

# MARCHING GENERATION OF SMOOTH STRUCTURED AND HYBRID MESHES BASED ON METRIC IDENTITY

Jochen Wild<sup>1</sup>, Peter Niederdrenk<sup>2</sup>, Thomas Gerhold<sup>2</sup>

<sup>1</sup>DLR, Institute for Aerodynamics and Flow Technology, Lilienthalplatz 7, D-38108 Braunschweig, Germany

<sup>2</sup>DLR, Institute for Aerodynamics and Flow Technology, Bunsenstr.10, D-37073 Göttingen, Germany

## ABSTRACT

Elliptic differential equations are derived for the generation of structured meshes and difference equations for the generation of smooth hybrid meshes from metric identity. A parabolic procedure is used to march the solution of the difference equations simultaneously for both types of meshes away from surface patches meshed by quads or triangles. An aerodynamic application for the ONERA M6 wing demonstrates how the blocking at the wing tip is simplified by using both types of meshes instead of a purely structured mesh. On the other hand it is shown that a large number of points can be saved compared to a purely hybrid grid.

Keywords: mesh generation, structured, hybrid, elliptic

## 1. INTRODUCTION

As long as the variations of the field variables to be resolved by the numerical solution to a set of differential equations are of the same order of magnitude in all directions the use of unstructured meshes is appropriate 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 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 unstructured surface meshes.

Since our objective is to generate smooth meshes we will start from a set of elliptic equations. In a previous paper [1] 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. To solve these differential equations numerically they were parabolized with respect to the marching direction and subsequently discretized [2].

For unstructured meshes there exists no globally underlying computational space. Therefore, starting from the same basic principle means that we will begin with the metric identity in its discrete form to directly derive from it the set of algebraic equations to be solved numerically.

In the 80s marching procedures were used with the primary intention to speed up the generation process of three-dimensional structured meshes. We have added a flexible control of grid line spacing and orthogonality with no need for any initial algebraic or reference mesh [2]. With respect to the generation of hybrid meshes the majority of the publications are based on pure algebraic procedures. An exception is the paper of David Thompson [3], which makes use of a local two-dimensional expansion of the elliptic mesh generation equation originating from Patrick Knupp's work [4] on smoothing triangular surface meshes.

## 2. DIFFERENTIAL FORM OF METRIC IDENTITY

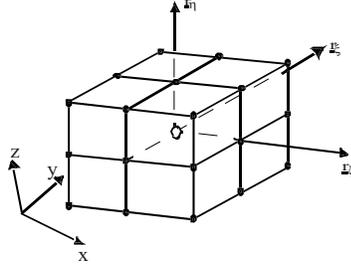


Fig. 1: Master cell

Dirichlet conditions on the boundary specify the solution of an elliptic equation. Figure 1 shows a structured master cell with 26 known mesh points on the cell boundary and just one unknown interior point. The position vector  $\underline{r}$  in physical space  $x, y, z$  is a function of the computational space variables  $\xi, \eta, \zeta$ . The normal onto a grid face  $\xi = \text{const}$  is given by the cross product of the vectors along the  $\eta$ - and  $\zeta$ -lines:  $\underline{r}_\eta \times \underline{r}_\zeta$ . The metric

identity in its differential form simply states that the sum over the face normal vectors in  $+\xi$  and  $-\xi$  direction together with the sum over the remaining face normal vectors in  $\eta$ - and  $\zeta$ -directions must vanish:

$$(\underline{r}_\eta \times \underline{r}_\zeta)_\xi + (\underline{r}_\zeta \times \underline{r}_\xi)_\eta + (\underline{r}_\xi \times \underline{r}_\eta)_\zeta = 0. \quad (1)$$

Defining normal face vectors as

$$\underline{S}_1 = \underline{r}_\eta \times \underline{r}_\zeta, \quad \underline{S}_2 = \underline{r}_\zeta \times \underline{r}_\xi, \quad \underline{S}_3 = \underline{r}_\xi \times \underline{r}_\eta \quad (2)$$

we decompose the contravariant vectors  $\underline{S}_i$  into the covariant base vectors  $\underline{r}_\xi, \underline{r}_\eta, \underline{r}_\zeta$  as shown below for  $\underline{S}_1$

$$\left[ \frac{1}{V} (\underline{S}_1^2 \underline{r}_\xi + \underline{S}_2 \underline{S}_1 \underline{r}_\eta + \underline{S}_3 \underline{S}_1 \underline{r}_\zeta) \right]_\xi + \dots = 0 \quad (3)$$

and expand the terms in brackets in order to cast the conservative form of the identity into a non-conservative equation for the position vector  $\underline{r}$ :

