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

UNSTRUCTURED HIGH ORDER GRIDS AND THEIR APPLICATION IN DISCONTINUOUS GALERKIN METHODS

F. Hindenlang, G. Gassner, T. Bolemann and C.-D. Munz

University of Stuttgart,
Pfaffenwaldring 21, 70569 Stuttgart, Germany, www.iag.uni-stuttgart.de
e-mail: {hindenlang,gassner,munz}@iag.uni-stuttgart.de

Key words: high order accuracy, discontinuous Galerkin, complex geometry, curved elements, surface reconstruction, free-stream preserving property

Abstract. *Discontinuous Galerkin (DG) methods are a prominent candidate for high order accurate schemes for advection dominated problems in three-dimensional complex geometries, since they sustain high order spatial accuracy even on general unstructured grids. To maintain the high order accuracy at curved wall boundaries, a high order representation of the elements near the wall surface is required, i.e. high order grids. Regarding three-dimensional curved geometries, the construction of such unstructured curved grids is a subtle task.*

In the first part of the paper, the implementation of the DG scheme is discussed, focusing on curved elements. Since integration can only be done on a linear reference element, mappings of curved element sides as well as the curved element volume are required.

In the second part, our approach for the resolution of curved boundary surfaces is presented. The concept of the approach is to rely on established unstructured grid generators for a basic grid, and to develop an independent tool to provide high quality information for the curved boundaries directly from the CAD model. The grid consists of hexahedra, prisms, pyramids and tetrahedra with straight edges. Via a CAD tool, we access the CAD definitions of the computational domain to detect all element corner points on the boundary surfaces, and assign for each of these points a set of 'exact' normal vectors. The normal vectors lead to curved element edges and further to a local surface mapping of the element sides. Finally, the volume mapping of the curved element is found, based on the curved element sides. We demonstrate the applicability of this approach with an unstructured grid of a complex wing-body-nacelle configuration, having multiple edges, corners and intersecting sub-surfaces.

1 INTRODUCTION

In industrial applications, geometries are three-dimensional and typically comprise curved surfaces, curved borders and sharp edges. Here meshing itself becomes an issue and therefore, unstructured grids are needed. Most of the unstructured grid generators provide high quality grids consisting of hexahedra, prisms, tetrahedra and pyramids, whereas the element edges are in general straight lines, at most having an additional mid-point.
When using high order methods, wall boundary conditions need a high order representation of the wall normal. Bassi and Rebay[1] showed, that in the case of curved boundaries a high order DG discretization with straight-sided 2D elements yields low order accurate and even physically wrong results. To overcome this issue, they propose to use at least elements with parabolic shaped sides on the boundary.
Approaches for high order grids should use CAD definitions to guarantee a correct approximation of the geometry. Existing high order grid generators[6] are very promising, however meshing of complex geometries with unstructured hybrid grids has not reached the level of commercial grid generators. Since curved elements are only needed near curved boundaries, a completely local approach, independent of the grid generator, is presented.

2 THE DG SCHEME ON CURVED ELEMENTS

In this part, we derive the DG scheme on an arbitrarily shaped element Q for the Euler equations in conservation form

$$U_t + \vec{\nabla} \cdot \vec{F}(U) = 0. \tag{1}$$

U is the vector of conserved variables , and $\vec{F}(U)$ the advection fluxes. For the DG formulation, we approximate the solution in element Q by a modal discontinuous polynomial approximation

$$U(\vec{x},t)\big|_Q \approx U^{h,Q}(\vec{x},t) = \sum_{j=1}^{N} \hat{U}_j^Q(t)\varphi_j^Q(\vec{x}), \tag{2}$$

where φ_j^Q are *orthonormal* basis functions, which are constructed via Gram-Schmidt orthogonalization. The approximation (2) is inserted in (1), the equations are multiplied by a test function $\phi \in \{\varphi_j^Q\}_{j=1,\dots,N}$ and integrated over the element Q

$$\left\langle U_t^{h,Q} + \vec{\nabla} \cdot \vec{F}(U^{h,Q}), \phi \right\rangle_Q = 0. \tag{3}$$

Integration by parts leads to the DG formulation

$$\left\langle U_t^{h,Q} - \vec{\nabla} \cdot \vec{F}(U^{h,Q}), \phi \right\rangle_Q + \left(H(U_\pm^h, \vec{n}), \phi^-\right)_{\partial Q} = 0, \tag{4}$$

with the numerical flux function resolving the discontinuous solutions on the element sides $U_\pm^h$ by an approximate Riemann flux

$$\vec{F}(U^h) \cdot \vec{n}\big|_{\partial Q} \approx H(U_\pm^h, \vec{n}). \tag{5}$$

The inner products are defined as volume integral and the surface integral, which is divided into a sum over all element sides

$$\langle a,b\rangle_Q = \int\limits_Q ab\,d\vec{x}\;, \quad (a,b)_{\partial Q} = \int\limits_{\partial Q} ab\,dS = \sum_s \int\limits_{\partial Q_s} ab\,dS\,. \tag{6}$$

