V European Conference on Computational Fluid Dynamics
ECCOMAS CFD 2010
J. C. F. Pereira and A. Sequeira (Eds)
Lisbon, Portugal,14-17 June 2010

AIRCRAFT CONTROL SURFACE DEFLECTION USING RBF-BASED MESH DEFORMATION

Andreas K. Michler

German Aerospace Center,
Institute of Aerodynamics and Flow Technology
Lilienthalplatz 7, 38108 Braunschweig, Germany
e-mail: andreas.michler@dlr.de

Key words: Mesh Deformation, Radial Basis Functions

Abstract. *In this paper, mesh deformation based on radial basis functions is employed for deflecting aircraft control surfaces. A technique is presented which locally restricts the mesh deformation to the vicinity of the deflecting component and thereby saves computational time. In addition to the data sites consisting of the mesh points on the control surface with known displacements, further data sites are specified, and the centers for the interpolants are selected by means of an adaptive process. Choosing the centers for the displacements in each direction separately, termed sequential uni-variate adaptation, has proven to be superior to the approach with the same centers for all directions. It also permits to impose additional data sites for each direction independently, according to the respective boundary conditions. These techniques are used for the deflection of aileron and horizontal tail of the SDM generic fighter aircraft. By making use of overlapping mesh blocks and allowing in-plane movements on their faces, large control surface deflections are enabled, even if small spanwise gaps between control surface and parent component are modelled.*

1 INTRODUCTION

Aircraft control surface deflections change the surrounding flow and consequently modify aerodynamic lift, drag and moment coefficient. This is exploited for trimming and maneuvering purposes: for trimming, the deflections are aimed at balancing the loads acting on the aircraft in order to achieve a specific flight condition; for maneuvers, the changes in aerodynamic coefficients are used to induce a motion of the aircraft. With the advent of ever more powerful computers, flow around complex aircraft in trimmed states and during maneuvers can be accurately simulated.[1,19,23,24] To this end, multiple disciplines need to be taken into account. Computational Fluid Dynamics (CFD) may be coupled with flight mechanics and, in case the elastic deformation of the structure is to be respected, Computational Structural Mechanics (CSM). One major requirement is the adaptation of the CFD mesh to the changes in the geometry. This may be achieved

by different methods such as mesh regeneration, the Chimera technique for overlapping mesh blocks and various mesh deformation techniques.

While avoiding mesh distortion with ill-shaped elements in case of large deflections, mesh regeneration often lacks robustness and suffers from being computationally expensive. Furthermore, element connectivities are not preserved, which may be important for error assessment and for the use of adjoint calculations (cf.[14]). In contrast, the Chimera technique offers an efficient alternative for rigid-body movements of components, which are embedded and moved along in mesh blocks overlapping with the background grid.[13,17] Flow variables are interpolated at each iteration of the solver between adjacent points of different mesh blocks. However, besides interpolation routines, this requires an efficient algorithm for point search, and the boundaries of the different mesh blocks, i.e. auxilliary geometrical features, have to be defined and the respective meshes generated. This may prove difficult in case the configuration's CAD description is no longer available. Mesh deformation techniques, however, can be applied to the existing mesh, both for rigid-body motion and for torsion and bending of a component. They have been used extensively in CFD, e.g. to account for the deformation of elastic components,[2,3,15,18] to realize the relative motion of components for unsteady simulations[4] or for CFD-flight mechanics coupled simulations,[3,19,24] and to modify the design for optimization purposes.[5] Mesh deformation preserves element connectivities and, depending on the technique, can be easily implemented and efficient. It is usually only applied once during preprocessing.

Because of these advantages, a mesh deformation technique is employed in the present work. It relies on radial basis function (RBF) interpolation and can be classified as a point-by-point scheme (cf.[6,18]). As opposed to techniques based on element connectivities (modelling springs or solid body elasticity, see e.g.[16,25]), these techniques exclusively use point coordinates and may therefore be applied to any mesh type, e.g. structured, unstructured, hybrid and multi-block meshes. Based on the displacement of the deflecting surface mesh points, an interpolant is calculated and then evaluated on the mesh points of the entire volume mesh.

RBFs have been applied to various areas of science and engineering, e.g. for the solution of partial differential equations,[7,20] for surface reconstruction,[8] and for scattered data modelling and approximation, occurring e.g. in the field of fluid-structure interaction for interpolation of interface data and deformation of the CFD mesh.[2,9,10,15,18] More recently, they have been used for aerodynamic shape optimization.[5] RBFs have many compelling features. According to,[7] RBFs have the potential as a universal grid-free method, as they allow for arbitrary scattered data, and generalize easily to several space dimensions. Approximations based on smooth RBFs are highly efficient in approximating smooth functions.[11] However, RBF methods also experience a number of problems. For the popular *global radial basis functions* (cf.[26]), the coefficient matrix of the linear system to be solved is dense. Furthermore, the coefficient matrix is prone to ill-conditioning, and this is significantly influenced by e.g. the number and location of centers used for the interpolant. Finally, evaluating the interpolant often turns out to be computationally

expensive, since it involves constructing large numbers of sums; hence the term *summation problem*.

Remedies for these problems have been developped (see e.g.[26] for an overview), but only some of these techniques have found their way to the area of mesh deformation: Jacobsson and co-workers have used varying shape parameters and employed a reduction of centers to arrive at an approximately equi-distant distribution;[5] Rendall et al. have used greedy algorithms for center selection.[10]

The present work focuses on the application of RBF-based mesh deformation to the deflection of control surfaces of aircraft configurations. A technique is presented, termed *locally confined RBF-based mesh deformation*, which restricts the mesh deformation to the vicinity of the deflecting control surface by evaluating the interpolant exclusively in a specified geometry and imposing suitable additional boundary conditions. Moreover, it reduces the number of data sites for the interpolants by using an adaptive strategy. Furthermore, a variant of the adaptation scheme used for mesh deformation (see e.g.,[10] Algorithm 2) is introduced. Rather than to use the centers for all displacements during adaptation, it is shown that different sets can be advantageously employed for different displacements. This strategy is called *sequential uni-variate adaptation*. Additionally, this strategy can be combined with different sets of data sites for different directions, corresponding to different boundary conditions. Moreover, parameter studies were conducted to investigate the influence of the numbers of initial centers obtained by point reduction and of the number of data sites added during the adaptive process. These techniques are demonstrated on three test cases involving the deflection of control surfaces of the Standard Dynamic Model (SDM), a generic fighter aircraft.[21] By employing overlapping Chimera mesh blocks and different additional boundary conditions for different displacements, large deflection angles can be achieved, even for cases in which small gaps in spanwise direction between control surface and component (e.g. aileron and wing) are modelled.

The outline of this paper is as follows. Section 2 presents the theory for mesh deformation based on radial basis functions and comments on different aspects and choices made for the applications. Section 3 details the method to locally restrict the mesh deformation. Section 4 gives an overview of the adaptation strategies used for the applications and presents the idea of using uni-variate adaptations sequentially. In section 5 the applications are presented. Conclusions are drawn in section 6.