$$\begin{aligned} & \underline{S}_1^2 \left\{ \underline{r}_{\xi\xi} + \frac{V}{\underline{S}_1^2} \left[ \left( \frac{\underline{S}_1^2}{V} \right)_\xi + \left( \frac{\underline{S}_1 \underline{S}_2}{V} \right)_\eta + \left( \frac{\underline{S}_1 \underline{S}_3}{V} \right)_\zeta \right] \underline{r}_\xi \right\} + \\ & \underline{S}_2^2 \left\{ \underline{r}_{\eta\eta} + \frac{V}{\underline{S}_2^2} \left[ \left( \frac{\underline{S}_2 \underline{S}_1}{V} \right)_\xi + \left( \frac{\underline{S}_2^2}{V} \right)_\eta + \left( \frac{\underline{S}_2 \underline{S}_3}{V} \right)_\zeta \right] \underline{r}_\eta \right\} + \\ & \underline{S}_3^2 \left\{ \underline{r}_{\zeta\zeta} + \frac{V}{\underline{S}_3^2} \left[ \left( \frac{\underline{S}_3 \underline{S}_1}{V} \right)_\xi + \left( \frac{\underline{S}_3 \underline{S}_2}{V} \right)_\eta + \left( \frac{\underline{S}_3^2}{V} \right)_\zeta \right] \underline{r}_\zeta \right\} + \\ & 2\underline{S}_1 \underline{S}_2 \underline{r}_{\xi\eta} + 2\underline{S}_2 \underline{S}_3 \underline{r}_{\eta\zeta} + 2\underline{S}_3 \underline{S}_1 \underline{r}_{\zeta\xi} = 0. \end{aligned} \quad (4)$$

Terms in square brackets represent the so-called control functions.  $V$  is the cell volume. Only when we prescribe these control functions the metric identity becomes a mesh generating equation. The most simple prescription, setting all control functions to zero, yields the face weighted Laplace equation in computational space corresponding to the linear Laplace equation for the mesh faces  $\xi, \eta, \zeta$  in physical space, when we interchange dependent and independent variables.

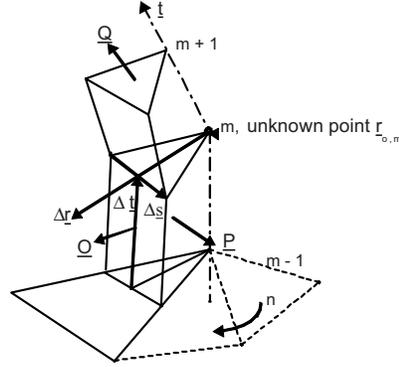


Fig. 2: Discrete triangular prism

$$\nabla_{x,y,z}^2 \underline{\xi} = 0, \quad \underline{\xi} = \{\xi, \eta, \zeta\} \quad (5)$$

### 3. PARABOLIC STRUCTURED MESH GENERATION

In order to parabolize equation (4) we choose  $\zeta$  as the independent variable in marching direction and split the line-wise second derivative of  $r$  with respect to  $\zeta$  into a difference of first order up- and downwind derivatives:

$$\underline{r}_{\zeta\zeta} = \underline{r}_{\zeta}^d - \underline{r}_{\zeta}^u. \quad (6)$$

Treating  $\underline{r}_{\zeta}^d$  as a source term yet to be specified and approximating the second line-wise derivatives by central differences we obtain

$$\begin{aligned} \underline{r}_{i,j,k} = & \left\{ \underline{S}_1^2 \cdot (\underline{r}_{i-1,j,k} + \underline{r}_{i+1,j,k}) + \dots \right. \\ & \left. + \underline{S}_3^2 \cdot (\underline{r}_{i,j,k-1} + \underline{r}_{\zeta}^d) + \dots \right\} / \{ 2\underline{S}_1^2 + 2\underline{S}_2^2 + 2\underline{S}_3^2 \} \end{aligned} \quad (7)$$

On each face  $k = const$  this equation will be solved iteratively by some relaxation scheme. The source term  $\underline{r}_{\zeta}^d$  is specified in terms of an outward spacing times a unit vector. Near body surfaces, e.g. across boundary layers, the preferred outward direction is usually the solution dependent local

normal to faces  $k = const$ . While marching in strictly local normal direction out of concave surfaces will inevitably lead to a crossing of grid lines, it is the ellipticity of our equation in the lateral directions, which usually provides sufficient dissipation to prevent the grid from folding. Such inherent dissipation scales with  $\underline{S}_1^2/\underline{S}_3^2$  and  $\underline{S}_2^2/\underline{S}_3^2$  in  $i$ - and  $j$ -direction, respectively, which - under the assumption of complete orthogonality - is just the square of the scaling factors proposed by Steger and Chan [5].

For further details, especially for the controls in the lateral and marching directions, we refer the reader to [2].

#### 4. METRIC IDENTITY FOR HYBRID MESH GENERATION

Since there is no globally underlying computational space for unstructured meshes, we start from the metric identity in its discrete form. As shown in figure 2 let us consider locally a closed collection of  $n$  triangles around a common point on three levels  $m-1, m, m+1$ .

As in the structured case we assume all points on the cell's bounding faces to be known and thus encounter a local elliptic problem for just one unknown central point. Defining normal vectors to the prism's faces,  $\underline{Q}$  in lateral,  $\underline{P}$  in circumferential and  $\underline{Q}$  in marching direction, respectively, the metric identity simply states again that the sum over all outward pointing face vectors must vanish:

$$\sum \left[ \underline{Q}_{n,m} + \frac{1}{4} \underline{Q}_{n,m+1} - \frac{1}{4} \underline{Q}_{n,m-1} \right] = \mathbf{0}. \quad (8)$$

The face vectors  $\underline{Q}$ ,  $\underline{P}$ ,  $\underline{Q}$  are defined via cross products of the line vectors  $\underline{\Delta r}$ ,  $\underline{\Delta s}$ ,  $\underline{\Delta t}$

$$\underline{Q} = \underline{\Delta s} \times \underline{\Delta t}, \quad \underline{P} = \underline{\Delta t} \times \underline{\Delta r}, \quad \underline{Q} = \underline{\Delta r} \times \underline{\Delta s} \quad (9)$$

where  $\underline{\Delta r}$  is from central point to the midpoint between neighbored corners,  $\underline{\Delta s}$  points anti-clockwise in circumferential direction and  $\underline{\Delta t}$  is in the outward body normal direction.

