Journal of Graph Algorithms and Applications

http://www.cs.brown.edu/publications/jgaa/

vol. 5, no. 5, pp. 59–91 (2001)

Shelling Hexahedral Complexes for Mesh Generation

Matthias Müller–Hannemann

Technische Universität Berlin
Fachbereich Mathematik, Sekr. MA 6-1,
Straße des 17. Juni 136,
D 10623 Berlin, Germany
http://www.math.tu-berlin.de/~mhannema
mhannema@math.tu-berlin.de

Abstract

We present a new approach for the generation of hexahedral finite element meshes for solid bodies in computer-aided design. The key idea is to use a purely combinatorial method, namely a shelling process, to decompose a topological ball with a prescribed surface mesh into combinatorial cubes, so-called hexahedra. The shelling corresponds to a series of graph transformations on the surface mesh which is guided by the cycle structure of the combinatorial dual. Our method transforms the graph of the surface mesh iteratively by changing the dual cycle structure until we get the surface mesh of a single hexahedron. Starting with a single hexahedron and reversing the order of the graph transformations, each transformation step can be interpreted as adding one or more hexahedra to the so far created hex complex.

Given an arbitrary solid body, we first decompose it into simpler subdomains equivalent to topological balls by adding virtual 2-manifolds. Second, we determine a compatible quadrilateral surface mesh for all created subdomains. Then, in the main part we can use the shelling of topological balls to build up a hex complex for each subdomain independently. Finally, the combinatorial mesh(es) are embedded into the given solids and smoothed to improve quality.

Communicated by T. Nishizeki, R. Tamassia and D. Wagner:
submitted January 1999; revised August 2000 and July 2001.

The author was partially supported by the special program "Efficient Algorithms for Discrete Problems and Their Applications" of the Deutsche Forschungsgemeinschaft (DFG) under grant Mo 446/2-3.

1 Introduction

In recent years, global competition has led to an increasing demand to reduce the development time for new products. One step in this direction would be a more efficient computer simulation of the technical properties of prototypes for such products. For many years the finite element method has been successfully applied by engineers in simulations. As a prerequisite such a method needs a tool which converts a CAD model into a finite element mesh model suitable for a numerical analysis.

Therefore, various algorithms for the generation of meshes have been developed, mostly decomposing surfaces into triangles and solid bodies into tetrahedra, for surveys see [2, 3]. In many applications, however, quadrilateral and hexahedral meshes have numerical advantages. The potential savings gained from an all-hexahedral meshing tool compared to an analysis based on tetrahedral meshing may be enormous (with estimations in the range of 75% time and cost reductions [37]). On the other hand, the generation of hexahedral meshes turns out to be much more complex than for tetrahedral meshes. Recent years showed many research efforts and brought up several approaches, but up to now, hexahedral mesh generation for an arbitrary 3D solid is still a challenge.

In this paper, we propose a new method for all-hexahedral mesh generation which mainly exploits combinatorial properties of such a mesh. A preliminary description (without proofs) can be found in [21].

Geometric vs. combinatorial meshes. We distinguish between geometric and combinatorial meshes. A *geometric mesh* is a partition of some given domain into subdomains, in our context into hexahedra, i. e. regions combinatorially equivalent to cubes. In contrast, a *combinatorial hexahedral mesh*, is only a decomposition of the given domain into an abstract (cell) complex of combinatorial cubes but without an explicit embedding into space.

Fixed surface meshes. In many applications, in particular in structural mechanics simulations, high mesh quality is required near the boundary of the solid and is much more important than "deep inside the domain". Therefore, it is often preferred to start the volume meshing subject to a fixed quadrilateral surface mesh of an excellent quality. Moreover, the complete solid may consist of several components, for example, solid parts of different material or a solid and its complement within a larger box. Many subdomains may also arise for algorithmic reasons: we may want to divide the domain into smaller and simpler regions either to facilitate the meshing process for each part or to allow parallelism which can be crucial for some large-scale applications. In all such cases, it is usually essential to have a compatible mesh at the common boundary of adjacent components, that is the surface meshes must be compatible. The only solution for this problem we can envision is to prescribe the surface mesh for each component.

Thurston [36] and Mitchell [18] independently characterized the quadrilateral surface meshes which can be extended to hexahedral meshes. They showed

that for a volume which is topologically a ball and which is equipped with an all-quadrilateral surface mesh, there exists a combinatorial hexahedral mesh without further boundary subdivision if and only if the number of quadrilaterals in the surface mesh is even. Furthermore, Eppstein [10] used this existence result and proved that a linear number of hexahedra (in the number of quadrilaterals) are sufficient in such cases. Unfortunately, all these results are not completely constructive and it remains unclear whether they can be extended to constructive methods for geometric meshes. There are quite simple solids with natural looking quadrilateral surface meshes, for example the quadratic pyramid problem of Schneiders [30], where only rather complicated combinatorial meshes are known, but no geometric mesh with an acceptable quality is available.

Related work. We briefly review approaches to hexahedral mesh generation. For a more complete survey, online information and data bases on meshing literature see [29] and [26].

Most commercial systems rely on a mapping approach where the domain must be divided into simple shapes which are meshed separately. For example, *isoparametric mapping* [8] is a method for generating hexahedral meshes which is robust for block-type geometries, but does not work well for more complicated general volumes. Various techniques based on object feature recognition have been developed to automate the subdivision of an object into simpler parts. *Virtual decomposition* [40] separates the volume into mappable subvolumes by the creation of virtual surfaces (2-manifolds) inside the volume. Another method uses the *medial axis* of a surface and *midpoint subdivision* [28, 27]. Compatibility between adjacent subregions and mesh density is then modeled within an integer programming formulation [16]. Unfortunately, solving such integer programming problems is NP-hard, even for quadrilateral surface meshes [20], and such models often rely on a very restricted set of meshing primitives, so-called *templates*.

Grid-based [31] and *octree-based* [32] methods start with a perfect grid which is then adapted to the object's boundary by an isomorphism technique. The adaption step is difficult and often leads to badly shaped hexahedra near the boundary.

Plastering [7, 4] is an advancing front based method which starts from a quadrilateral surface mesh. It maintains throughout the algorithm the *meshing front*, that is a set of quadrilateral faces which represent the boundary of the region(s) yet to be meshed. The plasterer selects iteratively one or more quadrilaterals from the front, attaches a new hexahedron to them, and updates the front until the volume is completely meshed or the algorithm gives up and the remaining voids are filled with tetrahedra.

Whisker weaving [34, 35] also meshes from a quadrilateral surface mesh inward. But in contrast to plastering it first builds the combinatorial dual of a mesh and constructs the primal mesh and its embedding only afterwards. This method is based on the concept of the so-called *spatial twist continuum (STC)* [23, 24]. The STC is an interpretation of the geometric dual of a hexa-

hedral mesh as an arrangement of surfaces, the *sheets*. More precisely, the mesh dual is the cell complex induced by the intersection of the sheets. A fundamental data structure for an STC is a *sheet diagram* which represents the crossings of one sheet with other sheets. Whisker weaving starts with incomplete sheet diagrams based on the surface mesh. It seeks to complete the sheet diagrams by a set of rules which determine the local connectivity of the mesh. The creation of invalid mesh connectivity has been observed in the plastering and whisker weaving algorithms. Heuristic strategies have been developed to resolve such invalidities [35]. Calvo & Idelsohn [6] recently presented rough ideas of a recursive approach which inserts layers of hexahedra separating the whole domain into two subdomains.

We finally mention that some methods relax the meshing problem by allowing mixed elements, that is they try to mesh with mostly hexahedra, but include tetrahedra [37], or pyramids and wedges [17], as well.

A new approach. We suppose that a solid body is described by polygonal surface patches. For the reasons given above, the new approach presented in this paper uses an all-quadrilateral surface mesh as a starting point. But before doing the surface meshing, we first decompose a complex body into simpler subregions by adding internal 2-manifolds. Our method only requires that these subregions are topological balls (to avoid difficulties with holes and voids). It is not necessary, although desirable, that these regions are "almost convex." Roughly speaking, we mean by *almost convex* that a region should not deviate from a convex region by too much (in particular, the local dihedral angles are not too big).

The insertion of additional internal 2-manifolds creates *branchings*, i. e. edges which belong to more than two polygonal patches. All-quadrilateral surface meshing in the presence of branchings can be achieved by first solving a system of linear equations over $GF(2)$, and then using a network flows [22] or an advancing front based method like paving [5].