2 MESH DEFORMATION BASED ON RADIAL BASIS FUNCTIONS

Mesh deformation based on radial basis functions (RBFs) constitutes a multi-variate interpolation problem: Given values of three[i] functions at distinct data sites, the RBF interpolation coefficients are to be found such that interpolation conditions are fulfilled. For mesh deformation, these function values are the displacements of the points of the

[i]Three dimensional cases are exclusively considered in this work.

deforming surface mesh in the three coordinate directions. Based on those function values, an interpolant is constructed and then evaluated at other points, usually the mesh points of the volume mesh.

In mathematical terms, we are given a set of data $\mathbb{D}_d = \{(\vec{x}_i, \vec{f}_i),\ i = 1, 2, \ldots, N_d,\ \vec{x}_i \in \mathbb{R}^3, \vec{f}_i \in \mathbb{R}^3\}$ at N_d distinct data sites, where $\vec{x}_i = [x_i, y_i, z_i]^T$ are the coordinate vectors and $\vec{f}_i = [f_{x,i}, f_{y,i}, f_{z,i}]^T$ the corresponding function values. We are to find continuous functions $s_x(\vec{x}), s_y(\vec{x}), s_z(\vec{x}) : \mathbb{R}^3 \to \mathbb{R}$ which interpolate the function values at the given data sites[ii]:

$$s(\vec{x}_i) = f_i, \quad i = 1, 2, \ldots, N_d. \tag{1}$$

We assume s to be linear combinations of RBFs Φ_j, augmented with a linear polynomial $p(\vec{x})$ to enable exact recovery of rigid body translations and rotations[iii]:

$$s(\vec{x}) = p(\vec{x}) + \sum_{j=1}^{N_d} \Phi_j \beta_j, \tag{2}$$

with $p(\vec{x}) = \alpha_1 + \alpha_2 x + \alpha_3 y + \alpha_4 z$. The radial basis functions $\Phi_j : \mathbb{R}^3 \to \mathbb{R}$ are defined by shifting a single basic function ϕ to the data sites[iv]:

$$\Phi_j(\vec{x}) = \phi(\|\vec{x} - \vec{x}_j\|_2). \tag{3}$$

For finding the coefficients $\vec{\alpha} = [\alpha_1, \alpha_2, \alpha_3, \alpha_4]^T$, $\vec{\beta} = [\beta_1, \beta_2, \ldots, \beta_{N_d}]^T$, the interpolation conditions (1) as well as four additional conditions have to be satisfied:

$$\sum_{i=1}^{N_d} \beta_i = 0, \qquad \sum_{i=1}^{N_d} \beta_i x_i = 0, \qquad \sum_{i=1}^{N_d} \beta_i y_i = 0, \qquad \sum_{i=1}^{N_d} \beta_i z_i = 0. \tag{4}$$

This yields the linear system

$$\boldsymbol{A}\vec{c} = \vec{b}, \tag{5}$$

where $\vec{c}$ is the coefficient vector $\vec{c} = [\vec{\alpha}^{\,T}, \vec{\beta}^{\,T}]^T$, $\vec{b} = [0, 0, 0, 0, f_1, f_2, \ldots, f_{N_d}]^T$ the vector of the right-hand side (RHS). $\boldsymbol{A}$ is the coefficient matrix

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{Q}^T \\ \boldsymbol{Q} & \boldsymbol{\Phi} \end{bmatrix}, \tag{6}$$

where $\boldsymbol{0}$ is a 4×4-matrix consisting of zeros, $\boldsymbol{Q}$ an $N_d \times 4$-matrix with row vectors $\vec{q}_i = [1, x_i, y_i, z_i]$ and $\boldsymbol{\Phi} = (\Phi_{ij}) = (\phi(\|\vec{x}_i - \vec{x}_j\|_2))$ an $N_d \times N_d$-matrix $(i, j = 1, 2, \ldots, N_d)$.

[ii] For simplicity, the indices x, y, z are omitted in most of the remainder of the paper.

[iii] This augmentation also ensures well-posedness for radial basis functions such as thin plate splines (TPS).

[iv] $\|\cdot\|_2$ signifies the ℓ_2-norm.

Equation (5) is set up for each coordinate direction individually and solved for the coefficient vectors. Once they are obtained, the interpolants are evaluated at N_e evaluation points $\vec{x}_k,\ k = 1, 2, \ldots, N_e$, yielding function values $f_{e,k}$. In matrix notation, this can be concisely written as follows[v]:

$$\vec{f_e} = \begin{bmatrix} \boldsymbol{Q_e} & \boldsymbol{\Phi}_e \end{bmatrix} \begin{Bmatrix} \vec{\alpha} \\ \vec{\beta} \end{Bmatrix}, \tag{7}$$

where $\boldsymbol{Q_e}$ is an $N_e \times 4$-matrix of the same structure as $\boldsymbol{Q}$, but constructed with the evaluation points $\vec{x}_k$ and $\boldsymbol{\Phi_e} = (\Phi_{e,ki}) = \phi(\|\vec{x_k} - \vec{x}_i\|_2)$ an $N_e \times N_d$-matrix, built with both evaluation points $\vec{x}_k$ and data sites $\vec{x}_i$.

2.1 Remarks

Choice of radial basis functions. In this work, the two *global radial basis functions* thin plate splines (TPS) and multi-quadrics (MQs) are used. For most of the applications in this work involving volume mesh deformation, the TPS is applied:

$$\Phi_j(\vec{x}) = \begin{cases} (\|\vec{x} - \vec{x}_j\|_2)^2 \ln(\|\vec{x} - \vec{x}_j\|_2) & \text{if} \quad \vec{x} \neq \vec{x}_j, \\ 0 & \text{else.} \end{cases} \tag{8}$$

TPS are chosen as they have been found to result in deformed meshes of high quality[6] and yield rather well-conditioned coefficient matrices compared to other RBFs.[11] For TPS, an augmentation with a linear polynomial is needed to ensure well-posedness (cf.[26]). For MQ, no augmentation is necessary (cf.[26]). The MQ is defined as

$$\Phi_j(\vec{x}) = \sqrt{1 + (\epsilon \cdot \|\vec{x} - \vec{x}_j\|_2)^2}, \tag{9}$$

where ϵ is a parameter controlling the shape of the multi-quadric as well as the conditioning of the coefficient matrix $\boldsymbol{A}$ and the accuracy of the numerical solution (cf.[12]). Following,[12] MQs are used in this work for surface mesh deformation with a fixed shape parameter as large as $\epsilon = 10^4$, so as to linearly interpolate between given data sites and to conserve the planarity of surfaces.

Choice of data sites for the interpolant. Since the sizes of the evaluation matrix $\boldsymbol{\Phi_e}$ as well as the coefficient matrix $\boldsymbol{A}$ and its conditioning depend on the data sites selected for the interpolant, it is advantageous to use a subset $\mathbb{D}_c = \{(\vec{x}_j, \vec{f_j}),\ j = 1, 2, \ldots, N_c\},\ \mathbb{D}_c \subseteq \mathbb{D}_d$ with N_c so-called *centers* $\vec{x}_j$, as long as the extrapolation quality at the remaining data sites is not sacrificed. To this end, a greedy algorithm for the selection of centers is applied, which is outlined in section 4.

Coefficient matrix, conditioning and solution method. The coefficient matrix $\boldsymbol{A}$ of equation (5) is dense for global RBFs such as TPS and MQ. For large number of centers N_c, the solution by direct methods is prohibitive, as they have a cost of $\mathcal{O}(N_c^3)$.

