Adaptive and Feature-Preserving Subdivision for High-Quality Tetrahedral Meshes

D. Burkhart[1], B. Hamann[2] and G. Umlauf[3]

[1]University of Kaiserslautern, Germany
burkhart@cs.uni-kl.de
[2]University of California, Davis, CA 95616
hamann@cs.ucdavis.edu
[3]HTWG Konstanz, Germany
umlauf@htwg-konstanz.de

Abstract

We present an adaptive subdivision scheme for unstructured tetrahedral meshes inspired by the $\sqrt{3}$-subdivision scheme for triangular meshes. Existing tetrahedral subdivision schemes do not support adaptive refinement and have traditionally been driven by the need to generate smooth three-dimensional deformations of solids. These schemes use edge bisections to subdivide tetrahedra, which generates octahedra in addition to tetrahedra. To split octahedra into tetrahedra one routinely chooses a direction for the diagonals for the subdivision step. We propose a new topology-based refinement operator that generates only tetrahedra and supports adaptive refinement. Our tetrahedral subdivision algorithm is motivated by the need to have one representation for the modeling, the simulation and the visualization and so to bridge the gap between CAD and CAE. Our subdivision algorithm design emphasizes on geometric quality of the tetrahedral meshes, local and adaptive refinement operations, and preservation of sharp geometric features on the boundary and in the interior of the physical domain.

Keywords: solid modeling, tetrahedral subdivision

**ACM CCS:** I.3.5 [Computer Graphics]: Curve, surface, solid, and object representations.

1. Introduction

Tetrahedral meshes are an important representation for a large number of applications including physical animations [KFCO06, CFL*07], numerical simulations [JSK08], solid deformations [SHW04], and scientific visualization [GSA*03]. For all of these applications the mesh quality in terms of the shape of the tetrahedra is of high importance. While long and thin tetrahedra can be useful in some applications, for example, video compression [Leh08] or simulation of anisotropic phenomena, in general, isotropic tetrahedra are preferred in most applications. In particular, the stability of a numerical simulation using the finite element or finite volume method is closely related to the quality of the underlying mesh [Kri92, PGH94]. An appropriate mesh quality is usually achieved by an optimization or adaptive refinement of the tetrahedral mesh.

The most commonly used tetrahedral meshing algorithms are based on Delaunay, octree, or advancing-front methods [She98, LS07, Sch07]. However, these methods cannot guaranty an appropriate mesh quality without an additional post-optimization to improve the mesh quality. Furthermore, these methods do not allow for an easy and flexible local refinement of the mesh in arbitrary mesh regions without destroying sharp features in the mesh that must be captured and preserved due to physical constraints.

In computer graphics subdivision surfaces have become a standard tool for shape modeling [DKT98], because of their efficiency and flexibility. They allow for a built-in adaptivity [Kob00], for example, integrated local optimizations [GU08], and special rules to preserve sharp features [HDD*94]. Therefore, we present a subdivision algorithm inspired by the $\sqrt{3}$-scheme to generate high-quality tetrahedral
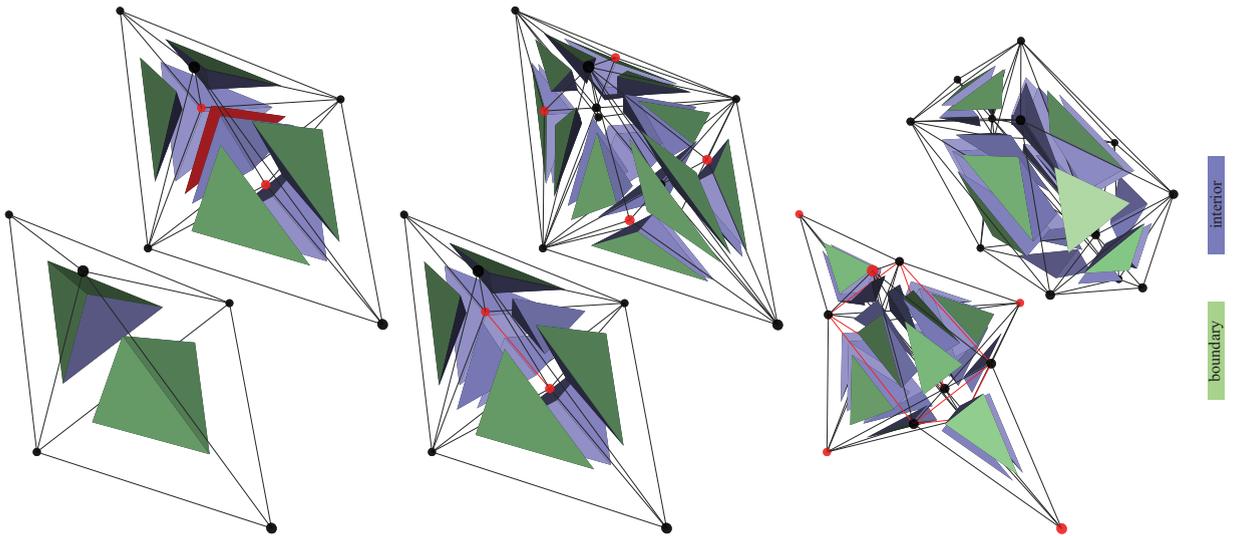
Figure 1: *An example for our tetrahedral subdivision scheme. Starting from the input mesh (bottom left) we show from left to right: the mesh after 1-4 splits, 2-3 face flips, boundary 1-3 splits, boundary edge flips, edge removals and geometric smoothing.*

meshes for numerical simulations or visualization. This subdivision algorithm can be used to refine a tetrahedral mesh adaptively while preserving all relevant sharp features.

In order to bridge the gap between CAD and CAE applications it is necessary that the underlying representation meets the requirements of both fields: exact interpolation of the bounding surface, preservation of geometric features on the bounding surface and in the interior, adaptive refinement, and quality of the tetrahedra.

First, we discuss in Section 2 previous approaches related to our method. Then we address in Section 3 the basic definition of our subdivision scheme, in Sections 4 and 5 the boundary and sharp feature rules, and in Section 6 the adaptive refinement strategy. We close in Sections 7 and 8 with our results and a brief outlook to future research directions.