Because of the practical difficulties indicated by the pyramid example, our approach does not attempt to extend any quadrilateral surface mesh to a hexahedral volume mesh. It seems that self-intersecting cycles in the dual of the surface mesh are the main source for these difficulties. It is a disadvantage of advancing front based methods that they usually generate many such cycles. In contrast, our network flow based mesher [22] tends to produce very regular meshes with only few self-intersecting cycles. To get rid of the remaining self-intersecting cycles we use two strategies. First, we use additional heuristics to avoid them in the surface meshing. Second, we introduce a method which modifies the surface mesh such that all self-intersections disappear. This modification will be explained in Section 5. The important feature of this method is that one can still use any quadrilateral surface mesher (provided it can handle branchings consistently and mesh the virtual internal surfaces without an explicit geometric embedding).

After this preparation, the core of our method is to build up a compatible combinatorial cell complex of hexahedra for a solid body which is topologically

Figure 1: Hexahedron with curved quadrilateral facets (left), a planar surface graph embedding and its combinatorial dual (right).

a ball and for which a quadrilateral surface mesh is prescribed. Such a surface mesh, taken as a graph, is simple, planar, and 3-connected.

The step-wise creation of the hex complex is guided by the cycle structure of the combinatorial dual of the surface mesh. Our method transforms the graph of the surface mesh iteratively by changing the dual cycle structure until we get the surface mesh of a single hexahedron. During the transformation process we keep the invariant that the surface mesh remains simple, planar, and 3-connected. Our main strategy is a successive elimination of dual cycles. Hence, algorithmically, we try to determine a cycle elimination scheme transforming the given surface mesh to the mesh of a single hexahedron.

Starting with a single hexahedron and reversing the order of the graph transformations, each transformation step can be interpreted as adding one or more hexahedra to the so far created hex complex. The embedding and smoothing of the combinatorial mesh(es) finishes the mesh generation process.

Very recently, Folwell & Mitchell [11] changed the original whisker weaving algorithm and incorporated a strategy which is similar to ours. In contrast to us, they have no restrictions on the elimination of a cycle and thereby allow that the surface mesh (as well as the intermediate hex complex) becomes degenerated. Based on the constructions in [18], the new version of whisker weaving applies a number of heuristics to convert a degenerated hex complex into a well-defined one.

Organization. We first introduce some basic terminology in Section 2. Then, in Section 3, we present our new approach for the meshing of topological balls and show its relations to the shelling of cell complexes. In Section 4, we characterize a few classes of surface meshes which have a cycle elimination scheme corresponding to a shelling. A general meshing scheme for arbitrary domains will be given in Section 5. Finally, in Section 6 we summarize the main features of our approach and give directions for future work.

2 Basic Facts and Terminology

Planar graphs. We need some basic graph theory, see for example [25]. *Drawing* a graph in a given space means representing nodes as points and edges as curves. A graph can be *embedded* into some space if it can be drawn such

that no two edges intersect except at a common node. A graph is *planar* if it has an embedding in the plane, or equivalently, an embedding on a sphere in three-dimensional space. The curves representing the edges of a planar, embedded graph partition the plane or the sphere, respectively, into connected components, called *faces*. For a planar graph G, the *(geometric) dual* G^* is constructed as follows. A node v_i^* is placed in each face F_i of G; corresponding to each (primal) edge e of G we draw a dual edge e^* which crosses e but no other edge of G and joins the nodes v_i^* which lie in the faces F_i adjoining e.

A graph is *connected* if there is a path between any two distinct nodes of G, and the graph is *simple* if it has neither loops nor parallel edges. A graph G with at least 4 edges is 3-*connected* if it is simple and cannot be disconnected by removing 1 or 2 nodes from G. The reason for this version of the definitions (here, we follow Ziegler [41]) is that it is invariant under planar duality. That is, if we have a planar embedding of a graph G and construct its dual graph G^*, then G is 3-connected if and only if G^* is 3-connected.

A *(convex) polyhedron* is the intersection of finitely many halfspaces in some $\mathbb{R}^d$, and it is a *polytope* if it is bounded. The famous theorem of Steinitz relates planar graphs and polytopes.

Theorem 2.1 (Steinitz' theorem). *G is the graph of a 3-dimensional polytope if and only if it is simple, planar and 3-connected.*

Hex complex. The bodies we want to mesh have more general, curved surfaces (but, of course, orientable surfaces). Therefore we use the term *(geometric) cell* to mean a bounded region in 3-dimensional space, bounded by a finite number of orientable 2-manifolds. In the following, cells of different dimension will appear: 0-dimensional cells, i. e. single points, called *vertices*; 1-dimensional cells, i. e. segments of curves between two vertices, the *edges*, and 2-dimensional cells in the form of *quadrilateral facets*, i. e. smooth 2-manifolds bounded by a cycle of four distinct edges. A *hexahedron* is a 3-dimensional cell which is a combinatorial cube. It is bounded by 6 distinct quadrilateral facets, 12 distinct edges, and 8 distinct vertices. The quadrilaterals pairwise share edges as depicted in Fig. 1. Ideally, hexahedra are polyhedra, but in this abstract setting we do not require that the edges are straight line segments and that the quadrilateral facets are planar.

A geometric cell complex of hexahedra, called *hex complex* for short, is a finite, non-empty collection $\mathcal{C}$ of distinct (openly disjoint) hexahedra and all their lower dimensional cells such that the intersection of any two members of $\mathcal{C}$ is either empty or a cell of both of them. Two hexahedra are *neighbored* if they share a quadrilateral. By definition, a hexahedron of a geometric cell complex has at most one neighbored hexahedron for each quadrilateral face (*unique neighbor property*).

Combinatorial hex complex. If cells of a hex complex are not embedded but abstract, combinatorial entities, we regard a cell as composed by its lower-dimensional cells. For an abstract cell C, we define the *cell lattice* as the the set of all lower-dimensional cells including the cell C itself and the empty cell,

partially ordered by inclusion. Abstract cells are *combinatorially equivalent* if their cell lattices are isomorphic.

For a hex complex given by abstract cells, a *combinatorial hex complex*, we have to demand explicitly the unique neighbor property (which is no longer implied by definition) to avoid degeneracies. Hence, a combinatorial hex complex is *non-degenerated* if it has the unique neighbor property. A combinatorial hex complex yields a *surface compatible combinatorial mesh* if each quadrilateral is contained in exactly one hexahedron, if it belongs to the surface mesh, and in exactly two hexahedra, otherwise.

Valid geometric meshes. For a vertex of a hexahedron the Jacobian matrix is formed as follows. For that, let $x \in \mathbb{R}^3$ be the position of this vertex and $x_i \in \mathbb{R}^3$ for $i = 1, 2, 3$ be the position of its three neighbors in some fixed order. Using edge vectors $e_i = x_i - x$ with $i = 1, 2, 3$ the Jacobian matrix is then $A = [e_1, e_2, e_3]$. The determinant of the Jacobian matrix is usually called *Jacobian*. A hexahedron is said to be *inverted* if one of its Jacobians is less or equal to zero. As the sign of a determinant depends on the order of its column entries, the latter definition is only useful for checking the quality of a hexahedron if the order of its neighbors is carefully chosen for each vertex. However, a consistent and fixed ordering of the vertices can easily be derived from the combinatorial hex complex by a graph search from some hexahedron lying at the bounding surface.

A hex complex is *compatible* with the quadrilateral surface mesh of a body if this surface mesh is the union of all quadrilateral facets which belong to exactly one hexahedron of the complex. A surface compatible hex complex is a *non-degenerated geometric mesh* if all of its hexahedra are embedded inside the domain and are non-inverted. A quadrilateral surface mesh can also be seen as a cell complex of quadrilaterals. The *dimension* $dim(\mathcal{C})$ of a cell complex is the largest dimension of a cell in $\mathcal{C}$. A cell complex is *pure* if all the inclusion-maximal cells have the same dimension. For example, hex complexes and quadrilateral surface meshes are pure. The *boundary complex* $\partial\mathcal{C}$ of a cell complex is formed by the set of all cells of $\mathcal{C}$ with a dimension lower than $dim(\mathcal{C})$.