[v] For meshes with a large number of points, i.e. with large N_e, assembling the matrices $\boldsymbol{C_e}$ and $\boldsymbol{\Phi_e}$ is prohibitive.

Therefore, they are only used in this work for surface mesh deformations, where typically $N_c \sim \mathcal{O}(10^2)$. For volume mesh deformation, the numbers of centers are larger, usually $N_c \sim \mathcal{O}(10^3)$. For these kind of problems, GMRES was applied.

The coefficient matrix is prone to ill-conditioning. For global RBFs, the condition number may be influenced e.g. by the number and location of centers as well as by shape parameters. The shape parameter of the MQ used in this work is taken to be very large, leading to rather well-conditioned matrices (cf.[7]). MQs are only applied to surface mesh deformation with a relatively small number of centers. As to the location of centers, its separation distance was found to affect the condition number (cf.[26]). As the number of centers grows, the separation distance of the centers generally decreases and the conditioning worsens. In this work, we consider the adaptation of centers, which affects both their number and distribution.

Evaluation. Once the interpolant is found, it can be evaluated on different point sets, e.g.

- at the remaining data sites to check the extrapolation quality at the data sites where the function values are given (and to decide which points to add in an adaptive process);
- at N_e evaluation points; these are usually the same for each direction; they might lie on a surface mesh in case of surface mesh deformation; or, in case of volume mesh deformation, they may be all or a subset of all mesh points (preferably in the vicinity of the deforming component).

Typically, for volume mesh deformation due to a deforming control surface, the number of centers is exceeded by far by the number N_G of grid points, i.e. the potential evaluation points. Hence we have $N_c \sim \mathcal{O}(10^2 \dots 10^3)$ as opposed to $N_e = N_G \sim \mathcal{O}(10^5 \dots 10^7)$. Since large numbers of sums have to be computed (see equation (7)) when evaluating the interpolant, this is called the *summation problem*. For mesh deformation, this problem is even more pronounced, since we have to evaluate three interpolants at N_e evaluation points. For an overview of remedies, the reader is referred to e.g.[26]

The method proposed in this paper seeks to reduce the number N_e of evaluation points to the points in the vicinity of the deflecting / deforming component. It is therefore termed *locally confined RBF-based mesh deformation* and is detailed in section 3.

3 LOCALLY CONFINED MESH DEFORMATION WITH CENTER ADAPTIVITY

The method proposed in this paper restricts the mesh deformation to the vicinity of the deforming or deflecting surface, thereby directly tackling the number of evaluation points N_e for the interpolants s_x, s_y, s_z. Furthermore, it ensures that other adjacent components are unaffected by the mesh deformation. The method consists of the following three ingredients:

1.) local confinement for the interpolant evaluations;

2.) imposing additional boundary conditions;
3.) center adaptation.
The first two ingredients are explained in the following, whereas the third one is detailed in section 4. In section 5 the method will be applied to three test cases involving control surface deflection.

Local confinement. The aim of this method is to confine the evaluation of the interpolants to a region around the deforming / deflecting surface, thereby evaluating the interpolant at $N_e = N_H$ mesh points of the vicinity, rather than at all N_G mesh points of the volume mesh. Since N_H may be considerably smaller than N_G, the proposed method has the potential to significantly reduce computational time.

To this end, a geometrical entity, containing this deforming / deflecting surface, is defined. All mesh points lying inside this entity are flagged differently from those lying outside. In principal, any geometrical entity which can contain such a surface may be chosen with the requirements that the relative location of mesh points be easily computable. This work focuses on mesh deformation for control surfaces, where the connection to the parent component[vi] is usually characterized by straight lines. Hence, the choice of a geometrical entity featuring planar surfaces is advantageous. Moreover, different extensions in different directions should be possible, so as to better adapt to the shape of the control surface. For these reasons, a hexahedron is selected.

Filtering the mesh points is straightforward. The hexahedral faces can be split into triangles, and their normal vectors constructed. For each mesh point, the location with respect to the triangle can then be tested. Although this filtering process increases the overall computational cost, it is sufficient to perform it only once for a deflecting surface and then store it for future evaluations. And since the number of mesh points in the hexahedron, N_H, may be a couple of orders of magnitude smaller than the total number of mesh points N_G, the overall reduction in computational time outweighs by far the increase due to filtering.

Additional boundary conditions. In order to avoid discontinuities of the displacements at the hexahedral faces, which may result in mesh distortion and negative elements, additional data sites with zero displacements are distributed approximately equi-distantly on the hexahedral faces. Thus, the interpolants are forced to become zero at these locations. By additionally choosing a spacing of about once or twice the length of the smallest element edge in that region, mesh distortion is expected to be negligible. Duplicate points have to be filtered out so as to avoid singular coefficient matrices.

Depending on the applications, different additional data sites may be chosen for different function values / displacement directions. This is in accordance with the idea of treating different function values independently, which is used for center adaptation and termed *sequential uni-variate center adaptation* (cf. section 4).

[vi] e.g. the wing in case of an aileron or the vertical tail in case of a rudder

4 CENTER ADAPTATION

The aim of center adaptation is to achieve sufficiently small interpolation errors on all N_d data sites, while taking only N_c centers, where N_c is preferably considerably smaller than N_d. Following,[10] the algorithm starts with a very small number of centers and then refines the space by adding those data sites where the interpolation error (or residual) is largest. This approach is a greedy algorithm, which always makes the best possible choice of the moment in disregard of any future consequences.

For a uni-variate interpolation problem, a basic pseudo-algorithm may look as follows[vii]:

(1) Choose a small set of $N_c^{(t)},\ t = 0$ initial centers.

(2) Set up and solve the linear system for the interpolation coefficients.

(3) Evaluate the interpolant at all N_d data sites and calculate the residual vector $\vec{r}^{(t)} = [r_1^{(t)}, r_2^{(t)}, \ldots, r_i^{(t)}, \ldots, r_{N_d}^{(t)}]^T$ with $r_i^{(t)} = |s^{(t)}(\vec{x}_i) - f(\vec{x}_i)|,\ i = 1, 2, \ldots, N_d$.

(4) Construct a vector $\vec{r}_s^{(t)}$ by sorting the residual vector $\vec{r}^{(t)}$, e.g., in decreasing order.

(5) Add the M_{add} worst data sites to the set of centers where $r_{s,M_{add}}^{(t)}$ is larger than a prescribed tolerance Tol_{add}.

(6) Test stopping criteria and, if none is met, increase the iteration counter t and continue with step **(2)**.

As to step **(6)**, an adaptation is successful if even the largest residual is smaller than a given tolerance Tol_{stop}, i.e. $r_{s,0}^{(t)} < Tol_{stop}$. Furthermore, limits on the number of iterations and on the total number of centers can be imposed.