2. Related Work

The presented work is related to surface subdivision, solid subdivision, mesh quality measures, mesh optimization, and adaptive refinement that we discuss in the sequel.
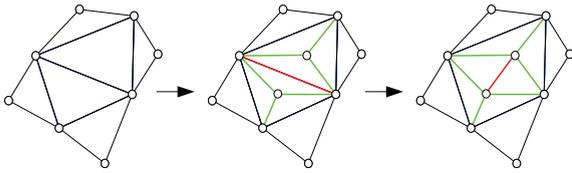
2.1. Subdivision surfaces

Subdivision surfaces are a powerful tool to model free-form surfaces of arbitrary topology. They were developed in 1978 [CC78, DS78] and are today a standard modeling tool in computer graphics [DKT98]. A subdivision surface is defined as the limit of an iterative refinement process, starting with a polygonal base mesh M_0 of control points. Iterating the subdivision on M_0 generates a sequence of refined

meshes $M_1, \ldots, M_n$, that converges to a smooth limit surface M_∞ [PR08]. This subdivision operator usually consists of a topological refinement operation followed by a geometrical smoothing operation. While the topological refinement inserts new vertices and faces or flips edges, the geometrical smoothing moves vertices to new positions.

In general, subdivision surfaces are smooth. However, real world models have sharp features where the surface normal is discontinuous. For this subdivision algorithms are tailored to also allow for sharp corners or creases. Examples for such special rules, where tagged edges will yield creases on the subdivision surface, are presented in [HDD*94, Kob00].

Subdivision surfaces are classified as interpolating or approximating. For approximating schemes the control points of M_i do, in general, not lie on M_{i+1}, $i \geq 0$. Examples of such schemes are the Doo-Sabin and the Catmull–Clark algorithm [CC78]. Both schemes are generalizations of uniform tensor-product B-Spline surfaces to arbitrary quadrilateral meshes. Approximating schemes for triangular meshes are the algorithm of Loop [Loo87] and $\sqrt{3}$-subdivision [Kob00]. The topological refinement operation of the latter is particularly designed for adaptive refinement, see Figure 2.

For interpolating schemes all control points of M_i are also in M_{i+1}, $i \geq 0$. Thus, the limit surface interpolates these points. The earliest interpolating subdivision scheme for surfaces is the butterfly scheme of Dyn et al. [DLG90] which triggered the subsequent development of other interpolating subdivision schemes. An example for such a scheme that uses the topological refinement operation of the $\sqrt{3}$-scheme followed by an interpolating geometrical smoothing operation is the interpolating $\sqrt{3}$-subdivision scheme of [LG00].

Figure 2: *1-3 split (green) and edge flip (red) operations used in $\sqrt{3}$-subdivision scheme for triangular meshes.*

2.2. Solid subdivision

In contrast to subdivision surfaces, subdivision solids have gained much less attention. One of the first algorithms was presented by Joy and MacCracken [JM96, MJ96] who used a generalization of Catmull–Clark subdivision to 3d-solids to define smooth deformations based on unstructured hexahedral meshes. As the topological refinement operation of this algorithm made it hard to analyze the smoothness of the resulting limit solid, Bajaj et al. [BSWX02] modified the operation to create deformations that are provably smooth everywhere except at the vertices of the base mesh M_0.

A subdivision scheme for tetrahedral meshes based on trivariate box splines was proposed by Chang et al. [CMQ02, CMQ03]. Their scheme is approximating or interpolating depending on the geometrical smoothing operation. Their topological refinement first splits every tetrahedron into four tetrahedra and one octahedron, see Figure 3(a). The resulting octahedra are then split along one of their diagonals into four tetrahedra. This choice of the diagonal can bias the resulting meshes towards a certain spatial direction. To remedy this effect Schaefer et al. [SHW04] use a topological refinement that splits the octahedra symmetrically into eight tetrahedra and six octahedra. Their geometric smoothing allows for globally C^2-continuous deformations, except along edges of M_0. The major drawback of these schemes is the use of tetrahedra and octahedra, which are not well-suited for finite volume simulations, and require a complicated data structure.

Another solid subdivision scheme was developed by Pascucci [Pas02]. This scheme allows for any type of polygonal elements and adaptive refinement. However, applying this
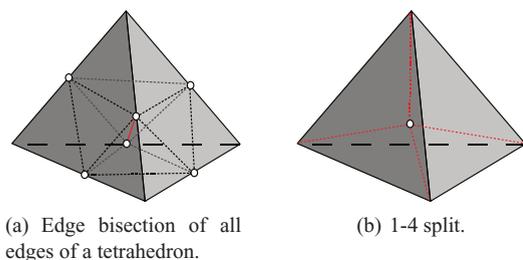
scheme to a tetrahedral mesh generates pyramids. So, the handling of the different polygonal elements requires complicated refinement and merging steps in each iteration.

Other topological refinement operators for tetrahedra are described in [Bey95, RM98]. Because these allow for an adaptive refinement, we discuss them in Section 2.5.
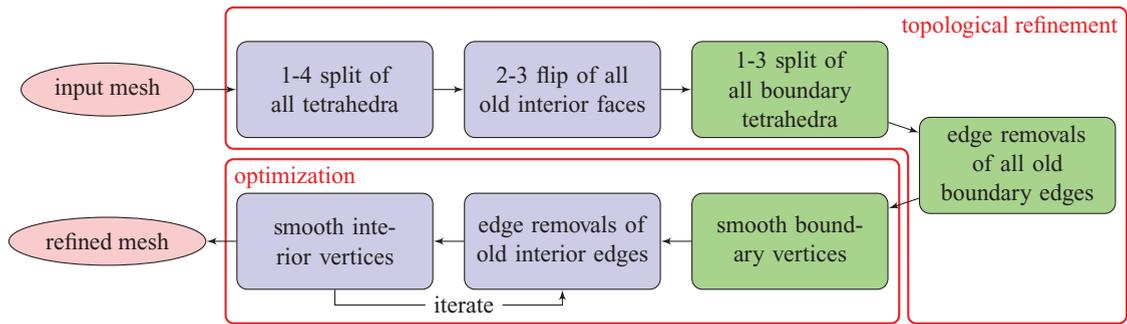
2.3. Mesh quality

Common quality measures are the minimum of a tetrahedron's six dihedral angles and the edge ratio of a tetrahedron's longest and shortest edge. The edge ratio measure is efficient to compute but has the drawback that some badly shaped tetrahedra (e.g. slivers) are measured to be good. Therefore, using angle criteria to determine the quality of a tetrahedron is more intuitive and precise because tetrahedra with large dihedral angles cause interpolation and approximation errors [She02] and tetrahedra with small dihedral angles lead to ill-conditioned stiffness matrices [Fri72, BA76]. The drawback of measuring the quality of a tetrahedron by its dihedral angles is that there are some badly shaped tetrahedra with good dihedral angles, for example, needle-shaped tetrahedra, but there is evidence that such badly shaped elements do neither hurt discretization error nor matrix conditioning [KS07]. For more details on the relation of shape and matrix conditioning we refer to [She02], which gives an extensive survey of different quality measures.