The volume fluxes are approximated using the same polynomial representation

$$\vec{F}(U)\big|_Q \approx \vec{F}^{h,Q}(U) = \sum_{j=1}^{N} \hat{\vec{F}}_j^Q \varphi_j^Q(\vec{x})\,, \tag{7}$$

as well as the surface normal flux of each element side

$$H(U_\pm^h, \vec{n})\big|_Q \approx H^{h,(s)}(U) = \sum_{j=1}^{N_s} \hat{H}_j^{(s)} \varphi_j^{\partial Q_s}(\vec{x})\,. \tag{8}$$

where $\varphi^{\partial Q_s}$ denotes an orthonormal modal basis on the side s. The semi-discrete formulation of the DG scheme for the vector of DOFs $\underline{\hat{U}}$ in element Q reads as

$$\underline{\underline{M}}^Q \underline{\hat{U}}_t - \sum_{d=1}^{dim} \underline{\underline{S}}_d^Q \underline{\hat{F}}_d^Q + \sum_s \underline{\underline{M}}^{\partial Q_s} \underline{\hat{H}}^{(s)} = 0. \tag{9}$$

with the mass matrix $\underline{\underline{M}}^Q$, the stiffness matrix $\underline{\underline{S}}_d^Q$ and the element side mass matrix $\underline{\underline{M}}^{\partial Q_s}$. These operators are defined as

$$M_{ij}\big|^Q = \left\langle \varphi_j^Q, \phi_i^Q \right\rangle_Q = \delta_{ij}\;, \quad S_{ij}\big|_d^Q = \left\langle \varphi_i^Q, \frac{\partial \phi_j^Q}{\partial x_d} \right\rangle_Q \;, \quad M_{ij}\big|_d^{\partial Q_s} = \left(\varphi_j^{\partial Q_s}, \phi_i^Q\right)_{\partial Q_s}. \tag{10}$$

Now we need to implement the volume and the surface integration (6) for each element type, e.g. tetrahedra, pyramids, prisms and hexahedra. An analytical integration is in general unfeasible, because the integration domain is supposed to be curved. A numerical Gauss-type integration is used instead. For all straight sided reference elements depicted in Figure 1, tensor-product Gauss-type integrations are found in Karniadakis and Sherwin,[7] which consist of different tensor-product Gauss-Jacobi integration points on the reference square / cube, including an exact integration of the linear transformations to triangles, prisms, pyramids and tetrahedra.

To perform the integration, a *volume mapping* $\vec{x} = \vec{x}(\vec{\xi})$ from the curved element to the reference element is needed. Regarding exemplarily the stiffness matrix entries, inserting the mapping yields to

$$\int\limits_Q \varphi_j(\vec{x})\,\frac{\partial \phi_j(\vec{x})}{\partial x_d}\,d\vec{x} = \int\limits_{Q_{gauss}} \varphi_j(\vec{x}(\vec{\xi}))\,\frac{\partial \phi_j(\vec{x}(\vec{\xi}))}{\partial x_d}\,J(\vec{\xi})\,d\vec{\xi} = \int\limits_{Q_{gauss}} P(\vec{\xi})\,d\vec{\xi},$$

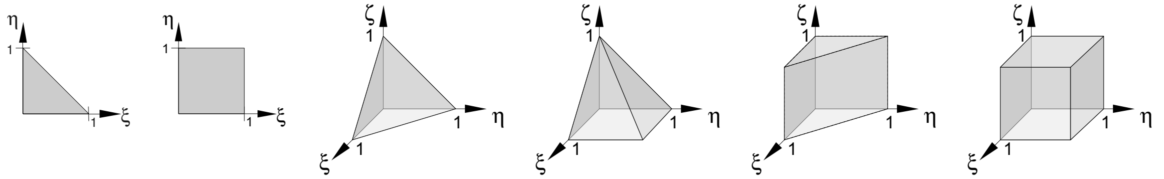

Figure 1: straight sided reference elements

with the Jacobian of the mapping $J(\vec{\xi}) = \left|\frac{\partial x_j(\vec{\xi})}{\partial \xi_i}\right|$. We confine the mapping $\vec{x}(\vec{\xi})$ to be a *polynomial* in $\vec{\xi}$, so $P(\vec{\xi})$ remains a polynomial. Polynomials can be integrated exactly by an adequate Gauss quadrature. The same approach is applied to the surface integral, requiring a surface mapping for each triangular or quadrangular element side. The construction of these polynomial mappings is described in the following sections.
We note that for efficiency, the operators in (10) are pre-computed via Gauss-type integration, and then combined with a nodal projection of the fluxes $\underline{\hat{F}}_d^Q, \underline{\hat{H}}^{(s)}$ in (9), leading to a quadrature free mixed nodal and modal approach, see Gassner et al.[5]

3 HIGH ORDER GRID CONSTRUCTION

3.1 Volume Mapping

The volume mapping is defined, following Figure 2, by the surface mapping of all element sides, and the surface mapping in turn is defined by curved element edges.

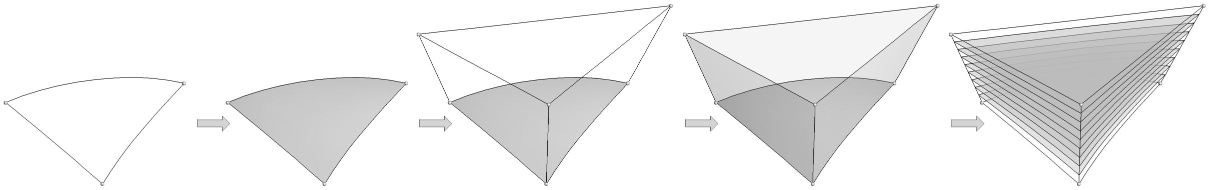

Figure 2: Construction of mappings: from curved element edges to surface and volume mappings

1. All element edges lying on the curved boundary are *uniquely* defined by a polynomial curve. The uniqueness of the edges guarantees C^0 continuity of the high order grid. The polynomial curve is represented by Bernstein-Bézier basis functions.[4] The inner edges remain straight lines. The construction of these curved element edges is described in section 3.2.

2. Triangle and quadrangle mappings are also defined by Bernstein-Bézier basis functions.[4] They have the convenient property, that the edge control points remain the same, and inner control points can be expressed in terms of the edge control points, see section 3.3. This is done for all element sides having at least one curved edge.

3. The volume mapping is a blending of all element sides. It is again a polynomial, represented with Bernstein-Bézier basis functions for tetrahedra, prisms and hexahedra,[8] as well as nodal tensor-product basis functions for pyramids. We impose the volume mapping to match *exactly* all element side mappings. This yields to an under-determined equation system, which is solved using a pseudo-inverse, resulting in the full volume mapping.