Since mesh deformation is a multi-variate interpolation problem, it was therefore suggested to combine the residuals of all three directions, i.e. $\vec{r}_x^{(t)}$, $\vec{r}_y^{(t)}$, $\vec{r}_z^{(t)}$, into a combined residual vector $\vec{r}_{comb}^{(t)}$ where an element is calculated as $r_{comb,i}^{(t)} = \sqrt{\left(r_{x,i}^{(t)}\right)^2 + \left(r_{y,i}^{(t)}\right)^2 + \left(r_{z,i}^{(t)}\right)^2}$. [10] $\vec{r}_{comb}^{(t)}$ is sorted (step **(4)**), and this sorted vector then determines which data sites to include for the next iteration. The above algorithm offers several options, e.g. (a) what initial point distribution to choose; (b) how many points to add at each iteration; (c) whether center adaptation occurs concurrently for all or sequentially for different function values. Regarding option (a), a rudimentary point reduction scheme was applied, which results in approximately equi-distant point distribution. As to option (c), for problems of e.g. fluid-structure interaction as treated in,[10] the standard approach seems to be to use the same set of centers at one iteration for all directions (abbreviated "MV"). The alternative approach is called here *sequential uni-variate center adaptation* (SQUV). It might be especially advantageous for problems of control surface deflection, since "pure" rotation of mesh points does not involve mesh point movement parallel to the rotation axis and hence does not need an interpolant in that direction. Along these lines, different data

[vii] The superscript t denotes the iteration counter.

sites can be prescribed for different function values, corresponding to boundary conditions particularly tailored to different directions. These issues are addressed in section 5.

5 APPLICATIONS

The SDM generic fighter aircraft configuration[21] was used for the three different test cases of this paper. The full model (cf. figure 1-left) consists of a fuselage, wings with ailerons, a vertical tail plane with a rudder and horizontal tail planes (HTs). Ailerons, rudder and HTs act as the SDM's control surfaces, and they have planar surfaces. The rear part of the fuselage is of cylindrical shape. The original configuration without gaps between components was used for the first test case, where the aileron was to be deflected. Thus, blending in chord- as well as spanwise direction was required. For a more realistic setting, a gap of about 1.4% of the half-span width was introduced in spanwise direction (cf. figure 1). Here, only blending in chordwise direction takes place. Furthermore, by making use of the Chimera technique with its overlapping mesh blocks, mesh distortion in the gap due to aileron deflection was alleviated by relaxing the boundary conditions imposed on the enclosing hexahedron. The third test case describes the deflection of the HT with the requirements that the planarity of the HT's surfaces as well as the cylindrical shape of the rear fuselage be conserved (see appendix A). Their adherence can be verified by visual inspection.

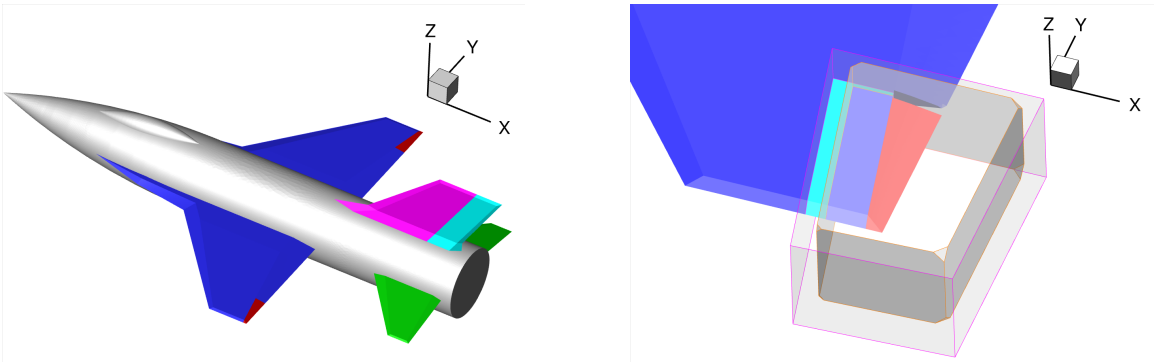

Figure 1: Left: SDM generic fighter aircraft configuration. Right: Overlapping Chimera blocks for the deflection of the aileron with gap in spanwise direction.

After CAD-preparation of the geometry, two unstructured tetrahedral meshes were generated. The mesh of the full model is a single-block mesh and consists of $623,387$ points, $3,367,882$ tetrahedral elements and has no gaps between components. The Chimera mesh block for the aileron used for the second test case has $918,699$ points and $5,108,630$ tetrahedrons. Its aerodynamic surfaces consist of the aileron (red color, cf. figure 1-right), a part of the wing (blue) and a part of the wing (cyan) which is shared with the Chimera block of the rest of the aircraft. In figure 1-right, the Chimera block for the aileron is displayed transparent and with magenta edges.

5.1 First test case: Deflection of the aileron

Parameter studies were conducted for this test case. To this end, mesh deformation was performed for the left aileron, deflected by $-12°$ around a rotation axis not aligned with the y-axis. For the surface mesh deformation, 58 mesh points at the aileron's intersection with the wing were taken as data sites, together with 18 points at the trailing edge, with the exception of the part close to the wing (to enable spanwise blending). The MQ's interpolation coefficients were determined (cf. appendix A) and then evaluated on the 240 surface mesh points of the aileron. For the volume mesh deformation, a hexahedron was constructed with two faces coinciding with one intersection line of aileron and wing each. 800 additional data sites were distributed over its faces.

Table 1 shows the convergence results of center adaptations for MV and SQUV for different numbers of initial centers, procured by the point reduction algorithm for control surface and hexahedron individually.[viii] $N^{(0)}_{c,\,CS}$ and $N^{(0)}_{c,\,Hex}$ denote the respective number of initial centers. t_{end} is the final iteration number, $N_{c,\,end}$ the total final number of centers. The tolerance for addition and the stopping tolerance were $Tol_{add} = Tol_{stop} = 8.e-05$. At each iteration, one point was added at most, i.e. $M_{add} = 1$. For each computation, the stopping tolerance was reached. It can be seen that an increase in the initial numbers of centers decreases the final iteration number, but that the total final number of centers mostly grow. The influence on the computational time is negligible. As to SQUV, the convergence is fastest for f_y, followed by the convergence for f_x. Also, different total final numbers of centers were needed, the smallest number of centers for f_y, followed again by f_x. Addition of the CPU times of every SQUV function value reveals that MV is more expensive than SQUV by more than a factor of two. A comparison of the results for SQUV-f_z and MV shows that the results are identical except for the CPU time. This finding is emphasized by the curves for f_z and MV of figure 2 (for $N^{(0)}_{c,\,CS} = 5$ and $N^{(0)}_{c,\,Hex} = 5$): The values of the maximum residual of f_z and the maximum residual of MV[ix] are identical, but shifted in the direction of the abscissa. That means that the error for f_z for MV dominates the errors for the other function values. However, with the numbers of centers and their location most suitable for f_z, the linear systems for the other function values has to be solved and the interpolants evaluated. This accounts for the increased CPU time.

In the next study, the maximum allowed number of points to be added, M_{add}, were varied for SQUV for f_z, while keeping the initial numbers of centers constant at $(N^{(0)}_{c,\,CS}, N^{(0)}_{c,\,Hex}) = (5, 5)$. The results are listed in table 2. It can be seen that increasing M_{add} generally decreases the final iteration number, increases the total final number of centers, but again decreases the CPU time. A reduction of CPU time by a factor of more than 6 can be

[viii] All calculations were performed on a local PC with four Intel(R) Core(TM)2 Quad processors, 2.66 GHz and 3760 MB memory.