As dihedral angles are inefficient to compute, often the minimum sine of a tetrahedron's six dihedral angles are used instead. For the rest of this paper, whenever we refer to the quality of tetrahedra, we use the minimum sine of the signed dihedral angle. For better readability, we show the real angle values in all figure, where the dihedral angle of an equilateral tetrahedron of $\approx 71°$ is the optimal value.

2.4. Mesh optimization

If a mesh contains distorted elements, a numerical simulation tends to be less accurate and the numerical solution is more expensive. Mesh optimization algorithms address this problem as they take an existing mesh and improve its quality. There are two methods to optimize a mesh: smoothing and topological changes. A smoothing operation moves a vertex to a new location with the objective to improve the local shape of the tetrahedra incident to that vertex. Topological changes are usually local operations that remove certain elements from a mesh and replace them with better ones occupying the same space.

For triangular meshes, every topological operation can be obtained by a combination of edge flips, vertex insertions into a face or an edge, and edge collapses. For tetrahedral meshes, there exist edge flips and face flips [dIG85]. The simplest case of an edge flip is a 3-2 edge flip shown in

(a) Edge bisection of all edges of a tetrahedron.

(b) 1-4 split.

Figure 3: *Edge bisection and tetrahedral 1-4 split.*

Figure 4: *The tetrahedral subdivision scheme. The green shaded boxes represent the $\sqrt{3}$-scheme on the boundary.*
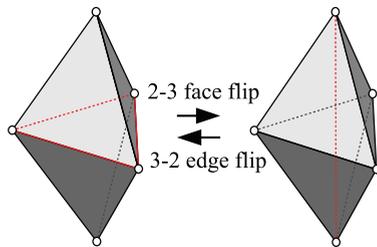
Figure 5: *2-3 face flip and 3-2 edge flip.*
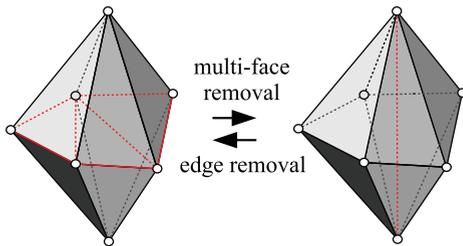
Figure 6: *Multi-face and edge removal for tetrahedra.*

Figure 5, where the red edge is flipped and the red face is inserted instead. The 2-3 face flip is the inverse operation. Their generalizations are called edge removal and multi-face removal, see Figure 6. In [KS07, Leh08] these generalized flip operations are analyzed in detail.

The most popular smoothing approach is Laplacian smoothing, which moves a vertex to the center of its one-ring neighbors [Fie88]. The advantage of Laplacian smoothing is its simplicity and computational efficiency, but it cannot guaranty to improve the quality since it is a heuristic not related to quality measures. However, Laplacian smoothing is by far the most common technique, especially as it works well for convex regions. Unfortunately, in concave regions, it can create bad or even inverted tetrahedra.

A smoothing approach that produces better results— with the drawback of higher computational costs—is optimization-based smoothing [BS97, AB97]. This technique moves vertices in order to locally optimize the quality of incident tetrahedra. Several researchers have implemented optimization based smoothing techniques. There are algorithms that basically use Laplacian smoothing and choose optimization based smoothing when necessary [Fre97, CTS98]. It has also been shown that these techniques can be parallelized [FJP99].

2.5. Adaptivity

For many applications it is necessary to adaptively refine the tetrahedra to a required level of detail. In numerical simulations, for instance, in areas of strong force gradients a finer mesh is required than in other areas. Applying free-form deformations to solid models based on tetrahedral mesh deformations, it is necessary to adaptively refine tetrahedra, where the deformation varies strongly. Finally, for visualization it is often desired to have finer tetrahedra on the boundary.

The most common technique for adaptive refinement of triangular meshes is the so-called red-green refinement [BSW83], where one, two, or three edges of a triangle are split to achieve transitions between different refinement levels. For triangles, there are $2^3 = 8$ possibilities to split the edges, i.e., up to symmetry there are four different cases. Here, the refinement operator of $\sqrt{3}$ subdivision is much more advantageous, as it only consists of triangular 1-3 splits and edge flips. For tetrahedral meshes, there is a refinement similar to red-green refinement based on edge bisection [RM98, Bey95].

But as tetrahedra have six edges, there are $2^6 = 64$ different cases and up to symmetry eleven different cases remain. Therefore, adaptive refinement for tetrahedral meshes with these rules is a complex operation. A simpler approach for adaptive refinement is longest-edge bisection [RN00], if one splits one edge after the other. However, these approaches focus on the topological refinement and do not address geometrical smoothing. This becomes particularly important,

if on the boundary of the solid surfaces and sharp features must be preserved during simulation, modeling and mesh optimization.

3. Tetrahedral Subdivision

For our tetrahedral subdivision scheme we define the topological refinement operation and the geometrical smoothing operation separately. Each of these two operations is split into elementary steps which are interleaved to achieve optimal mesh quality. The schematic sequence of these elementary steps is shown in Figures 4 and 1, where the changes are highlighted in red. It starts with four elementary topological refinement steps, followed by the geometrical optimization consisting of two elementary geometrical smoothing steps that bracket an additional elementary topological refinement step. The last two steps are iterated to improve the mesh quality.

Our tetrahedral subdivision schemes was inspired by $\sqrt{3}$-subdivision for triangular meshes whose topological refinement operation can be factored into triangular 1-3 splits followed by triangular edge flips. Instead of triangular 1-3 splits we use tetrahedral 1-4 splits (Figure 3(b)) to achieve adaptivity. This splitting operation has not been used before for tetrahedral subdivision or mesh generation in general. Instead of triangular edge flips we use 2-3 face flips (Figure 5) and edge removals (Figure 6) to control the vertex valence. On the boundary we use the standard $\sqrt{3}$-scheme.