Proceeding as in the structured case we decompose the face normal vectors into the line vector directions using the box product

$$V = \langle \underline{\Delta r} \quad \underline{\Delta s} \quad \underline{\Delta t} \rangle \quad (10)$$

as shown below for the face normal  $\underline{Q}$

$$\begin{aligned}
V \cdot \underline{O} &= \langle \underline{O} \ \underline{\Delta s} \ \underline{\Delta t} \rangle \underline{\Delta r} + \langle \underline{\Delta r} \ \underline{O} \ \underline{\Delta t} \rangle \underline{\Delta s} + \langle \underline{\Delta r} \ \underline{\Delta s} \ \underline{O} \rangle \underline{\Delta t} \\
&= [\underline{O}(\underline{\Delta s} \times \underline{\Delta t})] \underline{\Delta r} + [\underline{O}(\underline{\Delta t} \times \underline{\Delta r})] \underline{\Delta s} + [\underline{O}(\underline{\Delta r} \times \underline{\Delta s})] \underline{\Delta t} \\
&= (\underline{O}\underline{O}) \underline{\Delta r} + (\underline{O}\underline{P}) \underline{\Delta s} + (\underline{O}\underline{Q}) \underline{\Delta t}
\end{aligned} \tag{11}$$

and rewrite the metric identity in terms of line vectors with coefficients consisting of scalar products of face vectors:

$$\begin{aligned}
&\sum_n \left\{ [\underline{O}\underline{O}\underline{\Delta r} + \underline{O}\underline{P}\underline{\Delta s} + \underline{O}\underline{Q}\underline{\Delta t}]_{n,m} \frac{1}{V_{n,m}} \right. \\
&+ \frac{1}{4} [\underline{Q}\underline{O}\underline{\Delta r} + \underline{Q}\underline{P}\underline{\Delta s} + \underline{Q}\underline{Q}\underline{\Delta t}]_{n,m+1/2} \frac{1}{V_{n,m+1/2}} \\
&\left. - \frac{1}{4} [\underline{Q}\underline{O}\underline{\Delta r} + \underline{Q}\underline{P}\underline{\Delta s} + \underline{Q}\underline{Q}\underline{\Delta t}]_{n,m-1/2} \frac{1}{V_{n,m-1/2}} \right\} = \mathbf{0}.
\end{aligned} \tag{12}$$

Compared with the expanded differential form of the identity for structured meshes (4), which split into the Laplacian terms and the control functions, the corresponding decomposition of the discrete form of the identity (12) yields nothing similar. We are still faced with 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

$$\sum_n \underline{O}_n = \mathbf{0} \tag{13}$$

into line vectors  $\underline{\Delta r}$  and  $\underline{\Delta s}$  yields

$$\sum_n \frac{1}{V_n} (\underline{O}_n^2 \underline{\Delta r}_n + \underline{O}_n \underline{P}_n \underline{\Delta s}_n) = \mathbf{0}. \tag{14}$$

Expressing also the coefficients in terms of line vectors

$$\underline{\Delta t} = \text{const}, \quad \underline{O}_n^2 = \underline{\Delta s}_n^2, \quad \underline{O}_n \underline{P}_n = -\underline{\Delta r}_n \underline{\Delta s}_n \tag{15}$$

we obtain in two dimensions

$$\sum_n \frac{1}{V_n} [\underline{\Delta \bar{s}}_n^2 \underline{\Delta r}_n - (\underline{\Delta \bar{s}}_n \underline{\Delta \bar{r}}_n) \underline{\Delta s}_n] = \mathbf{0}. \tag{16}$$

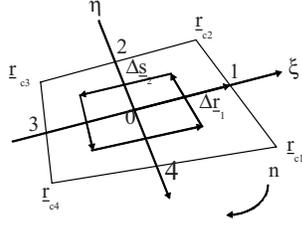


Fig. 3: planar quadrangle

that all coefficients of the Laplace equation are evaluated from cell boundary points only, i.e. the central point is never involved. Thus the averaged line vectors for forming the coefficients are readily defined from opposing boundary points:

$$\Delta \bar{s}_n = \frac{1}{2}(r_{n+1} - r_{n-1}), \quad \Delta \bar{r}_n = \frac{1}{2}(r_n - r_{n+2}) \quad (17)$$

with  $n$  being cyclic according to figure 3.

With the so defined coefficients equation (16) becomes identical to the difference approximation of the Laplace equation in computational space. Since the above relations hold for quads only, we still have to generalize the averaging of line vectors in order to formulate the coefficients. For that purpose we take into account the angle dependence in idealized multi-cornered cells as shown in figure 4 for a hexagon.

