49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition
4 - 7 January 2011, Orlando, Florida

AIAA 2011-1267

Smooth Mixed Meshes for Acceleration of RANS CFD in Aircraft Analysis and Design

J. Wild[1]
DLR German Aerospace Center, Braunschweig, 38108, Germany

An applicable method for generating mixed meshes around simple and medium complex configurations is presented. Core part of the mesh generation strategy is a smooth meshing of the boundary layer region by structured hexahedrons and prisms applying smoothing based on solving face-weighted Laplace equations. A preconditioning of the point set for triangulation by shear and edge-weighted average affine invariant transformation is proposed to stabilize and accelerate volume triangulation. In order to generate a smooth transition from the highly stretched structured boundary layer cells to the volume triangulation an anisotropic volume triangulation is used, where the derivation of the correct metrics is a key component. Benefits of the mesh generation strategy are outlined comparing calculations on different meshes to experimental data for the ONERA M6 wing.

Nomenclature

$\mathbf{A}$	=	affine invariant transformation matrix
$\mathbf{B}$	=	n-dimensional basis
c	=	local wing chord
c_p	=	local pressure coefficient
$\mathbf{e}$	=	eigenvector
$\mathbf{E}$	=	eigenvector matrix
$\mathbf{M}$	=	symmetric metric tensor
M	=	Mach number
Re	=	Reynolds number
$\mathbf{o}, \mathbf{p}, \mathbf{q}$	=	face normal vectors in prismatic grid notation
$\mathbf{P}$	=	N x n point set
$\mathbf{s}$	=	face normal vector in structured grid notation
$\mathbf{S}$	=	shear transformation matrix
$\Delta\mathbf{r}, \Delta\mathbf{s}, \Delta\mathbf{t}$	=	edge vectors in radial, circumferential and marching direction in prismatic grid notation
$\mathbf{T}$	=	transformation matrix
V	=	volume
$\mathbf{x}$	=	point coordinate vector
α	=	angle of attack
φ	=	sweep angle
λ	=	eigenvalue
ξ, η, ζ	=	computational coordinates in structured grid

I. Introduction

Numerical design optimization is an emerging design tool for real world applications in aerodynamics. The state of the art of 3D Navier-Stokes solvers already show a good predictive capability even for more complex 3D aircraft configurations as shown for example in Ref. 1-3.

Optimization based on high-level computational fluid dynamics (CFD) tools using Reynolds-averaged Navier-Stokes equations (RANS) in the meantime is widely used for 2D airfoil design. Today the step towards 3D wing optimization is to be done. Although already practical for simple configurations like wing-body, the effort of

[1] Research Scientist, Institute of Aerodynamics and Flow Technology, Lilienthalplatz 7. AIAA Senior Member

calculating more complex con-figurations is still too high for application within an optimization environment. The main limiting factor hereby is the number of grid points necessary for accurate flow simulation using hybrid unstructured grids consisting of tetrahedral and prismatic elements. E.g. for the configuration used in the 1st AIAA Drag Prediction Workshop[1] the medium hybrid unstructured grid used giving promising results contained approximately 12 million grid points. This resulted in a computational effort of 100 hours CPU-time on a NEC SX5 high performance computer for a given flow condition. Even the grid generation for this case took about 1 day turn-around time. This computational effort is clearly unacceptable in terms of an optimization process, where multiple configurations have to be evaluated. As a rough estimate, when assuming 200 evaluations for a converged optimization, this will lead linearly to 200 days grid generation and 800 CPU-days flow calculation. Even assuming a parallel processing of the flow on 4 processors will result in more than 1 year turn-around time for the optimization.

Regarding applications in high-lift optimization the situation worsens. Today, hybrid unstructured meshes for transport aircraft configurations contain about 20-30 Million grid points. A mesh is usually generated over night using an advancing front method for high quality meshes. Even with the new computer technology providing massively parallel LINUX cluster systems, 1 to 5 days are needed for a fully converged flow simulation. Using such meshes would require 40 to 1000 days of computing, depending on the optimization algorithm

The big shortcoming of hybrid unstructured grids is the low anisotropy of surface triangles resulting in a large number of grid points, which is agglomerated through the number of prismatic layers for the boundary layer resolution. This low anisotropy leads to an unnecessary high resolution in span direction, especially for high aspect ratio wings. Recovering the experience with the application of structured grids, it is known, that the aspect ratios of the surface quadrilaterals can be much higher, additionally resulting in well aligned body conforming meshes. The shortcoming of an overall use of structured meshes is the increasing complexity of the targeted configurations, where structured meshing reaches its limitations, mainly due to grid topology issues.

In the past the author et al.[4,5] reported a new strategy to generate mixed meshes containing smooth structured hexahedral and prismatic elements for the resolution of boundary layers by a parabolic marching routine based on the face-weighted Laplace equation. In the following the method is briefly described. New extensions are presented concerning the interface to the field triangulation and a preconditioning method for the Delaunay field triangulation. These extensions have been crucial in order to accelerate and stabilize the process. Finally the method is applied to a high-lift wing-body configuration including the use within an optimization environment.

II. Mixed Mesh Generation Methods

As long as the variations of the field variables to be resolved by the nu-merical solution to a set of differential equations are of the same order of magnitude in all directions the use of hybrid unstructured meshes is appro-priate and furthermore avoids the manual labor of subdividing the field into blocks usually needed for structured meshes. On the other hand an adequate resolution of very thin layers with large normal gradients requires a layered structure of the mesh. Such meshes may be generated by marching away from a fixed mesh face, say the body surface in case of a boundary layer, or away from a floating mesh face in case of a wake. When the mesh on the initial face is unstructured, the resulting three-dimensional mesh extruded from it consists of prisms of triangular cross-sections and is called hybrid. A structured mesh on the initial face, of course, leads to a fully structured three-dimensional mesh.

Structured meshes are easily clustered differently in all grid line directions. Since they need far less points for directional clustering than hybrid unstructured meshes, they add efficiency to the numerical solution procedure of the differential equations. On the other hand their generation suffers from the amount of work to be invested to achieve the blocking for complex geometries. To ease the situation we combine both kinds of surface meshes starting the three-dimensional marching generation of mixed meshes simultaneously from neighboring patches with structured and un-structured surface meshes.

For the generation of these mixed meshes we integrated surface and volume triangulation methods into the DLR structured mesh generator MegaCads[6]. We used available methods, in fact the TRIANGLE code of Shewchuk[7] and the DELAUNDO code of Mueller et al.[8] for surface mesh generation in parametric space. For the current investigation we use the SIMMETRIX mesh generation code due to its anisotropic capabilities for volume triangulation.

A. Structured Hexahedral Boundary Layer Meshes

Like in most structured mesh generation methods the 3D structured mesh blocks are initially generated by an algebraic interpolation method. Depending on the block boundaries we use either tri-linear or transfinite interpolation[10]. Since our objective is to generate smooth meshes we additionally apply smoothing with the use of elliptic equations. In the work of Niederdrenk[11] beginning with the metric identity in its differential conservative form the well known Poisson equations for structured meshes were derived including a precise analytical definition of all 9 control functions. The metric identity in its differential form simply states that the sum over the face normal vectors in the grid conforming +ξ and -ξ direction together with the sum over the remaining face normal vectors in η- and ζ-directions must vanish. A decomposition using the face normal vectors leads to a system of differential equations