3.1. Topological refinement

To avoid the generation of octahedra in the mesh we use the 1-4 split (Figure 3(b)), which is the analog of the triangular 1-3 split (Figure 2). One 1-4 split inserts a new vertex at the barycenter of a tetrahedron and connects it to the tetrahedron's vertices. The benefit of this operation is the implicit support for adaptive refinement. Unfortunately, the tetrahedra become flatter and the valence of the vertices increases.

To decrease the valence, $\sqrt{3}$-subdivision uses edge flips. For tetrahedral meshes there are flipping operations that flip faces or edges. Since the number of tetrahedra incident to an edge is arbitrary, an tetrahedral edge flip can become arbitrarily complex. Therefore, we use 2-3 face flips (Figure 5) to remove all old interior faces. As 2-3 face flips can create inverted tetrahedra, we assume that the input mesh fulfills the geometric properties of [Leh08], i.e. for two face-adjacent tetrahedra the line connecting the two remaining vertices intersects the common face, see Figure 5. For all subsequent iterations our algorithm optimizes the signed dihedral angles in the refined mesh with edge removal operations. Thus, we can guaranty that no inverted tetrahedra are left after the optimization.

For our subdivision scheme, the combination of 1-4 split and 2-3 face flips is important for a uniform refinement. Re-

fining the tetrahedral mesh only with 1-4 splits, the tetrahedra do not get smaller, but only flatter.

A mesh with n tetrahedra and n_b boundary faces has $4n$ tetrahedra after the 1-4 split and $n_b + 3(4n - n_b)/2$ tetrahedra after 2-3 face flips. Unfortunately, these operations still increase the valence of an interior vertex v with n_t incident tetrahedra and $n_e = n_t/2 + 2$ incident edges. The number of tetrahedra incident to v increases by the factor three, because each tetrahedron is split into four tetrahedra, but only three are incident to v. Thus, the number of incident edges becomes $3n_e - 4$. Note that n_b and n_e are even.

To reduce the valence further we perform additional edge removals of interior edges of the mesh. Because there is no unique way to identify removable edges we control these edge removals by the subsequent smoothing operation.

3.2. Geometric smoothing

Traditional subdivision algorithms for surfaces use simple stencils for geometric smoothing. As the purpose of the proposed tetrahedral subdivision scheme is for numerical simulations, we use a geometric smoothing operation that is directly related to a quality measure of the mesh. To achieve the best trade-off between mesh quality of optimization based smoothing and efficiency of Laplacian smoothing we combined both methods. To find the best location for a vertex v, we first determine the quality vector $(q_1, \ldots, q_n)$ of all tetrahedra $(t_1, \ldots, t_n)$ incident to v. In principle, q_i denotes an arbitrary quality measure for a tetrahedron t_i; in practice, we use the quality measure described in Section 2.3. We denote by $q_{min} = \min(q_1, \ldots, q_n)$ the quality of the worst incident tetrahedron. If $q_{min} > \alpha$ for a pre-defined quality level α (e.g., 0.3), no optimization is triggered. Otherwise, the vertex is first moved using Laplacian smoothing and the quality vector and q_{min} are re-calculated. If q_{min} is decreased, the old vertex location of v is recovered. If now $q_{min} > \alpha$, the smoothing stops. Otherwise, optimization based smoothing is used to find a better location for v, cf. [Fre97].

To guide the choice of edge removals to reduce the vertex valence we also use the quality vectors. An edge is removed when the quality of its worst incident tetrahedron would be improved as a consequence of the removal (cf. [KS07]).

4. Boundary

The operations described in the previous section work only for interior tetrahedra. The boundary of a tetrahedral mesh is a triangular mesh. Therefore, we use on the boundary the $\sqrt{3}$-scheme which consists of triangular 1-3 splits, triangular edge flips and geometrical smoothing, see Figure 2.

To propagate the topological refinement of the $\sqrt{3}$-scheme on the boundary to the interior of the tetrahedral mesh we use the following strategy. First, we insert a new vertex at the barycenter of each boundary face. This operation is a

(a) Edge trisection of one edge of a tetrahedron.

(b) Tetrahedral 1-3 split.
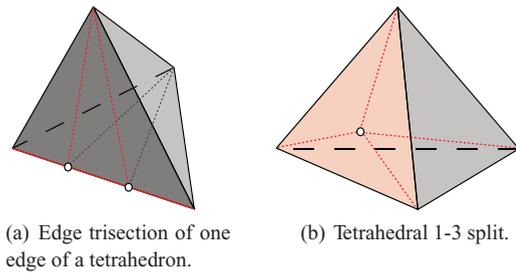
Figure 7: *Edge trisection and tetrahedral 1-3 split.*
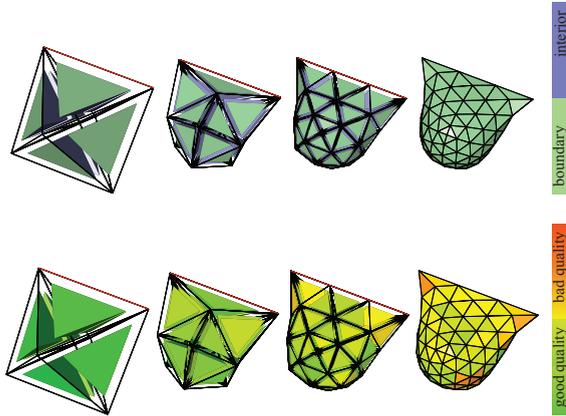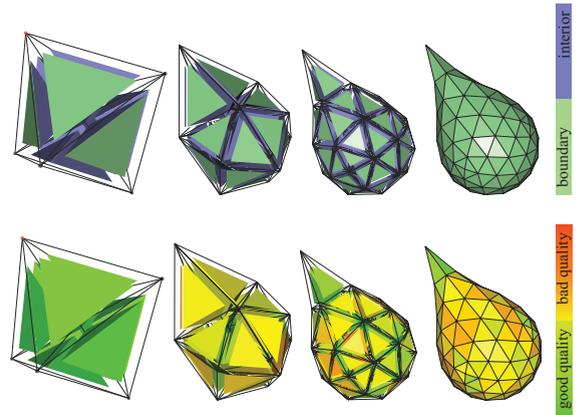

Figure 8: *Three steps of the subdivision scheme with preservation of a sharp crease (red edge). The upper row shows the exploded view of the mesh. The lower row visualizes the quality of the tetrahedra measured with the minimum dihedral angle criterion from red (bad quality, $0°$) to green (good quality, $\approx 71°$).*