Shellings. Shellability of cell complexes has been widely studied in a very general context and is usually defined in a recursive way (on the dimension of the complex) [41]. A *shelling* of a pure d-dimensional cell complex $\mathcal{C}$ is a linear ordering $(F_1, F_2, \ldots, F_k)$ of the set of inclusion-maximal cells, which is arbitrary for $dim(\mathcal{C}) = 0$, but for $dim(\mathcal{C}) > 0$ has to satisfy the following two conditions:

(i) the boundary complex ∂F_1 has a shelling, and

(ii) for every $1 < j \leq k$, the boundary complex ∂F_j has a shelling $(G_1, G_2, \ldots, G_t)$ such that

$$F_j \cap \bigcup_{i=1}^{j-1} F_i = G_1 \cup G_2 \cup \cdots \cup G_r,$$

for some $1 \leq r \leq t$.

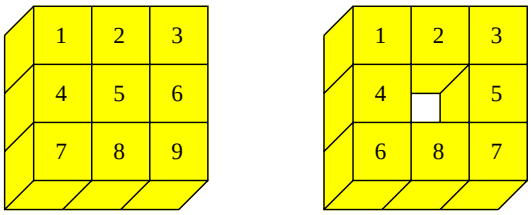

Figure 2: Example of a shellable(left) and a non-shellable (right) hex complex.

A complex is *shellable* if it has a shelling.

For the purpose of shelling hex complexes (or surface meshes) this can be simplified as the shellability of a single hexahedron and its lower-dimensional cells is obvious. Hence, for a hex complex, the concept of shellability can be defined as follows. A *shelling of a hex complex* $\mathcal{C}$ with k hexahedra is a linear ordering $H_1, H_2, \ldots, H_k$ of the hexahedra such that for $1 < j \leq k$ the intersection of the hexahedron H_j with the previous hexahedra is the non-empty union of quadrilateral faces of H_j and these quadrilaterals are connected with respect to the dual graph of H_j, that is

$$H_j \cap \bigcup_{i=1}^{j-1} H_i = Q_1 \cup Q_2 \cup \cdots \cup Q_r,$$

for some dually connected quadrilaterals $Q_1, \ldots Q_r$ of H_j, and $1 \leq r \leq 6$.

In Fig. 2, the left hex complex is shellable in the order of the numbered hexahedra, for example. However, the right hex complex is not shellable. To see this, just note that a beginning of a shelling sequence can never be completed. For example, a shelling sequence could start in the order $1, 2, \ldots, 7$, but the addition of hexahedron no. 8 would violate the condition that the quadrilaterals of the intersection of this hexahedron with the previous ones must be connected.

3 Shelling and Meshing Topological Balls

Our goal is to develop a purely combinatorial approach for hexahedral mesh generation starting from a fixed quadrilateral surface mesh. This implies that we can only use and therefore have to exploit the combinatorial structure of the surface mesh.

Throughout this section we restrict our discussion to the case that our input is a solid body which is topologically a ball. We further assume that a fixed all-quadrilateral surface mesh has been determined for this body. A non-degenerated surface mesh (which we assume) is combinatorially a simple, planar and 3-connected graph. Although the shape of the surfaces we consider is usually more general than that of a convex polytope, the combinatorial structure is not (by Steinitz' theorem).

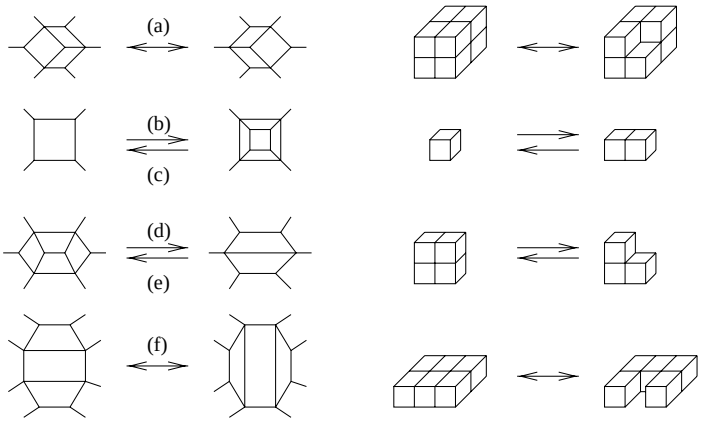

Figure 3: Local graph transformations (a) – (f) (left side) and examples of their application (right side).

Intuitive idea. Shelling a hex complex can be interpreted as building up the complex by successive additions of hexahedra. We would like to do just the opposite, that is to decompose a hex complex by taking away a hexahedron one after another. Of course, in the beginning no hex complex is available to us, all we have is the surface mesh. But if a compatible hex complex were available, then a reversed shelling order would give the desired decomposition. In general, for a given shelling order $H_1, H_2, \ldots, H_k$, the reversed order $H_k, H_{k-1}, \ldots, H_1$ is not again a shelling order (as, for example, in some step H_j may have empty intersection with $H_k, H_{k-1}, \ldots, H_{j+1}$), but it preserves that after j deletions the remaining hex complex is still topologically a ball.

The key observation is that each legal shelling step changes the planar surface graph of the so far created hex complex in a very restrictive way. Namely, such a shelling step corresponds to a local graph transformation which preserves that the surface graph is simple, planar, and 3-connected. The basic idea is to maintain this as an invariant also for the real problem where we do not know the hex complex.

What we are looking for is a *series of graph transformations* on the given surface graph which preserves that the surface mesh is a simple, planar and 3-connected graph. In other words, throughout the decomposition process the surface mesh corresponds to a topological ball. If this process ends up with the surface graph of a single hexahedron, the series of graph transformations yields a *"topology preserving reversed shelling"* of a hex complex. In the following paragraphs, we will make these general ideas more precise.

Shelling steps as graph transformations. Let us suppose for a while that a hex complex is already known for some body. Then we could try to shell this complex. The basic observation is that each shelling step, that is, the transition from a hex complex with k to $k+1$ hexahedra, can be interpreted as a local

Figure 4: Quadrilaterals forming part of a self-intersecting dual cycle (left) and an example of a simple dual cycle (right) where the enclosed quadrilaterals a and b are disconnected with respect to the dual graph (although they share a primal vertex).

graph operation on the surface graph except for the very first shelling step with $k = 0$. Consider the six "local" transformations (a) - (f) shown in Fig. 3 on subgraphs of the surface graph. A transformation is applicable only if the subgraph has at least the edges shown in Fig. 3 (which ensures that each node has minimum degree 3 after the transformation). The operations represent the transformation of the current surface graph for all possible ways to add a single hexahedron to the shelling sequence in a legal way. The basic properties of these transformations are summarized in the next lemma.

Lemma 3.1. *Let G be a simple, planar and 3-connected graph whose faces are all quadrilaterals. Then any application of one of the six operations in Fig. 3 preserves the following invariants for the resulting graph G': G' remains simple, planar, 3-connected, all its faces are quadrilaterals, and the parity of the number of quadrilaterals remains unchanged.*

Observe that each graph transformation is reversible. It can either be interpreted as adding or as deleting of a hexahedron to a hex complex.

Reversed shelling orders as series of graph transformations. As mentioned above, the reversed order of a shelling is no shelling order in general. In contrast, not only all single graph transformations in Fig. 3 are reversible, a whole series is.

Lemma 3.2. *Suppose we are given a shellable hex complex with surface graph $G = G_0$. Then there is a series of graph transformations $g_1, g_2, \ldots, g_k$ from Fig. 3 which transform G successively to $G_1, G_2, \ldots, G_k$ such that G_k is isomorphic to the graph of a single hexahedron. The reversed order of these graph transformations corresponds to a shelling of the hex complex.*

In other words, topology preserving shelling of a hex complex can be seen as applying a series of graph operations on a planar graph. At first glance, this does not help too much as we usually do not know a hex complex compatible to the surface we want to mesh, and so cannot determine which operation we should apply and in what order. So we are faced with the problem of "shelling

an unknown complex," which seems to be intractable. However, we will combine single graph operations to larger units which will help to characterize classes of surface meshes where this concept is useful.

Dual cycle decomposition. Let G be the graph of a quadrilateral mesh and G^* its combinatorial dual. We say that two adjacent dual edges are *opposite to each other* if and only if they correspond to opposite sides of a quadrilateral, i. e. if they are not neighbored in the cyclic adjacency list of their common node. Hence, the four adjacent edges to each dual node can be partitioned into two pairs of opposite edges.