[ix] For comparison purposes, not the maximum of the combined residuals is taken, but the maximum of the maxima of the residuals for the different function values.

achieved with $M_{add} = 50$. However, the decrease in CPU time is monotonic only up to $M_{add} = 25$, thereafter the CPU time frequently increases again. Nevertheless, using M_{add} larger than one seems to be an effective means for reducing the CPU time.

	$N^{(0)}_{c,\,CS}$	$N^{(0)}_{c,\,Hex}$	t_{end}	$N_{c,\,end}$	CPU time in $[s]$
multi-variate	5	5	338	348	4.538e+01
	15	15	308	338	4.342e+01
	25	25	298	348	4.691e+01
	35	35	283	353	4.816e+01
uni-variate (f_x)	5	5	95	105	8.7e-01
	15	15	76	106	8.2e-01
	25	25	66	116	8.8e-01
	35	35	55	125	9.1e-01
uni-variate (f_y)	5	5	32	42	1.4e-01
	15	15	18	48	1.4e-01
	25	25	14	64	1.6e-01
	35	35	11	81	1.7e-01
uni-variate (f_z)	5	5	338	348	1.999e+01
	15	15	308	338	1.880e+01
	25	25	298	348	1.992e+01
	35	35	283	353	2.089e+01

Table 1: Comparison of convergence for MV and SQUV ($M_{add} = 1$).

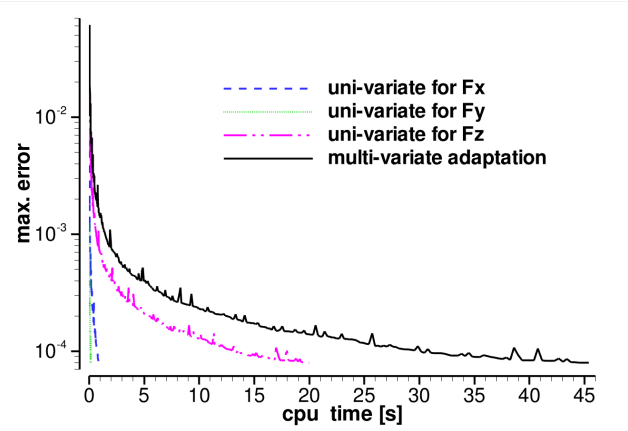

Figure 2: Comparison of convergence between SQUV and MV ($N^{(0)}_{c,\,CS} = 5$, $N^{(0)}_{c,\,Hex} = 5$, $M_{add} = 1$).

Deformed meshes were produced for both MV and SQUV for $(N^{(0)}_{c,\,CS}, N^{(0)}_{c,\,Hex}) = (10, 10)$ and $M_{add} = 10$. The deflection angle was again $-12°$. Figure 3 shows the quality histograms of the deformed meshes with respect to the undeformed one. As quality measure Q_{mesh}, the mean ratio for tetrahedral elements[x] was taken (cf.[22]). Only minor quality

[x]The mean ratio is defined as $Q_{mesh} = \frac{12\sqrt[3]{9V^2}}{\sum_{0\le i\le j\le 3} l_{ij}^2}$, where V denotes the volume of the tetrahedron, l_{ij} the length of an edge.

deteriorations are visible, for $Q_{mesh} < 0.55$ none at all. This means that SQUV results in a deformed mesh of comparable quality as MV.

M_{add}	t_{end}	$N_{c,\,end}$	CPU time in $[s]$
1	338	348	19.990
5	80	407	8.470
10	45	442	6.080
15	33	499	5.620
20	26	505	4.970
25	22	538	4.710
30	21	563	5.820
35	17	558	4.020
40	14	562	3.220
45	15	600	5.490
50	12	588	3.200
55	13	650	4.620

Table 2: Convergence behavior for addition of multiple centers at each iteration (aileron deflection of $-12°$, SQUV for f_z, $(N^{(0)}_{c,\,CS}, N^{(0)}_{c,\,Hex}) = (5, 5)$).

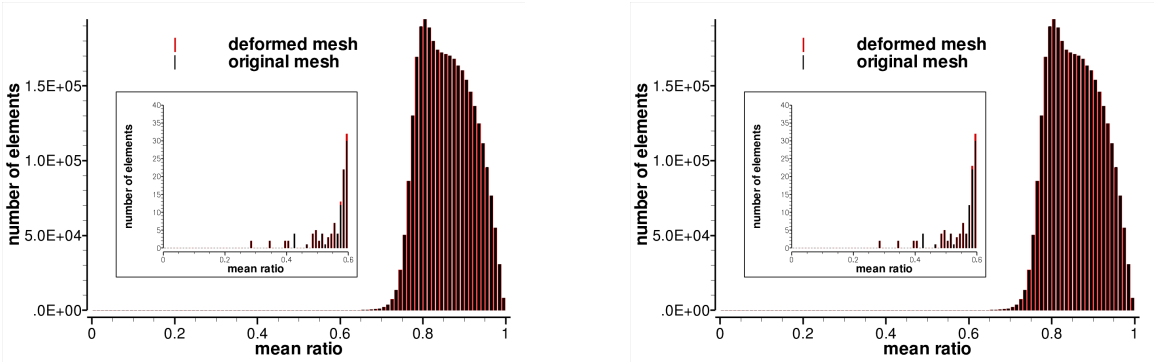

Figure 3: Mesh quality comparison of deformed (red) to original mesh (black) for MV (left) and SQUV (right).

For the interpolant of f_x of the above mesh deformation by SQUV ($N_c = 214$), the CPU times were measured. The filtering of the mesh points took 0.06 seconds, and $1,937$ mesh points were located inside. The evaluation of the interpolant on these mesh points took 0.03 seconds, whereas 8.39 seconds on all mesh points.

5.2 Second test case: Deflection of the aileron within a Chimera block

For this test case, mesh deformation was performed for aileron deflections of $-12°$ and $-30°$ inside the Chimera mesh block. This resulted in maximum deflections of the aileron's trailing edge of about 1 and 2.6 times the gap width in z-direction or about 10 and 24 times the edge length of a typical tetrahedron in that area, respectively. For the generation of input data, 405 mesh points on the connection to the wing and 200 points

on the trailing edge were extracted. The former stayed fixed, while the latter underwent rotation. The rotation axis is the same as in section 5.1, but this time no blending in spanwise direction occurs. With the above points, MQ interpolants were constructed and then evaluated on the $17,952$ surface mesh points of the aileron.

As to the volume mesh deformation, additional data sites were chosen such that the overlapping zone of the two Chimera mesh blocks are conserved. Due to the small gap between aileron and wing, severe mesh distortion in that region was likely to occur under the requirement that no mesh point move on the respective face of Chimera mesh block. Therefore, the local confinement technique was applied with a little tweak. All but one face of the hexahedron were constructed so as to lie inside the Chimera mesh block (one face being aligned with the connection line of aileron and wing). The other, wing root facing side of the hexahedron was chosen to extend beyond the corresponding face of the Chimera mesh block (both are perpendicular to the y-axis). Additional data points were then distributed over all hexahedral faces but the latter one. Only for f_y, data sites were placed in a section of the face of the Chimera mesh block facing the root, which was cut by the hexahedron. Thereby, only in-plane movement of the mesh points inside the hexahedron on this face of the Chimera mesh block was enabled. This resulted in $4,696$ additional data sites for the y-direction, but only $3,672$ for the x- and the z-direction.

