Grid Generation and Grid Conversion by Subdivision Schemes

Karl-Heinz Brakhage

Institute for Geometry and Applied Mathematics
RWTH Aachen University
D-52056 Aachen
brakhage@igpm.rwth-aachen.de

Abstract

In this paper we present new algorithms for optimal fitting with subdivision schemes. We use an iterative numerical scheme for solving the arising linear system for the least squares problem. This scheme avoids the explicit set up of the normal equations and the amount of work for one approximation step is proportional to the number of vertices of the subdivision surface. Together with this algorithms subdivision schemes can be used for modelling and grid generation in aeronautical projects. To make the schemes really usable for this purposes their rules have to be modified in such a way that the limit surfaces have the desired properties. Furthermore fast algorithms for evaluation and approximation of existing surfaces have to be developed.

1. Introduction

Subdivision surfaces are normally used for modeling in CAGD and in Computer Graphics. Our aim is to solve interpolation and approximation problems with this methods. Thus we do not only need values at certain points but also need the coefficients of the involved mesh points to end up with a sparse linear system for an initial mesh. In case of the Catmull-Clark scheme we give a detailed analysis to set up the system needed for solving the least squares problem, which can be solved efficiently by iterative methods. Furthermore we will demonstrate how this concept is used for the generation of numerical grids for realistic wing-fuselage configurations in the Collaborative Research Center SFB401 Flow Modulation and Fluid Structure Interaction at Airplane Wings at the RWTH Aachen.

In the literature on subdivision (see [1] for example) it is a widespread opinion that it is too time consuming to solve the full least squares problem for the approximation with Catmull-Clark schemes. Thus very often local quasi-interpolation is used instead. We will disprove this opinion and show
that the use of adaptive iterative methods for the solution of the least squares problem leads to a tremendous reduction of computation time. We use CGLS, a Conjugate Gradient method for linear Least Squares (also called CGNR in [2]). This method makes extensive use of the sparsity of the system matrix. If \( n \) is the number of points to be approximated then this method needs \( O(n) \) time.

At least fifty years ago the first papers on subdivision were published. In 1947 de Rham’s first articles on corner cutting appeared [3]. He suggested to do corner cutting with ratio 1 : 1 : 1 (compare Figure 1 left side). The outer polygon \( \{P_0, P_1, \ldots, P_{n-1}\} \) is subdivided two times with the edge ratio 1:1:1. The result is the inner curve (polygon \( \{R_0, R_1, \ldots, R_{2n-1}\} \)). We use the sequence \( P, Q, R, \ldots \) instead of \( P^0, P^1, P^2, \ldots \) to reduce the number of indices. Afterwards the outer parts (corners) are cut away. The resulting polygon is the input for the next same step. Since the type of the refinement rule for this task is a vertex split we call it a vertex split scheme. Later de Rham generalized the scheme to make the limit curve GC-1 (this is C-1 (continuously differentiable) after reparameterization). The relevance of such algorithms to curve generation became clear with the paper of the graphics artist G. Chaikin [4] in 1974.

![Figure 1. Corner cutting of de Rham 1947 (left side) and Cubic edge splitting scheme (two steps both)](image)

Further investigations showed that uniform B-spline curves of any degree would have such a subdivision construction. This is shown for the cubic case in Figure 1 on the right hand side. In this method the edges are split into two pieces and hence it is called edge split scheme. We have new vertices on the edges and recomputed old ones. In this case the corresponding formulae are
\[ Q_0 = (P_{n-1} + 6 \cdot P_0 + P_1)/8 \]
\[ Q_1 = (P_0 + P_1)/2 \]
\[ \vdots \]
\[ Q_{2i} = (P_{i-1} + 6 \cdot P_i + P_{i+1})/8 \]
\[ Q_{2i+1} = (P_i + P_{i+1})/2 \] (1)

In each step all edges are split into two parts. The new points on the edges are constructed by ratio 1 : 1 of the adjacent vertices. The old vertices are recomputed with the ratio 1 : 6 : 1.