Figure 9: *Three steps of the subdivision scheme with a sharp corner (red vertex left).*

Figure 10: *Two subdivision steps with adaptive refinement. Here, the interpolating boundary scheme was used.*
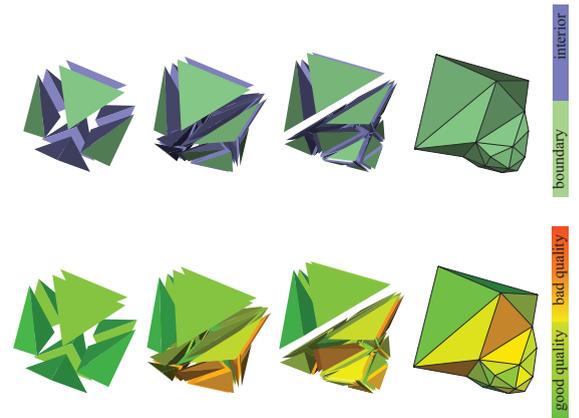
so-called tetrahedral 1-3 split, see Figure 7(b). It is the equivalent to a triangular 1-3 split, but instead of three new triangles it generates three new tetrahedra. Second, we flip all old boundary edges. After the 1-4 split there are always at least two tetrahedra incident to a boundary edge. Thus, we use edge removals to flip boundary edges. Note that one of the red edges in Figure 6 (left) becomes the new boundary edge connecting triangle barycenters.

The classical $\sqrt{3}$-scheme [Kob00] is an approximating subdivision scheme. We use its geometric smoothing operation for the boundary of the tetrahedral mesh. Examples of this are shown in Figures 8 and 9. The top row shows an exploded view of the meshes and the bottom row a visualization of the mesh quality. For the visualization of the tetrahedral quality the color hue of the HSV model is linearly interpolated from $0°$ (bad quality) to $120°$ (good quality).

An interpolating $\sqrt{3}$-scheme was proposed in [LG00]. They used the same topological refinement, but different

geometric smoothing to create an interpolating scheme. We implemented these different smoothing rules as well, so that our framework allows to choose between interpolation and approximation. See Figure 10 for an example.

5. Sharp Features

To allow for sharp features on the boundary of a tetrahedral mesh we use a variant of $\sqrt{3}$-subdivision that is capable of modeling corners and creases, see [Kob00]. On the one hand, to preserve creases in a $\sqrt{3}$-mesh, tagged edges must not be flipped during subdivision. On the other hand, two steps of $\sqrt{3}$-subdivision trisect each edge. Therefore, tagged edges are left unchanged in every odd iteration, and are trisected in every even iteration. For the even iterations this yields topologically the same mesh as the standard $\sqrt{3}$-scheme. For the geometric smoothing linear interpolation is used.

For sharp creases in tetrahedral meshes, we also use edge trisection as shown in Figure 7(a), where the red edge is trisected. For sharp creases we also leave tagged boundary edges unchanged in every odd iteration. In every even iteration we apply edge trisection to all tetrahedra incident to a tagged boundary edge. We also use linear interpolation for the geometric smoothing. An example for a sharp crease on the boundary is shown in Figure 8, where one edge is tagged (red). In the first iteration the edge is preserved, while in the second iteration the edge is trisected. Note that the quality is decreased near the feature.

Figure 9 shows three subdivision steps with one sharp corner. This is achieved on the boundary and in the interior by skipping the geometric smoothing operation for all tagged vertices.

For interior features, tagged interior edges can be handled the same way to allow for interior creases. Similar to boundary surfaces, interior feature surfaces can be modeled based on the $\sqrt{3}$-surface subdivision scheme with transitions to the solid tetrahedral mesh on both sides.

6. Adaptive Refinement

The subdivision scheme presented so far increases the number of tetrahedra by approximately a factor of six after one complete subdivision step. This is better than edge bisection-based subdivision where the number of tetrahedra grows with a factor of eight. However, the growth is still exponential. Adaptive refinement is a strategy to avoid this, because it allows to subdivide a mesh in different regions to different refinement levels without generating low-quality tetrahedra.

Adaptive refinement for triangular meshes splits only those triangles that belong to a mesh area of high importance for the simulation or whose neighborhood is highly curved. Thus, for automatic adaptivity mesh generation depends heavily on the application problem.

The strategy of adaptive refinement for our tetrahedral subdivision scheme is as follows: If a tetrahedron is selected to be not refined, the 1-4 split is not applied. All other subsequent steps of the topological refinement are only applied if none of the involved tetrahedra are selected. The geometric smoothing operation is applied to all vertices.

Figure 10 shows two steps of an adaptive refinement. The top-most tetrahedra are un-split, the tetrahedra in the middle are split once and the bottom-most tetrahedra are split twice. In this example, for the boundary we used the interpolating smoothing operation.

7. Results

The examples in the previous figures are designed to demonstrate the functionality and the features of the proposed sub-
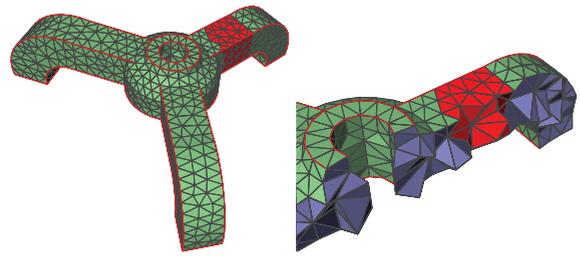
Figure 11: *Tripod model. Courtesy getfem++ [get09].*

division scheme. In this section, we demonstrate our subdivision scheme for the tripod mesh in Figure 11. The statistics of the initial mesh are shown in the first column of Table 1. The initial model has larger tetrahedra in the interior and smaller tetrahedra on the boundary, which is preserved during the refinement. To preserve the sharp features of this tripod model, the 293 red highlighted edges are handled as feature edges.

The first example shown in Figure 12 illustrates two steps of uniform subdivision of the complete tripod model. The first row shows the boundary with red highlighted sharp feature edges, the second row shows a cutaway view, the third row shows the quality distribution in the cutaway view, and the fourth row shows a histogram of the minimum dihedral angles of the model. The feature edges are not highlighted in the third column, to illustrate the edge trisection in the second subdivision step. Note that the feature edges are not split in the first step. The initial model has 2,799 tetrahedra, 23,480 tetrahedra after the first step and 161,028 tetrahedra after the second step. The individual elementary steps have the following impact on the tetrahedral mesh:

1. The 1-4 split creates $4 \times 2,799 = 11,196$ tetrahedra.
2. All old 4,743 interior faces are removed by a 2-3 flip, which creates 4,743 additional tetrahedra. This yields 21,096 tetrahedra in total.
3. All 1,710 boundary faces are split by a 1-3 split ($3 \times 1,710 = 5,130$).
4. All boundary edges are flipped by the edge removal, which replaces m tetrahedra by $2m - 2$ tetrahedra.
5. During the optimization steps, the first 565 boundary vertices are smoothed.
6.&7. Iteratively interior edges are flipped and interior vertices are smoothed, which creates the remaining tetrahedra.

