Volume Mesh Generation for Numerical Flow Simulations using Catmull-Clark and Surface Approximation Methods

Michael Rom[1,2] and Karl-Heinz Brakhage[1]

[1] Institut für Geometrie und Praktische Mathematik, RWTH Aachen, 52056 Aachen, Germany, {rom,brakhage}@igpm.rwth-aachen.de
[2] German Research School for Simulation Sciences, 52425 Jülich, Germany

Summary. We present a new technique for the semi-automatic generation of volume meshes which can be used for numerical flow simulations. Our aim is to end up with a fine block-structured volume mesh connected to a smooth surface mesh. For this purpose we start with a polyhedron giving a rough approximation of the target surface geometry which can be of arbitrary genus. To this initial polyhedron we apply an alternating iterative process of Catmull-Clark subdivisions and approximations of the target surface. Usually, such a surface is given by a collection of trimmed B-spline surfaces so that the best approximation results can be achieved by projecting the points of the Catmull-Clark limit surface, which converges to uniform bi-cubic B-spline patches, onto it and subsequently recomputing the control mesh. If we construct another polyhedron surrounding the initial polyhedron, both being automatically connected to each other, we can perform three-dimensional Catmull-Clark subdivision to the flow field between the inner and the outer surface.

1 Introduction, Significance and Related Work

Numerical flow simulations require high-quality volume meshes. The generation of such meshes as well as the construction of the object around or through which the flow should be simulated can be very difficult and time-consuming. We have developed a new promising approach by combining established methods for 2D- and 3D-subdivision with new algorithms for the limit point computation of a subdivision surface, the projection of the limit points onto given B-spline surfaces and the approximation of these given surfaces to get new surface mesh control points (= vertices). The process is illustrated in Fig. 1. The user only has to construct a simple inner initial surface polyhedron, being roughly similar to the target surface of arbitrary genus, and an outer surrounding polyhedron by giving the coordinates, the face connectivity and optionally edges which should be treated like creases. These two surfaces are connected automatically leading to a coarse volume mesh representing the flow field.

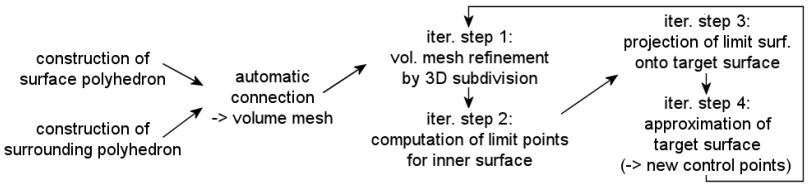

Fig. 1. Process of volume mesh generation

The subsequent iterative process can be stopped if the approximation of the target surface is satisfying and the volume mesh is fine enough.

For the 2D-subdivision we use the scheme presented by Catmull and Clark [1]. Due to the convergence of the Catmull-Clark limit surface to uniform bi-cubic B-spline patches (see [1]) we get a smooth C^2-surface with the exception of points where no tensor-product topology is given (C^1 there). The results of Stam's analysis of the subdivision matrix [2] with an extension for modeling creases [3] allow for the pre-calculation of the limit surface points at each refinement level. We apply 3D-subdivision by using a combination of the schemes presented by Joy and MacCracken [4] and Bajaj et al. [5].

2 Background on Catmull-Clark Subdivision

The Catmull-Clark scheme [1] can be applied to surface meshes that are not regular rectangular grids. A refinement step is defined by the following rules:

1. Add a point for each face: $\mathbf{F} = \frac{1}{N}\sum_{i=0}^{N-1}\mathbf{P}_i$, average of the N face vertices.
2. Add a point for each edge: $\mathbf{E} = \frac{1}{4}(\mathbf{F}_{left} + 2\mathbf{E}_c + \mathbf{F}_{right})$, weighted average of the two adjacent face points $\mathbf{F}_{left}$, $\mathbf{F}_{right}$ and the edge midpoint $\mathbf{E}_c$.
3. Move each old vertex: $\mathbf{P}_{new} = \frac{1}{N}(\tilde{\mathbf{F}} + 2\tilde{\mathbf{E}}_c + (N-3)\mathbf{P})$, weighted average of the vertex $\mathbf{P}$, the average $\tilde{\mathbf{F}}$ of the adjacent face points and the average $\tilde{\mathbf{E}}_c$ of the adjacent edge midpoints; N: number of edges connected to $\mathbf{P}$.