The first subdivision schemes for surfaces were introduced in 1978. Doo and Sabin [5] established a quadratic subdivision scheme which is a direct extension of the vertex split method of de Rham / Chaikin. Furthermore they analyzed the behavior of such schemes. This was done by writing the subdivision process in form of matrix multiplications and studying the eigenvalues and eigenvectors of the subdivision matrices. For further information on the history of subdivision schemes we refer to the survey paper of Malcolm Sabin: Recent Progress in Subdivision: a Survey (6)

Catmull and Clark [7] published their descriptions of quadratic and cubic subdivision surfaces in 1978 too. The latter one regards to the curves generated according to (1). Contrary to tensor-product splines this scheme can be applied to meshes that are not regular rectangular grids (see Figure 2). The number of edges that meet at a vertex is called the valence of that point. We denote it with \( N \). The (new) face points are the barycenters of the vertices of each face. This is denoted by averaging with the weights 1 for every face vertex. Next the edge points are computed using the new face points according to the given weights. In the last vertex step the (old) vertices are recomputed. Again the weights are given in the figure. Notice
that in every case we have to divide by the sum of the weights. After the vertex steps (face→vertex, edge→vertex and vertex→vertex) illustrated in Figure 2 the new edges are build by splitting the old ones (edge→edge) and connecting the new face points with the new edge points (face→edge). The new faces are the faces inside the old ones (face→face). The right plot of Figure 3 can be thought of as an example for a pentagon. The centroid is the new face vertex, the vertices with weight 6 are the new edge vertices and the the vertices with weight 1 are the recomputed vertices. In a regular rectangular grid the valence of all interior vertices is \( N = 4 \). Subdivision schemes that have the same coefficients/rules in every level are called stationary.

In our modelling and grid generation concepts (see [8]) we are interested in ending up with smooth untrimmed B-spline patches. Thus for our purposes the Catmull-Clark scheme is the method of choice. We summarize its crucial properties:

- The surfaces can be of arbitrary genus since the subdivision rules can be carried out on a mesh of arbitrary topological type.
- After one subdivision step all faces are quadrilaterals.
- Except at extraordinary vertices (vertices of valence \( N \neq 4 \)) the limiting surface converges to uniform bi-cubic B-Spline patches. Hence the surface is \( C^2 \) except at extraordinary vertices.
- The number of extraordinary vertices is fixed after the first subdivision step.
- After two subdivision steps all faces have at most one extraordinary vertex.
- Near an extraordinary vertex the surface can be shown to have a well defined tangent plane at the limit point, but the curvature there is generally not well defined.
- The subdivision rules can be modified in such a way that they generate infinitely sharp creases as well as semi-sharp creases, i.e. creases whose sharpness can vary from zero (meaning smooth) to infinite. See de Rose et. all [9] (1998) for more details.

Because at least after one subdivision step there are only quadrilaterals we show the rules for this case in Figure 3. The weight for the recomputed extraordinary vertex of valence \( N \) is \( A(N) = 4N^2 - 7N \). Again we have to
divide by the sum of the weights in every case. The correspondence to cubic B-splines can easily be seen. Notice that for the regular case $A(4) = 36$.

2. Analysis of subdivision schemes

The ideas of Doo and Sabin [5] using eigenanalysis of the subdivision matrix were advanced by Ball and Storry from 1984 to 1988 (in [10], [11] and [12]) and later by Stam [13] in 1998. Stam additionally gave an algorithm for evaluating the Catmull-Clark-Scheme (and its derivatives) at arbitrary points. He used a choice of ordering for the control vertices that the main part of the subdivision matrix has a cyclical structure. Hence the discrete Fourier transform can be used to compute its eigenstructure. For the modified rules given in [9] this analysis is very technical. It is currently being investigated in more detail and implemented in a master thesis [14] at our institute. We restrict the explanation to the unmodified rules but give examples for the modified ones.

For our practical objects we can restrict the valence $N$ of the extraordinary vertices to $N \in \{3, 5, 6\}$. For these cases we use a more intuitive numbering and pre-compute the eigenstructure with numerical methods. Our aim is to solve interpolation and approximation problems for this methods. Thus we do not only need values at certain points but also need the coefficients of the involved mesh points. For this reason we additionally build up tables for the interpolation and approximation schemes for valence $N \in \{3, 4, 5, 6\}$.