The averaged values of  $\Delta s_n$  are defined by the formulae,

$$\begin{aligned} \Delta \bar{s}_1 &= \left[ \cos \varphi_1 \Delta s_1 + \cos \varphi_2 \Delta s_2 + \dots + \cos \varphi_N \Delta s_N \right] \frac{2}{N} \\ \Delta \bar{s}_2 &= \left[ \cos \varphi_1 \Delta s_2 + \cos \varphi_2 \Delta s_3 + \dots + \cos \varphi_N \Delta s_1 \right] \frac{2}{N} \\ &\vdots \\ \Delta \bar{s}_N &= \left[ \cos \varphi_1 \Delta s_N + \cos \varphi_2 \Delta s_1 + \dots + \cos \varphi_N \Delta s_{N-1} \right] \frac{2}{N} \end{aligned} \quad (18)$$

with  $\varphi_N = \frac{2\pi}{N}(n-1)$

i.e. we measure the true  $\Delta s_n$ , multiply them with the cosines of the angles of an ideally regular cell and add the terms up to obtain the average of the coefficient forming line vector  $\Delta \bar{s}_n$ . Analogous formulae for  $\Delta \bar{r}_n$  only

In order to define the coefficients indicated by overbars as suited averages such that equation (16) will represent the discretized Laplace equation, we compare them first to the discretized form of the Laplace equation for quadrangles. Since that equation does not contain a variable volume, we assume  $V_n$  to be constant.

From structured meshes we know

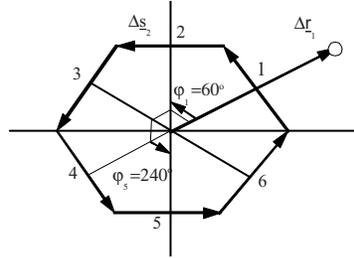


Fig. 4: Idealized hexagon

contain the midpoints of the cell's edges marked as an open circle in figure 4. The unknown central point does not enter.

Having introduced the averaged coefficients into the identity we compared our solutions with those of Winslow's discretized Laplace equation [6] for a number of example meshes consisting of quads or hexagons and found

both solutions in full agreement. Since the only structured meshes made up of triangles lead to quadrangles and hexagons, our averaging procedure is not proved but only believed to hold for any collection of triangles forming an  $N$ -cornered cell.

In an earlier paper about „Winslow smoothing on two-dimensional unstructured meshes“ Patrick Knupp [4] introduced pointwise a locally uniform computational space, expanded the derivatives about each mesh node in a Taylor series and solved the resulting coupled system. His procedure seems to require essentially more computational work.

Applying the averaged line vectors in three dimensions to formulate averaged face vectors according to definitions (9) the latter show some remarkable properties:

1. the face vectors  $\underline{\bar{Q}}_n$  in body normal direction and
2. in consequence, also the local cell volumes  $\bar{V}_n$  on the central level  $m$  both become constant for all  $n$  just as in the Laplace equation for structured meshes,

$$\underline{\bar{Q}}_n = \underline{\bar{Q}} = const \quad \bar{V}_n = \underline{\bar{Q}} \cdot \underline{\Delta t} = const \quad (19)$$

3. also, sums over mixed scalar products of averaged face normal vectors vanish

$$\sum \underline{\bar{Q}}_n \bar{P}_n = \sum \underline{\bar{Q}}_n \underline{\bar{Q}} = \sum \underline{\bar{Q}} \bar{P}_n = \mathbf{0}. \quad (20)$$

Expanding the differences in body normal direction about the central level  $m$  in equation (12)

$$\begin{aligned} (\underline{\bar{Q}} \underline{\bar{Q}} \Delta t)_{m+1/2} - (\underline{\bar{Q}} \underline{\bar{Q}} \Delta t)_{m-1/2} &= (\underline{\bar{Q}} \underline{\bar{Q}})_m (t_{m-1/2} - 2t_m + t_{m+1/2}) \\ &+ \left( (\underline{\bar{Q}} \underline{\bar{Q}})_{m+1/2} - (\underline{\bar{Q}} \underline{\bar{Q}})_{m-1/2} \right) \Delta t_m + \dots \end{aligned} \quad (21)$$

we discard the difference of the coefficients between upper and lower level in accordance with the structured Laplace equation. Thus, all averaged coefficients are to be evaluated on the central level  $m$  only.

With this expansion and accounting for the above mentioned properties of the coefficients we solve equation (12) for the unknown central point  $\underline{r}_0$  on level  $m$ :

$$\begin{aligned} \underline{r}_{0,m} \sum \left( \bar{\underline{O}}_n \bar{\underline{O}}_n + \frac{1}{2} \bar{\underline{Q}} \bar{\underline{Q}} \right) = & \quad (22) \\ \sum \left[ \bar{\underline{O}}_{n,m} \bar{\underline{O}}_{n,m} \underline{rm}_{n,m} + \bar{\underline{O}}_{n,m} \bar{\underline{P}}_{n,m} \Delta s_{n,m} + \frac{1}{4} \bar{\underline{Q}}_m \bar{\underline{Q}}_m (\underline{t}_{m-1/2} + \underline{t}_{m+1/2}) \right. \\ & \left. + \frac{1}{4} \bar{\underline{O}}_{n,m} \bar{\underline{Q}}_{n,m} (\Delta r_{m+1/2} - \Delta r_{m-1/2}) + \frac{1}{4} \bar{\underline{Q}}_{n,m} \bar{\underline{P}}_{n,m} (\Delta s_{m+1/2} - \Delta s_{m-1/2}) \right], \end{aligned}$$

where  $\Delta r_{n,m}$  is defined as the midpoint between two adjacent corner points according to the sketch in figure 5

$$\Delta r_{n,m} = \underline{rm}_{n,m} - \underline{r}_{0,m} \quad \text{with} \quad \underline{rm}_{n,m} = 1/2(\underline{rc}_{n,m} + \underline{rc}_{n+1,m}). \quad (23)$$

On the right hand side of equation (22) the coefficients in the first line result from the decomposition of the face vectors  $\underline{Q}_{n,m}$ , while the coefficients in the three following lines result from the decomposition of the face vectors  $\underline{Q}_{n,m}$ .

