ADAPTIVE FREE-FORM DEFORMATION PARAMETERIZATION FOR GEOMETRY AND MESH MORPHING IN AERONAUTICAL APPLICATIONS

Jiří Hradil

Institute of aerospace engineering

Faculty of mechanical engineering

Brno university of technology

Technická 2896/2
616 69 Brno
Czech republic

hradil@fme.vutbr.cz

Abstract. This article describes adaptive parameterization procedure for geometry and mesh deformations. The Free-Form Deformation (FFD) parameterization based on NURBS is enhanced to enable adaptive feature. The idea is to automatically refine the parameterization grid in appropriate regions of the object geometry. That result in better precision of the object description. Both surface and internal unstructured mesh are considered. The adaptive parameterization is compared with regular FFD parameterization on reverse engineering of the airfoil geometry.

Keywords: FFD, adaptive parameterization, mesh deformation, CFD, unstructured meshes.

1 Introduction

It is hard to imagine aerodynamic design of modern aircrafts without the use of CFD simulation methods. Their benefits are known for quite long time and they are widely used to supplement or even replace wind tunnel testing in aircraft design.

As the progress of computer hardware power increases rapidly it enables more and more detailed simulations to be performed in practical use. Powerful computer availability is also one of the reasons for growing popularity of the aerodynamic shape optimization techniques which results in significant time savings in design cycle.

One of the key steps in optimization process is parameterization of the geometry. Parameterization defines possible object shapes and shape changes by a set of parameters which are used as design variables during the optimization process. The number of optimization parameters has major influence on the computational time cost. It is crucial to have suitable parameterization for each specific task. Wrong parameterization can slow down the optimization process or even prevent the optimization algorithm from finding the optimum solution, since the parameterization could not be able to generate optimum shape. FFD parameterization method was chosen to deal with all of these problems.

To solve this adaptive parameterization method is proposed. Such parameterization automatically adapts the original parameterization to the specific optimization problem and thus improves the solution both in terms of accuracy and rate of convergence.

Mesh morphing is another feature that can be managed by using FFD parameterization, which is capable of morphing both the geometry of the object and the computational mesh at once. Thanks to this procedure it is not necessary to create new mesh every time the geometry is changed and therefore significantly speed up the optimization process itself. The idea is to generate the computational mesh only once at the beginning.

2 Free-Form Deformation parameterization

The FFD parameterization is rather complicated but also very powerful method. It was developed for computer graphics for morphing images (e.g. Boubekeur et al. [1]) and deforming models, first published by Sederberg and Parry [2]. It is usually linked with polynomial and spline parameterization techniques as in [3], [4]. It is ideal for parameterization of objects of high geometry complexity.

The FFD algorithm treats the model (mesh) as rubber that can be stretched, compressed, twisted, tapered or bent and yet preserve its topology. FFD makes it possible to deform only part of the domain of interest while the rest of the geometry remains intact and the transition between deformed and unreformed parts is smooth. It belongs between the parameterization methods that deform existing shapes.

2.1 FFD parameterization procedure:

The FFD parameterization procedure has four main steps:

a.) Construction of the Parametric Lattice

b.) Embedding the object within the Lattice

c.) Deforming the Parametric Volume

d.) Evaluating the Effects of the Deformation

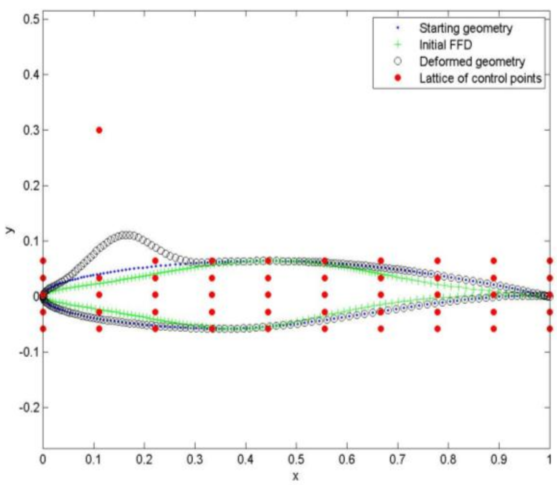

Figure 1: Demonstration of deformation of airfoil geometry

a.) Construction of the Parametric Lattice

As described in Amoiralis and Nikolos [3] a 1D, 2D or 3D lattice is constructed around/in the object that should be deformed. This defines parametric coordinate system. Nodes of the lattice are used as control points to define NURBS volume that contains the object to be deformed. NURBS polynomials are defined in each lattice direction u, v, w. Constraints of polynomial degrees:

$$1 \leq p \leq a \quad 1 \leq m \leq b \; 1 \leq n \leq c$$

Where p, m, n define degree of the basic polynomial function in corresponding direction, a+1, b+1, c+1 are numbers of the control points in each direction.