For surfaces and even for valences $N \leq 6$ the subdivision matrices are of high dimension (see Table 1). Therefore we explain the ideas for surfaces and give the analysis for curves. The main difference between curves and surfaces is the existence of extraordinary points. The polygon for the curve subdivision can be evaluated everywhere at each step by well known B-spline algorithms whereas the surface mesh can not.
Table 1. Dimension of subdivision matrices at extra ordinary points

<table>
<thead>
<tr>
<th>valence</th>
<th>rows</th>
<th>columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>23</td>
<td>14</td>
</tr>
<tr>
<td>4 regular</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>27</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>29</td>
<td>20</td>
</tr>
<tr>
<td>(N)</td>
<td>(2N + 1 + 7 + 9)</td>
<td>(2N + 1 + 7)</td>
</tr>
</tbody>
</table>

The control vertex structure near an extraordinary point is not a simple rectangular grid, thus all faces that contain extraordinary vertices cannot be evaluated as uniform B-splines. Since after one subdivision step all faces are quadrilaterals and after two steps all faces have at most one extraordinary vertex, we assume for our mesh that each face is a quadrilateral and contains at most one extraordinary vertex. Figure 4 shows that the region where the surface can not be evaluated with standard methods is scaled down in every subdivision step.

Figure 4. Behavior near an extraordinary point of valence \(N = 3\).

Since we can evaluate the surface at extraordinary vertices the remaining problem is to demonstrate only how to evaluate a patch corresponding to a face with just one extraordinary vertex, such as the dark region shown in Figure 4. Analogue to curves (compare Figure 5) we introduce parameter values and define a surface patch \(x(u, v)\) over the unit square \([0, 1] \times [0, 1]\) such that the point \(x(0, 0)\) corresponds to the extraordinary vertex. We can evaluate the surface at those vertices \((x(0, 0))\) as a linear combination of the circumfluent vertices (the same ones we need for the re-computation step vertex→vertex during subdivision, but with different weights). Additionally we can evaluate \(x(u, 1)\) for \(u \in [0, 1]\) and \(x(1, v)\) for \(v \in [0, 1]\) as regular B-spline part. The remaining problem is the evaluation \(x(u, v)\) in the rest of the unit square. This problem is solved in the following way. First we do just enough subdivision steps such that \((u, v)\) corresponds to a regular part at that stage. We then pick the right vertices and do the
evaluation as regular B-spline.

Figure 5. Cubic edge splitting scheme – one step and limit curve $x(t)$.

We will give the analysis for curves now. Figure 5 shows one step of the cubic edge splitting scheme and the limit curve $x(t)$ in $t \in [0, 1]$. The control points $P_0, P_1, P_2, P_3$ determine the curve between $C_0 = (P_0 + 4 \cdot P_1 + P_2)/6$ and $C_2 = (P_1 + 4 \cdot P_2 + P_3)/6$. We say that $C_0 = x(0)$ corresponds to $P_1$ and $C_2 = x(1)$ to $P_2$. After one subdivision step $C_1 = x(0.5)$ can be computed as $C_1 = (Q_1 + 4 \cdot Q_2 + Q_3)/6 = (P_0 + 23 \cdot P_1 + 23 \cdot P_2 + P_3)/48$. Imagining $P_1$ as an extraordinary vertex we can not compute $x(t)$, $t \in [0, 1]$ by standard B-Spline methods cause we have to cross $P_1$ and for this thought experiment we have an analogon to the surface case. The subdivision matrix $A_{00}$ corresponding to $P_1$ is given by

$$A_{00} = \begin{pmatrix} 1/2 & 1/2 & 0 \\ 1/5 & 2/5 & 1/5 \\ 0 & 1/2 & 1/2 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \end{pmatrix} = A_{00} \begin{pmatrix} P_0 \\ P_1 \\ P_2 \end{pmatrix}. \quad (2)$$

The eigenvalues of $A_{00}$ are 1, 1/2 and 1/4. For every of our subdivision schemes there is a unique largest eigenvalue 1, all eigenvalues are real and the matrices are diagonalizable (equal algebraic and geometric multiplicities for all eigenvalues). One matrix that has as columns the eigenvectors of $A_{00}$ is

$$V_{00} = \begin{pmatrix} 1 & -1 & -2 \\ 1 & 0 & 1 \\ 1 & 1 & -2 \end{pmatrix} \Rightarrow V_{00}^{-1} = \begin{pmatrix} -1/2 & 2/3 & 1/6 \\ 1/2 & 0 & 1/2 \\ 1/6 & 1/3 & -1/6 \end{pmatrix} \quad (3)$$