The initial tetrahedral mesh contains 967 nodes. The increment of the nodes is caused by the 1-4 splits creating 2,799

Table 1: *Statistics of initial tripod mesh and after the subdivision.*

	Initial Mesh	Example 1 Step 1	Example 1 Step 2	Example 2 Step 1	Example 2 Step 2
Nodes	967	5,476	34,086	5,910	7,658
Edges	4,621	31,521	202,809	6,292	16,410
Tetrahedra	2,799	23,480	161,028	4,074	12,089
Worst minimum dihedral angle	$7,4°$	$6,1°$	$4,9°$	$6,8°$	$5,9°$
Average minimum dihedral angle	$42,0°$	$36,1°$	$28,4°$	$38,3°$	$30,6°$
Best minimum dihedral angle	$64,1°$	$65,2°$	$65,7°$	$64,1°$	$65,5°$
Average vertex valence	8	12	16	9	16
Maximal vertex valence	50	150	223	114	165
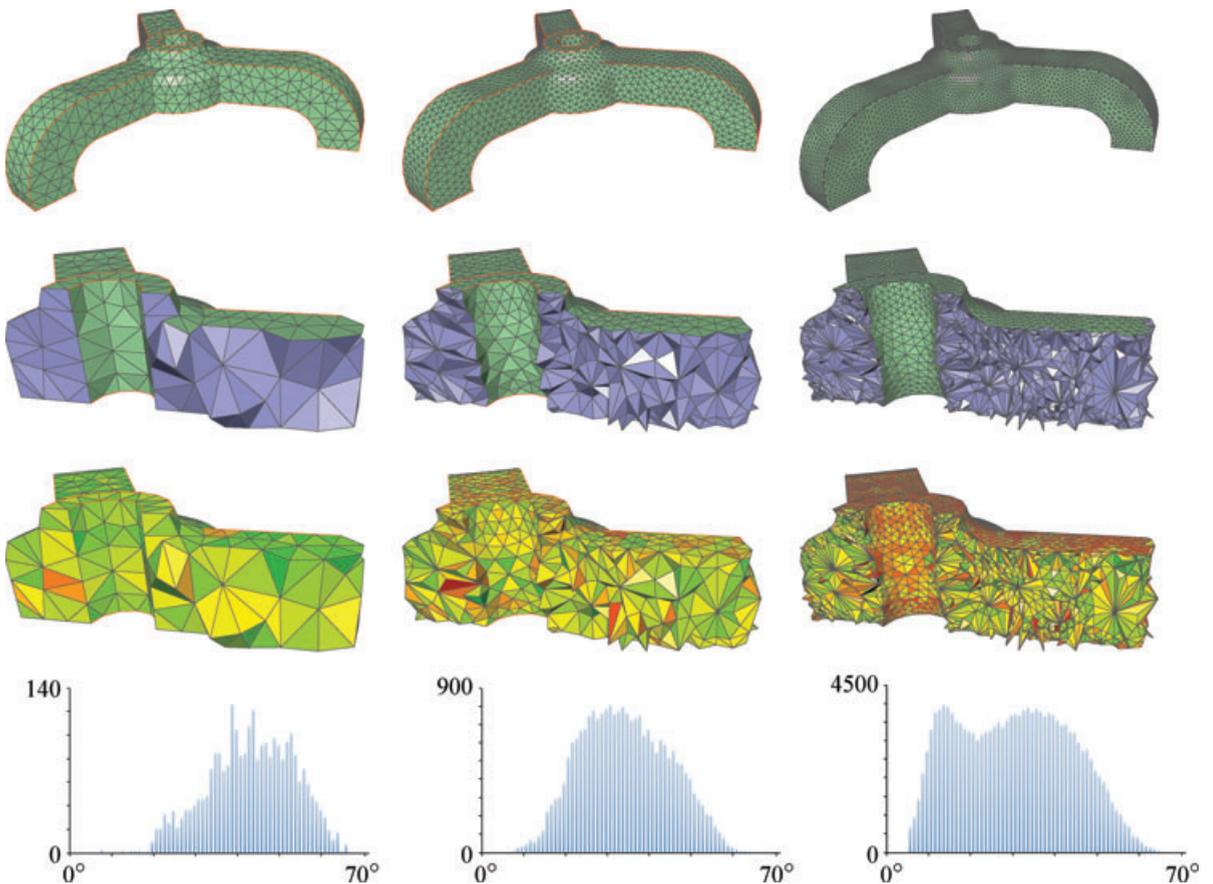
Figure 12: *Two subdivision steps of the tripod model without adaptive refinement. The first column shows the original model, the second column the model after the first step and the third column the model after the second step. The row shows the whole model, the second row a cutaway view of the model, the third row the quality distribution (minimum dihedral angles) and the fourth row shows the histogram of the minimum dihedral angles. The ordinate of the histogram represents the amount of tetrahedra with corresponding minimum dihedral angle.*
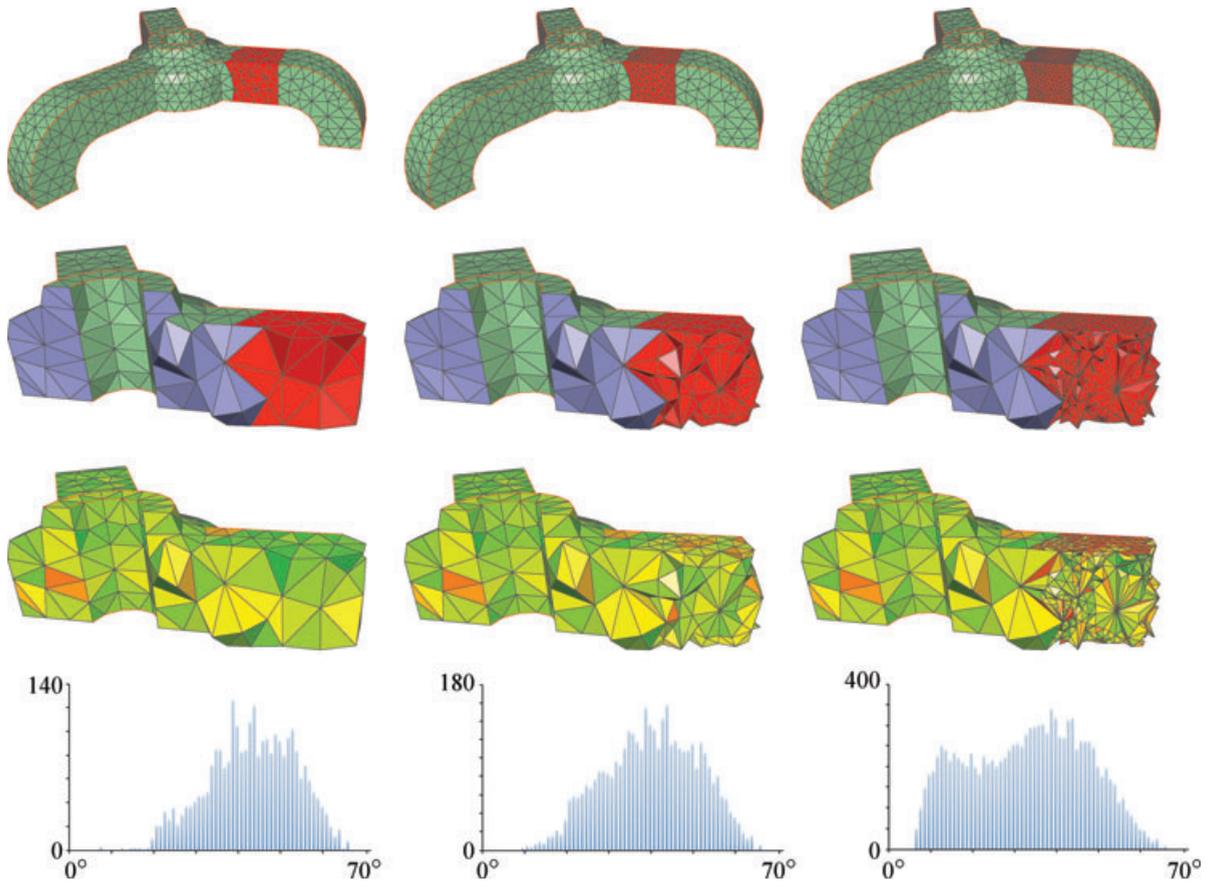
Figure 13: *Two subdivision steps of the tripod model with adaptive refinement. The first column shows the original model, the second column the model after the first step and the third column the model after the second step. The row shows the whole model, the second row a cutaway view of the model, the third row the quality distribution (minimum dihedral angles) and the fourth row shows the histogram of the minimum dihedral angles. The ordinate of the histogram represents the amount of tetrahedra with corresponding minimum dihedral angle.*

