing span. In the present case, the wing span is increase in 10% with regard to its original length. The wings, before and after the span alterations, are shown in Fig. 16. The alteration in the wing span as stated in Section 3.6 is made by a scaling process, but only



Figure 15 – Twist applied to the CRM wing. The wing in black indicates the original geometry and the red one is the modified wing.

in the spanwise direction. By scaling only the wing span, the wing aspect ratio is easily computed in relation to its original value, once, the mean wing chord aerodynamic is constant. This aspect is important because we can measure impact of the wing aspect ratio in the geometry aerodynamics with great accuracy. The last wing parameter modified in the present study is the wing dihedral. As stated before, the modification in the dihedral angle are made by simply adding or subtracting a value in the FFD coordinates. The dihedral angle can be positive and negative. Usually, a positive dihedral angle improves the lateral stability of the an airplane and a negative one is not common, but in certain cases they are employed to get a better airplane maneuverability. The present FFD implementation can address both situations. The alterations in this wing parameter are shown in Fig. 17.

## 5. Concluding Remarks

In many FFD programs it is necessary that the user creates the box manually, *i.e.*, provides the FFD box coordinates as program input. Herein, a simple FFD algorithm is shown that can parameterize and alter the geometry without FFD coordinates. The program only requires the number of control points and which design variables the user want to alter. Even though, creating could cause some problems influences the process of optimization or alteration of wing geometry, leading to undesirable results. Therefore, this process can facilitate the engineer or researcher job, once the user does not need to control many parameters. Moreover, if one needs to study different geometries this process will save time, once there is no need to create one FFD box for each geometry. This is a task that was wrap inside the FFD box.

The results herein demonstrate the capabilities of the present FFD geometry parameterization. Regarding an airliner wing, the program can alter and control the main features that influences the aerodynamics wing. Therefore, the FFD algorithm can deal with complex problem optimizations, where many wing parameters are considered.

## 6. Acknowledgments

The authors acknowledge the support for the present research provided by Conselho Nacional de Desenvolvimento Científico e Tecnológico, CNPq, under the Research Grant No. 309985/2013-7. The work is also supported by Fundação de Amparo à Pesquisa do Estado de São Paulo, FAPESP, under Research Grant No. 2013/07375-0. The support provided by Fundação Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, CAPES, through a graduate scholarship for the first author is also greatly appreciated. The work is further supported by the computational resources of the Center for Mathematical Sciences Applied to Industry (CeMEAI), also funded by FAPESP under the Research Grant No. 2013/07375-0.

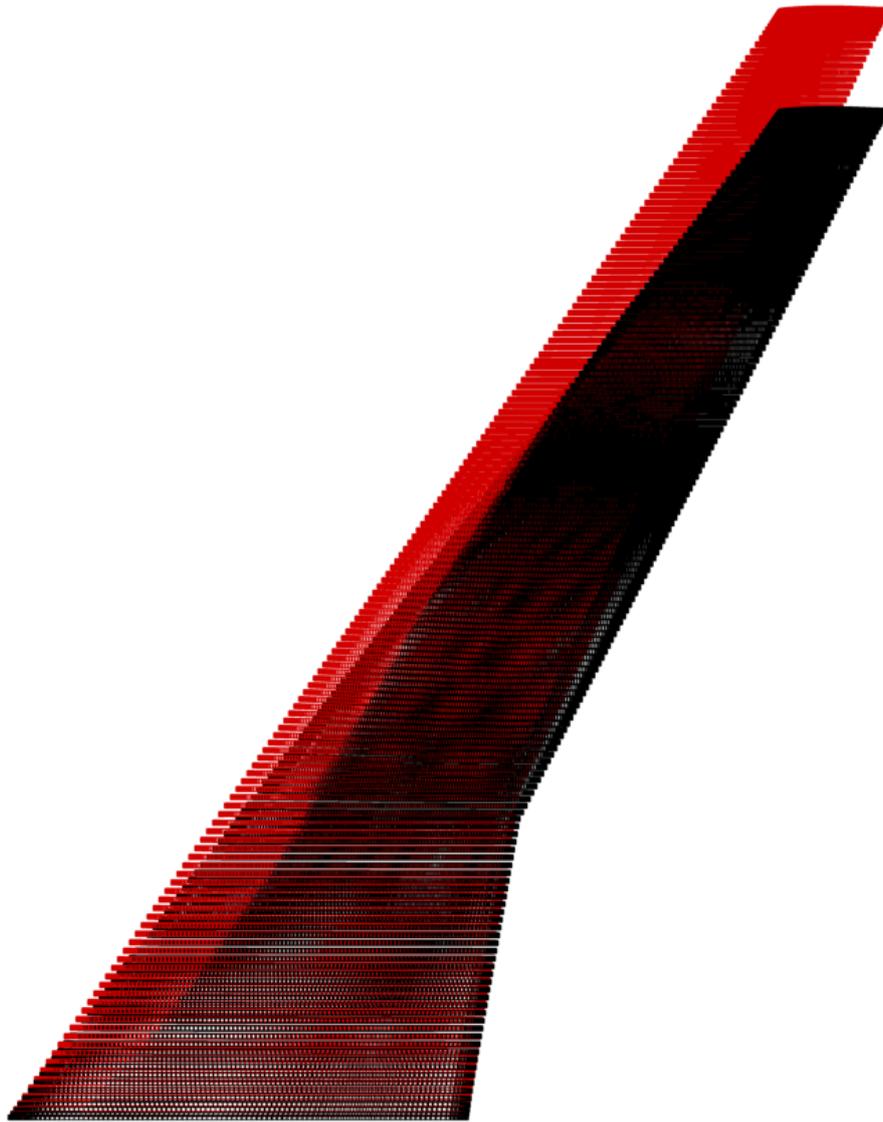


Figure 16 – Modification in the wing span. The wing in black indicates the original geometry and the red one is the modified wing.