The dual graph $G^* = (V^*, E^*)$ can be decomposed in a canonical way into a collection of edge-disjoint cycles, say into $C_1, \ldots, C_k$, by putting a pair of adjacent edges e_1^*, e_2^* into the same cycle if they are opposite to each other. In other words, for each quadrilateral the edges which are dual to opposite sides are contained in the same cycle. With respect to the labeling in Fig. 1, the cycle decomposition for a single hexahedron yields the cycles $C_1 = \{a, b, c, d\}$, $C_2 = \{a, e, c, f\}$, and $C_3 = \{b, f, d, e\}$ (cycles taken as node sequences).

Observe that by transitivity two edges may belong to the same cycle even if they are neighbored in the cyclic adjacency list of some node. Hence, these dual cycles can be non-simple or *self-intersecting*, see Fig. 4.

Note that the set of dual cycles is well-defined and unique (in the sense that two cycles are equivalent if they have the same set of edges) and can be easily determined in linear time. We call this set the *canonical dual cycles* of G^*, and by a dual cycle we will henceforth always mean a canonical dual cycle.

Cycle elimination. Next we introduce the concept of a cycle elimination. To get an intuitive idea of the use of cycle eliminations for the meshing see Fig. 5. The elimination corresponds to the removal of a complete layer of hexahedra, that is in STC terminology to the removal of a complete sheet.

Let us suppose that the edges of a dual cycle are ordered within a cyclic list such that two edges are consecutive if and only if they are adjacent and opposite to each other with respect to their common node. The cyclic order of the edge list induces an orientation of the cycle. With respect to such an orientation, a simple dual cycle C separates the dual vertices $V^* \setminus V^*(C)$ into vertices $V^*_{C,\ell}$ on the "left hand side" of C and vertices $V^*_{C,r}$ on the "right hand side."

Given a planar graph G, its dual G^*, and a dual cycle C of G^*, the elimination of C transforms G to G' and G^* to G'^* in the following way. The new graph G' is obtained from G by contracting each primal edge corresponding to a dual edge contained in C, and removing parallel edges afterwards. The graph G'^* is then defined as the combinatorial dual of G'. An equivalent way to describe the elimination of a dual cycle is to remove all edges of C from G^*, and to replace for each node v^* of C the two remaining incident edges, say (u^*, v^*) and (w^*, v^*), by the new dual edge (u^*, w^*), and finally remove all vertices of C. In this second definition, dualization of G'^* gives the new primal graph G'.

In order to maintain the above mentioned invariants of the surface graph and to yield a connection to shellings (which will be explained below), cycle

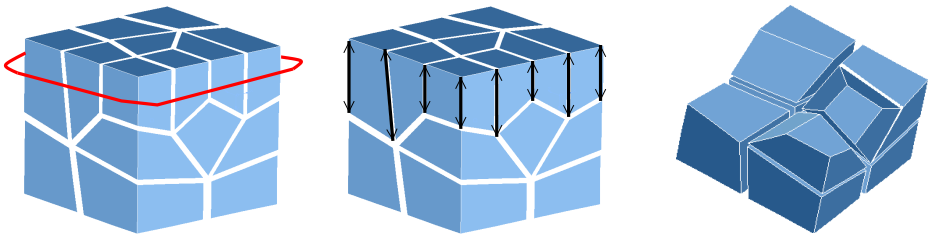

Figure 5: The elimination of a dual cycle (left) corresponds to the contraction of primal edges (middle) and leads to the removal of a sheet of hexahedra (right). In this example, we have taken the nine quadrilaterals on the top facet as the connected set of enclosed quadrilaterals.

eliminations have to be restricted. A *feasible elimination* of a dual cycle C from G^* requires that

1. C is a simple cycle,
2. at least one of the two subgraphs of G^* induced by $V^*_{C,\ell}$ and $V^*_{C,r}$, respectively, is connected, and
3. G'^* is 3-connected.

Note that 3-connectivity can be checked in linear time [13]. So we can test in linear time whether a cycle can be eliminated in a feasible way or not.

Cycle elimination corresponds to shelling operations. The concept of a feasible elimination is closely connected to shelling and graph transformations. It turns out that a feasible cycle elimination is equivalent to a series of graph operations and therefore also to shelling operations.

Suppose that G' is the surface mesh of some hex complex. If we (re)insert a dual cycle C which has been eliminated in a feasible way from graph G, this reinsertion can be interpreted as adding a complete layer (sheet) of hexahedra. Namely, if we assume that the subgraph G^* induced by the quadrilaterals $V^*_{C,\ell}$ on the left hand side of C is connected, then there is a one-to-one correspondence between the quadrilaterals belonging to the set $V^*_{C,\ell}$ and the hexahedra belonging to the added layer.

Lemma 3.3. *Let G be the graph of a quadrilateral mesh and G^* be its combinatorial dual. Let C be a dual cycle which can be eliminated in a feasible way from G^*. If $V^*_{C,\ell}$ ($V^*_{C,r}$) is connected then there is a sequence of exactly $|V^*_{C,\ell}|$ ($|V^*_{C,r}|$) graph operations from Fig. 3 which transforms G to G'.*

Proof: We first observe that the 2-dimensional cell complex formed by the union of the quadrilaterals contained in $V^*_{C,\ell}$ is shellable. This follows from a characterization of the shellability of 2-dimensional (pseudo)-manifolds [9]. In

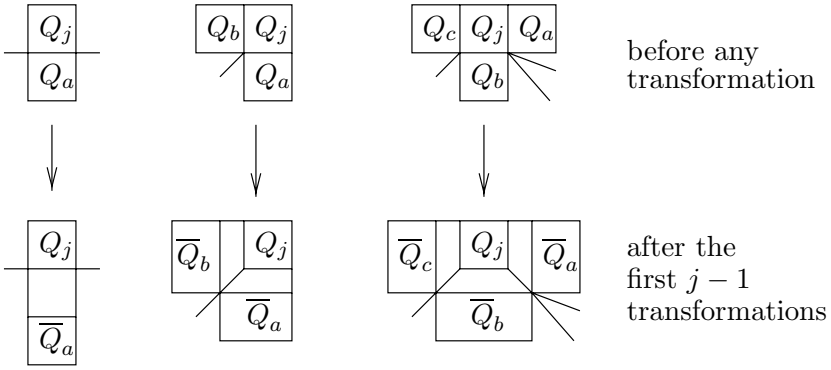

Figure 6: The different cases of local subgraphs before a hexahedron is placed on top of quadrilateral Q_j (Q_a, Q_b and Q_c come before Q_j in the shelling order). The number of quadrilaterals sharing a corner with Q_j and lying "between" Q_a and Q_b (or "between" Q_b and Q_c) can be an arbitrary non-negative number (including zero!).

our setting this characterization asserts that the cell complex is shellable if the planar primal graph G_ℓ induced by these quadrilaterals can be embedded in a 2-ball or 2-sphere M in such a way that $M \setminus G_\ell$ is the union of $|V^*_{C,\ell}|$ pairwise disjoint open 2-balls whose boundaries are the bounding cycles of the quadrilaterals. Clearly, these conditions are fulfilled as the complete quadrilateral mesh is embedded in a 2-manifold and so the quadrilaterals contained in $V^*_{C,\ell}$ are embedded in a 2-ball by our assumption that the dual subgraph induced by $V^*_{C,\ell}$ is connected.

We claim that any shelling order $(Q_1, Q_2, \ldots, Q_k)$ of quadrilaterals (with $k = |V^*_{C,\ell}|$) determines a possible series of graph operations to transform G' to G. The general idea is to place a hexahedron "on top" of each of quadrilaterals Q_j in the shelling order. At each step, we maintain the property that the newly created hexahedron contains the quadrilateral Q_j as the "bottom facet," and exactly one quadrilateral facet from each hexahedron which has been placed on a quadrilateral Q_i adjacent to Q_j with $i < j$. Thus, all these faces disappear from the surface graph. At the same time, the addition of a hexahedron creates new quadrilateral facets, namely the "top facet", denoted by $\overline{Q}_j$ and all the remaining facets of the hexahedron which are not shared with a previously added hexahedron.