Elements are curved only when elements are linked – e.g. sharing at least one edge – with the wall boundary ('local curving'). If the elements are highly stretched or skewed, additional curving of inner elements would be required[6,9] ('global curving') to avoid negative or very small Jacobians. We found that using near-wall tetrahedra or pyramids generally leads to much smaller Jacobians than using prismatic extrusions of the surface grid, i.e. prisms and hexahedra. Smaller angles on the element corners are the reason. In most cases, when using prismatic elements, local curving is sufficient.

3.2 Curved Edge Construction

First let us introduce the Bernstein-Bézier basis functions and the Bézier curve, shown in Figure 3.

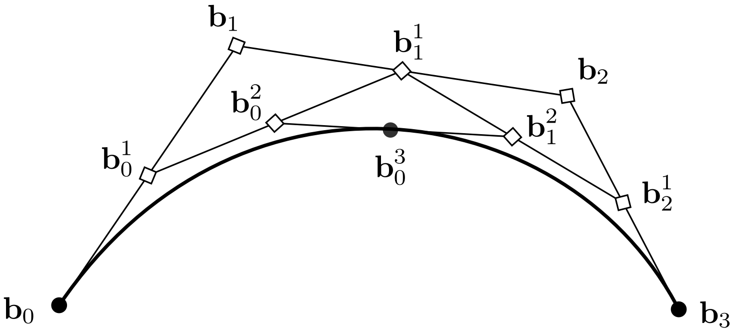

Figure 3: The Bézier curve and its construction via the De Casteljau algorithm, $\vec{x}_b(t = 0.5)$

Bézier curves (b) are expressed by

$$\vec{x}_b(t) = \sum_{i=0}^{n} \vec{b}_i B_i^n(t)\,, \quad \vec{x}_b, \vec{b}_i \in \mathbb{R}^3\,, \tag{11}$$

where $\vec{b}_i$ are the coordinates of the control points and form the control polygon. The i-th Bernstein polynomial of degree n reads as

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \tag{12}$$

with the binomial coefficient

$$\binom{n}{i} = \frac{n!}{i!(n-i)!}\,. \tag{13}$$

In Figure 3, one can easily see that the first and the last control point coincide with the end points and that here, the control polygon is tangential. The derivative of the Bézier curve is defined as[4]

$$\frac{\partial \vec{x}_b}{\partial t} = \dot{\vec{x}}_b(t) = n \sum_{i=0}^{n-1} \Delta \vec{b}_i B_i^{n-1}(t)\,, \tag{14}$$

where $\Delta\vec{b}_i = \vec{b}_{i+1} - \vec{b}_i$ are the vectors from one control point to the other.
Given an element edge with two end points $\vec{p}_1, \vec{p}_2$ and two tangential vectors $\vec{t}_1, \vec{t}_2$, a Bézier curve of degree $n = 3$ is well-defined. The four control points are found from equation (11)

$$\vec{x}_b(t=0) = \vec{p}_1 = \vec{b}_0 \quad , \quad \vec{x}_b(t=1) = \vec{p}_2 = \vec{b}_3 \tag{15}$$

and from equation (14)

$$\dot{\vec{x}}_b(t=0) = \vec{t}_1 = 3(\vec{b}_1 - \vec{b}_0) \rightarrow \vec{b}_1 = \vec{p}_1 + \frac{1}{3}\vec{t}_1 \tag{16}$$

$$\dot{\vec{x}}_b(t=1) = \vec{t}_2 = 3(\vec{b}_3 - \vec{b}_2) \rightarrow \vec{b}_2 = \vec{p}_2 - \frac{1}{3}\vec{t}_2\,. \tag{17}$$

For the simple case of a straight line segment, control points lie on the line and are equally spaced,

$$\vec{b}_i^{line} = \frac{i}{n}(\vec{b}_n - \vec{b}_0). \tag{18}$$

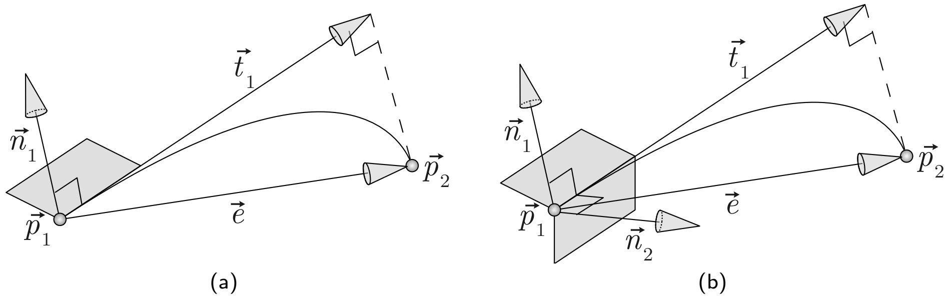

Figure 4: Tangential vector by projection and by cross product

The CAD tool, described in section 4, will supply a set of normals on each boundary surface grid point. For the calculation of the tangential vectors, two principal cases, depicted in Figure 4, have to be considered,

(a) one normal at the end point:
This occurs when a grid point lies inside the CAD surface or on the edge between adjacent tangential CAD surfaces. The tangential vector of the element edge is found by projection of the straight edge $\vec{e}$ onto the normal $\vec{n}_1$, see Figure 4a

$$\vec{t}_1 = \vec{e} - (\vec{n}_1 \cdot \vec{e})\vec{n}_1 \quad , \quad \vec{e} = \vec{p}_2 - \vec{p}_1\,. \tag{19}$$

(b) two normals at the end point:
This occurs when the a grid point lies on the sharp edge between adjacent CAD surfaces. The tangential direction is found by their cross product, and its length by projection of the straight edge see Figure 4b

$$\vec{t}_1 = (\vec{e} \cdot \vec{\tau})\vec{\tau} \quad , \quad \vec{\tau} = \frac{\vec{n}_1 \times \vec{n}_2}{|\vec{n}_1 \times \vec{n}_2|} \,. \tag{20}$$

This approach guarantees that arbitrary sharp edges are automatically detected and represented as piecewise Bézier curves, in particular with G^1 continuity at the common grid points.