and thus

$$V_{00}^{-1} A_{00} V_{00} = \text{diag} \left\{ 1, \frac{1}{2}, \frac{1}{4} \right\} = \Lambda_0. \quad (4)$$
On the other hand this yields \( A_{00} = V_{00} \Lambda_0 V_{00}^{-1} \). Furthermore we have \( A_{00}^n = (V_{00} \Lambda_0 V_{00}^{-1})^n = V_{00} \Lambda_0^n V_{00}^{-1} \) and \( \lim_{n \to \infty} A_{00}^n = \text{diag} \{1, 0, 0\} =: \Lambda_0^\infty \). Now we can conclude

\[
A_{00}^\infty := \lim_{n \to \infty} A_{00}^n = V_{00} \Lambda_0^\infty V_{00}^{-1} = \begin{pmatrix}
\frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\
\frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\
\frac{1}{6} & \frac{2}{3} & \frac{1}{6}
\end{pmatrix}.
\tag{5}
\]

If we apply the subdivision for the curve case on \( P_0, P_1, P_2 \) this yields

\[
C_0 = x(0) = \frac{1}{6} P_0 + \frac{2}{3} P_1 + \frac{1}{6} P_2
\tag{6}
\]

and we have proven the consistency to uniform B-spline curves. Due to the fact that \( \lambda_0 = 1 > \lambda_1 = 1/2 > \lambda_2 = 1/4 \) we can already conclude from the given inverse in (3) that

\[
\begin{align*}
x(0) &= \frac{1}{6} P_0 + \frac{2}{3} P_1 + \frac{1}{6} P_2 \\
x'(0) &= \alpha_1 \left( -\frac{1}{4} P_0 + \frac{1}{2} P_2 \right) \\
x''(0) &= \alpha_2 \left( -\frac{1}{6} P_0 + \frac{3}{2} P_1 - \frac{1}{6} P_2 \right)
\end{align*}
\tag{7}
\]

If we want to do the same computation for \( C_1 = x(0.5) \) we have to enlarge the subdivision matrix \( A_{00} \) to \( A \) with

\[
A = \begin{pmatrix}
A_{00} & 0 \\
A_{10} & A_{11}
\end{pmatrix} = \begin{pmatrix}
\frac{1}{2} & \frac{1}{2} & 0 & 0 \\
\frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 0 \\
0 & \frac{1}{2} & \frac{1}{2} & 0 \\
0 & \frac{1}{3} & \frac{1}{4} & \frac{1}{5}
\end{pmatrix}.
\tag{8}
\]

The new rows directly follow from the usual B-spline knot-insertion rules for curves and surfaces. We make use of the block structure to compute the eigenvalues and eigenvectors of \( A \). First it induces

\[
V = \begin{pmatrix}
V_{00} & 0 \\
V_{10} & V_{11}
\end{pmatrix} \quad \text{and} \quad \Lambda = \begin{pmatrix}
\Lambda_0 & 0 \\
0 & \Lambda_1
\end{pmatrix}
\tag{9}
\]

as structure for the equation \( AV = V \Lambda \). Inserting (9) and expanding the last equation yields

\[
\begin{pmatrix}
A_{00} V_{00} & 0 \\
A_{10} V_{00} + A_{10} V_{10} & A_{11} V_{11}
\end{pmatrix} = \begin{pmatrix}
V_{00} \Lambda_0 & 0 \\
V_{10} \Lambda_0 & V_{11} \Lambda_1
\end{pmatrix}
\tag{10}
\]
and we can compute the remaining eigenvalues and the eigenvector block $V_{11}$ with $A_{11}$. For surfaces $A_{11}$ is a $7 \times 7$ matrix and independent of the valence $N$. Hence it can be computed in advance and used for any valence. The remaining part is the extension $V_{10}$ of the eigenvectors in $V_{00}$. $V_{10}$ is a $7 \times 2N + 1$ matrix and can be computed by solving the $2N + 1$ linear systems

$$A_{10} V_{00} + A_{11} V_{10} = V_{10} \Lambda_0$$  \hspace{1cm} (11)

which in more detail is $(A_{11} - \Lambda_0(j) \cdot I_7) V_{10}(j) = -A_{10} V_{00}$. In the last equation we have used $\Lambda_0(j)$ for the $j^{th}$ diagonal element of $\Lambda_0$, $I_7$ for $7 \times 7$ identity matrix and $V_{10}(j)$ for the $j^{th}$ column of $V_{10}$.