$$\begin{aligned}
&\mathbf{s}_1^2\left\{\mathbf{x}_{\xi\xi}+\frac{V}{\mathbf{s}_1^2}\left[\left(\frac{\mathbf{s}_1^2}{V}\right)_\xi+\left(\frac{\mathbf{s}_1\mathbf{s}_2}{V}\right)_\eta+\left(\frac{\mathbf{s}_1\mathbf{s}_3}{V}\right)_\zeta\right]\mathbf{x}_\xi\right\}+\\
&\mathbf{s}_2^2\left\{\mathbf{x}_{\eta\eta}+\frac{V}{\mathbf{s}_2^2}\left[\left(\frac{\mathbf{s}_2 s_1}{V}\right)_\xi+\left(\frac{\mathbf{s}_2^2}{V}\right)_\eta+\left(\frac{\mathbf{s}_2\mathbf{s}_3}{V}\right)_\zeta\right]\mathbf{x}_\eta\right\}+\\
&\mathbf{s}_3^2\left\{\mathbf{x}_{\zeta\zeta}+\frac{V}{\mathbf{s}_3^2}\left[\left(\frac{\mathbf{s}_3\mathbf{s}_1}{V}\right)_\xi+\left(\frac{\mathbf{s}_3\mathbf{s}_2}{V}\right)_\mu+\left(\frac{\mathbf{s}_3^2}{V}\right)_\zeta\right]\mathbf{x}_\zeta\right\}+\\
&2\mathbf{s}_1\mathbf{s}_2\mathbf{x}_{\xi\eta}+2\mathbf{s}_2\mathbf{s}_3\mathbf{x}_{\eta\zeta}+2\mathbf{s}_3\mathbf{s}_1\mathbf{x}_{\zeta\xi}=0.
\end{aligned} \tag{1}$$

with the $\mathbf{s}_i$ being the face normal vectors, V the cell volume and the $\mathbf{x}$ represent the physical coordinates along the grid lines in the three computational coordinates ξ, η, and ζ.

$$\mathbf{s}_1=\mathbf{x}_\eta\times\mathbf{x}_\zeta\ ,\quad \mathbf{s}_2=\mathbf{x}_\zeta\times\mathbf{x}_\xi\ ,\quad \mathbf{s}_1=\mathbf{x}_\xi\times\mathbf{x}_\eta \tag{2}$$

This system of equations can easily be separated into the face weighted Laplace equation in computational space and so-called control functions. Prescribing the control functions, the metric identity becomes a mesh generating equation. This is used either for elliptic smoothing or for an advancing layer method through these differential equations parabolized with respect to the marching direction and subsequently discretized[12]. By this the generation of the initial algebraic mesh is not necessary.

B. Smooth Prismatic Boundary Layer Meshes

For unstructured meshes there exists no globally underlying computational space. Therefore, starting from the same basic principle we begin with the metric identity in its discrete form to directly derive from it the set of algebraic equations to be solved numerically. The derivation of the equations has been shown in a previous paper of the author *et al.*[4]. Proceeding as in the structured case we decomposed the face normal vectors into the line vector directions and rewrote the metric identity in terms of line vectors with coefficients consisting of scalar products of face vectors. Compared with the expanded differential form of the identity for structured meshes, which split into the Laplacian terms and the control functions, the corresponding decomposition of the discrete form of the identity yields nothing similar, still leaving the problem to isolate the Laplacian smoothing part from the pure identity. The problem area is, of course, confined to the unstructured mesh in the body conforming surfaces. Decomposing the discrete identity for planar surfaces results in the definition of suited averages, such that the discrete Laplace equation is represented. To verify this we compared them to the discretized form of the Laplace equation for structured quadrangles and hexagonal cells[13]. So our averaging procedure is not proven but only believed to hold for any collection of triangles forming an *N*-cornered cell.

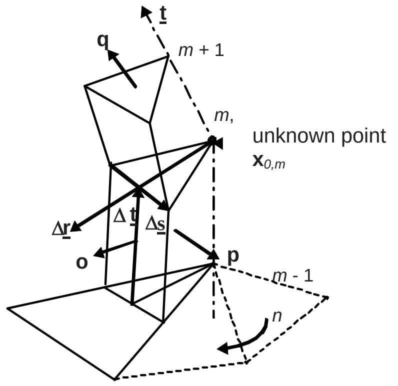


Figure 1. Discrete triangular prism

First we define the edge vectors in radial, circumferential, and marching direction according to Figure 1

$$\Delta\mathbf{r}_{n,m}=1/2\left(\mathbf{x}_{n,m}+\mathbf{x}_{n+1,m}\right)-\mathbf{x}_{0,m},\quad \Delta\mathbf{s}_{n,m}=\mathbf{x}_{n+1,m}-\mathbf{x}_{n,m},\quad \Delta\mathbf{t}_{m+1/2}=\mathbf{x}_{0,m+1}-\mathbf{x}_{0,m} \tag{3}$$

The face vectors are defined similar to the structured case

$$\mathbf{o}=\Delta\mathbf{s}\times\Delta\mathbf{t},\ \mathbf{p}=\Delta\mathbf{t}\times\Delta\mathbf{r},\ \mathbf{q}=\Delta\mathbf{r}\times\Delta\mathbf{s} \tag{4}$$

For the averaging we derived that for obtaining the Laplacian equation the face vectors have to be replaced by averaged face vectors based on averaged edge vectors in radial and circumferential direction

$$\Delta\bar{\mathbf{s}}_n = \frac{2}{N}\sum_{i=1}^{N}\cos\varphi_{n+i-1}\Delta\mathbf{s}_i\,,\ \Delta\bar{\mathbf{r}}_n = \frac{2}{N}\sum_{i=1}^{N}\cos\varphi_{n+i-1}\Delta\mathbf{r}_i\,,\ \varphi_j = \frac{2\pi}{N}\left(j-1\right) \tag{5}$$

Finally we solve equation (6) for the unknown central point $\mathbf{x}_0$ on level *m*:

$$\begin{aligned}
&\mathbf{x}_{o,m}\sum_n\left(\bar{\mathbf{o}}_{n,m}\bar{\mathbf{o}}_{n,m}+\frac{1}{2}\bar{\mathbf{q}}_m\bar{\mathbf{q}}_m\right)= \\
&\sum_n\left[\frac{1}{2}\bar{\mathbf{o}}_{n,m}\bar{\mathbf{o}}_{n,m}\left(\mathbf{x}_{n,m}+\mathbf{x}_{n+1,m}\right)+\bar{\mathbf{o}}_{n,m}\bar{\mathbf{p}}_{n,m}\Delta\mathbf{s}_{n,m}+\frac{1}{4}\bar{\mathbf{q}}_m\underline{\bar{Q}}_m\left(\mathbf{x}_{o,m-1}+\mathbf{x}_{o,m+1}\right)\right. \\
&\left.+\frac{1}{4}\bar{\mathbf{o}}_{n,m}\bar{\mathbf{q}}_{n,m}\left(\Delta\mathbf{r}_{n,m+1/2}-\Delta\mathbf{r}_{n,m-1/2}\right)+\frac{1}{4}\bar{\mathbf{q}}_{n,m}\bar{\mathbf{p}}_{n,m}\left(\Delta\mathbf{s}_{n,m+1/2}-\Delta\mathbf{s}_{n,m-1/2}\right)\right],
\end{aligned} \tag{6}$$

using the average of the edge vectors of two adjacent layers

$$\Delta\mathbf{s}_{n,m+1/2} = 1/2\left(\Delta\mathbf{s}_{n,m}+\Delta\mathbf{s}_{n,m+1}\right),\quad \Delta\mathbf{r}_{n,m+1/2} = 1/2\left(\Delta\mathbf{r}_{n,m}+\Delta\mathbf{r}_{n,m+1}\right) \tag{7}$$

C. Intermediate layer to connect to volume triangulation

In order to be able to triangulate the volume the outer hull of the hexahedrons of the structured mesh must be converted to a triangulation. First attempts to generate pyramidal elements on top of the hexahedrons led to intersections in concave regions that were only removed by using very flat pyramidal cells.

The way chosen now is a division of the outermost hexahedrons into pyramids and tetrahedrons using the center point. By this method the interface surface grid formed by the outer layer is as smooth as the hexahedral layer itself.

Figure 2 shows the sequence of dividing the outer layer first into pyramids, afterwards into tetrahedrons to form a smooth outer hull for the volume triangulation.