NURBS uses knot vectors, where:

$$\mathbf{U} = (u_0, u_1, ..., u_q), q = a + p + 1$$
$$\mathbf{V} = (u_0, u_1, ..., u_r), r = b + m + 1$$
$$\mathbf{W} = (u_0, u_1, ..., u_s), s = c + n + 1$$

Values of **U** knot vector are calculated as

$$u_i = \begin{pmatrix} 0 & 0 \le i \le p \\ i - p & p < i \le (q - p - 1) \\ q - 2.p & (q - p - 1) < i \le q \end{pmatrix}$$

and unified with range of x coordinates of parametric u coordinate. This knot vector has p multiple (same value) members at the beginning and at the end. **V** and **W** vectors are calculated similarly.

NURBS basic functions N are defined for every direction of the parametric volume. N for u direction is calculated with standard recursive formula.

$$N_{i,p}(u) = \frac{u - u_i}{u_{i+p} - u_i} N_{i,p-1}(u) + \frac{u_{1+p+1} - u}{u_{i+p+1} - u_{i+1}} N_{i+1,p-1}(u)$$
$$N_{i,0}(u) = \begin{pmatrix} 1 & u_i \le u < u_{i+1} \\ 0 & otherwise \end{pmatrix}$$

Where u is vector of coordinates of geometry (points) that is to be deformed (calculated one by one in loop), i is position in knot vector and u(i..) are coordinates in knot vector.

The Cartesian coordinates of a geometry points within the 3D volume with parametric coordinates u,v,w are calculated using

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

Where R is Cartesian coordinate vector of a point [*u*,*v*,*w*] in a parametric space, P_{ijk} is vector of control points coordinates (x_i, y_i, z_i,) in *u* direction and G_{ijk} is matrix of weights. For 1D:

$$R(u) = \frac{\sum_{i=0}^{a} G_i.P_i.N_{i,p}}{\sum_{i=0}^{a} G_i.N_{i,p}}$$

b.) Embedding the object within the Lattice

The object to be deformed is described by set of Cartesian coordinates (x,y,z), to use FFD a corresponding parametric coordinates (u,v,w) have to be found.

The embedding step means that It is needed to find such parametric coordinates that the product R(u,v,w) will be equal or in specified tolerance to the Cartesian object coordinates. So an inverse problem needs to be solved in this step.

Amoiralis and Nikolos [3] used quadtree 2D and octree 3D search algorithms. Other author used for that the golden section method or Newton's method or other.

Several approaches to the embedding problem were tested by the author. So called "Secant method" turned out as the fastest of them.

c.) Deforming the Parametric Volume

In this step the lattice of control points is changed or/and the weights are modified.

d.) Evaluating the Effects of the Deformation

The deformed coordinates R are calculated.

Huge advantage of this method is that it enables deformations of whole object within the NURBS volume. For example wing surface can be deformed with its inner structure (spars, beams, fuel tank) and also with whole computational domain for flow simulation around it. All the deformations are smooth and the topology remains the same. An obvious opportunity for this method is in aeroelastic optimizations (FEM-CFD).

Most computational effort during FFD parameterization is in calculating the NURBS volume, which does not change in the optimization process and needs to be calculated only once. As a result, in the next optimization iteration the deformation of complete wing mesh could take only seconds, maybe minutes.

3 Adaptive FFD parameterization

For aerodynamic optimization of complex geometry (e.g. complete aircraft) many parameters are needed to describe its shape. This of course means that the computational costs of such a optimization task are high.

The smart solution is to use as few parameters as possible. The idea is to have the parameters in appropriate regions of the geometry. In other words the parameterization that is adapted to the object's shape.

Even more advanced approach is described in this paper. Adaption of the NURBS based FFD parameterization during the optimization cycle. The optimization is started with very small number of optimization parameters (lattice control points) and thus progresses fast without the risk of falling into some local minima. In principle the lattice of control points is being refined in selected areas after defined number of the optimization iterations until the convergence is reached and the optimum found.

This approach is tested on simple reverse engineering of the airfoil geometry and compared with regular FFD parameterization. The goal is to get RAE 2822 airfoil geometry from the NACA 0012 airfoil geometry while using conjugate direction optimization method to find the optimal lattice control points vertical positions. Adaption criteria that decide where to add control points is based on certain level of cost function gradient, that is evaluated for each interval between two neighboring lattice columns.