For both deflections, the convergence results of SQUV are listed in table 3 ($M_{add} = 8$, $(N^{(0)}_{c,\,CS}, N^{(0)}_{c,\,Hex}) = (20, 20)$, $Tol_{add} = Tol_{stop} = 8.e-05$; all adaptations stopped because the stopping tolerance was reached).

		t_{end}	$N_{c,\,end}$	cpu-time in $[s]$
$-12°$	uni-variate (f_x)	35	320	40.59
	uni-variate (f_y)	22	210	30.5
	uni-variate (f_z)	110	916	338.16
$-30°$	uni-variate (f_x)	87	729	172.88
	uni-variate (f_y)	63	537	88.88
	uni-variate (f_z)	160	1 315	1010.72

Table 3: Comparison of convergence for SQUV for aileron deflections of $-12°$ and $-30°$.

Again, the adaptation for f_y requires the least number of centers and the least cpu time. This is despite the fact that $1,024$ data sites more were used for the adaptation of this function value. Comparison of the results for different deflection angles shows that different numbers of centers are necessary to achieve the specified stopping tolerance and that, hence, different cpu time is needed. For a larger deflection angle, more centers are required, regardless of the function value to be adapted for. This might be due to the fact that for a larger deflection angle, the variations in the function values from the one imposed on the aileron to the ones additionally imposed are larger and that more centers are required to account for that. The histograms of the mesh quality are shown in the above figure 4. The deflection of $-12°$ can be seen not to cause severe quality degradation: no deterioration is visible for $Q_{mesh} < 0.6$. For a deflection of $-30°$, the deterioration is

much more severe; still, the deformed mesh does not possess elements with $Q_{mesh} < 0.4$.

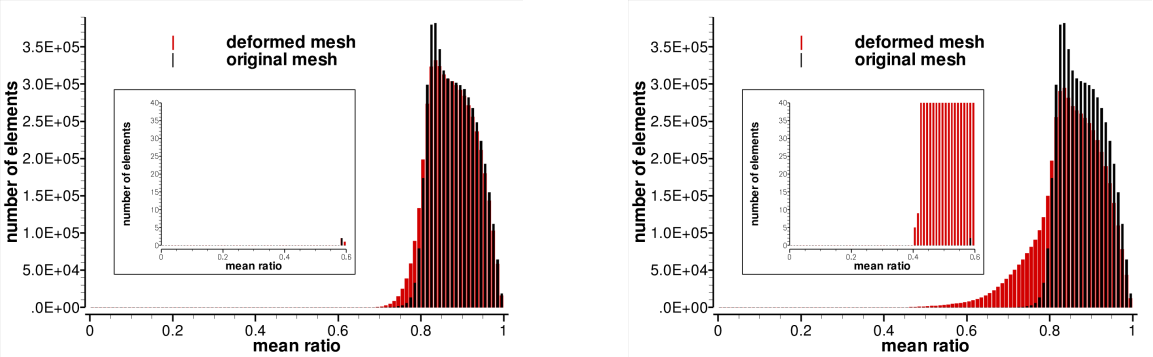

Figure 4: Mesh quality comparison of deformed to original mesh. Aileron deflection of $-12°$ (left) and $-30°$ (right).

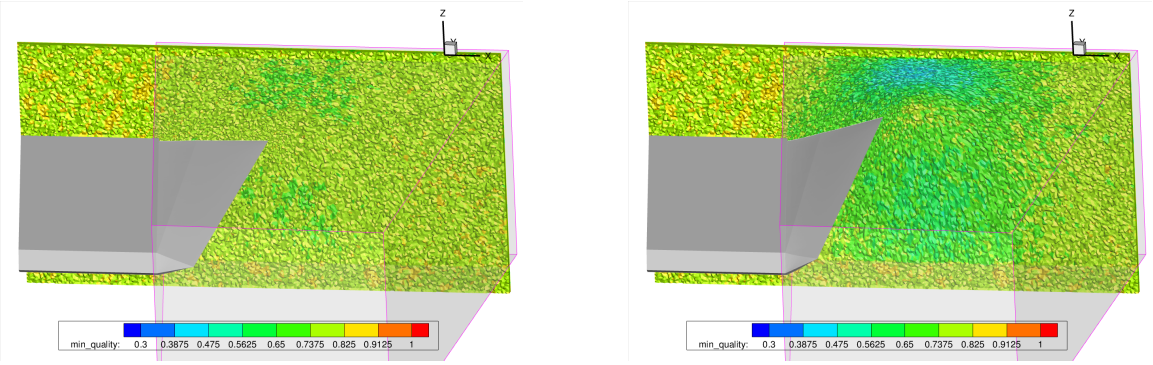

Figure 5: Portion of the tetrahedral volume mesh in the gap between aileron and Chimera face, colored by mesh quality. The deflection angles are $-12°$ (left) and $-30°$ (right). Aileron and part of the wing are shaded (dark grey), the hexahedron is transparent (magenta edge).

In figure 5, it can be seen that for both deflection angles, no change in mesh quality and (vaguely) no element distortion occurs on the portion[xi] of the tetrahedral volume mesh facing the wing's tip outside the hexahedron. Furthermore, it can be seen that the wing is not influenced by the deformation and that the planarity of the surfaces of the aileron is conserved. The locally confined mesh deformation is hence successful in both cases. The findings based on the quality histograms translate to this figure, too. No severe quality degradation for $-12°$ can be observed; for $-30°$, however, there is, especially for the elements near the deflected aileron near the upper face of the hexahedron. Here, a compression of elements can be seen. Overall, allowing in-plane motion of mesh points on a Chimera face successfully alleviates the degradation of element quality even for large deflection angles.

[xi] This portion was created by the visualization system using the so-called *value blanking* option, i.e. omitting parts of the surface or volume mesh within ranges of specified coordinates.

5.3 Third test case: Deflection of the HT

This test case deals with the volume mesh deformation due to the rotation of the HT, which is attached to the fuselage. For the generation of input data, surface mesh deformation was this time considerably more involved than for the previous test cases, since rotation of the mesh points of the HT entails a movement of the mesh points of the adjacent panel of the fuselage, where mesh nodes where allowed to move. Mesh movement was to occur in such a way that the planarity of the HT's surfaces and the circular shape of the fuselage panel be conserved. Details for the generation of input data are given in appendix A. The HT surface mesh had $11,722$ points, whereas the surface mesh of the adjacent fuselage panel consisted of $5,353$ points. The rotation axis was chosen to be the y-axis, and the center of rotation was set to be at about 60% with respect to the HT's root chord. Input data for a deflection of the HT of $+12°$ was generated. The data sites of the surface mesh totaled $16,816$[xii], whereas the number of additional data sites of the hexahedron amounted to $9,080$ (the hexahedral face pointing towards the other side of the left HT was spared). The surface mesh of the HT and of the adjacent fuselage panel as well as the hexahedron and the additional data sites can be seen in figure 6. Note that the hexahedron does not extend beyond the rear part of the fuselage so as to prevent the corresponding surface mesh points from moving and consequently from deforming the shape.