D. Metric representations of interface to volume triangulation

The purpose is to adopt a volume triangulation in a way that the cells at the boundaries match the size and the orientation of the underlying hexahedral and prismatic elements. The usually high stretching of the hexahedral cells requires the use of an anisotropic triangulation method. For anisotropic meshing we need to store the information in terms of tensors. The metrics can be stored either as a tensor describing the ellipsoids of density prescription or as transformation tensors describing the transformation into the metric space. We prefer to use the storage of the metric itself. The metric is then a symmetric positive tensor which can be decomposed into the eigenvector matrix **E** and eigenvalues λ_i according to

$$\mathbf{M} = \mathbf{E}\cdot diag\left(\lambda_i\right)\cdot\mathbf{E}^T \tag{8}$$

The duality to the transformation **T** into metric space is given by

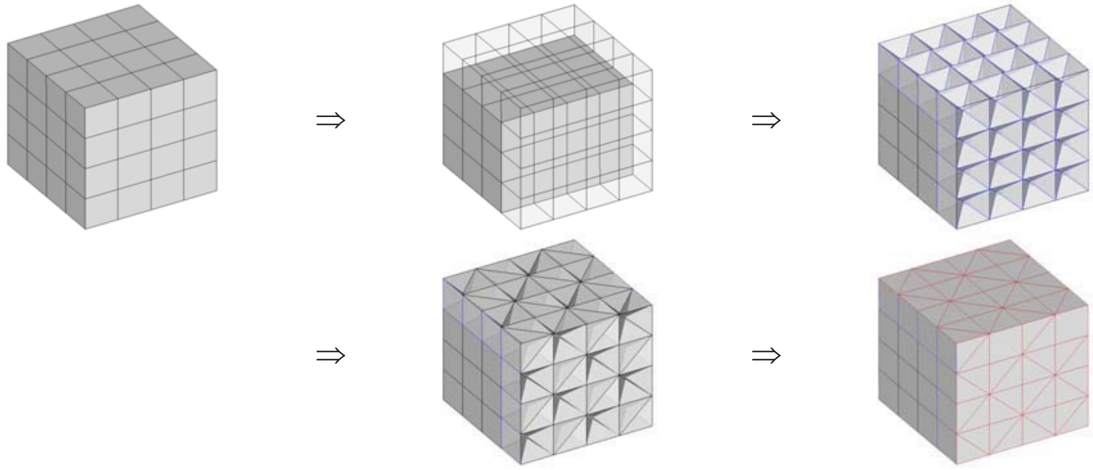


Figure 2. Division of outermost structured hexahedrons into pyramids and tetrahedrons to achieve a smooth interface surface for tetrahedral mesh generation

$$\mathbf{T} = diag\left(\sqrt{\lambda_i}\right)\mathbf{E}^T \tag{9}$$

The metrics resulting from the surrounding cells, including the block-structured hexahedral cells must be specified. There are three ways to derive space metrics from the surrounding elements

1. *Hexahedral metrics*

For the attachment to hexahedral elements we use the principal axis of the hexahedron for the specification of the metrics. For this purpose we form the non-orthogonal basis $\mathbf{B}_H$ by averaging the edge vectors along the i, j, and k-directions in structured notation.

$$\mathbf{B}_H = \left[\mathbf{e}_i \quad \mathbf{e}_j \quad \mathbf{e}_k\right]; \quad \begin{aligned} \mathbf{e}_i &= \frac{1}{4}\sum_{j=1}^{2}\sum_{k=1}^{2}\left(\mathbf{x}_{i+1,j,k} - \mathbf{x}_{i,j,k}\right) \\ \mathbf{e}_j &= \frac{1}{4}\sum_{i=1}^{2}\sum_{k=1}^{2}\left(\mathbf{x}_{i,j+1,k} - \mathbf{x}_{i,j,k}\right) \\ \mathbf{e}_k &= \frac{1}{4}\sum_{i=1}^{2}\sum_{j=1}^{2}\left(\mathbf{x}_{i,j,k+1} - \mathbf{x}_{i,j,k}\right) \end{aligned} \tag{10}$$

The corresponding symmetric metrics is then simply obtained by

$$\mathbf{M}_H^{-1} = \mathbf{B}_H\mathbf{B}_H^T$$

2. *Affine invariant*

The affine variant metric[14] describes the least square fit of an ellipse in 2D, or an ellipsoid in 3D, through a given set of points, centred at the centre point of the point set. The affine invariant metric is well defined if the number of non-collinear, resp. non-coplanar, points is larger than $2N$+1.

$$\begin{aligned} \mathbf{M}_{A_{ij}}^{-1} &= \frac{1}{n}\sum_{k=1}^{n}\left(\mathbf{x}_{ik} - \mathbf{x}_{ic}\right)\left(\mathbf{x}_{jk} - \mathbf{x}_{jc}\right) \\ \mathbf{x}_{ic} &= \frac{1}{n}\sum_{k=1}^{n}\mathbf{x}_{ik} \end{aligned} , \tag{11}$$

where N is the dimension of the space. This method is suitable for prismatic or pyramidal elements.

3. *Riemannian Invariants*

In case of neighbouring cells being tetrahedrons, the metric is derived by using the Riemannian invariant metric of the neighbouring cell. The Riemannian invariant metric is the metric that transforms an arbitrary triangle in 2D or a tetrahedron 3D in an ideal one with unity edge length (Figure 3). This leads to solving a system of N! equations of the form

$$\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \mathbf{M}_R \left(\mathbf{x}_i - \mathbf{x}_j\right) = \mathbf{1}, \quad i < j; \quad i, j = 1\ldots N+1, \tag{12}$$

where again N is the space dimension.

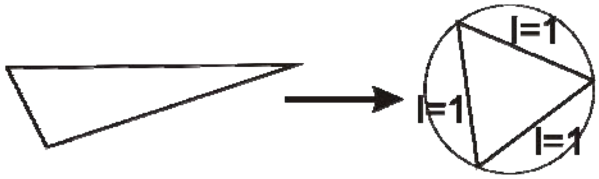


Figure 3. Definition of Riemannian invariant transformation

E. Preconditioning of Volume Triangulation

In general, volume triangulation methods, even if they are formulated as anisotropic, suffer difficulties if the aspect ratio on the boundary is too high, which usually results in a high computational effort in terms of both time and memory. In order to reduce this effort we use a geometric preconditioning of the surface mesh for the triangulation algorithm. We call it preconditioning since it is a global transformation of the initial boundary triangulation in order to make reduce the anisotropy for the volume triangulation.

A first transformation is a shear transformation, which is important for the application to swept high aspect ratio wings of transport aircraft.

$$\mathbf{P}_1 = \mathbf{S}^{-1}\mathbf{P} \tag{13}$$

where **P** is the set of boundary points and ***S*** is the shear transformation for the wing sweep. Assuming the spanwise coordinate to be *y* the shear transformation is given by

$$\mathbf{S} = \begin{bmatrix} 1 & \sin\varphi & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{14}$$

where φ is the wing sweep angle. This transformation shears the surface elements and results in an intermediate mesh where the shear has been removed from the surface cells close to the wing, making them more rectangular.

The second item to deal with is the high aspect ratio of the surface triangles. We apply an edge-weighted mean affine invariant transformation. For each surface triangle an affine invariant transformation

$$\tilde{\mathbf{p}}_\mathbf{i} = \mathbf{A}^{-1}\mathbf{p}_\mathbf{i} \tag{15}$$

is defined by a matrix **A** reflecting the anisotropy of the surface element

$$A_{ij} = \frac{1}{n}\sum_{k=1}^{n}\left(\mathbf{x}_{k,i} - \mathbf{x}_{c,i}\right)\left(\mathbf{x}_{k,j} - \mathbf{x}_{c,j}\right) \tag{16}$$

where *n* is the number of points of the surface element and $\mathbf{x}_c$ is the center point. In fact the affine invariant is the quadratic form of an ellipsoid passing through the points of the surface element. By this the matrix reflects the size and stretching of the element. We now calculate a mean edge-weighted average of these matrices of all surface elements.

$$\overline{\mathbf{A}} = \sum_{i=1}^{m} \frac{n}{\sum_{j=1}^{n}\left(\mathbf{x}_j - \mathbf{x}_c\right)^2}\mathbf{A}_\mathbf{i} \tag{17}$$