In our thought experiment for curves (see (8)) we have $A_{11} = (1/8) = \Lambda_1$ and $V_{11} = (1)$. Solving equation (11) for $V_{10}$ we finally get $V_{10}$, $V = \Lambda$ and $A^n = (V \Lambda V^{-1})^n = V \Lambda^n V^{-1}$. Repeating the above process we can compute $x(0.25)$ etc. But if we want to compute values in $(0.5, 1)$ we have to compute $Q_4$ and thus need another subdivision step given by an usual B-spline knot-insertion rule. Notice that we do not need further vertices as input. Our matrix for such a step is

$$\hat{A} = \begin{pmatrix} A_{00} & 0 \\ A_{10} & A_{11} \\ A_{20} & A_{21} \end{pmatrix} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/8 & 3/4 & 1/8 & 0 \\ 0 & 1/2 & 1/2 & 0 \\ 0 & 1/8 & 3/4 & 1/8 \\ 0 & 0 & 1/2 & 1/2 \end{pmatrix}. \hspace{1cm} (12)$$

Let us summarize the results from above. With

$$P := \begin{pmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{pmatrix}, \quad Q := \begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \\ Q_3 \end{pmatrix} \quad \text{and} \quad \hat{Q} := \begin{pmatrix} Q_0 \\ Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \end{pmatrix}$$  \hspace{1cm} (13)

the first steps of the subdivision processes in matrix form are $Q = A P$ and $Q = \hat{A} P$.

Since $A$ is a square matrix we can iterate with it. Thus if we want to compute values inside $t \in [1/2^{i+1}, 1/2^i]$ we have to do $i$ subdivision steps with $A$ and than one with $\hat{A}$. This results in a general subdivision matrix

$$S = \hat{A} A^i = \hat{A} V \Lambda^i V^{-1}. \hspace{1cm} (14)$$
After that we have to pick the right points to compute the desired value by normal B-spline evaluation.

Figure 6 shows the setting for surfaces. For the computation of the limit point corresponding to vertex 0 we need the $2N + 1 = 7$ vertices numbered from 0 to $2N = 6$. For area I we need the additional 7 (this number is independent of $N$) vertices from $2N + 1 = 7$ to $2N + 7 = 13$. For the computation inside the areas II, III and IV we need another 9 vertices from $2N + 8 = 14$ to $2N + 16 = 22$. For valence $N > 3$ we have a vertex 7 which does not exist in the case $N = 3$ where we have to identify it with vertex 1.

### 3. cgls – a cg-method for linear least squares

We now describe how to build the matrices for the approximation of given surfaces with our Catmull-Clark meshes. We can compute the values $L_i$ of the limit surface and the corresponding coefficients $c_j$ of the involved vertices for arbitrary wanted points. $L_i$ can be written as $L_i = c_j^T P_i^i$ where we have collected the $c_j$ in the vector $c_i$ and the associated vertices in $P_i^i$. For a given surface $s$ we can project $L_i$ onto it: $L_i \rightarrow L_i^s$. Know we have the desired equation of form

$$c_i^T P_i^i = L_i^s.$$  (15)

Sampling enough of these collocation points we end up in a sparse linear system for interpolation or an over-determined sparse (linear) system for
approximation. Normally we use the latter one and solve it with CGLS, a Conjugate Gradient method for linear Least Squares (also called CGNR in [2]). To apply CGLS for solving the normal equations we only need effective methods to multiply the system matrix $M$ and its transpose $M^T$ with vectors. It should be noted that our vectors are vectors of 3d-vectors. In (15) we can see how the sparse multiplication with $M$ is done for one row. From Table 1 we know that for valence $N \leq 6$ we have at most 20 entries in each row of $M$. This number is independent of the number of control vertices ($\#V$). For the multiplication with $M^T$ we have to construct coefficient vectors for each row of $M$ to get an analogue form to (15). Again the number of non-zero entries is independent of $\#V$ but it depends on the ratio (number of sample points)/$\#V$.