To solve equation (22) within a marching procedure we need at least three layers. Providing Dirichlet boundary conditions on the first layer initially from a known surface mesh and on the second and third layer from an algebraic forecast we iterate the solution to convergence on the intermediate layer. After each such iteration the algebraic forecast on the outer third layer will be updated. Once the solution is known on the intermediate layer, it becomes the boundary condition of the first layer for the next marching step and so forth.

The algebraic forecast needs a prescription of the spacing and the local marching direction. The latter follows from an angle weighted superposition of normal vectors to those faces around a central point, which limit the domain of visibility. Details are described in the appendix.

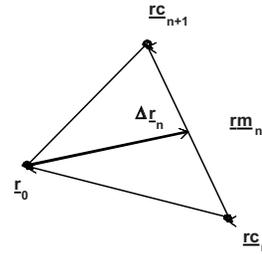


Fig. 5: Midpoint vector  $\underline{rm}_n$

## 5. A GENERIC TEST EXAMPLE FOR HYBRID MESH GENERATION

The test example consists of a planar delta wing with extremely sharp edges and a wedge on top of it. Despite its geometric simplicity the generic configuration poses a severe test case, since we expect pure Laplacian smoothing to pull the mesh over the sharp edges and corners and also pure algebraic marching out of concavities to generate overlapped meshes.

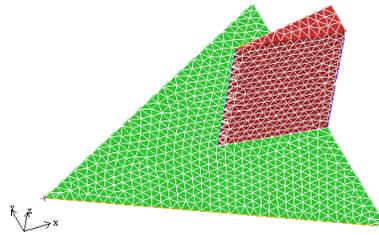


Fig. 6: Test configuration

The surface mesh is coarse (all together 1515 points, 3026 triangles), rather regular but not symmetric. Points have not been clustered towards the edges and corners - as would be preferable to better resolve these areas in fluid flow calculations - in order not to mask the cell stretching around the sharp edges expected to appear in the body conforming mesh surfaces to be generated.

Having marched the solution over 20 layers off the body surface with 20 iterations per layer the two figures below show cuts through the hybrid mesh in the symmetry plane  $y = 0$  and in a section at  $x = 0.8$ .

As is seen from figure 7 marching out off concave corner regions does not cause any problems, even not ahead of the sharp wedge starting from a point of mixed strong convex and concave curvature, where the front edge of the wedge meets the wing in the plane  $y = 0$ .

But here the curvature of the cut lines changes sign when approaching the outer hull (figure 7 left) and the resulting surface mesh (figure 8) becomes unacceptable. Around those three points of mixed strong convex/concave curvature, where the wedge meets the wing, the cells are badly stretched in the direction of the convexity and are compressed in the direction of the concavity.

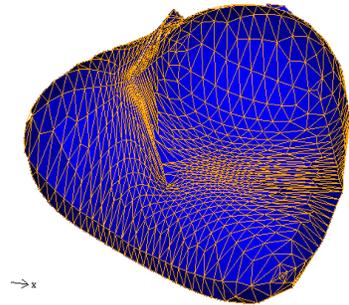


Fig. 8: Outer surface mesh (pure Laplace)

Looking for a remedy to suppress these excessively stretched cells we return to the planar case and consider just a single convex cell. For given corner points the central point (see figure 9) follows from the solution to the equation

$$\sum_n [\Delta \bar{s}_n^2 \Delta r_n - (\Delta \bar{s}_n \Delta \bar{r}_n) \Delta s_n] = 0 \tag{24}$$

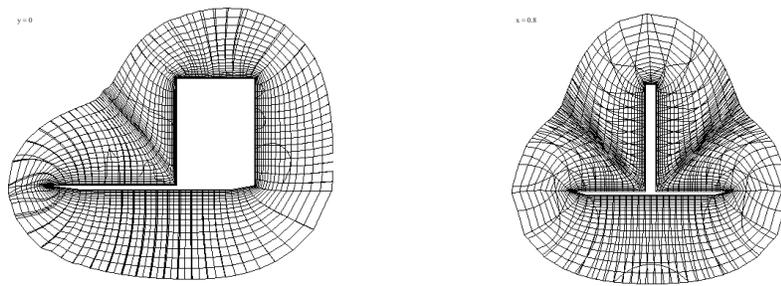


Fig. 7: Mesh cuts at  $y = 0$  (left) and  $x = 0.8$  (right)

as a superposition of forces  $\Delta r$  acting in the radial directions and forces  $\Delta s$  acting parallel to the cell's edges weighted by our Laplacian coefficients. The dark black arrows show the resulting force per edge acting on the central point. In figures 9 to 11 the magnitude of the arrow-heads is drawn proportional to the magnitude of the forces and the central point is marked as a circle filled black.