Hence, the graph transformation starts with operation (b) of Fig. 3 applied to the first quadrilateral Q_1 in the shelling order. For the j-th element Q_j of the shelling order ($1 < j \leq k$), the shelling conditions guarantee that the intersection of Q_j with quadrilaterals which come earlier in the shelling order is a connected set of edges. Moreover, the following is a crucial observation about a correct shelling order. If Q_a and Q_b are quadrilaterals which come before Q_j in the shelling order and both share an edge with Q_j incident to a common

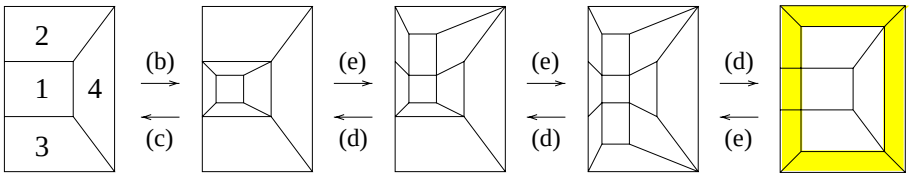

Figure 7: Graph transformations corresponding to a cycle elimination/reinsertion according to the shelling order (1,2,3,4).

vertex v, then all other quadrilaterals incident to this vertex v also precede Q_j in the given shelling order. Using this observation, one can show by induction that the local graph structure at Q_j of the surface graph after the first $j-1$ transformations ($j > 1$) looks as depicted in Fig. 6.

Hence, if the intersection of Q_j with preceding quadrilaterals is just a single edge shared with Q_a, we can apply transformation (e) of Fig. 3 using the facet Q_j and the adjacent quadrilaterals of the surface graph which has been created in the a-th step. If the intersection are two edges shared with Q_a and Q_b, we apply transformation (a) of Fig. 3 using the facet Q_j and the two adjacent quadrilaterals at the common corner of Q_a and Q_b in the current surface graph. Similarly, if the intersection contains three edges, we apply transformation (d) of Fig. 3. Note finally, that for any shelling order it is impossible that Q_j is completely surrounded by preceding quadrilaterals. The resulting sequence of graph operations transforms G' to G, i. e. it inserts a dual cycle. Reversing the order of transformations and each single operation yields the claimed sequence of transformations. □

The shelling order for the enclosed quadrilaterals can be found in linear time using an algorithm of Danaraj and Klee [9]. Basically, this algorithm adds a quadrilateral to the initial shelling sequence in a greedy fashion if the shelling conditions are fulfilled. Danaraj and Klee's algorithm can be simplified in the special case of quadrilateral cells as we can check in constant time whether a quadrilateral can be added to an initial shelling sequence or not.

Fig. 7 shows a small example of a dual cycle which encloses four quadrilaterals and the corresponding graph transformations of a (re)insertion ("from left to right") or an elimination ("from right to left").

Note that in the example of Fig. 5 we have taken the nine quadrilaterals on the top facet as the set of quadrilaterals enclosed by the eliminated cycle. The quadrilaterals on the opposite side of the cycle are also connected with respect to the dual graph. However, if we take this other set of quadrilaterals as the basis for our sequence of graph transformations this would lead to a different hex complex.

Perfect cycle elimination schemes. A *perfect cycle elimination scheme* of a dual graph G^* of a quadrilateral surface mesh is an order $C_1, C_2, \ldots, C_k$ of its

k canonical cycles such that the first $k-3$ cycles with respect to this order can be eliminated one after another in a feasible way, and such that the remaining 3 cycles form the dual surface graph of a single hexahedron. In particular, this means that the last 3 cycles are simple and cross pairwise exactly twice (two cycles *cross* if they share a common node).

The core algorithm. Whenever we know a perfect cycle elimination scheme for some graph G^*, we can iteratively apply Lemma 3.3 in reversed elimination order to give an explicit construction of a hex complex compatible to the prescribed surface mesh. This yields an algorithm for combinatorial meshes of topological balls which runs in two phases: In the first phase we determine a cycle elimination scheme, and in the second phase we build up a hex complex by adding sheets in reversed order of the elimination.

Example: the "double fold" problem revisited. We conclude this section with the example of the so-called "double fold" problem to illustrate how our core algorithm works. The same example has been used to study the whisker weaving algorithm [34]. The geometry of this example is simply a cube, but the surface mesh is quite irregular (several nodes of degree five and three), see Fig. 8. The Figs. 9 to 12 show the successive elimination of dual cycles from the original surface mesh to the mesh corresponding to a single hexahedron, and the construction phase. The first cycle to be eliminated encloses 9 dual nodes, the second 3, the third 2, and the fifth and sixth only one node. Hence, in the reversed order, the series of hexahedra to add to the first one is 1,1,2,3,9.

Figure 8: The "doublefold" example (the middle picture is the left cube with the front faces removed so you can see through to the back faces from the inside). The hex complex consists of 17 hexahedra.

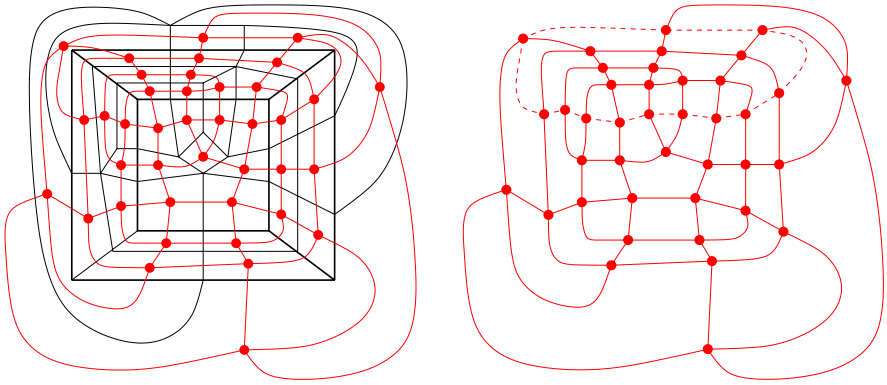

Figure 9: The primal and dual graph of the surface mesh of the "double fold" example, dual nodes are drawn as solid circles (left); the first dual cycle of a perfect elimination scheme is dashed, it encloses 9 dual nodes (right).

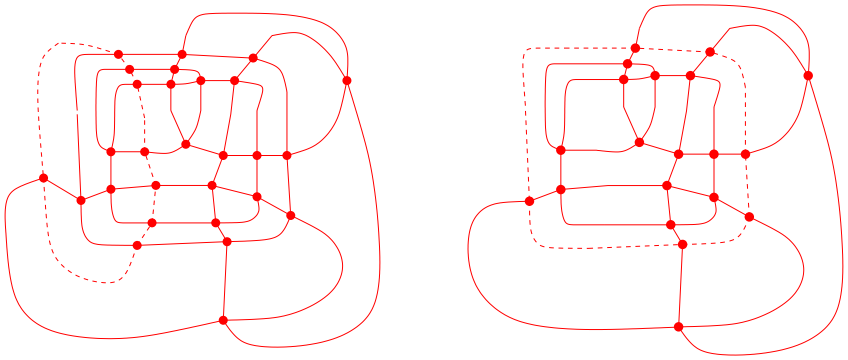

Figure 10: The dual graph after the first and second elimination. The cycle to be eliminated next is again dashed. The dashed cycle on the left side encloses 3 dual nodes, on the right side 2 dual nodes (on the unbounded side of this drawing!).

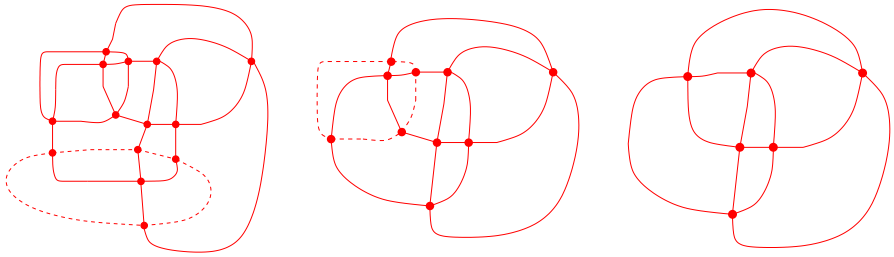

Figure 11: The dual graph after the third, forth and fifth elimination. The right figure shows final configuration of the remaining three cycles.

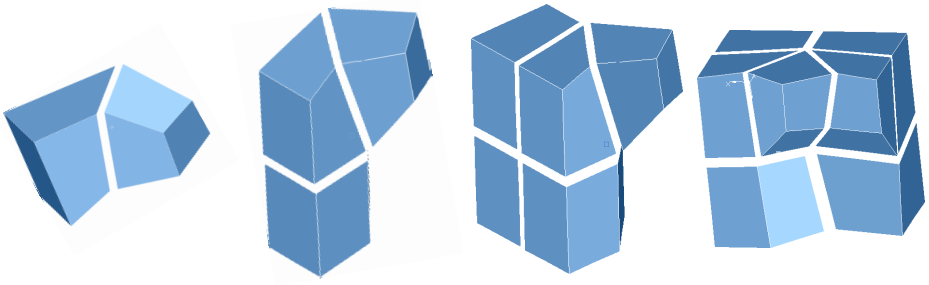

Figure 12: Construction of the hex complex for the "doublefold" example. The figures show from left to right the intermediate steps where hexahedra are added in reversed order of the elimination scheme.