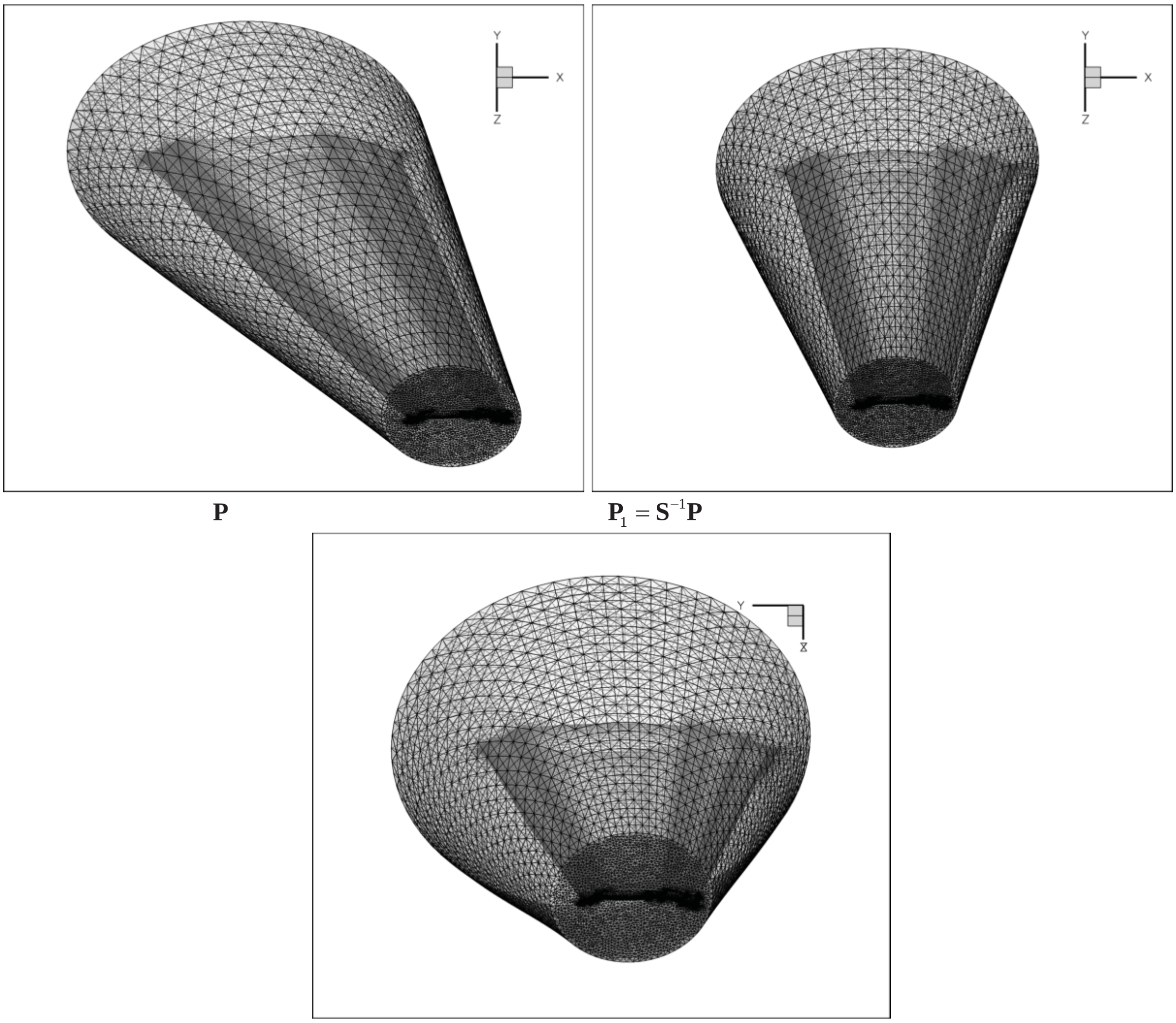
The advantage of the edge weighting is that the large regular triangles at the outer domain boundary don’t mask the high anisotropy of the small surface elements at the interface. The final point array used for the Delaunay triangulation is now

$$\overline{\mathbf{P}} = \overline{\mathbf{A}}^{-1}\mathbf{S}^{-1}\mathbf{P} \tag{18}$$

In Figure 4 the meshing box around the outboard portion of a transport aircraft high-lift wing is shown for all steps of the preconditioning. The points-of-view are identical excluding a necessary swapping of the axes due to the affine invariant transformation where the sorting of the axes is according to the eigenvalues.

F. Anisotropic Volume Triangulation

For the volume triangulation of the preconditioned boundary grid we use the SIMMETRIX[15] MeshSim suite v7.1, which allows for anisotropic volume meshing. The mesh generation follows an octree approach and is able to adjust to pre-specified anisotropic metrics at the boundaries of the meshing domain. The mesh generation software provided by SIMMETRIX has been linked into the MegaCads grid generator.



P

$\mathbf{P}_1 = \mathbf{S}^{-1}\mathbf{P}$

$\overline{\mathbf{P}} = \overline{\mathbf{A}}^{-1}\mathbf{S}^{-1}\mathbf{P}$

Figure 4. Meshing box around outboard high-lift wing: top left) original geometry; top right) shear transformation applied; bottom) shear and edge-weighted affine invariant transformation applied.

III. Applications of Mixed Meshes to the ONERA M6 wing

The aerodynamic configuration where we demonstrate the capabilities of the chosen mixed mesh approach is the well known ONERA M6 wing where measurements at transonic conditions are available for a range of angles of attack[16]. In order to assess the different ingredients of the meshing procedure three different meshes are generated and used in the calculations. In all cases a special a priori knowledge of shock positions was neglected to pronounce more the differences in the different approaches. First, as a reference, a hybrid mesh consisting only of prismatic and tetrahedral elements was generated using the available software CENTAUR[17] (Figure 5). This represents the state of the art of mesh generation used at our institute and generates high-quality, but isotropic meshes. The second mesh uses the mixed mesh approach without passing anisotropic into the tetrahedral space (Figure 6). Finally, the third grid uses the complete approach including the generation of anisotropic tetrahedrons using the metric information from hexahedral and prismatic elements at the interfaces (Figure 7).

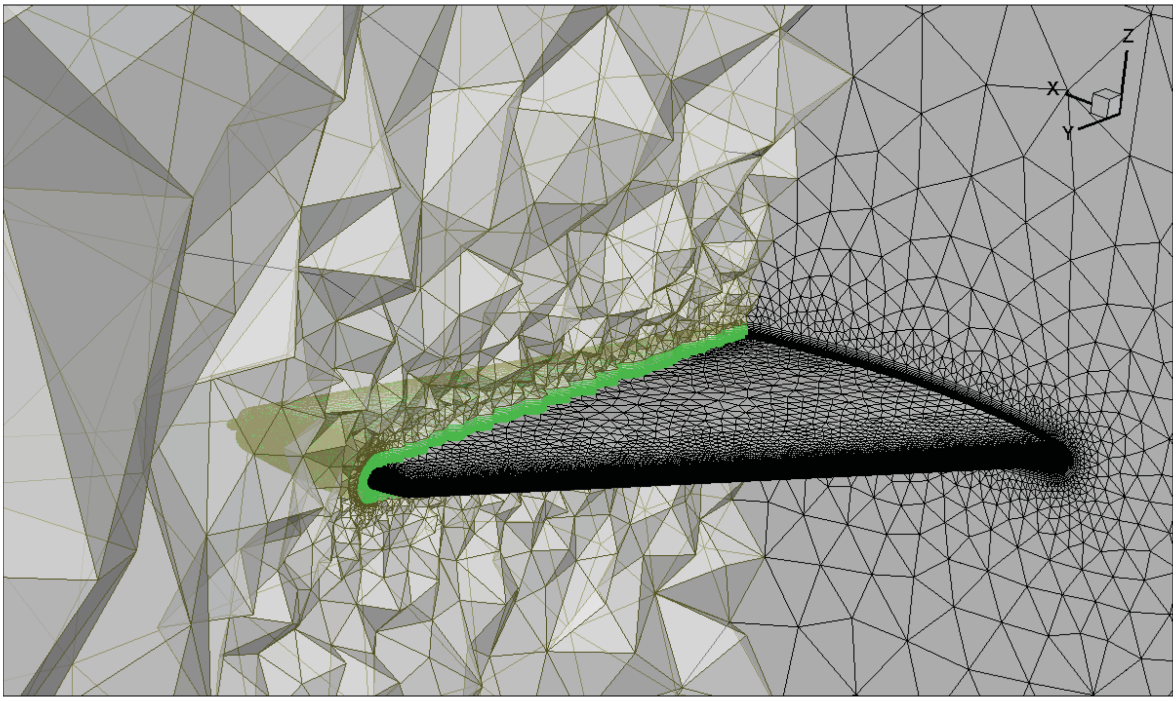


Figure 5. Reference hybrid mesh around the ONERA M6 wing (O-type boundary layer grid); prisms: green; tetrahedrons: red.