Switching to a cell with large differences in edge lengths putting one point to the left and five boundary points to the right hand side (see fig. 10) the solution to our Laplace equation being identical to Winslow's solution pulls over the right end of the convex hexagon. This is somewhat surprising, since

the extremum principle should hold for Laplace's equation. Obviously the extremum principle being defined for the continuous differential case does not hold for the discrete and distorted cell.

The Laplacian solution seems to deteriorate due to the large forces parallel to the edges. Reducing these forces, when the cell becomes strongly stretched, i.e. when the angle between  $\Delta r$  and  $\Delta s$  becomes small, by a correction formula in an engineering fashion

$$\Delta s_{red} = \Delta s \cdot (1 - 0.25 \cdot \cos^2(\Delta r, \Delta s)) \tag{25}$$

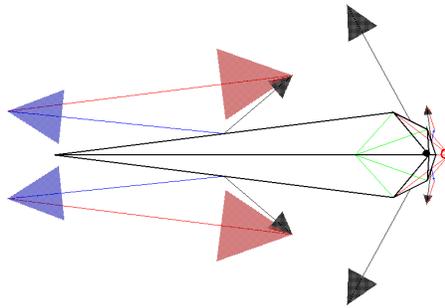


Fig. 11: Laplacian cell with reduced  $\Delta s$ -forces

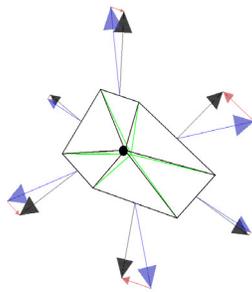


Fig. 9: Forces acting on the central point of a planar cell

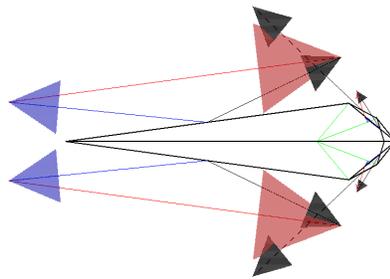


Fig. 10: Folded hexagon, central point pulled over right end of cell

drives the central point from its position outside the hexagon (fig. 11, open circle) back into the convex polygon (fig. 11, black filled circle) preventing the cell from folding.

Applying the simple correction formula to the three-dimensional example the extreme stretching disappear (fig. 12) and the distribution of cells becomes rather smooth, as is also to be seen from front and rear.

While the body is symmetric, its surface triangulation is not. Therefore the full mesh has been generated. Slight deviations from symmetry are observable in the outer surface mesh.

Sure, there is still some larger stretching around the sharp edges of the wing, which could have been mitigated significantly by clustering points towards the edges on the body surface mesh to better resolve these regions. This would lead to cells being stretched in the direction of the edges on the body surface counteracting the present stretching of the cells on the outer hull perpendicular to the edges.

## 6. GENERATION OF MIXED MESHES

The shortcoming of purely hybrid grids is the low anisotropy of surface triangles resulting in a large number of surface grid points, which is agglomerated throughout the prismatic layers for the boundary layer resolution. Especially for the high aspect ratio wings the low anisotropy of the unstructured surface mesh leads to an unnecessary high resolution in the span-wise direction. Recalling the experience with the application of structured grids, it is known that the aspect ratios of surface quadrilaterals can be orders of magnitude higher, additionally resulting in well aligned body conforming meshes. On the other hand an increasing complexity of the configuration to be meshed limits the application of pure structured grids mainly because of intricate grid topology.

In the mixed mesh approach we use simple block structured boundary layer grids with highly stretched hexahedral cells wherever possible. Topologically difficult regions limiting the application of pure block structured grids are meshed by adjoining layers of prisms of triangular cross-section. The outer flow field is meshed using tetrahedral elements. The connection between the tetrahedral elements in the outer inviscid flow domain and the hexahedral elements in the boundary layer region is accomplished by egg-carton like pyramids. Since the structured grid approach offers direct control on point distributions by the bottom up approach, a smooth transition from the structured into the unstructured part can be achieved. It has already been shown that applying this grid technique can speed up Navier-Stokes flow computations by up to 90% without losing solution accuracy, mainly due to the saving of points [7].

The grid generator used is the formerly purely structured DLR grid generator MegaCads [8], which allows for parametric construction of block structured grids. Besides basic CAD features like surface construction, projection and intersection, it is capable of constructing smooth grids by applying elliptic techniques as well as parabolic advancing front meshing for wall orthogonal boundary layer grids [2]. The whole grid generation process is stored in a process description file and can be re-run for modified geometries, as long as the CAD-topology is retained. Due to the parametric capabilities it is possible to guarantee grid quality in terms of smoothness, resolution and grid line angles even for larger geometrical changes. To account for the necessity to include the grid generator in an

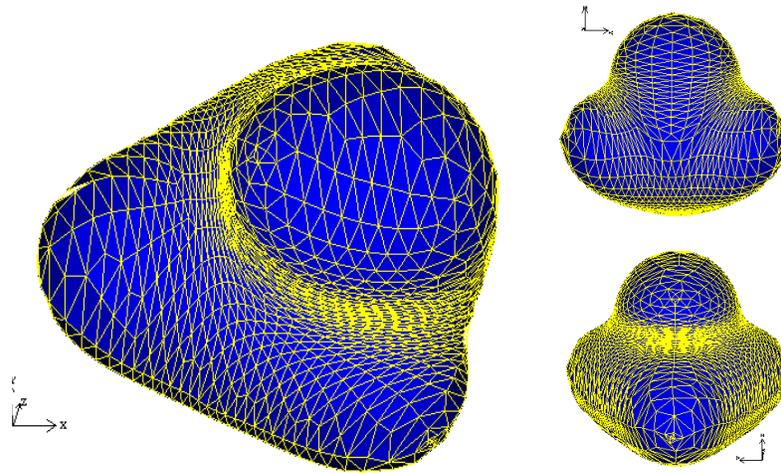


Fig. 12: Laplacian mesh with reduced  $\Delta_s$ -forces

optimization loop, MegaCads can be run in a non-interactive mode, repeating the grid generation process for the actual geometry.