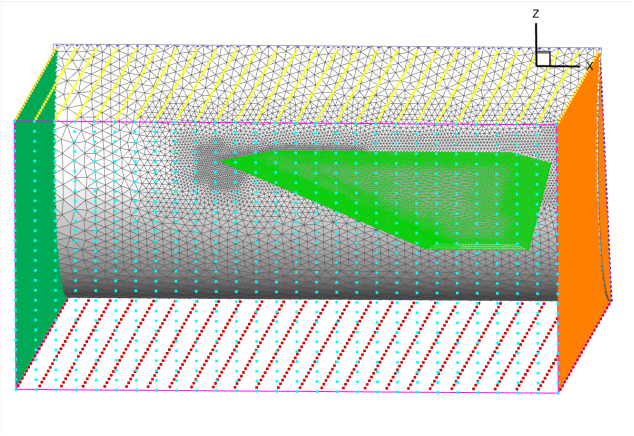

Figure 6: Surface mesh of the HT and the adjacent panel of the fuselage as well as the hexahedron and the additional data sites distributed over its faces.

The settings for volume mesh deformation with SQUV were: $(N^{(0)}_{c,\,CS}, N^{(0)}_{c,\,Hex}) = (100, 100)$, $M_{add} = 12$, $Tol_{add} = Tol_{stop} = 8.e-05$. A limit of 2 200 centers was imposed. Table 4 shows the convergence results. No SQUV has reached the prescribed tolerance, hence all SQUVs end with the same total number of centers and the same iteration number. Differences in cpu time are rather small and may be attributed to different sets of centers

[xii] after subtraction of the points for the intersection line in order to avoid duplicate points

and, correspondingly, different RHSs, which may have affected the solution of the linear system by GMRES. The resulting maximum error is the lowest for the spanwise direction, followed by the error in x-direction. Figure 7 shows the quality histogram of the deformed mesh in comparison to the one of the undeformed mesh. It can be seen that, although there is degradation of mesh quality, the deformed elements still have a quality larger than $Q_{mesh} = 0.4$.

	t_{end}	$N_{c,\,end}$	cpu-time in $[s]$	max. error
uni-variate (f_x)	167	2200	1452.15	3.1952e-04
uni-variate (f_y)	167	2200	1478.61	1.8296e-04
uni-variate (f_z)	167	2200	1460.43	3.8354e-04

Table 4: Results for SQUV for HT deflections of 12°.

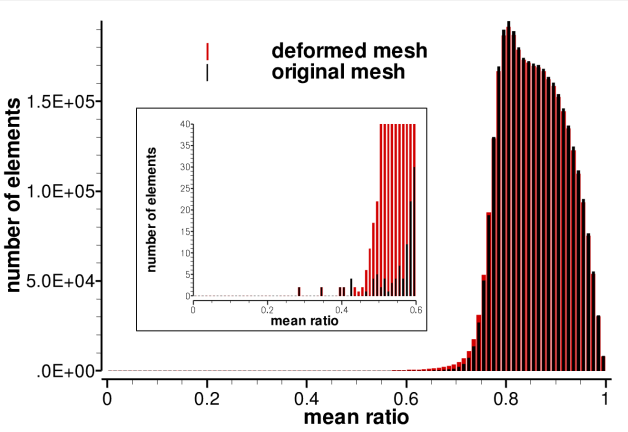

Figure 7: Quality histograms for undeformed and deformed mesh (HT deflection of 12°).

Figure 8 shows a portion of the tetrahedral volume mesh, colored by mesh quality, surrounding the undeformed (left) and deformed (right) HT and the fuselage. The HT is displayed shaded without surface mesh, so as to spot irregularities on the surfaces due to deformation. The surface meshes of the adjacent fuselage panels are also visible. In the smaller pictures in the right upper corner, the SDM configuration is displayed from the rear with the same shading and mesh settings (wing not visible). It can be seen that the surface mesh on the fuselage only deforms in the adjacent panel (bordered by black line). Furthermore, the circular shape of the fuselage is kept, which is visible from the smaller pictures, and no movement of mesh points occurs at the rear panel. As to the HT, the planarity of its surfaces seems to be conserved, and no bending and gap between HT and fuselage is visible. As can be concluded from the change in color of the tetrahedral elements, quality degradation occurs, but is still above 0.4 in the mesh portion displayed in figure 8.

The cpu time was taken for evaluating the interpolant for f_x ($N_c = 2,200$). The filtering of mesh points consumed 0.05 seconds, and 80,261 mesh points were flagged for evaluation. This took 11.15 seconds, in contrast to 86.18 seconds on the whole mesh.

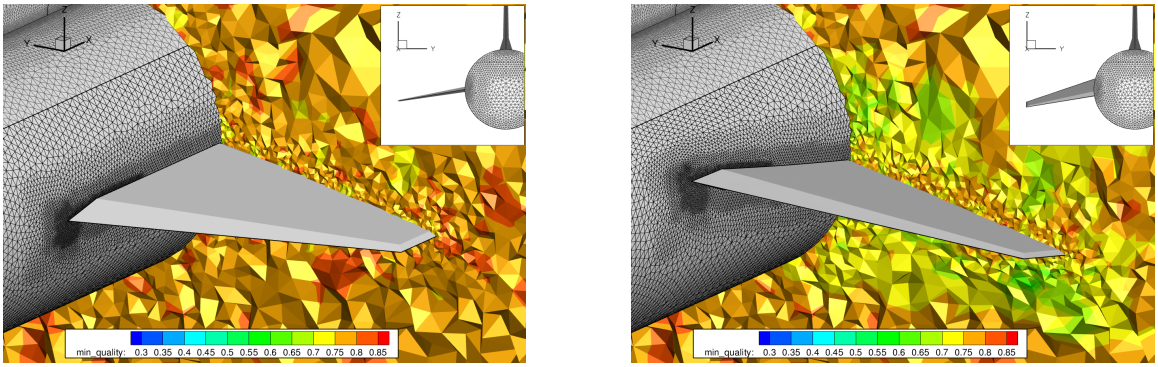

Figure 8: Mesh quality for a portion of the tetrahedral volume mesh for the undeformed (left) and the deformed mesh (right; HT deflection of 12°) as well as surface meshes on fuselage panels.

6 CONCLUSIONS

Locally confined RBF-based mesh deformation has been successfully applied to three test cases involving the control surface deflections of a generic fighter aircraft. It restricts mesh deformation to the compounds of a surrounding geometrical entity and reduces computational time for the evaluation process. Additional data sites have to be specified, and an adaptive process is used to select the centers. Adapting the centers for each displacement independently after one another, termed sequential uni-variate center adaptation, has been shown to be faster than using the same centers for every displacement, while achieving a comparable mesh quality. Along these lines, different additional data sites can be speficied for the different displacements. Thereby, and by exploiting the requirements for a Chimera mesh block, these techniques have been shown to enable large control surface deflections in the presence of a small gap between components. The influence of using different numbers of initial equi-distantly distributed centers has been observed to have a negligible effect on the computational time, whereas addition of multiple centers during one iteration is beneficial.

7 ACKNOWLEDGEMENTS

The author has appreciated the insightful discussions with R.C. Swanson during his stay at $C^2A^2S^2E$ in 2009. He is grateful to his colleagues R. Heinrich, T. Schwarz and S. Langer for discussing various new ideas, and to M. Orlt and S. Albensoeder for revealing the intrinsics of mesh quality issues. The author is thankful to M. Ghoreyshi of Liverpool University for providing the original SDM geometry. The use of Dr. Langer's linear algebra toolbox and of Dr. Albensoeder's mesh quality analysis and improvement program is gratefully acknowledged.