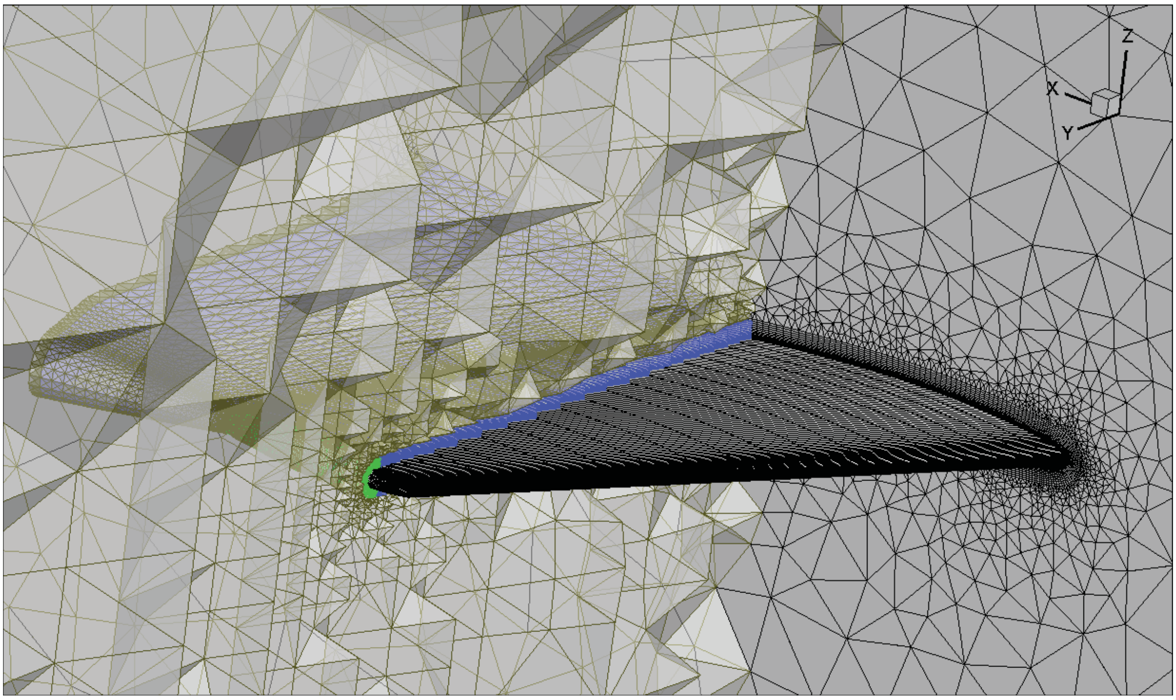


Figure 6. Application of mixed mesh method with isotropic tetrahedrons to the ONERA M6 wing (C-type boundary layer and wake grid); hexahedrons: blue; prisms: green; tetrahedrons: red.

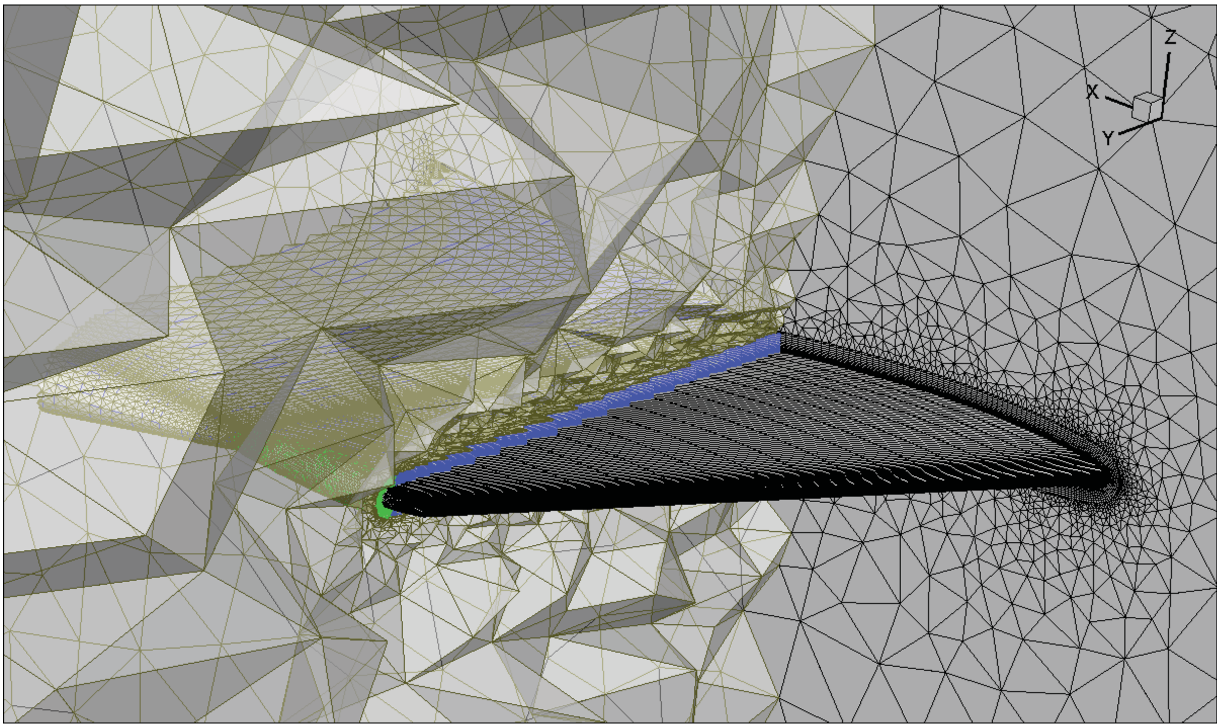


Figure 7. Application of mixed mesh method with anisotropic tetrahedrons to the ONERA M6 wing (C-type boundary layer and wake grid); hexahedrons: blue; prisms: green; tetrahedrons: red.

The comparison in this work concentrates on the benefits of using structured high-y stretched hexahedral elements for shear layer resolution. The surface resolution has been chosen similar to what has been used in structured or unstructured meshing practice. The wall normal resolution is the same for all meshes with 32 structured/prismatic layers and a first wall spacing for obtaining dimensionless wall distances around y^+=1. Due to the different methodology the mixed meshes allow for a resolution of the wing wake with C-type hexahedral grids while the hybrid approach is only capable for resolving this region with an O-type topology. No emphasis has been put on resolving possible shock positions since this would represent an a priori knowledge of the flow. Additionally a possible weak shock capturing can pronounce the differences in the different grid structures.

The sizes of the meshes are reported in Table 1. The mixed meshes both only have slightly more than 20% of the points used in the hybrid grid making use of a high stretching in span direction. The adoption of the tetrahedrons to the metrics of the structured elements itself generates an only small overhead of about 2.5% in terms of points and about 12% in terms of tetrahedrons.