Finally, the new face points are connected to the new adjacent edge points.

These 2D-rules are applied to the inner and the outer surface in iteration step 1 (see Fig. 1), while the space in between is refined by 3D-subdivision. We use the 3D-rules given in the list below demanding the volume mesh to contain only hexahedra composed of eight vertices, six faces and twelve edges. This is guaranteed if the inner surface mesh exclusively consists of quadrilaterals before connecting it to the outer one. Since the first Catmull-Clark subdivision always provides a quadrilateral mesh, this requirement can be fulfilled easily.

1. Add a point for each cell: $\mathbf{C} = \frac{1}{8}\sum_{i=0}^{7}\mathbf{P}_i$, average of the eight cell vertices.
2. Add a point for each face: $\mathbf{F} = \frac{1}{4}(\mathbf{C}_{left} + 2\mathbf{F}_c + \mathbf{C}_{right})$, weighted average of the two adjacent cell points $\mathbf{C}_{left}$, $\mathbf{C}_{right}$ and the face centroid $\mathbf{F}_c$.

3. Add a point for each edge: $\mathbf{E} = \frac{1}{N}(\tilde{\mathbf{C}} + 2\tilde{\mathbf{F}}_c + (N-3)\mathbf{E}_c)$, weighted average of the edge midpoint $\mathbf{E}_c$, the average $\tilde{\mathbf{C}}$ of the adjacent cell points and the average $\tilde{\mathbf{F}}_c$ of the adjacent face points; N: number of adjacent faces.
4. Move each old vertex: $\mathbf{P}_{new} = \sum_{\mathbf{U}_i \in ring(\mathbf{P})} \frac{3^{d-dim(\mathbf{U}_i,\mathbf{P})}}{4^d\ val(\mathbf{P})}\, val(\mathbf{U}_i, \mathbf{P})\, \mathbf{U}_i$ with $ring(\mathbf{P})$ denoting the set of vertices $\mathbf{U}_i$ connected to $\mathbf{P}$ by a cell, a face or an edge. The vertex $\mathbf{P}$ itself is stored in $ring(\mathbf{P})$, too. $val(\mathbf{P})$ is the number of cells containing $\mathbf{P}$, whereas $val(\mathbf{U}_i, \mathbf{P})$ gives the number of cells containing both $\mathbf{U}_i$ and $\mathbf{P}$. $dim(\mathbf{U}_i, \mathbf{P})$ specifies the connection between $\mathbf{U}_i$ and $\mathbf{P}$ (0 if $\mathbf{U}_i = \mathbf{P}$, 1 if $\mathbf{U}_i$ and $\mathbf{P}$ lie on a common edge, 2 if they only lie on a common face or 3 if they only lie on a common cell).

The first three of these 3D-rules were presented by Joy and MacCracken [4]. Since their vertex recomputation formula (rule 4) does not consider an adjustment for the number of edges a vertex is connected to, we use the equation developed by Bajaj et al. [5] instead.

Finally, the new cell points are connected to the new adjacent face points. In turn, these are connected to the new adjacent edge points.

A volume meshing example is given in Fig. 2. Starting from a simple wing configuration (1), an outer polyhedron surrounding the wing is generated manually (2). These two surfaces are connected by edges and faces automatically (3), so that 3D-subdivision can be applied (4, mesh after two subdivisions).