3.3 Blending Edges to Surfaces

Bernstein-Bézier basis functions are also available for triangles and quadrangles. We restrict ourselves to cubic polynomials, since the edges are of degree $n = 3$.

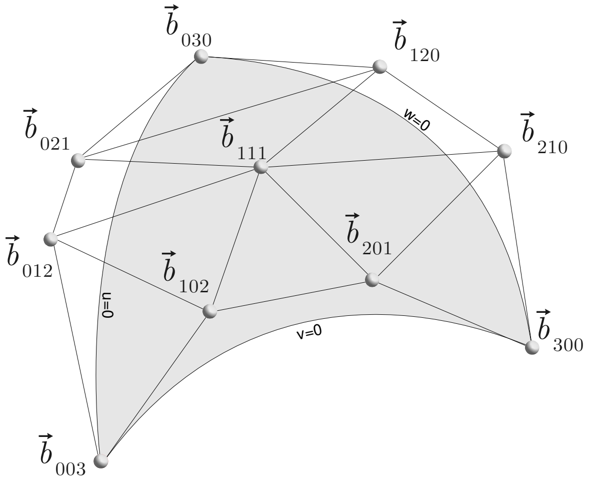

Figure 5: cubic Bézier triangle

The representation of a Bézier triangle (bt) is written as

$$\vec{x}_{bt}(u,v,w) = \sum_{ijk}^{n} \vec{b}_{ijk} B^n_{ijk}(u,v,w)\,, \quad 0 \le i+j+k \le n\,, \quad \vec{x}_{bt}, \vec{b}_{ijk} \in \mathbb{R}^3\,, \tag{21}$$

where n is again the polynomial degree, and u, v, w are tri-variate parameter directions. The surface remains bivariate, since $u + v + w = 1$ must hold. B^n_{ijk} are the triangular Bernstein-Bézierbasis functions defined as

$$B^n_{ijk}(u,v,w) = \binom{n}{ijk} u^i v^j w^k\,, \tag{22}$$

with the trinomial coefficients

$$\binom{n}{ijk} = \frac{n!}{i!j!k!} \,. \tag{23}$$

$\vec{b}_{ijk}$ are control points of the surface in physical space, shown in Figure 5. Now, all control points of the edges are known from the section 3.2, the only unknown control point is $\vec{b}_{111}$. Symmetry suggests a formula of the type

$$\vec{b}_{111} = (1-\alpha)E + \alpha V \,, \tag{24}$$

with

$$E = \frac{1}{6}(\vec{b}_{012} + \vec{b}_{021} + \vec{b}_{102} + \vec{b}_{201} + \vec{b}_{120} + \vec{b}_{210}) \,, \quad V = \frac{1}{3}(\vec{b}_{003} + \vec{b}_{030} + \vec{b}_{300}) \,. \tag{25}$$

The specific choice $\alpha = \frac{1}{2}$ reproduces quadratic polynomials exactly as shown by Farin[3] and is used in this work.

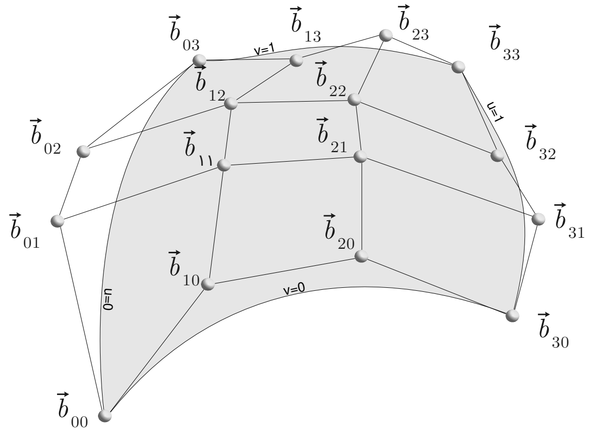

Figure 6: cubic Bézier quadrangle

The representation of a Bézier quadrangle (bq) is written as a tensor product

$$\vec{x}_{bq}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{n} \vec{b}_{ij} B_i^n(u) B_j^n(v) \,, \tag{26}$$

where B_i^n, B_j^n are the Bezier-Bernstein basis functions from (12) and $\vec{b}_{ij}$ are control points of the surface in physical space, shown in Figure 6.

Again, the control points of the edges are known, and the inner points are found by a bi-linear blending of the edges,[4]

$$\vec{b}_{ij} = \vec{b}_{ij}^{u} + \vec{b}_{ij}^{v} - \vec{b}_{ij}^{uv} \,, \tag{27}$$

with

$$\vec{b}^{u}_{ij} = (1 - \frac{i}{n})\vec{b}_{0,j} + \frac{i}{n}\vec{b}_{n,j} \tag{28}$$

$$\vec{b}^{v}_{ij} = (1 - \frac{j}{n})\vec{b}_{i,0} + \frac{j}{n}\vec{b}_{i,n} \tag{29}$$

$$\vec{b}^{uv}_{ij} = (1 - \frac{i}{n})(1 - \frac{j}{n})\vec{b}_{0,0} + \frac{i}{n}(1 - \frac{j}{n})\vec{b}_{n,0} \tag{30}$$

$$+ (1 - \frac{i}{n})\frac{j}{n})\vec{b}_{0,n} + \frac{i}{n}\frac{j}{n}\vec{b}_{n,n} \,. \tag{31}$$

The surface mapping is done for all elements touching curved surfaces with at least one edge. At inner element sides, only one unique surface mapping is constructed, which guarantees C^0 continuity at the element interface and equal normal vectors for both element sides.