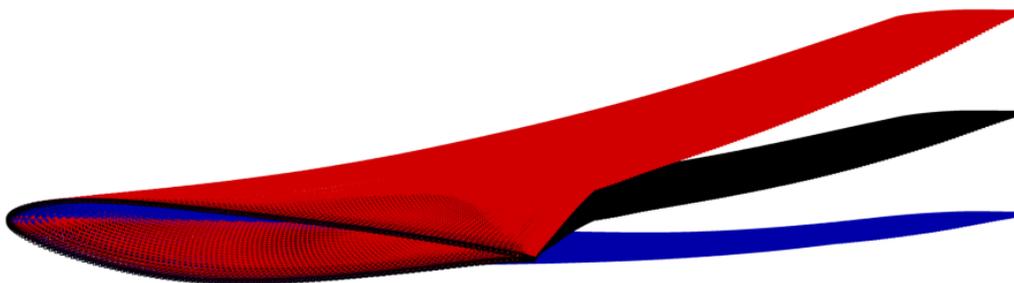


Figure 17 – Dihedral of the CRM wing. The wing in black indicates the original geometry and the red one is the modified wing with positive angle and the blue one is with the negative one.

## 7. Contact Author Email Address

Marco Aurélio Leonel Matunaga: marco.matunaga@gmail.com

João Luiz F. Azevedo: joaoluiz.azevedo@gmail.com

## 8. Copyright Statement

The authors confirm that they, and/or their company or organization, hold copyright on all of the original material

included in this paper. The authors also confirm that they have obtained permission, from the copyright holder of any third party material included in this paper, to publish it as part of their paper. The authors confirm that they give permission, or have obtained permission from the copyright holder of this paper, for the publication and distribution of this paper as part of the ICAS proceedings or as individual off-prints from the proceedings.

## References

- [1] Z. Lyu, G. K. W. Kenway, and J. R. R. A. Martins, "Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark," *AIAA Journal*, vol. 53, pp. 968–985, Apr. 2015.
- [2] K. Telidetzki, L. Osusky, and D. W. Zingg, "Application of Jetstream to a Suite of Aerodynamic Shape Optimization Problems," in *52nd Aerospace Sciences Meeting*, American Institute of Aeronautics and Astronautics, Jan. 2014.
- [3] M. Meheut, A. Dumont, G. Carrier, and J. E. Peter, "Gradient-Based Optimization of CRM Wing-Alone and Wing-Body-Tail Configurations by RANS Adjoint Technique," in *54th AIAA Aerospace Sciences Meeting*, (San Diego, CA), American Institute of Aeronautics and Astronautics, Jan. 2016.
- [4] G. K. W. Kenway and J. R. R. A. Martins, "Multipoint High-Fidelity Aerostructural Optimization of a Transport Aircraft Configuration," *Journal of Aircraft*, vol. 51, pp. 144–160, Jan. 2014.
- [5] J. A. Samareh, "Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization," *AIAA Journal*, vol. 39, no. 5, pp. 877–884, 2001.
- [6] T. W. Sederberg and S. R. Parry, "Free-Form Deformation of Solid Geometric Models," in *Proceedings of the 13th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '86, (New York, NY, USA), Aug. 1986.
- [7] G. K. Kenway, G. J. Kennedy, and J. R. Martins, "A CAD-Free Approach to High-Fidelity Aerostructural Optimization," in *13th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference 2010*, (Fort Worth, TX), Sep. 2010.
- [8] J. Samareh, "Multidisciplinary Aerodynamic-Structural Shape Optimization using Deformation (MAS-SOUD)," in *8th Symposium on Multidisciplinary Analysis and Optimization*, (Reston, Virginia), Sept. 2000.
- [9] T. D. Economou, F. Palacios, S. R. Copeland, T. W. Lukaczyk, and J. J. Alonso, "SU2: An Open-Source Suite for Multiphysics Simulation and Design," *AIAA Journal*, vol. 54, no. 3, pp. 828–846, 2016.
- [10] H. Gagnon and D. W. Zingg, "Two-Level Free-Form and Axial Deformation for Exploratory Aerodynamic Shape Optimization," *AIAA Journal*, vol. 53, no. 7, pp. 2015–2026, 2015.
- [11] Z. Lyu, *High-Fidelity Aerodynamic Design Optimization of Aircraft Configurations*. PhD thesis, The University of Michigan, 2014.
- [12] T. Akenine-Möller, E. Haines, and N. Hoffman, *Real-Time Rendering*. CRC Press, 2008.
- [13] P. Shirley and S. Marschner, *Fundamentals of Computer Graphics*. USA: A. K. Peters, Ltd., 2009.
- [14] T. P. Yeh and J. M. Vance, "Applying Virtual Reality Techniques to Sensitivity-Based Structural Shape Design," *Journal of Mechanical Design, Transactions of the ASME*, vol. 120, no. 4, pp. 612–619, 1998.