4 Classes of Surface Meshes with a Perfect Elimination Scheme

As our approach relies on finding a perfect elimination scheme we would like to characterize the surface meshes of topological balls which admit such a scheme. While the decision problem whether a single cycle can be eliminated in a feasible way is solvable in polynomial time, the complexity status for the recognition of a perfect elimination scheme is still open (to our knowledge).

There are some obvious necessary conditions. By definition, existence of a perfect cycle elimination scheme requires that all dual cycles are simple. We will show in Section 5 how to guarantee this property for our surface meshes. Simplicity of a dual cycle implies even length, and all cycles being simple implies an even number of quadrilaterals. The latter is sufficient for the existence of a combinatorial mesh, but not for the existence of a perfect elimination scheme in general. To see a small counter-example, consider a 3-connected planar dual graph consisting of three canonical cycles which cross each other pairwise exactly four times (hence, we cannot reach the desired final state wherein three cycles cross pairwise twice).

We also note that non-shellable hex complexes exist. Furch's knotted hole ball is a well-known example of a hex complex (see Fig. 13) which is non-shellable [42]. However, it is insightful to check that the corresponding surface mesh has a perfect elimination scheme, and so can be decomposed into some hex complex, which is, of course, different from the non-shellable one. In other words, this shows that a fixed surface mesh may have different decompositions leading to shellable and non-shellable hex complexes.

Cycle arrangements. Consider a collection of k simple, closed Jordan curves in the plane such that no three curves meet in a point, which we call a *cycle arrangement*. We may associate in the obvious way a planar graph G^* to such a collection of curves (considered as a dual graph): The vertex set is formed by the intersection points between curves, and the edges are induced by the

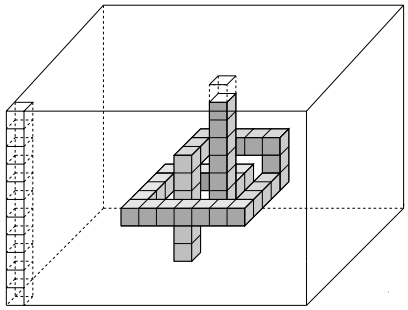

Figure 13: Furch's "knotted hole ball": a pile of $(k_1 \times k_2 \times k_3)$ hexahedra where hexahedra are missing along a knotted curve except for a single hexahedron at the top layer. This hex complex is non-shellable, but the surface mesh has a perfect elimination scheme.

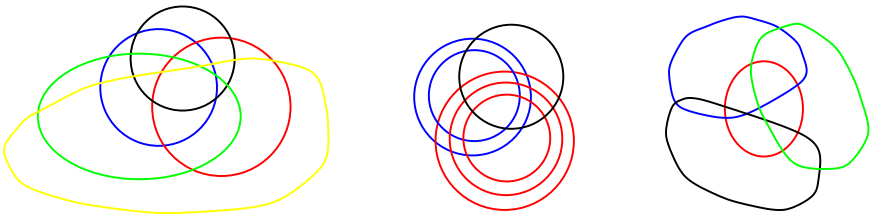

Figure 14: Examples of cycle arrangements of surface meshes: a zonotopal arrangement (left), a zonotopal arrangement with three equivalence classes of parallel neighbors (middle), and an arrangement of Class III (right). Note that the elimination order is not arbitrary in the latter example.

segments between intersection points. If G^* is simple and 3-connected, then so is the primal graph. As the degree of each dual vertex is four, such a primal graph corresponds to a quadrilateral surface mesh. Hence, any cycle arrangement such that the associated planar graph is simple and 3-connected is a *cycle arrangement of a surface mesh.*

Next we consider different classes of cycle arrangements which have perfect elimination schemes, examples are shown in Fig. 14.

Class I: "zonotopal cycle arrangements". Zonotopes are special polytopes which can be defined in many equivalent ways [41], for example, as the image of a higher-dimensional cube under an affine projection, or as the Minkowski sum of a set of line segments. It is well-known that zonotopes where all surface facets are quadrilaterals always have decompositions into hexahedra [33].

Moreover, such zonotopes constitute a class of polytopes for which every cycle order leads to a perfect elimination scheme (the "zones" of a zonotope

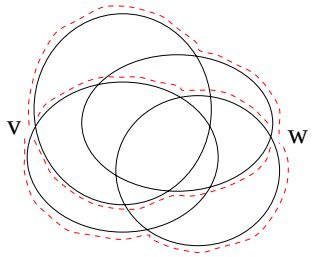

Figure 15: Zonotopal cycle arrangements are 4-connected: there exist four disjoint paths (dashed) between v and w.

correspond to the dual cycles). Certainly, for the existence of an elimination scheme it is sufficient that the surface graph of some mesh is combinatorially equivalent to that of a zonotope.

The dual cycles in graphs arising from zonotopes with $k+1$ zones have a nice combinatorial property: The dual cycles cross each other pairwise exactly twice, i. e. each dual cycle has length $2k$, and if a dual cycle is traversed, the first k intersections with other cycles appear in the same order as the second k intersections. We call an arrangement of three or more simple cycles which all have this ordering property a *zonotopal cycle arrangement.*

Lemma 4.1. *A zonotopal cycle arrangement has a perfect elimination scheme and every cycle order yields such a scheme.*

Proof: The only zonotopal cycle arrangement with three cycles is the arrangement of a single hexahedron, and so trivially has a perfect elimination scheme.

Any zonotopal cycle arrangement with more than three cycles is not only 3-connected (as we require for an elimination), it is even 4-connected. To see this, consider an arbitrary pair of vertices $v, w \in G^*$ and construct four internally vertex disjoint paths using the cycles through v and w as indicated in Fig. 15.

Hence, for a zonotopal cycle arrangement with $k+1$ cycles, we may select an arbitrary cycle C for elimination and the resulting graph remains 4-connected. For a feasible elimination we have still to check that, for some orientation of C, the dual vertex set $V^*_{C,\ell}$ on the left hand side from C is connected. To see this, consider any two dual vertices $v^*, w^* \in V^*_{C,\ell}$. Denote by C_{v^*} and C_{w^*} dual cycles going through (that is containing) v^* and w^*. If the two cycles are identical or intersect at some third vertex $s^* \in V^*_{C,\ell}$, there is a path from v^* to w^*. If, however, such an intersection between C_{v^*} and C_{w^*} does not exist within $V^*_{C,\ell}$, we get a contradiction to the order of the intersections with C in a zonotopal cycle arrangement, see Fig. 16. □

Class II: "zonotopal cycle arrangements with parallel cycles". We say that a cycle C_1 is a *parallel neighbor* of another cycle C_2 of the same length in the graph G^* if their node sets are disjoint but for each node v^* of C_1 there is a node w^* of C_2 such that the edge (v^*, w^*) belongs to G^*.

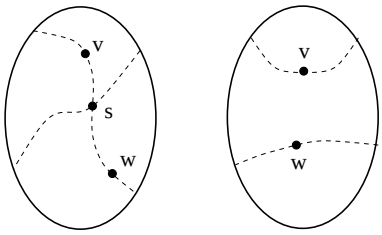

Figure 16: Vertices inside a cycle C of a zonotopal arrangement are always connected (left); otherwise the intersections with C appear in the wrong order (right).

Lemma 4.2. *If the dual cycles are partitioned into equivalence classes of parallel neighbors, then a mesh has a perfect elimination scheme if representing cycles of these equivalence classes form a zonotopal cycle arrangement. In this case, any elimination order is possible provided it keeps cycles from three different equivalence classes to the end.*

Proof: Just note that the class of zonotopal cycle arrangements with parallel neighbor cycles is closed under elimination of an arbitrary cycle (provided that the remaining arrangement has at least three equivalence classes). Then, basically the same arguments as in the proof of Lemma 4.1 show that such a cycle arrangement is 4-connected, and that the dual vertex set $V^*_{C,\ell}$ on the left hand side from an arbitrary cycle C is connected. □

Note that although the order of eliminations is almost arbitrary, a different hexahedral mesh will result for different orders.