The hybrid mesh generation procedure described in detail above in this paper has been implemented in MegaCads.

For the generation of the unstructured grid part the 3D triangulation software NETGEN of Schöberl [9] was incorporated into the framework of MegaCads as a Black-Box tool. The NETGEN software offers constrained Delaunay volume triangulation of a given triangulated surface mesh, which in the following application is the outer hull of the pyramidal interface layer and the outer prisms' faces wherever applied. The pyramidal elements connecting the structured with the unstructured mesh part are extruded from the hexahedral elements based on edge lengths in order to achieve a smooth transition of the control volumes for the reduction of numerical errors and/or instabilities.

## 7. APPLICATION

The ONERA M6 wing, measured by Schmidt and Valpin [11], is used for demonstration of the semi-structured grid approach using the described parabolic marching algorithms. The blunt trailing edges are closed, as this is commonly applied for CFD for this case. The rounded wing tip is replaced by a sharp cut at the end of the wing. This is done to demonstrate the prismatic capabilities. In purely structured grid generation this kind of wingtip would either lead to degenerated cells at the extension of the trailing edge or to an arbitrary closing of the wing tip within the range of one cell. Applying semi-structured grid generation the wing tip is meshed with triangular prisms (figure 13).

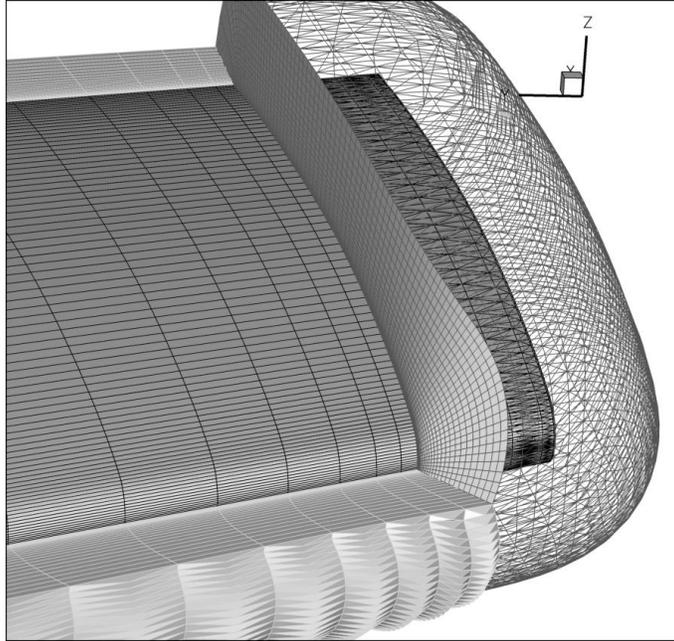


Fig. 13: Parabolic prism grid around the ONERA M6 wing tip

For comparison purposes a second purely hybrid mesh has been generated using the CENTAUR mesh generator [10]. Figure 14 shows the surface meshes for both mesh generation strategies. Both meshes have 32 cell layers normal to the wall to resolve the boundary layer and the vertical resolution is similar between both types of meshes. The mixed grid has about 215.000 points and 6.000 surface elements on the wing, while the purely hybrid grid has 1.400.000 points with about 80.000 surface elements, since for the unstructured surface mesh generation only a maximum anisotropic aspect ratio of 2 is allowed. For the semi-structured grid the maximum aspect ratio of the surface quads is around 170.

Figure 15 shows a comparison of the calculated surface pressure coefficient at two different spanwise locations for both meshes. It is observed that the solution quality is similar and the shock at the outer section is resolved better by the mixed mesh approach.

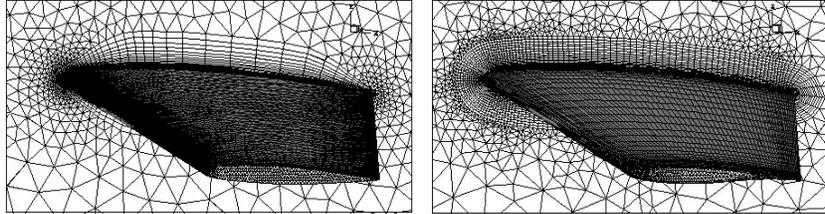


Fig. 14: Surface meshes for hybrid (left) and mixed meshes (right) around the ONERA M6 wing

## 8. CONCLUSIONS

In analogy to the differential equations of structured grids the face-weighted Laplace equation in discrete form has been derived from metric identity. Hybrid meshes are generated embedding the solution of the difference equation in a marching procedure with sufficient algebraic control of spacing in the structured direction. The ellipticity of the equation in the unstructured body conforming directions provides enough dissipation to prevent the mesh from folding when marching out of concave body surface areas. A simple coefficient modification avoids the formation of highly stretched distorted cells prone to overlap.