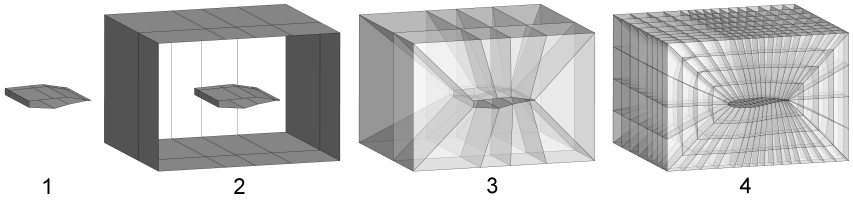

Fig. 2. Volume meshing example (3 and 4: translucent view)

3 Limit Points, Projection and Approximation

A point $\mathbf{L}_i$ of the Catmull-Clark limit surface corresponding to a control point $\mathbf{P}_i^{(k)}$ of the mesh is defined by $\mathbf{L}_i = \lim_{k\to\infty} \mathbf{P}_i^{(k)}$, i.e., the position after $k \to \infty$ subdivisions. For the approximation of given surfaces with our Catmull-Clark meshes we can pre-compute the values $\mathbf{L}_i$ of the limit surface (see iter. step 2 in Fig. 1) by using Stam's algorithm for evaluating the Catmull-Clark scheme at arbitrary points (see [2]) and our extension [6] for the modified rules given in [3]. The limit points $\mathbf{L}_i$ can be written as $\mathbf{L}_i = \mathbf{c}_i^T\, \mathbf{V}^i$ where we have collected the weights c_j of the involved vertices $\mathbf{P}_j^{(k)}$ in the vicinity of the control point $\mathbf{P}_i^{(k)}$ in the vector $\mathbf{c}_i$ and the associated vertices $\mathbf{P}_j^{(k)}$ in $\mathbf{V}^i$.

For a given surface s we can project $\mathbf{L}_i$ onto it: $\mathbf{L}_i \rightarrow \mathbf{L}_i^s$ (iter. step 3 in Fig. 1). Usually, the surface s is given by a B-spline representation defined by $\mathbf{x}(u,v) = \sum_{r=0}^{m} \sum_{s=0}^{n} \mathbf{x}_{r,s} N_{r,p}(u) N_{s,q}(v)$ where $(m+1) \times (n+1)$ gives the number of control points and p and q denote the degree of the B-spline basis functions $N_{r,p}(u)$ and $N_{s,q}(v)$, respectively. Together with knot vectors in u- and v-direction $\mathbf{x}(u,v)$ and the partial derivatives $\mathbf{x}_u(u,v)$ and $\mathbf{x}_v(u,v)$ can be evaluated by applying de Boor's algorithm [7]. Given such a representation, we search for parameter values u_{L_i} and v_{L_i} for each limit point $\mathbf{L}_i$ such that $\|\mathbf{x}(u_{L_i}, v_{L_i}) - \mathbf{L}_i\|_2 = \min_{u,v} \|\mathbf{x}(u,v) - \mathbf{L}_i\|_2$. With initial values (u_0, v_0) we can formulate an iterative process to solve this minimization problem. The tangent plane is given by $\mathbf{t}(u_k, v_k) = \mathbf{x}(u_k, v_k) + \Delta u_k\, \mathbf{x}_u(u_k, v_k) + \Delta v_k\, \mathbf{x}_v(u_k, v_k)$. The problem now is equivalent to dropping a perpendicular from $\mathbf{L}_i$ to the tangent plane. It is given by $\mathbf{L}_i + \mu_k \mathbf{n}_k = \mathbf{x}(u_k, v_k) + \Delta u_k \mathbf{x}_u(u_k, v_k) + \Delta v_k \mathbf{x}_v(u_k, v_k)$ where $\mathbf{n}_k$ denotes the unit normal to the tangent plane. We solve this for $(\Delta u_k, \Delta v_k)$ and compute the update using a damping factor λ_k:
$u_{k+1} = u_k + \lambda_k \Delta u_k, \;\; v_{k+1} = v_k + \lambda_k \Delta v_k$.

Finally, we get an equation of the form $\mathbf{c}_i^T\, \mathbf{V}^i \stackrel{!}{=} \mathbf{L}_i^s$ with $\mathbf{L}_i^s = \mathbf{x}(u_{L_i}, v_{L_i})$. We use the notation $C\mathbf{V} \stackrel{!}{=} \mathbf{L}^s$ for the approximation problem $\|C\mathbf{V} - \mathbf{L}^s\|_2 \rightarrow$ min and apply it for a single equation of this system, too. Sampling enough projection points we end up with an over-determined sparse linear system for approximation (iter. step 4 in Fig. 1) which we solve by applying the **c**onjugate **g**radient method for linear **l**east **s**quares (CGLS or also called CGNR [8]). This leads to new control point positions with better approximation properties for the next iteration loop.