Class III: "cycles cross pairwise twice plus 3-connectedness". The condition "all cycles cross pairwise exactly twice" is to weak for a guarantee of a perfect elimination scheme, for an example see the picture on the right in Fig. 14 without the "inner cycle."

We consider the following recursively defined class of cycle arrangements, called *class III*. The only member of class III with three cycles is the arrangement of a hexahedron. A cycle arrangement with $k+1$ cycles belongs to class III, if it can be derived from a member of class III with k cycles by the insertion of one simple cycle which crosses all others exactly twice and the resulting graph is 3-connected.

Lemma 4.3. *A cycle arrangement of class III has a perfect elimination scheme.*

Proof: By definition, a cycle arrangement which is member of class III can be constructed by the stepwise insertion of cycles. We claim that the reversed order of the insertion yields a perfect elimination scheme. Hence, if we eliminate a cycle, 3-connectivity is asserted. The only non-trivial fact to show is that either the vertex sets $V^*_{C,\ell}$ on the left hand side or on the right hand side $V^*_{C,r}$ are

connected. (It is not hard to construct an example where one of the two sets, say $V^*_{C,\ell}$, is disconnected.)

Assume that both sets are disconnected. This means that there are disconnected nodes $v, w \in V^*_{C,\ell}$ and $s, t \in V^*_{C,r}$. Denote by C_v, C_w, C_s, C_t cycles going through the vertices v, w, s, t. As v, w are not connected within $V^*_{C,\ell}$, the cycles C_v and C_w cannot intersect within $V^*_{C,\ell}$. Similarly, the cycles C_s and C_t do not cross within $V^*_{C,r}$. As cycles cross pairwise, this implies that C_v and C_w intersect in $V^*_{C,r}$, and C_s and C_t intersect in $V^*_{C,\ell}$.

Suppose that neither C_s nor C_t intersect C_v within $V^*_{C,\ell}$. This would imply that both cycles intersect C_v in $V^*_{C,r}$, in contradiction to the assumption that s and t are not connected in $V^*_{C,r}$. Hence, we may assume that at least one of both cycles, say C_s, intersects C_v in $V^*_{C,\ell}$. This in turn means that C_s crosses C_w in $V^*_{C,r}$. Now we get a contradiction. Either C_t crosses C_w in $V^*_{C,\ell}$ which would mean that v and w are connected in $V^*_{C,\ell}$ (via portions of C_v, C_s, C_t and C_w), or C_t crosses C_w in $V^*_{C,r}$, but then s and t are connected in $V^*_{C,r}$ (via portions of C_s,C_w and C_t). □

5 Hexahedral Meshing for Arbitrary Domains

In this section we consider the meshing of arbitrary domains. We do not restrict our discussion to one of the many different CAD formats and possibilities to encode surfaces. However, we will assume that the CAD input model for our algorithm describes a solid body by polygonal surface patches.

The general algorithm. For arbitrary domains, we propose the following approach which consists of five major steps:

1. Decompose the whole domain into subdomains which are topologically balls and "almost convex."
2. Quadrangulate the surface mesh with half of the required density. Replace each quadrangle by four new ones.
3. Cancel self-intersecting dual cycles.
4. Do for each subdomain
   (a) Search for a perfect cycle elimination scheme.
   (b) Build up a hex complex.
5. Embed the hex complex and perform mesh smoothing.

In the following we will explain the steps of our general scheme.

Step 1: Decomposition into subdomains. In the very first step we have to decompose the given body into simpler subdomains which are topologically

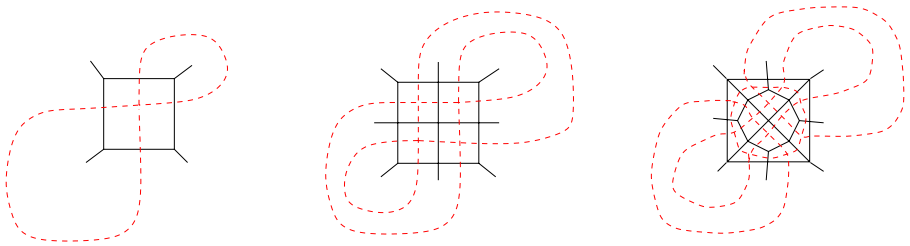

Figure 17: Getting rid of a self-intersecting dual cycle (left): the dual cycle is first duplicated (middle), and then the surface mesh is locally transformed (right).

balls and should be "almost convex." In general, this is a difficult problem which offers many degrees of freedom in how to subdivide a body. Solution methods depend very much on the used CAD data structures and geometric representations of the input model. The corresponding details go beyond the scope of this paper. We only mention that we add internal 2-manifolds using ideas of White et al. [40], but with weaker requirements for the resulting subregions. It is not necessary, although desirable, that these regions are "almost convex." Asking for a subdivision into convex regions in the usual mathematical sense would be too strict, as otherwise there would be no subdivision into finitely many regions for concave surfaces. So, roughly speaking, we mean by *almost convex* that a region should not deviate from a convex region by too much.

Step 2: Surface mesh quadrangulation. We can also be brief with remarks on the surface mesh generation, as any method which works consistently with branchings (because of the insertion of internal 2-manifolds) can be chosen. In our experiments, we used the surface meshing tool as described in [19, 22] without changes.

Step 3: Cancel self-intersecting dual cycles. We have to ensure that all dual cycles are simple as this is necessary for the existence of a perfect elimination scheme. Our preliminary paper [21] as well as Folwell & Mitchell [11] describe how to modify a quadrilateral surface mesh in order to remove self-intersections for a single domain. The main idea of the latter approach is to collapse a quadrilateral where a self-intersection occurs into two edges. This may result in degenerated vertices with degree two, but such a situation can be resolved by a so-called "pillowing" technique which places an additional ring of quadrilaterals around such a degenerated vertex. However, without a modification both methods have a serious drawback as they do not work in the presence of branchings. If branchings occur, the problem is that at least some dual cycles belong to several subdomains and cannot be modified independently. Hence, it may happen that self-intersections are removed in one subdomain but new self-intersections are created in some other subdomain.

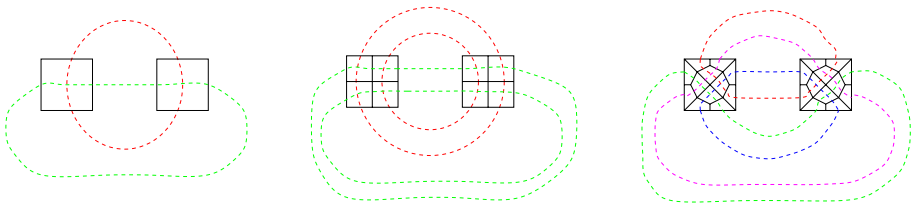

Figure 18: Simplicity of a pair of simple cycles (before duplication, left) is maintained if we apply the graph transformation at all crossings.

In this paper, we propose an alternative method which can also be applied in the presence of branchings. It works as follows: After decomposing the domain of our solid body into topological balls, we use the quadrilateral surface mesher with only half the required mesh density. Then we subdivide each quadrilateral into four new ones (by halving all edges) to meet the required mesh density.

This replacement duplicates all dual cycles. In particular, all quadrilaterals where self-intersections occur appear in pairs, see the middle part of Fig. 17. So it is possible to change the surface mesh locally at all such places, by replacing four quadrilaterals with 12 new ones, see the right part of Fig. 17. Obviously, the surface graph remains planar, simple and 3-connected, and we do not change the structure of other cycles than those going through such quadrilaterals. Most importantly, it serves its primary purpose to resolve each existing self-intersection. If a quadrilateral where a self-intersection occurred belongs to exactly one subdomain, the transformation cannot create a new self-intersection, otherwise it might do so with respect to the other subdomain. More precisely, if we apply the transformation to a quadrilateral where to simple dual cycles cross with respect to some subdomain (the common quadrilaterals of simple dual cycles are called *crossings*) then this transformation would create a self-intersection at one of the new quadrilaterals (if the two dual cycles are not changed in their remaining parts). Hence, we need some additional step for each pair of simple cycles affected by a transformation for some other subdomain. Fortunately, whenever necessary, we can avoid the complication of creating self-intersecting cycles if we apply the same local transformation at *all* four-tuples of quadrilaterals corresponding to crossings, for an example with two crossings see Fig. 18. (As the proposed transformation typically degrades the surface mesh quality we do not apply this transformation everywhere.) It is not hard but insightful to verify that this modification creates always simple cycles, for any number of crossings.