4 CAD Tool for Normal Vectors

The CAD tool is written in Visual Basic 8 and uses the Microsoft .Net 2.0 framework. It is connected to CATIA via a scripting interface (CAA-API), thus commands to load a model, do geometric operations and extract CAD definitions are made directly accessible for the CAD tool.

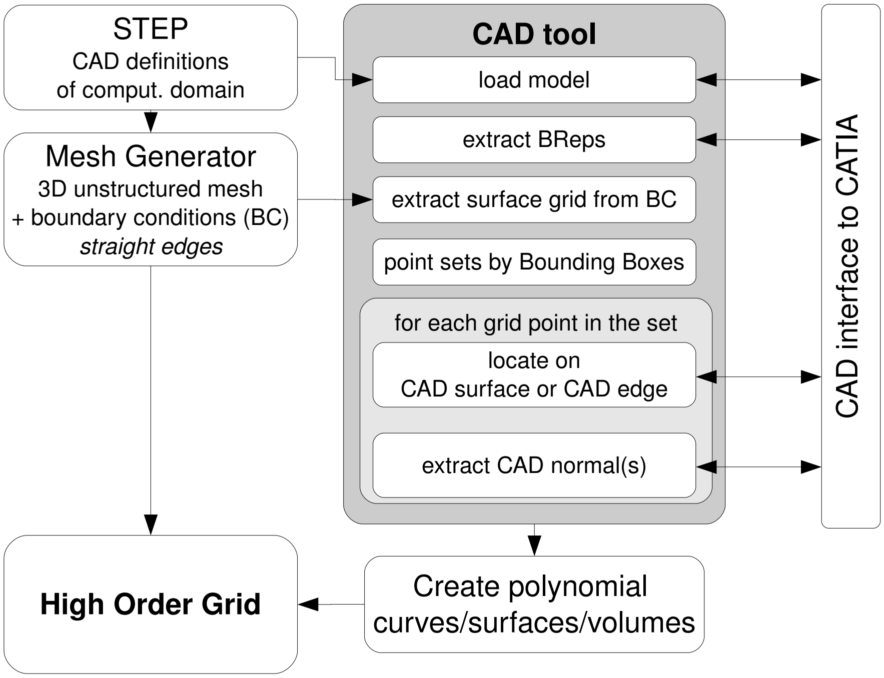

Figure 7: Flowchart of the high order grid construction

The workflow to construct a high order grid is shown in Figure 7. The geometry is normally defined in CAD and can be exported to a STEP file ('.stp'), a widely used standard exchange format. The grid generator imports the STEP file and provides, in general, straight-edged elements and boundary conditions (BCs).
In a preliminary step, the CAD tool reads the 3D grid, and extracts all grid points lying on boundaries, as well as the connectivity of the surface grid. Then the STEP file is loaded and the topology is analyzed, and for each CAD surface, a bounding box is created. For each CAD surface, the distance to the CAD surface and its edges is measured to decide whether or not the grid point lies on the surface or edge. To minimize the computational effort, only grid points inside the bounding box are considered. If the grid point lies inside the surface, one exact normal vector is evaluated at the grid point. Multiple exact normal vectors are found for CAD edges or corners. A unique Face ID is attached to every CAD surface, providing additional information to the normal vector. This facilitates curved edge construction afterwards, especially at sharp edges and corners. Once all grid points are checked, we construct the curved element edges, surface and volume mappings, as described in the previous sections.

5 APPLICATION TO CURVED GEOMETRIES

5.1 Free-Stream Preserving

CFD codes have to guarantee the so-called free-stream preserving property. It says that for a constant initial condition and a constant flux, the solution has to remain constant for all times. Choosing

$$\vec{\vec{F}}(U) = \vec{c} = \text{const.} \tag{32}$$

and following the integration by parts

$$\left\langle \vec{\nabla}\cdot\vec{F}(U),\phi_i\right\rangle_Q = \left\langle \vec{F}(U),\vec{\nabla}\phi_i\right\rangle_Q - \left(\vec{F}(U^h_-)\cdot\vec{n},\phi_i\right)_{\partial Q} \tag{33}$$

the condition reduces to

$$0 = \left\langle \vec{c},\vec{\nabla}\phi_i\right\rangle_Q - \left(\vec{c}\cdot\vec{n},\phi_i\right)_{\partial Q}\ , \tag{34}$$

which must hold *discretely*, i.e. both terms have to be integrated exactly, in particular for *curved elements.*
Inserting the volume mapping $\vec{x}(\vec{\xi})$ of degree P_c into the volume integral, we get

$$\left\langle \vec{c},\vec{\nabla}\phi_i\right\rangle_Q = \iiint\limits_{\xi_1\xi_2\xi_3} \underbrace{\vec{c}\cdot\vec{\nabla}\phi_i(\vec{x}(\vec{\xi}))}_{\leq(P-1)P_c}\cdot\underbrace{J(\vec{\xi})}_{\leq 3P_c-1}\,d\vec{\xi}, \tag{35}$$

where P is the polynomial degree of the basis functions. The number of 1D Gauss points needed for integration is given by maximum polynomial degree in one coordinate direction.

$$P_{\text{Gauss1D}} \leq PP_c + 2P_c - 1 \rightarrow N_{GP1D} \geq \frac{P_{\text{Gauss}}-1}{2}\,. \tag{36}$$