Table 1. Mesh sizes and characteristics of the meshes used for the ONERA M6 wing calculations

	points	surface elements	hexahedrons	tetrahedrons	prisms	pyramids
mixed-anisotropic	456212	32826	191828	467445	316352	41752
mixed-isotropic	445310	32826	191828	416421	316352	41752
hybrid	1916138	113702	-	974515	3443584	-

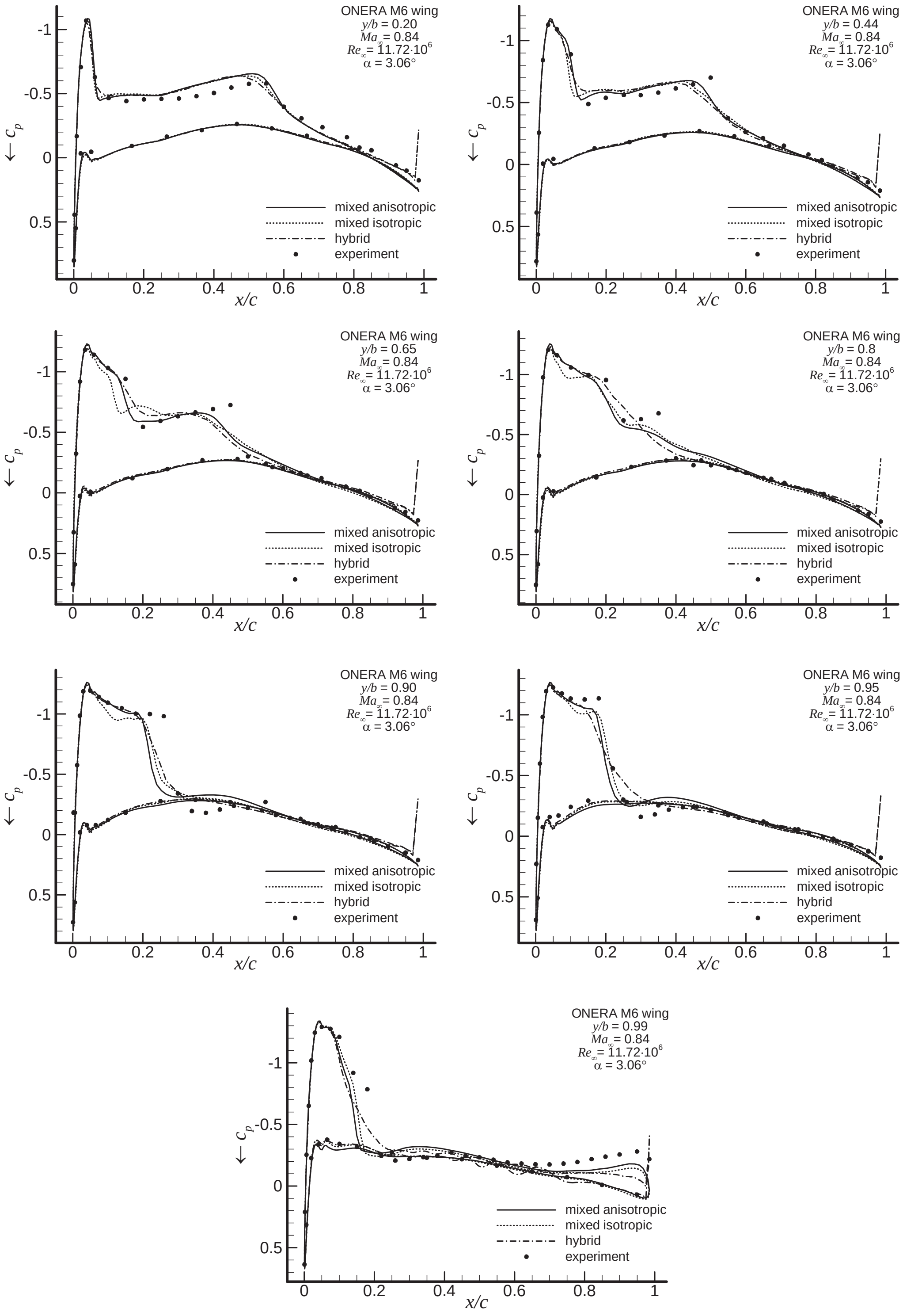
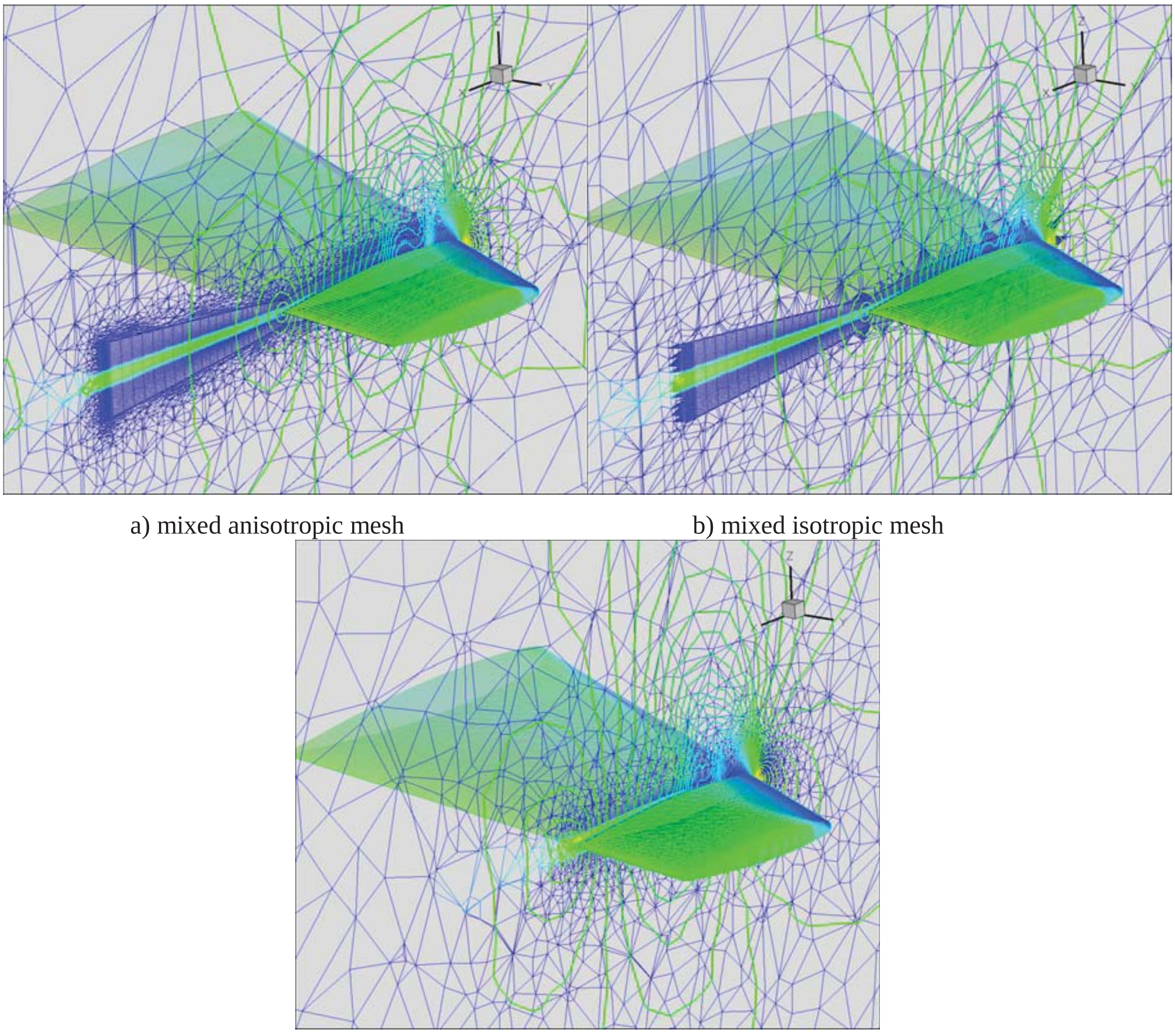


Figure 8. Comparison of the pressure distributions obtained on the three different grids to experimental data for the ONERA M6 wing at an angle of attack α=3°

CFD calculations have been performed using the DLR TAU code, which is an unstructured finite volume RANS solver second order in time and space. The unsteady equations are integrated in time by an explicit Runge-Kutta time stepping until a steady state is reached. Convergence is accelerated by applying multigrid and local time stepping. For turbulence modeling the one-equation model of Spalart and Allmaras is used. Calculations have been performed for transonic conditions M=0.84 at a Reynolds number of Re=11.7x10^6, at two distinct angles of attack of the experimental data base, α=3° and α=6°.