Let $A \in \mathbb{R}^{m \times n}$ with $m \geq n$. It is well known that the solution for the linear least squares problem $\|Ax - b\|_2 \rightarrow \min$ can be determined by solving the normal equations $A^T A x = A^T b$. Furthermore, if $\text{rang}(A) = n$ then $A^T A$ is symmetrical positive definite (spd). Unfortunately in most cases $A^T A$ is badly conditioned and for sparse $A$’s normally $A^T A$ is not sparse. The conjugate gradient method (cg) can be reformulated to resolve both problems. We use two residual vectors $r^{(k)} = b - Ax^{(k)}$ and $s^{(k)} = A^T b - A^T A x^{(k)} = A^T r^{(k)}$. With this notation we can formulate the algorithm as follows:

For $A \in \mathbb{R}^{m \times n}$ with $\text{rang}(A) = n$ and arbitrary starting vector $x^{(0)}$:

\[
\begin{align*}
    r^{(0)} &= b - Ax^{(0)} \\
    s^{(0)} &= A^T r^{(0)} \\
    d^{(0)} &= s^{(0)} \\
\end{align*}
\]

for $k = 0, 1, 2, \ldots$

\[
\begin{align*}
    \alpha_k &= \frac{\|s^{(k)}\|_2^2}{\|Ad^{(k)}\|_2^2} \quad // \text{store } Ad^{(k)} \\
    x^{(k+1)} &= x^{(k)} + \alpha_k d^{(k)} \\
    r^{(k+1)} &= r^{(k)} - \alpha_k Ad^{(k)} \\
    s^{(k+1)} &= A^T r^{(k+1)} \\
    \beta_k &= \frac{\|s^{(k+1)}\|_2^2}{\|s^{(k)}\|_2^2} \\
    d^{(k+1)} &= s^{(k+1)} + \beta_k d^{(k)}
\end{align*}
\]

until stop

We stop if a maximum number of steps $K$ is exceeded (emergency exit) or the residual becomes smaller than a given tolerance $\varepsilon$ ($\|s^{(k)}\|_2 \leq \varepsilon$).
Figure 7. Simplified wing tip: Start polyhedron for the Catmull-Clark-Method and polyhedron after three subdivision steps

4. Examples

In this section we give some examples from our aeronautical project. All examples show the vertex meshes and not the limit surfaces. Furthermore we do not project the subdivision surface onto our given constellation because we want to demonstrate the nice smoothing properties of the subdivision surfaces on a larger scale. For real applications an approximation step is done at the state of the lower parts of the figures and for flow calculations an exact evaluation of the limit surface is done.

Figure 7 shows the reparameterization of a wing tip. Some details of Figure 8 can be seen, especially the high grid quality at the extraordinary points (valence 3) of the wing tip. Furthermore it can be seen that already a very rough approximation leads to a high quality grid. Figure 8 shows a part of a wing-fuselage configuration. We use a non stationary subdivision process with tagged edges, in other words: some edges are treated specially for a few steps. For the first steps for instance we treat the transition curve from the wing to the fuselage with the masks of (1) as a curve. Only in the last step we use the surface masks. This leads to semi sharp part in that region. Tagged edges and their influence on the limit surface where
Figure 8. Wing-fuselage constellation: Start polyhedron for the Catmull-Clark-Method and constellation after three subdivision steps analyzed in [15], [16] and [9]. We use a modified version of the latter on. Figure 9 shows the approximation of an engine. The scalings of the different areas have been slightly modified to see some details, namely the sharp edges and the overall high grid quality.

5. Conclusion

We have presented the basic framework for modelling, interpolation, approximation and grid generation with Catmull-Clark methods in technical projects. The result is a watertight geometry of untrimmed surface patches. The isolines of them yield high quality block structured grids that are well suited for adaptive flow solvers. In this contents adaptation is simply func-
Figure 9. Approximation of an engine: Start polyhedron for the Catmull-Clark-Method and polyhedron after three subdivision steps.

tion evaluation. Our approach is based on the precalculation of coefficient masks for standard or modified Catmull-Clark methods. These coefficients can directly be used to efficiently solve the sparse linear systems.

6. Acknowledgments

This work has partially been performed with funding by the Deutsche Forschungsgemeinschaft in the Collaborative Research Center SFB 401 "Flow Modulation and Fluid-Structure Interaction at Airplane Wings" of the RWTH Aachen, University of Technology, Aachen, Germany.

References