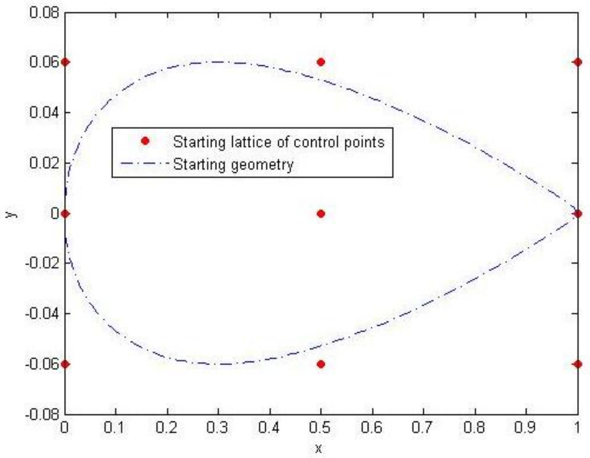

Figure 2: Start of the adaptive optimization case

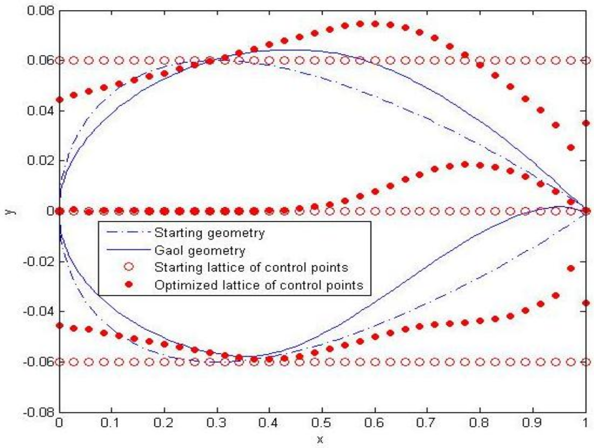

Figure 2: Regular optimization case

Figure 2 depicts starting layout of the “Adaptive” case. Layout as well as result of the “Regular” optimization case can be seen on Figure 3.

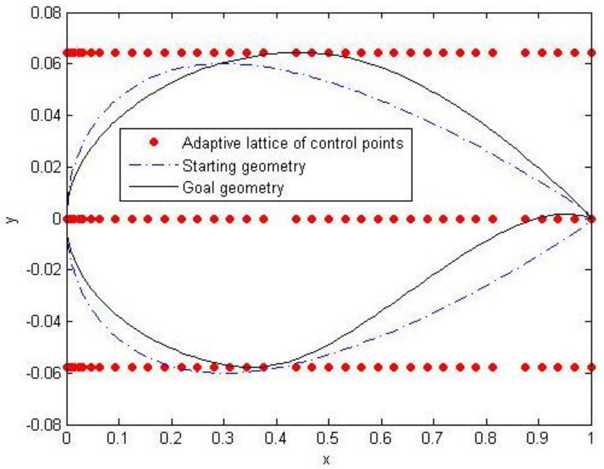

Figure 4: Adaptive optimization case

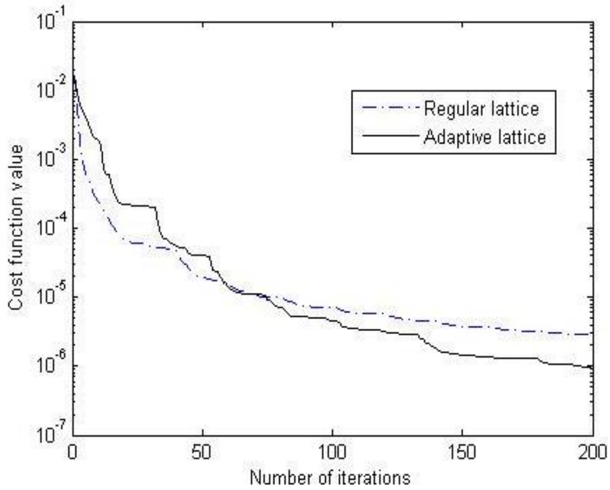

Figure 5: Convergence evolution, regular vs. adaptive

Figure 4 shows adapted (deformed) lattice of the “Adaptive” case. The adaptive procedure resulted in 35 lattice control points in x direction. The “Regular” case was matched to that number. As can be seen on the Figure 5 the adaptive approach managed to find better optimization result. The “Adaptive” case optimization met the same value of cost function as the “Regular” one (~70 iterations) while using fewer parameters. After each refinement a sharp fall of the cost function evolution is visible on the Adaptive lattice line.

4 Mesh morphing

Mesh morphing is modern way of adjusting the computational mesh on changes in object shape during the optimization, so there will be no need to generate the CFD mesh again after every iteration anymore.

Based on that, another improvement is proposed. Simplification of the optimization process will be achieved by manipulating the whole CFD mesh (or appropriate part of it) with the FFD parameterization. In other words the object shape (subject to the optimization cost function) will be deformed together with the volume mesh that surrounds it. Thanks to that the use of another mesh morphing program can be skipped.