Figure 8 shows a comparison of the computed pressure distributions at the 7 wing sections measured in the experiments for the first angle of attack. The directional alignment of the hexahedral cells results in a better shock resolution although the number of surface elements is less than for the hybrid grid. The usage of C-type grids for the wake resolution prevents the occurrence of fish-tails, artificial pressure peaks occurring in O-type grids due to the severe shearing of the prismatic elements. The necessity of the adoption of the tetrahedrons to the structured layer is seen by the occurrence of some pressure variations mainly within the supersonic area. For the last section very close



a) mixed anisotropic mesh

b) mixed isotropic mesh

c) hybrid mesh

Figure 9. Comparison of the computed flow fields around the ONERA M6 wing at an angle of attack α=3°; contours on surface: pressure; contour lines on planar slice: local Mach number; mesh in slice plane colored by total pressure loss.

to the tip the hybrid mesh doesn't capture the pressure distribution and seems unable to capture the wing tip vortex properly, in contrast to the mixed meshes. Figure 9 shows the computed flow fields by pressure contours on the wing, and local Mach number and total pressure loss contours in a planar slice through the mesh. The structured elements in the C-type grid much better capture the wake than the O-type structure obtained by hybrid meshing. The O-type grid diffuses the wake in short distance after the wing surface, while for the C-type grids the wake is resolved up to the end of the structured grid area. The deficiency using isotropic tetrahedral meshing in conjunction with stretched hexahedrons is a result of the supersonic area leaving the region of structured cells and by this being immediately diffused or dissipated.

Pressure distributions for the second angle of attack are shown in Figure 10. The tendencies reported above are more severe comparing the hybrid to the mixed meshes while the differences between the two mixed meshes are less pronounced but still existent. For this case the hybrid mesh solution shows for the outermost 20% of the wing span a shock induced separation that is neither predicted by the calculations on the mixed meshes nor visible form the experiments. It can be assumed that this is a result of the deficit in resolving the wing vortex and wake flow properly. For the mixed mesh computations the resolution of the reacceleration due to the wing tip vortex is in remarkable good agreement with the experiments. A better prediction using a hybrid mesh would only be possible with an even higher number of mesh points in this area.

A final remark should be made concerning the computational efficiency. In some cases different types of mesh elements or non-smooth meshes can affect the convergence of the CFD method. Figure 11 shows the convergences of the density residual and the lift and drag coefficients for the calculations at α=3°. It is observed that the differences of the mesh types have no impact on the convergence. In other words, point savings directly scale into savings of computational time and memory effort.

IV. Conclusion

A method for generating smooth mixed meshes for aerospace applications was outlined. Special target was to provide meshes with a small number of mesh points but giving the chance of obtaining accurate results by CFD computations, with the major focus of reducing the computational effort and enabling CFD for design optimization. Key ingredients for high quality meshes for viscous flows are elliptic smooth structured hexahedral and prismatic meshes for the resolution of shear layers at boundaries or in wake regions thus enabling C-type meshes in unstructured meshing. The special problem of matching the highly stretched hexahedral elements with the outer tetrahedrons have been tackled first by a preconditioning transformation to the mesh boundary and a proper derivation of the metrics of surrounding elements to be fed into an anisotropic tetrahedral mesh generation method. The benefits of the proposed mesh type have been demonstrated by comparing calculations on different meshes around the ONERA M6 wing at two different angles of attack. The mixed mesh approach was able to provide results of improved accuracy compared to classical hybrid mesh approached by only using 20% of the grid points. Additional superior behavior was found in the resolution of the wing wake and in the response on the wing tip vortex. The adjustment of the tetrahedrons to the metrics of the underlying elements give an improved accuracy for an only minor overhead of 2.5% more points in the grid. Analysis of the convergence showed that the point savings scale directly into savings of computational time and memory effort.

Acknowledgments

The author wants to thank Peter Niederdrenk, who has developed and implemented the theory of the face weighted Laplacian equations based on metric identity. He also is the father of the parabolic marching procedures the author uses throughout the mesh generation strategy.

The author also wants to thank Carl F. Ollivier-Gooch for his guidance during the work on volume triangulation methods and their extension to anisotropic meshing.

A last thank is towards the support group of SIMMETRIX in order to adapt the MeshSim code towards the very hard requirements for our type of meshes.

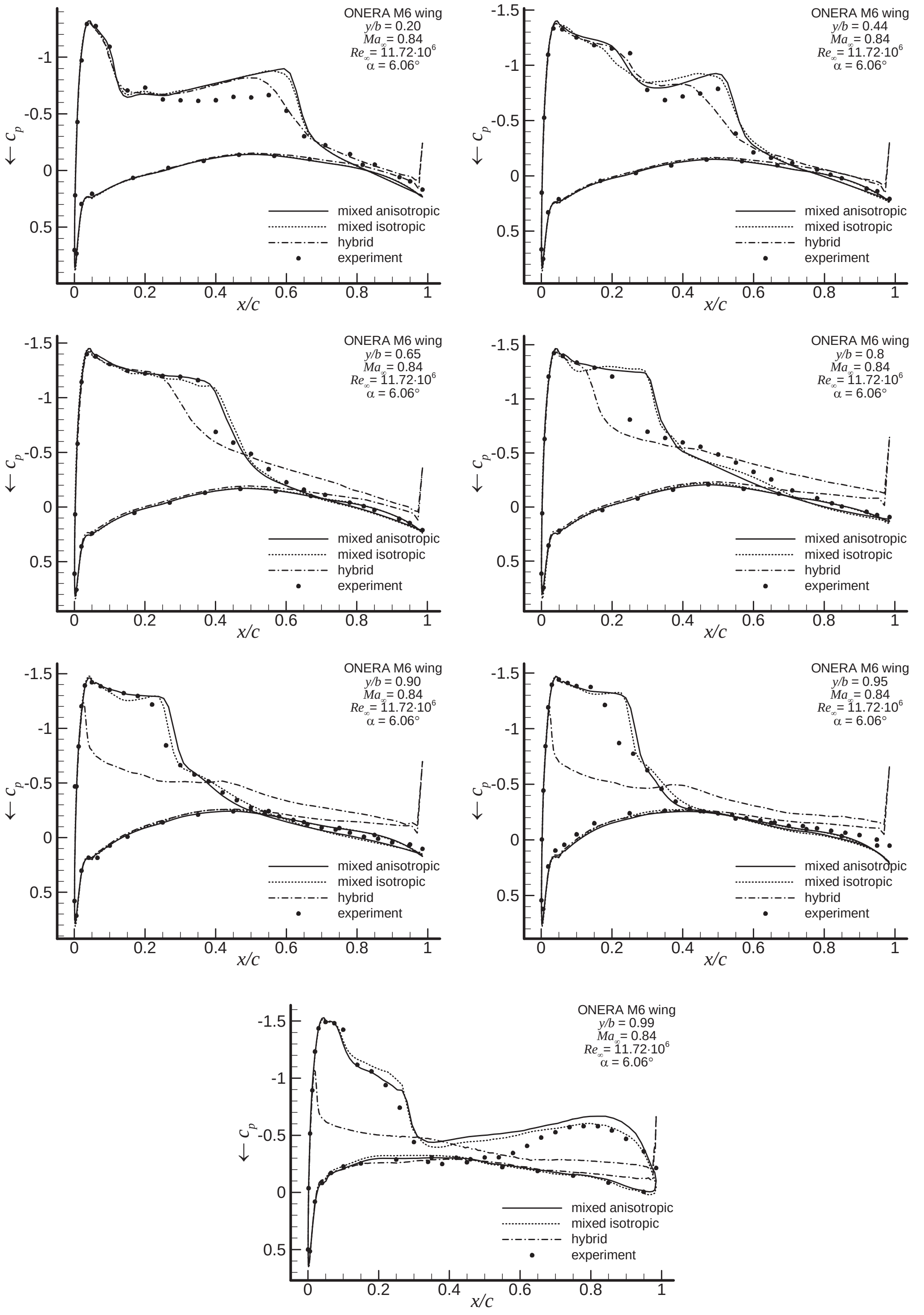


Figure 10. Comparison of the pressure distributions obtained on the three different grids to experimental data for the ONERA M6 wing at an angle of attack *α*=6°