new vertices and the 1-3 splits creating 1,710 new vertices on the boundary. This yields 5,476 vertices in total.

Table 1 shows the statistics of this example in the second and third column. As these statistics indicate we cannot completely guaranty to preserve the initial quality and to hold the vertex valence small. The maximal vertex valence increases from 50 to 223 after two steps of subdivision. However, less than 2% of the vertices have a valence larger than 50. The average vertex valence only increases from 8 to 16. The reason for this quality decrease is that we use only vertex smoothing and edge removal operations for the optimization of the mesh, and that we do not add additional optimization operations like vertex insertions or multi-face removals. The increase of the vertex valence is not problematic for finite element simulations, as long as vertices with high valence still have incident tetrahedra with acceptable minimum dihedral angels. However, if the objective is to minimize the vertex valence, the optimization step can be changed to use a different objective function, for example, spherical angle instead of dihedral angle.

The second example in Figure 13 shows two steps of the adaptive refinement of the red highlighted region containing 284 tetrahedra. The statistics of this example are shown in the last two columns of Table 1. As in the first example, the quality of the tetrahedra decreases and the maximal valence of the vertices increases from 8 to 16. Compared to the first example the mesh quality and the vertex valance are slightly better. This is due to the fact that globally less vertices are necessary to achieve locally a high resolution in the red highlighted mesh area.

8. Conclusion and Future Work

In this paper, we have presented a new subdivision scheme for tetrahedral meshes. The main contribution is the novel split operation. To refine the tetrahedra we use a 1-4 split

instead of edge bisection. This has two major advantages: we do not introduce octahedra into the mesh and the split implicitly supports adaptive refinement. A second key ingredient is that the presented subdivision scheme allows the modeling of sharp features such as corners and creases and can be either interpolating or approximating. We did not tackle the problem of analyzing the limit smoothness of the solid, because this requires the development of new techniques for volumetric cases, which is beyond the scope of this paper.

For the future we plan to analyze how this subdivision scheme can be used for animation, scientific visualization, solid modeling, and simulation of real world problems. At the moment we are working on a more efficient data structure for tetrahedral meshes, as this is the major bottleneck of the proposed framework.

Acknowledgments

References

[AB97] AMENTA N., BERN M.: Optimal point placement for mesh smoothing. In *Proc. of 8th ACM-SIAM Symposium on Discrete Algorithms* (1997), pp. 528–537.

[BA76] BABUSKA I., AZIZ A.: On the angle condition in the finite element method. *SIAM J. Num. Anal. 13* (1976), 214–226.

[Bey95] BEY J.: Tetrahedral grid refinement. *Computing 55*, 4 (1995), 355–378.

[BS97] BANK R. E., SMITH R. K.: Mesh smoothing using a posteriori error estimates. *SIAM J. Num. Anal. 34* (1997), 979–997.

[BSW83] BANK R., SHERMAN A., WEISER A.: Some refinement algorithms and data structures for regular local mesh refinement. *Sci. Comput.* (1983), 3–17.

[BSWX02] BAJAJ C., SCHAEFER S., WARREN J., XU G.: A subdivision scheme for hexahedral meshes. *The Visual Computer 18* (2002), 343–356.

[CC78] CATMULL E., CLARK J.: Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design 10*, 6 (1978), 350–355.

[CFL*07] CHENTANEZ N., FELDMAN B., LABELLE F., O'BRIEN J., SHEWCHUK J.: Liquid simulation on lattice-based tetra-

hedral meshes. In *Proc. Eurographics Symp. Computer Animation* (2007), pp. 219–228.