To demonstrate the capability of the FFD parameterization to handle mesh morphing the unstructured airfoil mesh case was selected. Lattice of control points was generated in the vicinity of the airfoil. Deformation of the airfoil (surface mesh) together with the CFD mesh was achieved by modifying the position of some of the control points (Figure 8). Visual inspection of this morphing does not reveal any critical issue with the mesh cells quality. Mathematical criteria for calculating the mesh quality, such as cell skewness and aspect ratio, is under development.

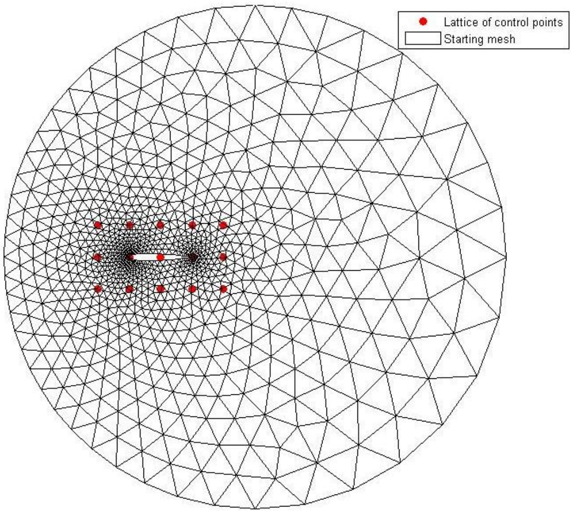

Figure 6: Unstructured mesh domain

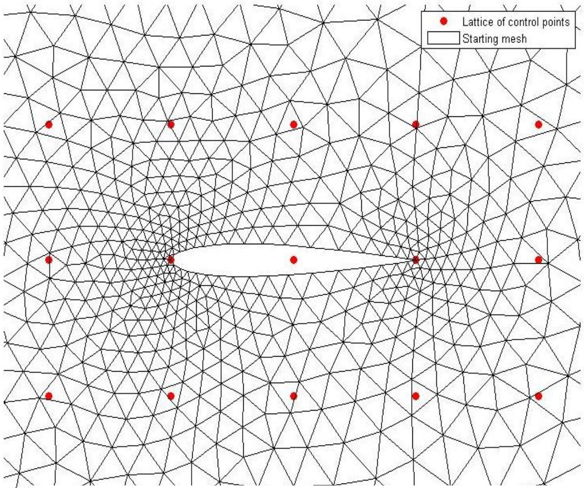

Figure 7: Detail of the mesh around NACA 0012 airfoil with initial lattice of control points

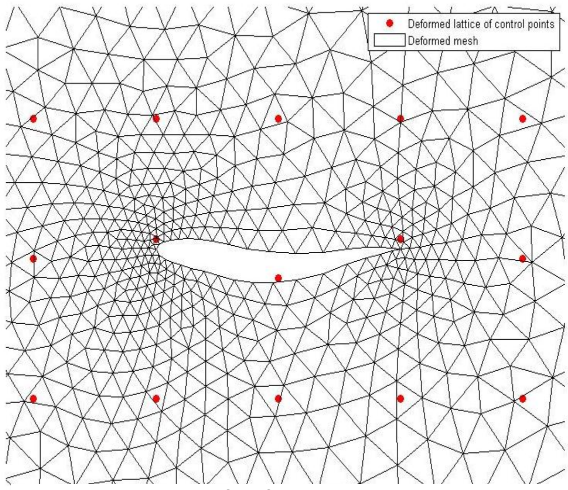

Figure 8: Detail of deformed mesh caused by moving some of the lattice control points

5 Conclusion

This document introduces the Free-Form Deformation parameterization and its Adaptive enhancement developed for use in aerodynamic shape optimizations in aeronautical applications.

At the beginning the description FFD parameterization is presented, adaptive FFD procedure is defined, successfully tested on airfoil reverse optimization case and compared to regular FFD parameterization. The final part of the article describes use of the FFD parameterization for mesh morphing.

Development of presented methods is focused on practical aeronautical optimization tasks and aims to incorporate use of adaptive FFD parameterization in adjoint optimization cases.

Acknowledgment

The research leading to these results has received support from the FP7 REGPOT project CEDESA (Centre of Excellence for the Design of Efficient and Safe Aircraft)

References

[1 Boubekeur T., Sorkine O. and Schlick Ch.: SIMOD: making Freeform Deformation Size-Intensive. Eurographics Symposium on Point-Based Graphics. 2007.

[2] Sederberg T. W., Parry S. R.: Free-Form Deformation of Solid Geometric Models. ACM 0-89791-196-2/86/008/0151. 1986.

[3] Amoiralis E. I. and Nikolos I. K.: Freeform Deformation Versus B-Spline Representation in Inverse Airfoil Design. Journal of Computing and Information Science in Engineering. 2008.

[4] Samareh J. A.: Aerodynamic Shape Optimization Based on Free-Form Deformation. AIAA paper 2004-4630. 2004.