The wing depicted in Fig. 3 is used for a test of our algorithm. Its geometry consists of two B-spline representations, a wing being C^1-connected to a wing tip. Figure 4 shows the initial polyhedron (top) and the result of the iterative process from Fig. 1 with 2D-subdivision only (middle and bottom).

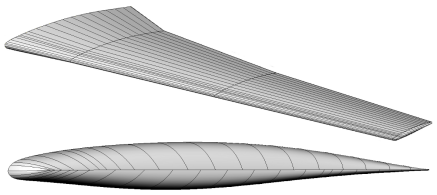

Fig. 3. Top: B-spline surface (wing + wing tip, knot-isolines visible), **bottom**: close-up of wing tip (knot-isolines vis.)

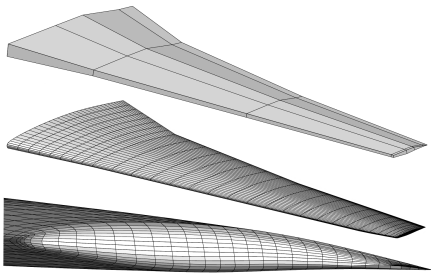

Fig. 4. Top: Initial polyhedron, **middle**: surface mesh after three iterations, **bottom**: wing tip after three iterations

4 Conclusion and Future Work

We have presented a fast iterative procedure for the generation of high-quality block-structured volume meshes with an inner surface well conforming to a given target surface. Apart from the manual generation of two initial polyhedra the whole process operates automatically.

We will evaluate the quality of our implementation by using the resulting volume meshes for numerical flow simulations and comparing the results to experimental data. From the sub-project *High Reynolds Number Aero-Structural Dynamics (HIRENASD)* of the SFB 401 *Flow Modulation and Fluid-Structure Interaction at Airplane Wings* at RWTH Aachen (see [9]) we have data from wind tunnel readings for the wing in Fig. 3 attached to a simplified fuselage. Furthermore, from the follow-up project *Aero-Structural Dynamics Methods for Airplane Design (ASDMAD)* (see [10]), a cooperation with Airbus, we have additional experimental results for a wing with a winglet.

Acknowledgments

The first author is funded by the German Research School for Simulation Sciences (GRS), Jülich, Germany.

References

1. J. Clark and E. Catmull. Recursively generated B-spline surfaces on arbitrary topological meshes. *CAD*, 10(6):350–355, 1978.
2. J. Stam. Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values. In *Proceedings of SIGGRAPH*, pages 395–404, 1998.
3. T. DeRose, M. Kass, and T. Truong. Subdivision Surfaces in Character Animation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques, ACM SIGGRAPH*, pages 85–94, 1998.
4. K. Joy and R. MacCracken. The Refinement Rules for Catmull-Clark Solids. Technical Report CSE-96-1, Department of Computer Science, University of California, Davis, 1996.
5. C. Bajaj, S. Schaefer, J. Warren, and G. Xu. A subdivision scheme for hexahedral meshes. *The Visual Computer*, 18:343–356, 2002.
6. M. Rom. Oberflächenreparametrisierung und Gittererzeugung für numerische Strömungssimulationen mit Hilfe von Catmull-Clark-Methoden. Diplomarbeit, RWTH Aachen, 2009.
7. C. de Boor. *A Practical Guide to Splines*. Springer, 1978.
8. Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2nd edition, 2003.
9. W. Schröder, editor. *Summary of Flow Modulation and Fluid-Structure Interaction Findings*. Notes on Numerical Fluid Mechanics and Multidisciplinary Design, Vol. 109. Springer, 2010.
10. J. Ballmann, K.-H. Brakhage et al. Aero-Structural Wind Tunnel Experiments with Elastic Wing Models at High Reynolds Numbers (HIRENASD - ASDMAD). 49th AIAA Aerospace Sciences Meeting, Orlando, Florida, 2011.