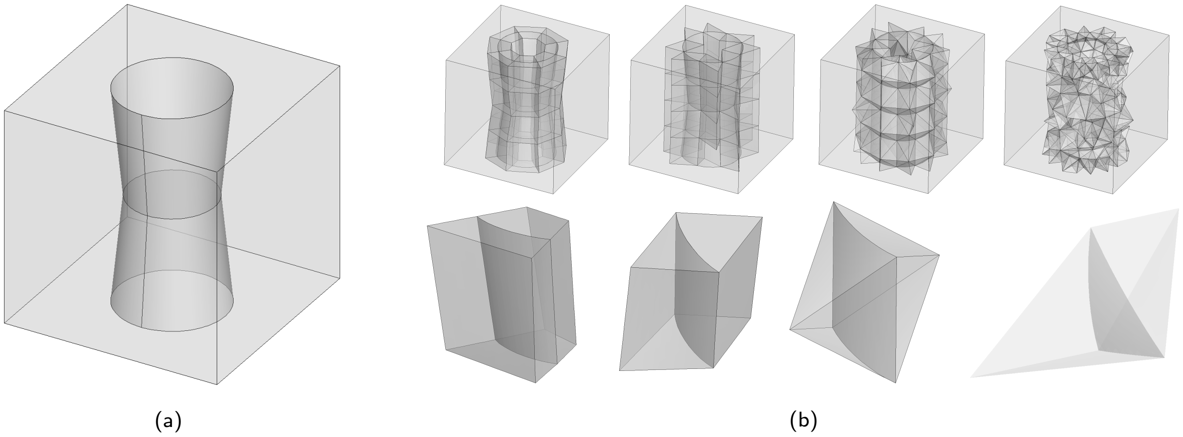

Figure 8: Test case geometry and curved elements, $P_c = 3$, $P = 3$

An element with $P = P_c = 3$ would require $N_{GP1D} = 7 \rightarrow 7^3$ Gauss points for integration. For the surface integral, we introduce the surface element $d\vec{S}$ and the non-normalized normal N

$$\vec{N}(u,v) = \frac{\partial \vec{x}_s(u,v)}{\partial u} \times \frac{\partial \vec{x}_s(u,v)}{\partial v} \qquad \vec{n} = \frac{\vec{N}}{|\vec{N}|} \qquad d\vec{S} = |\vec{N}(\vec{\xi})|\, d\xi_1\, d\xi_2$$

with $\vec{x}_s = \vec{x}_{bt}$ or $\vec{x}_s = \vec{x}_{bq}$ for a Béziertriangle (21) or quadrangle (26). Inserting the surface mapping $\vec{x}_s(u,v)$ of degree P_c into the surface integral, we get

$$(\vec{c} \cdot \vec{n}, \phi_i)_{\partial Q} = \iint\limits_{u\,v} \vec{c} \cdot \vec{n} \phi_i d\vec{S} = \iint\limits_{u\,v} \underbrace{\vec{c} \cdot \vec{N}(u,v)}_{\leq 2P_c - 1} \cdot \underbrace{\phi_i(\vec{x}_s(u,v))}_{\leq PP_c}\, d\xi_1\, d\xi_2\,. \tag{37}$$

Again, the number of 1D Gauss points is given as

$$P_{\text{Gauss1D}} \leq PP_c + 2P_c - 1\,. \tag{38}$$

Our implementation guarantees free-stream preserving to machine precision, when curved tetrahedra, pyramids, prisms or hexahedra are used, see Table 1. For the tests, a cube with a double frustum inside was meshed, and periodic boundary conditions were applied. The geometry is depicted in Figure 8a. In Figure 8b, only the curved elements are shown, furthermore a zoom of two adjacent elements.

	hexahedra	prisms	pyramids	tetrahedra
nElems	246	200	988	2575
$L_2(\rho)$	1.07E-012	2.58E-012	1.73E-012	4.23E-014

Table 1: Error in density, $P_c = 3$, $P = 3$

5.2 DLR-F6 wing-body-nacelle configuration

The DLR F6 is an idealized wing-body-nacelle configuration of a passenger jetliner, that was tested in multiple wind tunnels for comparison with numerical simulations. The CAD geometry as well as the experimental data can be found in Brodersen and Stürmer.[2] All mappings are represented by forth order polynomials. The volume grid ($\sim$ 340600 elements), is shown in Figure 10. The majority of the elements are linear tetrahedra ($\sim$ 280000). The fuselage is meshed with one layer of curved prisms ($\sim$ 3600), on the wing and nacelle, curved tetrahedra are used ($\sim$ 56000). Curved elements are more expensive to compute, thus the percentage of curved elements has to be minimized. In the grid, they are limited to the near surface region, a selection is depicted in Figure 11. The pictures in Figure 12 show the forth order surface mesh. It can be clearly seen that the CAD edges are detected overall and a smooth surface curving is achieved.
A useful measure for the quality of a curved element is the variation of the Jacobian $J(\vec{\xi})$, referred to as the *scaled Jacobian*[9]

$$J_{scaled} = \frac{\min_{\vec{\xi}\in Q} J(\vec{\xi})}{\max_{\vec{\xi}\in Q} J(\vec{\xi})} \,. \tag{39}$$

For linear elements (e.g. tetrahedron with straight edges) the Jacobian is constant, thus the scaled Jacobian equals 1. The scaled Jacobians range from -1 to 1, but only elements with $J_{scaled} > 0$ are valid, quality increases with $J_{scaled} \to 1$. The statistics for the mixed prism/tetrahedra grid are shown in Figure 9. All elements have positive Jacobians, whereas a scaled Jacobian smaller than 0.5 is found for 235 curved tetrahedra (0.45%) and 70 curved prisms (1.93%).