[CMQ02] CHANG Y., MCDONNELL K., QIN H.: A new solid subdivision scheme based on box splines. In *Proc. Solid Modeling* (2002), pp. 226–233.

[CMQ03] CHANG Y., MCDONNELL K., QIN H.: An interpolatory subdivision for volumetric models over simplicial complexes. In *Proc. Shape Modeling International* (2003), pp. 143–152.

[CTS98] CANANN S., TRISTANO J., STATEN M.: An approach to combined Laplacian and optimization-based smoothing for triangular, quadrilateral, and quad-dominant meshes. In *Proc. 7th Int. Meshing Roundtable* (1998), pp. 479–494.

[dIG85] DE ISLE E. B., GEORGE P.-L.: Optimization of tetrahedral meshes. *IMA Volumes in Mathematics and its Applications 75* (1985), 97–128.

[DKT98] DEROSE T., KASS M., TRUONG T.: Subdivision surfaces in character animation. In *SIGGRAPH* (1998), pp. 85–94.

[DLG90] DYN N., LEVIN D., GREGORY J.: A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graph. 9*, 2 (1990), 160–169.

[DS78] DOO D., SABIN M.: Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design 10*, 6 (1978), 356–360.

[Fie88] FIELD D. A.: Laplacian smoothing and Delaunay triangulations. *Com. App. Num. Meth. 4* (1988), 709–712.

[FJP99] FREITAG L., JONES M., PLASSMANN P.: A parallel algorithm for mesh smoothing. *SIAM J. Sci. Comput. 20* (1999), 2023–2040.

[Fre97] FREITAG L.: On combining Laplacian and optimization-based mesh smoothing techniques. In *Trends in unstructured mesh generation* (1997), pp. 37–43.

[Fri72] FRIED I.: Condition of finite element matrices generated from uniform meshes. *AIAA Journal 10* (1972), 219–221.

[get09] http://home.gna.org/getfem/, 10.07.2009.

[GSA*03] GREGORSKI B. F., SIGETI D. E., AMBROSIANO J. J., GRAHAM G., WOLINSKY M., DUCHAINEAU M. A., HAMANN B., JOY K. I.: Multiresolution representation of datasets with material interfaces. In *Hierachical and Geometrical Methods in Scientific Visualization*. Farin G., Hamann B., Hagen H., (Eds.). Springer-Verlag, Berlin, Germany, 2003, pp. 99–117.

[GU08] GINKEL I., UMLAUF G.: Local energy-optimizing subdivision algorithms. *Comp. Aided Geom. Design 25* (2008), 137–147.

[HDD*94] HOPPE H., DEROSE T., DUCHAMP T., HALSTEAD M., JIN H., MCDONALD J., SCHWEITZER J., STUETZLE W.: Piecewise smooth surface reconstruction. In *SIGGRAPH '94* (1994), pp. 295–302.

[JM96] JOY K., MACCRACKEN R.: *The refinement rules for Catmull-Clark solids*. Technical report cse-96-1, UC Davis, 1996.

[JSK08] JÄGER P., STEINMANN P., KUHL E.: Modeling three-dimensional crack propagation—a comparison of crack path tracking strategies. *Int. J. Num. Meth. Eng 76* (2008), 1328–1352.

[KFCO06] KLINGNER B., FELDMAN B., CHENTANEZ N., O'BRIEN J.: Fluid animation with dynamic meshes. In *SIGGRAPH '06* (2006), pp. 820–825.

[Kob00] KOBBELT L.: ($\sqrt{3}$) subdivision. In *SIGGRAPH* (2000), pp. 103–112.

[Kri92] KRIZEK M.: On the maximum angle condition for linear tetrahedral elements. *SIAM J. Num. Analysis 29*, 2 (1992), 513–520.

[KS07] KLINGNER B., SHEWCHUK J.: Agressive tetrahedral mesh improvement. In *Proc. 16th Int. Meshing Roundtable* (2007), pp. 3–23.

[Leh08] LEHNER B.: *Meshing Techniques for Image / Video Compression and Surface Reconstruction*. PhD thesis, TU Kaiserslautern, 2008.

[LG00] LABSIK U., GREINER G.: Interpolatory ($\sqrt{3}$) subdivision. In *Proc. Eurographics* (2000), pp. 131–138.

[Loo87] LOOP C.: *Smooth Subdivision Surfaces Based on Triangles*. Master thesis, University of Utah, 1987.

[LS07] LABELLE F., SHEWCHUK J.: Isosurface stuffing: fast tetrahedral meshes with good dihedral angles. *ACM Trans. Graph. 26*, 3 (2007), 1–57.

[MJ96] MACCRACKEN R., JOY K.: Free-form deformations with lattices of arbitrary topology. In *SIGGRAPH* (1996), pp. 181–188.

[Pas02] PASCUCCI V.: Slow growing volumetric subdivision. In *SIGGRAPH* (2002), pp. 251–251.

[PGH94] PARTHASARATHY V. N., GRAICHEN C. M., HATHAWAY A. F.: A comparison of tetrahedron quality measures. *Finite Elem. Anal. Des. 15* (1994), 255–261.

[PR08] PETERS J., REIF U.: *Subdivision Surfaces*. Springer, 2008.

[RM98] RUPRECHT D., MULLER H.: A scheme for edge-based adaptive tetrahedron subdivision. In *Mathematical Visualization* (1998), pp. 61–70.

[RN00] ROXBOROUGH T., NIELSON G. M.: Tetrahedron based, least squares, progressive volume models with application to freehand ultrasound data. In *Proce. Conf. Visualization* (2000), pp. 93–100.

[Sch07] SCHBERL J.: Netgen—an advancing front 2d/3d-mesh generator based on abstract rules. *Comput. Visual. Sci. 1* (2007), 41–52.

[She98] SHEWCHUK J. R.: Tetrahedral mesh generation by Delaunay refinement. In *Proc. 14th Annu. ACM Sympos. Comput. Geom* (1998), pp. 86–95.

[She02] SHEWCHUK J.: What is a good linear element? Interpolation, conditioning, and quality measures. In *Proceedings of the 11th International Meshing Roundtable* (2002), pp. 115–126.

[SHW04] SCHAEFER S., HAKENBERG J., WARREN J.: Smooth subdivision of tetrahedral meshes. In *Proc. Symp. Geometry Processing* (2004), pp. 147–154.