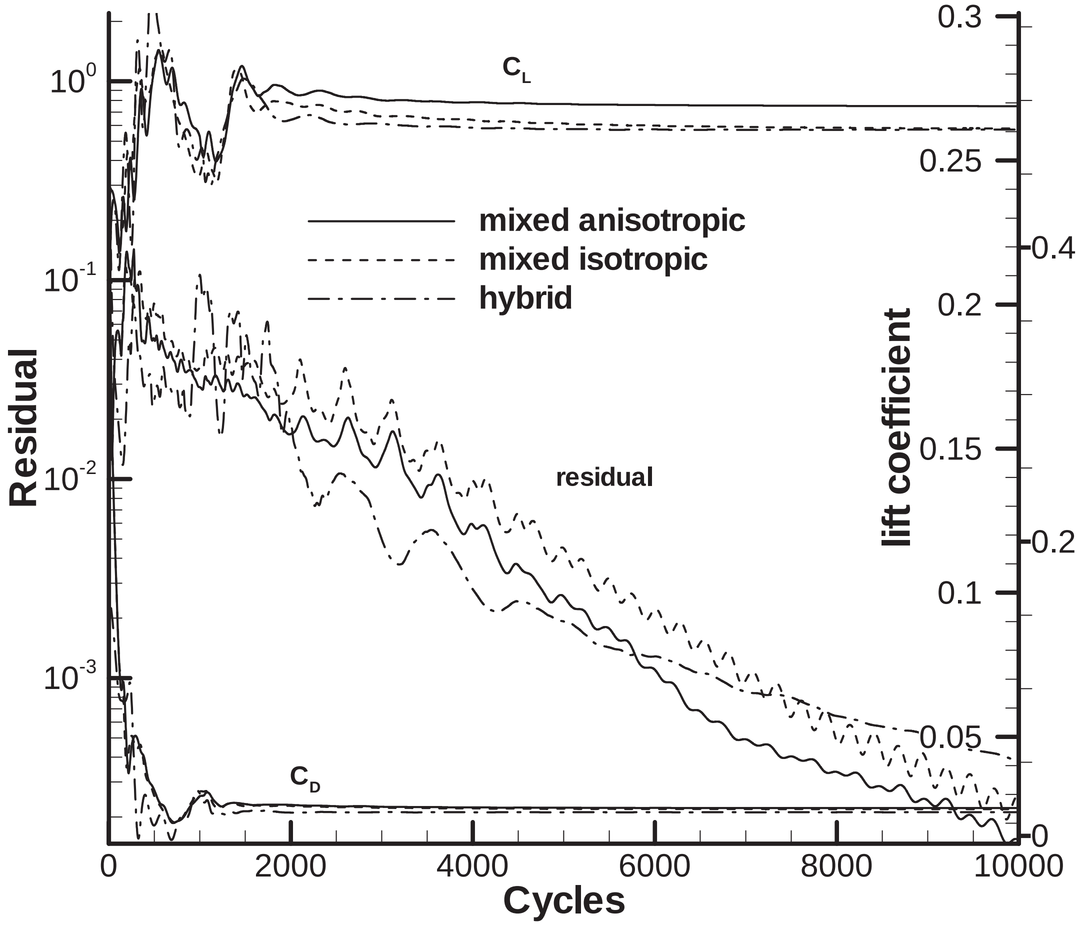


Figure 11. Comparison of the convergence rates of lift, drag and density residual of the computations at α=3° for the three different meshes

References

[1] D. Levy, T. Zickuhr, J. Vassberg, S. Agrawal, R. Wahls, S. Pirzadeh, M. Hemsch, “Data summary from the first AIAA computational fluid dynamics drag prediction workshop”, *Journal of Aircraft*, 2003, Vol. 40, no. 5, pp. 875-882.

[2] O. Brodersen, M. Rakowitz, S. Amant, P. Larrieu, D. Destarac, M. Sutcliffe, “Airbus, ONERA and DLR results from the 2nd AIAA drag prediction workshop”, AIAA Paper 2004-0391, 2004.

[3] R. Rudnik, R. Heinrich, B. Eisfeld, T. Schwarz, DLR contributions to code validation activities within the European high lift project EUROLIFT”, *New Results in Numerical and Experimental Fluid Mechanics IV*, Vol. 87 of Notes on Numerical Fluid Mechanics, 2004, eds. C. Breitsamer *et al.*, Springer-Verlag, Berlin Heidelberg, pp. 42-49.

[4] J. Wild, P. Niederdrenk, T. Gerhold: “Marching Generation of Smooth Structured and Hybrid Meshes Based on Metric Identity”,: *Proceedings of the 14th Int. Meshing Roundtable*, San Diego, CA, 2005, ed. B.W. Hanks, Springer, Berlin, pp. 109 – 128.

[5] J. Wild: “Application Of Smooth Mixed Meshes Based On Metric Identity In Aerospace Analysis And Design”,: *Proceedings of the 17th Int. Meshing Roundtable*, Pittsburgh, PA, 2008, ed. R.V. Grimella, Springer, Berlin, pp. 387 – 398.

[6] O. Brodersen, M. Hepperle, A. Ronzheimer, C.-C. Rossow, B. Schöning, “The Parametric Grid Generation System MegaCads”, *Proc. 5th Intern. Conf. On Numerical Grid Generation in Comp. Field Simulation*, National Science Foundation (NSF), 1996, pp. 353-362.

[7] J.R. Shewchuk, “Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator”, *First Workshop on Applied Computational Geometry*, ACM, 1996.

[8] J.D. Müller, P.L. Roe, H. Deconinck, “A frontal approach for internal node generation in Delaunay triangulations”, *International Journal for Numerical Methods in Fluids*, 1993, Vol. 17, no. 3, pp. 241-255.

[9] J. Schöberl, “NETGEN - An advancing front 2D/3D mesh generator based on abstract rules”, *Computing and Visualization in Science*, 1997, Vol. 1, pp. 41-52.

[10] L.E. Eriksson, “Generation of boundary-conforming grids around wing-body configurations using transfinite interpolation”, *AIAA Journal*, 1982, Vol. 20, no. 10, pp.1313–1320.

[11] P. Niederdrenk, “On the Control of Elliptic Grid Generation”, *Proc. 6th Intern. Conf. On Numerical Grid Generation in Comp. Field Simulations*, London, July 1998, eds. M. Cross et al., NSF Eng. Center, Mississippi, pp. 257

[12] P. Niederdrenk, O. Brodersen, “Controlled Parabolic Marching Grid Generation”, *Proc. 7th Intern. Conf. On* Numerical Grid Generation in Comp. Field Simulations, Whistler, British Columbia, Canada, Sept. 2000, eds. B. Soni et al., NSF Eng. Center, Mississippi, pp. 29

[13] A. Winslow, “Numerical Solution of the Quasi-Linear Poisson Equations in a Non-Uniform Triangle Mesh”, *J. Comp. Phys.*, 1967, Vol.2, pp. 149

[14] J. Hoschek, and D. Lasser, *Grundlagen der geometrischen Datenverarbeitung*, 2nd edition, Teubner, Stuttgart (1992).

[15] Simmetrix Inc., http://www.simmetric.com (accessed 2010)

[16] V. Schmidt and F. Charpin, Pressure Distributions on the ONERA M6 Wing at Transonic Mach Numbers, *Experimental data base for computer program assessment*, AGARD AR-138 (1979), B1.

[17] CentaurSoft, http://www.centaursoft.com (accessed 2010)