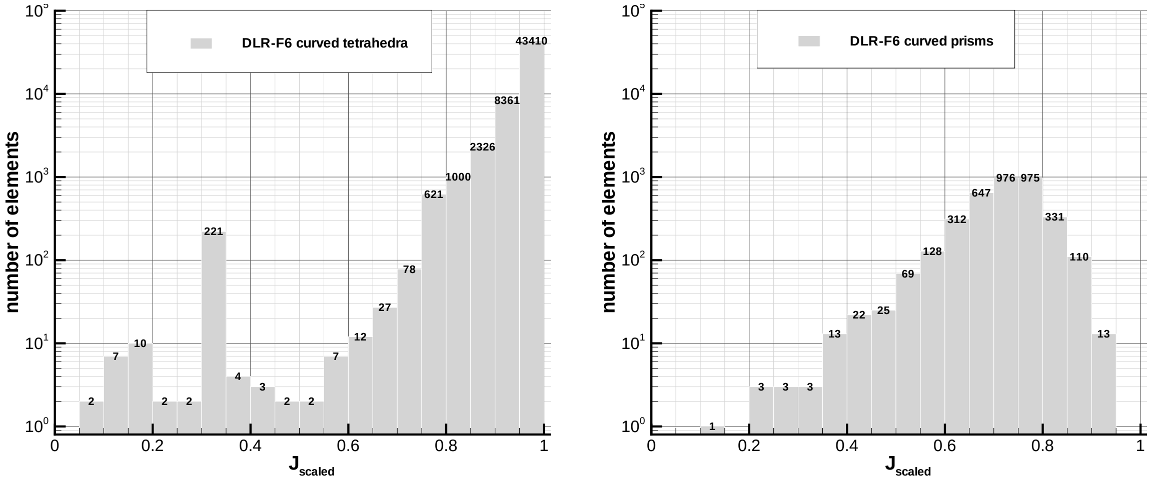

Figure 9: DLR-F6 configuration: scaled Jacobian statistics for curved tetrahedra and prisms

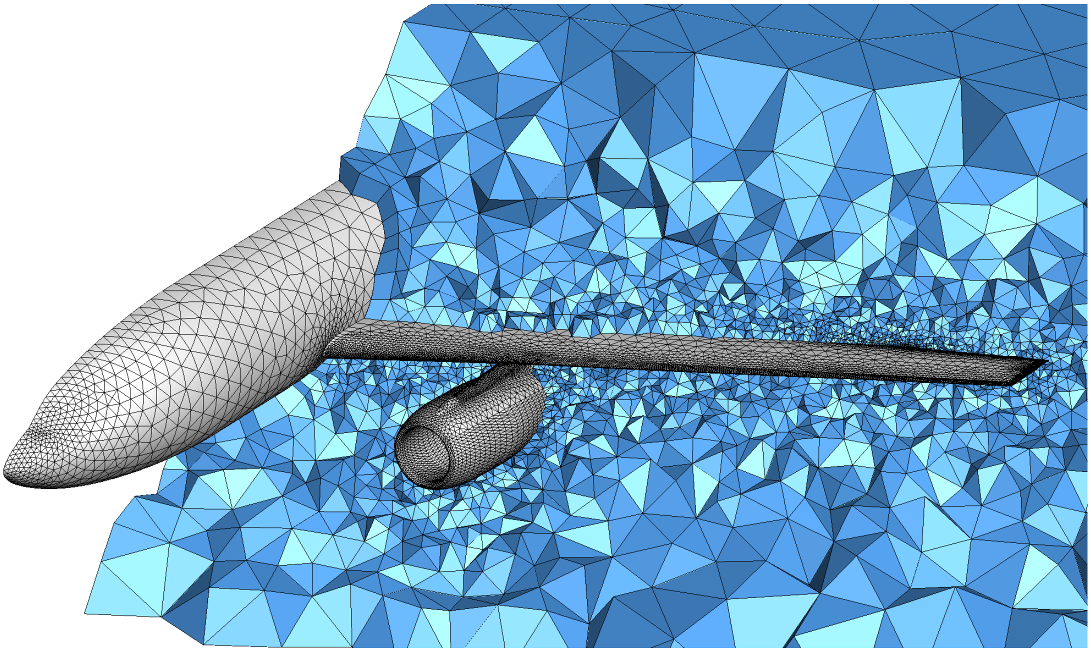

Figure 10: volume grid of the DLR-F6 configuration (tetrahedra and prisms)