REFERENCES

[1] Murman SM, Aftosmis MJ, Berger MJ. Simulations of a 6-DOF motion with a cartesian method. In *Proceedings of the 41st AIAA Aerospace Sciences Meeting,* January 6-9, 2003, Reno, NV; AIAA-2003-1246.

[2] Rendall TCS, Allen CB. Unified fluid-structure interpolation and mesh motion using radial basis functions. *International Journal for Numerical Methods in Engineering* 2007.

[3] Raveh DE. Maneuver load analysis of overdetermined trim systems. *Journal of Aircraft* 2008; **45**(1).

[4] Allen CB. Parallel universal approach to mesh motion and application to rotors in forward flight. *International Journal for Numerical Methods in Engineering* 2007; **69**:2126–2149.

[5] Jacobsson S, Amoignon O. Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Computers and Structures* 2007; **36**:1119–1136.

[6] de Boer A, van der Schoot MS, Bijl H. Mesh deformation based on radial basis function interpolation. *Computers and Structures* 2007; **85**:784–795.

[7] Sarra SA. Adaptive radial basis function methods for time dependent partial differential equations. *Applied Numerical Mathematics* 2005; **54**(1):79–94.

[8] Tobor I, Reuter P, Schlick C. Efficient reconstruction of large scattered geometric datasets using the partition of unitiy and radial basis functions. *Journal of WSCG* 2004; **12**(1-3).

[9] Rendall TCS, Allen CB. Improved radial basis function fluid-structure coupling via efficient localized implementation. *International Journal for Numerical Methods in Engineering* 2009; **78**:1188–1208.

[10] Rendall TCS, Allen CB. Efficient mesh motion using radial basis functions with data reduction algorithms. *Journal of Computational Physics* 2009; **228**:6231–6249.

[11] Fornberg B, Driscoll TA, Wright G, Charles R. Observations on the behavior of radial basis function approximations near boundaries. *Computers and Mathematics with Applications* 2002; **43**:473–490.

[12] Driscoll TA, Heryudono ARH. Adaptive residual subsampling methods for radial basis function interpolation and collocation problems. *Computers and Mathematics with Applications* 2007; **53**:927–939.

[13] Benek J, Steger JL, Dougherty FC. A flexible grid embedding technique with application to the Euler equations. In *Proceedings of the 6th Computational Fluid Dynamics Conference,* July 13-15, 1983, Danvers, MA; AIAA paper 83-1944.

[14] Peter J, Dwight RP. Numerical sensitivity analysis for aerodynamic optimization: a survey of approaches and applications. *Computers and Fluids*, submitted for publication.

[15] Heinrich R, Kroll N, Neumann J, Nagel B. Fluid-structure coupling for aerodynamic analysis and design: a DLR perspective. In *Proceedings of the 46th AIAA Aerospace Sciences Meeting and Exhibit,* January 7-10, 2008, Reno, NV; AIAA paper 2008-561.

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

[17] Schwarz TO. *Ein blockstrukturiertes Verfahren zur Simulation der Umströmung komplexer Konfigurationen,* PhD thesis, TU Braunschweig, 2006.

[18] de Boer A. *Computational fluid-structure interaction: spatial coupling, coupling shell and mesh deformation,* PhD thesis, TU Delft, 2008.

[19] Heinrich R, Michler A. Unsteady simuilation of the encounter of a transport aircraft with a generic gust by CFD flight mechanics coupling. In *Proceedings of the CEAS 2009 European Air and Space Conference,* October 26-29, Manchester, UK, 2009.

[20] Munoz-Gomez JA, Gonzalez-Casanova P, Rodriguez-Gomez G. Adaptive node refinement collocation method for partial differential equations. In *Proceedings of the Seventh Mexican International Conference on Computer Science (ENC'06),* 2006.

[21] Huang XZ. Wing and fin buffet on the standard dynamics model. *RTO Technical Report 26 - Verification and validation data for computational unsteady aerodynamics*, October 2000.

[22] Dompierre J, Labbé P, Guibault F, Camarero R.. Proposal of benchmarks for 3D unstructured tetrahedral mesh optimization. *Rapport CERCA R98-91*, September 1998.

[23] Schwarz T. RANS simulations with one and six degrees of freedom rigid body motions. In *Notes on Numerical Fluid Mechanics and Multidisciplinary Design (NNFM)*, Vol. 96. Springer, 2008.

[24] Michler A, Heinrich R. Numerical simulation of the elastic and trimmed aircraft. In *Notes on Numerical Fluid Mechanics and Multidisciplinary Design (NNFM)*, Vol. 112. Springer, 2010.

[25] Dwight RP. Robust mesh deformation using the linear elasticity equations. In *Computational Fluid Dynamics 2006,* Deconinck H, Dick E (eds). Springer, 2006.

[26] Fasshauer GE. *Meshfree Approximation Methods with Matlab* (1st edn). World Scientific: Singapore, 2007.

A SURFACE MESH DEFORMATION

In this work, surface mesh deformation is applied to components undergoing deflection and their neighbors with the aim of providing input data for the volume mesh deformation, i.e. surface mesh point coordinates and their displacements. To this end, surface mesh deformation based on RBFs in combination with rotation and projection of mesh points is used.

For the first two test cases (cf. sections 5.1 and 5.2), in order to achieve the deflection of the aileron, blending between fixed and rotated points (located at the aileron's trailing edge) is performed. These two point sets constitute the centers for the MQ interpolant, for which a shape parameter of $\epsilon = 10^4$ is chosen. The evaluation takes place on all the surface mesh points of the aileron.

The third test case (cf. section 5.3) involves the rotation of the horizontal tail (HT), which is attached to the fuselage, with the requirement of conserving the surface shapes as best as possible (planar surfaces for the HT, cylindrical shape for the rear part of the fuselage). Surface mesh deformation is applied to both HT and fuselage. For the HT, all mesh points are rotated first. Then the mesh points on the intersection line with the fuselage are projected onto the cylindrical fuselage (in extension of their HT surfaces). After that, the locally confined mesh deformation technique is applied to the HT's mesh points near the intersection line. To this end, a hexahedron is constructed enclosing the root region of the HT. No adaptation is necessary because of the small number of data sites and a direct solver is used. The latter consist of the rotated intersection points, with the differences between projected and rotated intersection points as displacements, and additional data sites with zero displacements on the tip-facing hexahedron face. The evaluation of the interpolant at the HT's mesh points within the hexahedron results in an elongation of the HT, such that no gap between HT and fuselage due to rotation is created. MQs with the same shape parameter as above are taken. Thereby, the planarity of the surfaces are conserved. Regarding the deformation of the fuselage, surface mesh deformation with TPS is applied. The data sites are the (original) intersection points, where as displacements the differences between projected intersection points and original ones have been taken, and the boundary nodes of the fuselage panel adjacent to the HT with zero displacements. Again, no adaptive process is necessary, and the linear system can be solved by a direct method. The evaluation takes place on all the mesh points of the fuselage panel. Thereafter, since bulging was found to occur, these mesh points are projected radially onto the cylindrical fuselage, thereby conserving the cylindrical shape.