The mixed mesh strategy has been applied for the ONERA M6 wing and has been compared to a purely hybrid mesh. It has been shown that the smooth mixed mesh approach is able to capture the flow physics by using only about 20% of the points of a standard hybrid grid.

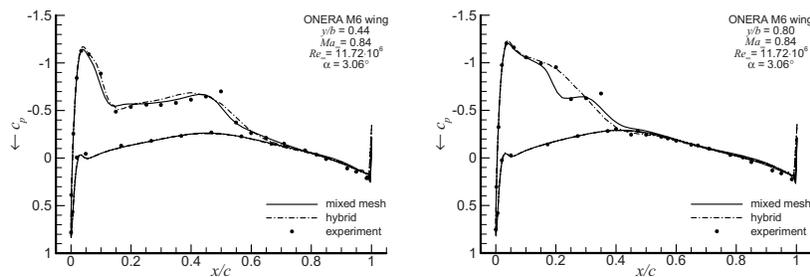


Fig. 15: Comparison of pressure distributions of the M6 wing at two different spanwise locations for a hybrid grid and a mixed mesh

## REFERENCES

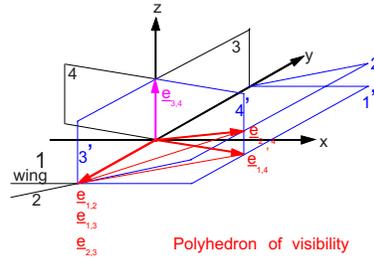
- [1] P. Niederdrenk, "On the Control of Elliptic Grid Generation", Proc. 6th Intern. Conf. On Numerical Grid Generation in Comp. Field Simulations, London, July 1998, eds. M. Cross et al., NSF Eng. Center, Mississippi, pp. 257
- [2] P. Niederdrenk, O. Brodersen, "Controlled Parabolic Marching Grid Generation", 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
- [3] D. Thompson, B. Soni, "Generation of Quad- and Hex-Dominant, Semi structured Meshes Using an Advancing Layer Scheme", Proc. 8th Intern. Meshing Roundtable, South Lake Tahoe, CA, USA, Oct. 1999, pp. 171
- [4] P.M. Knupp, "Winslow Smoothing on Two-Dimensional Unstructured Meshes", Proc. 7th Intern. Meshing Roundtable, Sandia Natl. Labs., 1998, pp. 449
- [5] W.M. Chan, J.L. Steger, "Enhancements of a Three-Dimensional Hyperbolic Grid Generation Scheme", Appl. Mathematics and Computations, 1992, Vol.51, pp. 181
- [6] A. Winslow, "Numerical Solution of the Quasi-Linear Poisson Equations in a Non-Uniform Triangle Mesh", J. Comp. Phys., 1967, Vol.2, pp. 149
- [7] J.Wild, "Acceleration of Aerodynamic Optimization Based on RANS-Equations by Using Semi-Structured Grids", Proc. ERCOFTAC Design Optimization: Methods & Applications, Athens (Greece), Paper ERCODO2004\_221, 2004
- [8] 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
- [9] J. Schöberl, "NETGEN - An advancing front 2D/3D mesh generator based on abstract rules", Computing and Visualization in Science, Vol. 1, 1997, pp. 41-52
- [10] CentaurSoft, "Welcome to CentaurSoft, The solution to grid generation issues for computational engineering problems", <http://www.centaursoft.com>, 2004
- [11] V. Schmitt, F. Charpin, "Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers", AGARD AR 138, 1979

## APPENDIX: LOCAL MARCHING DIRECTION

The definition of the local marching direction for hybrid meshes is readily explained by an example. With respect to the configuration shown in figure 6 let us consider the point, in which the left rear edge of the wedge

formed by faces 3 and 4 meets the trailing edge of the wing formed by faces 1 and 2 (see sketch below).

To define the polyhedron of visibility for a point we only use different unit normal face vectors, i.e. we discard multiples of identical unit normals. Edge vectors  $\underline{e}_{j,k}$  are formed along the intersection lines of all combinations of two faces, for instance  $\underline{e}_{3,4}$  between faces 3 and 4. The sense of direction of the edge vector follows from the sense of the most aligned face normal, e.g.  $\underline{e}_{3,4}$  from  $\underline{n}_1$  or  $\underline{e}_{1,3}$  from  $\underline{n}_4$ .



The polyhedron of visibility is given by only those edge vectors, which can see all faces, i.e. for which  $\underline{e}_{j,k} \cdot \underline{n}_i$  is not negative. In the example the edge vector  $\underline{e}_{3,4}$  is excluded, since it cannot see face 2. The remaining edge vectors  $\underline{e}_{1,4}$ ,  $\underline{e}_{2,4}$  and  $\underline{e}_{1,2}$  form the polyhedron of visibility.

The marching direction follows from a weighted superposition of normals to those faces forming the polyhedron of visibility

$$\underline{N} = \sum w_i \underline{n}_i .$$

In our example  $\underline{n}_3$  does not contribute to the polyhedron of visibility, so it does not appear in the above formula. The weights are taken proportional to the angle between the appertaining edge vectors, for instance  $\underline{n}_4$  is weighted by the angle between  $\underline{e}_{1,4}$  and  $\underline{e}_{2,4}$ .