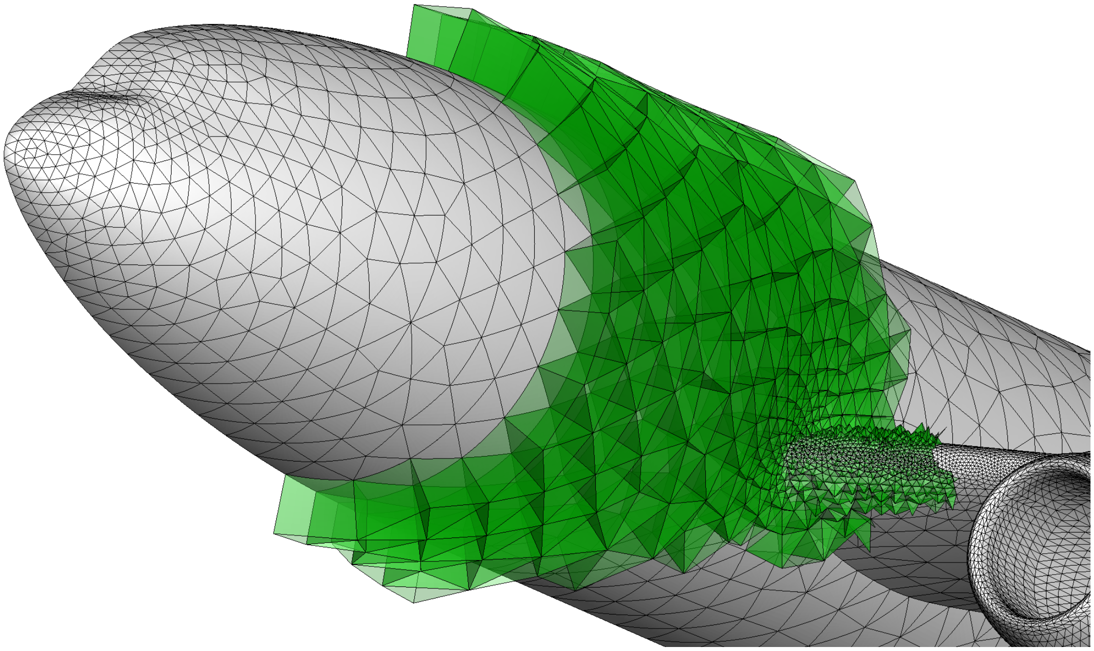

Figure 11: curved elements at the body surface

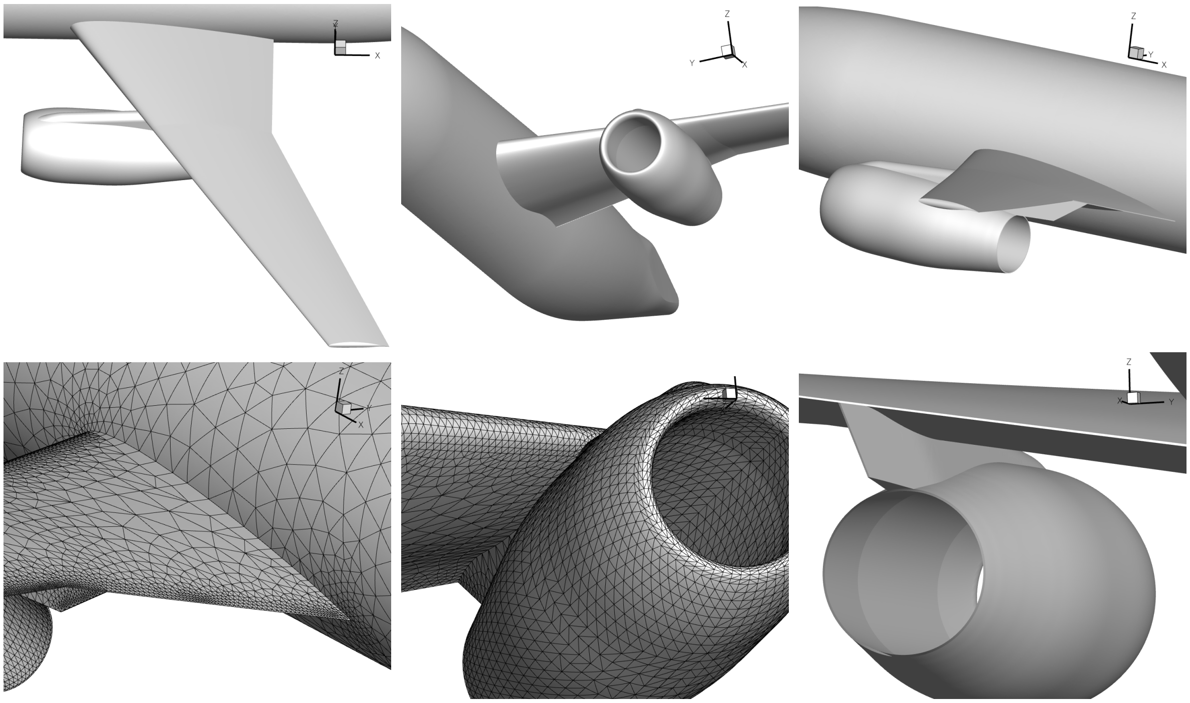

Figure 12: volume mapping of curved elements evaluated at the boundary surface

6 CONCLUSIONS

High order grid construction in complex three-dimensional geometries is required when high order methods are used. In this paper, a new approach is presented, relying on a CAD tool, which uses CAD definitions of the computational domain boundaries to find exact normals for the boundary grid points. The underlying grid can be generated by nearly every mesh generator, since only grids consisting of elements with straight edges are needed. We showed how the normals are used to define curved element edges. In particular, sharp edges at the borders of CAD surfaces are automatically found. Subsequently is was shown how the curved edges are extended to curved surface mappings and a furthermore to a curved volume mapping of the element. Until now, the volume mappings are restricted to forth order polynomials.

The DG scheme was derived for an arbitrarily shaped element. The approximation of the volume and surface integration are addressed by using polynomial mappings. Integration rules were found to guarantee the free-stream preserving property. A high order grid for a complex wing-body-nacelle reference configuration was presented and element Jacobians, a quality measure for curved elements, were analyzed.

7 ACKNOWLEDGEMENTS

We would like to thank the *Bundesministerium für Bildung und Forschung (BMBF)* and the *Cluster of Excellence SimTech* at the University of Stuttgart for the funding of projects related to this work.

REFERENCENCES

[1] F. Bassi and S. Rebay. High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys.*, 138(2):251–285, 1997.

[2] O. Brodersen and A. Stürmer. Drag prediction of engine–airframe interference effects using unstructured navier–stokes calculations. In *19th AIAA Applied Aerodynamics Conference*, number AIAA 2001-2414, Anaheim, California, 11-14 June 2001. AIAA. `http://aaac.larc.nasa.gov/tsab/cfdlarc/aiaa-dpw/Workshop2/DLR-F6-geom.html`.

[3] G. Farin. Smooth interpolation to scattered 3D data. In R. Barnhill and W. Böhm, editors, *Surfaces in CAGD*, pages 43–63. North-Holland, Amsterdam, 1983.

[4] G. Farin. *Curves and surfaces for CAGD: a practical guide.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[5] G. J. Gassner, F. Lörcher, C.-D. Munz, and J. S. Hesthaven. Polymorphic nodal elements and their application in discontinuous Galerkin methods. *Journal of Computational Physics*, 228(5):1573–1590, Mar. 2009.

[6] C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79:1309–1331, 2009.

[7] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp element methods for CFD.* Oxford University Press, New York, 1999.

[8] D. Lasser. Bernstein-Bézier representation of volumes. *Computer Aided Geometric Design*, 2(1-3):145–149, 1985.

[9] P.-O. Persson and J. Peraire. Curved mesh generation and mesh refinement using lagrangian solid mechanics. *Proc. of the 47th AIAA Aerospace Sciences Meeting and Exhibit*, 2009.