Step 4a: Search for a perfect elimination scheme. In principle, every perfect cycle elimination scheme allows us to build up a valid combinatorial mesh. Unfortunately we do not know an efficient algorithm to find such a scheme in the general case. Hence, we use a greedy strategy which iteratively eliminates some feasible cycle. However, the choice which cycle to eliminate next should not be arbitrary for several reasons. First, we note that, in general, different

elimination orders, lead to meshes of a different structure and size. Second, we also observe that two different geometric meshes can have the same combinatorial mesh, see Fig. 19. Hence, a careful cycle selection has to take the geometry of the surface into account.

Cycle selection. For that purpose, we determine for each dual edge the dihedral angle between the two quadrilateral faces which correspond to its endpoints. This, in turn, gives us a an initial classification for each primal edge as "sharp" or "plane" edges. A primal edge of the surface graph is a *sharp edge* if the dihedral angle is significantly smaller than 180 degrees.

For a simple dual cycle C, we use the term *neighboring primal cycles* to mean the two connected primal cycles induced by the union of all primal edges of the quadrilaterals corresponding to C which do not cross dual edges of C. We assign a *side elimination weight* to each dual cycle according to the number of sharp edges of the neighboring primal cycles, divided by the cycle length (number of edges). (In our implementation of side elimination weights, we classify a dihedral angle smaller than 120 degrees as sharp.) The weights can be used to define a preference order for dual cycles. A *first rule* is that a cycle should be preferred in the selection if it has a higher weight, i. e. if it has a higher quotient of sharp edges to the total number of cycle edges than some other cycle. We also keep track of the side for which the elimination weight has more sharp edges, as this side will be used as the enclosed set of quadrilaterals on which the construction phase adds hexahedra.

Moreover, we use a weight counting the number of sharp primal edges corresponding to edges of the dual cycle. A *second rule* says that one should eliminate a cycle only if this second weight is positive. The intuition behind this rule is that one should not eliminate a cycle which lies in a plane, as this may lead to a bad quality of the geometric mesh. (For that purpose, we use a different parameter classifying an edge as a plane edge, if the dihedral angle is larger than 170 degrees.)

After each cycle elimination we update this classification for all primal edges which are affected by the elimination. To be precise, if two primal edges become identified by contraction of a quadrilateral, the resulting edge is classified as a sharp edge if at least one of the edges was sharp before the identification. Hence, the weights of a dual cycle will change after eliminations.

Double cycle elimination. Suppose that some dual cycle C_m has two parallel neighbors, one on the right and one on the left side. Then it can certainly be eliminated in a feasible way. If, in addition, all edges of the corresponding left and right primal neighboring cycles are classified as sharp edges, the cycle C_m has a high preference to be selected for elimination. However, in such a situation it seems to be better to eliminate both the left and right neighboring parallel cycles simultaneously. In such a *double elimination* the set of enclosed quadrilaterals is just the dual cycle C_m in the middle. The elimination can be interpreted as removing a torus of hexahedra, see the left example of Fig. 19. Similarly to the statement of Lemma 3.3, one can prove that a double elimina-

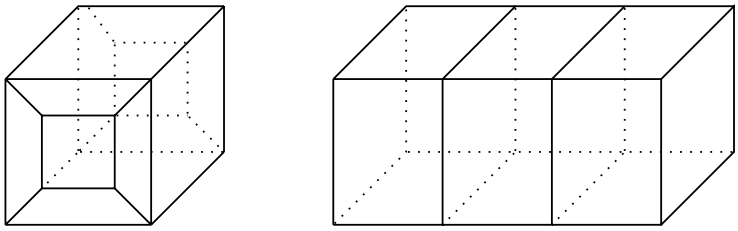

Figure 19: Examples of two solids with "obvious" decompositions into five and three hexahedra. Note that the surface meshes are combinatorially equivalent. Therefore, additional geometric information is necessary to yield the appropriate decompositions. For the left instance, a double cycle elimination is appropriate.

tion corresponds to a series of graph transformations from Fig. 3 (with $|V(C_m)|$ operations in total).

Adding new cycles. Of course, our greedy cycle selection can get stuck. We now sketch a strategy to resolve the situation that no cycle can be eliminated in a feasible way.

Even if no cycle can be eliminated, we can certainly still perform graph transformations (for which we have many degrees of freedom). Recall that all graph transformations of Fig. 3 can be reversed. Hence, through a series of such transformations we cannot only eliminate a dual cycle but also insert a new one. If no cycle can be eliminated in a feasible way, the idea is to add one or more new cycles in such a way to the current graph that at least one of the "old cycles" can be feasibly eliminated afterwards. We choose some cycle according to our selection criteria for elimination. The check for 3-connectivity after a testwise elimination tells not only a failure, it also gives us all 2-separators. To augment the local connectivity, a new cycle is inserted around a node of a 2-separator preferably in such a way that it crosses other cycles exactly twice. The latter heuristical placement is motivated by our characterization of shellable surface.

Step 4b: Building up the hex complexes. As soon as a perfect elimination scheme has been determined for some domain, we reverse the order of cycle eliminations and build up a hex complex layer by layer as described in Section 3.

Step 5: Embedding and smoothing. Up to this point we have only discussed how to find a combinatorial mesh. Embedding of a combinatorial hex complex into the prescribed surface such that all hexahedra are well-shaped is a non-trivial task. It is not even clear which conditions are sufficient for a "nice" embedding.

However, for convex domains we have been quite successful with a strikingly simple strategy, namely with a barycentric embedding algorithm (commonly also referred to as *Laplacian smoothing*).

This strategy has been inspired by a beautiful result of Tutte [38, 39] for

drawing planar graphs in the plane. Tutte showed that the barycentric straight-line embedding draws a 3-connected planar graph without crossings in the plane such that every facet is convex if the outer facet is prescribed as a convex polygon.

For our purposes, we consider a hex complex as a graph, fix the surface nodes according to the surface mesh, and for all other nodes the node position is determined as a convex combination (the barycenter) of the position of its neighbors. More precisely, the barycentric embedding determines vertex positions for all movable vertices from

$$p_v = \frac{1}{deg_v} \sum_{(v,w)\in E} p_w,$$

where we denote by p_v the position of vertex v in $\mathbb{R}^3$ and by deg_v the degree of vertex v. The barycentric embedding is the unique minimizer of the energy functional

$$E(p_1, \ldots, p_n) = \frac{1}{2} \sum_{(v,w)\in E} ||p_v - p_w||^2,$$

where $p_{k+1}, \ldots, p_n$ are fixed positions of vertices on the surface. A straightforward Gauß-Seidel iteration on the system of equations for the node positions converges quickly and is computationally very fast. The barycentric embedding can be interpreted as a special case of force-directed algorithms in graph drawing, see for example [1, Chapter 10].

In the experiments with convex domains mentioned in this paper, the initial layout obtained from this method was already excellent. However, for this simple to implement and comparably fast algorithm it is well-known that it might fail to produce valid meshes (indeed it is quite likely to fail for non-convex domains). Therefore, we implemented additional embedding algorithms for optimized mesh smoothing and untangling, i.e., for getting rid of inverted hexahedra, as described in the pioneering work of Freitag and Knupp [12, 14, 15].

Example: Model of a shaft. We have implemented a first prototype for our approach in C++. To illustrate first experimental results we use the CAD model of a shaft with a flange shown in Fig. 20. Several features make this model useful for testing purposes: it contains 8 holes, a cylindrical inlet in the lower part (not visible in the hidden surface description of Fig. 20), and concave surface patches. The input model is axis-symmetric, and so is our output – although our algorithm has no tools incorporated to detect and to exploit symmetry. The complex input model has first been decomposed into topological balls, see the left part of Fig. 20. Then our quadrilateral meshing algorithm [22] has been run according to some specified uniform mesh density. The resulting surface mesh has the nice property to be free of self-intersecting dual cycles for all subregions. For all subregions, our algorithm was successful to find a perfect cycle elimination scheme. Even more, the embedding of the hex complexes has led also to geometric meshes of a very good quality. Figs. 21,

22, 23 and 24 show the resulting meshes for different parts of the model. In total, the hexahedral mesh for the complete model, depicted in the right part of Fig. 20, consists of 7488 hexahedra.

6 Summary and Future Work

In this paper, we proposed a new approach to hexahedral mesh generation. This approach combines techniques from several fields, namely decompositions of non-convex regions into convex pieces from computational geometry, minimum cost bidirected flows for surface meshing [20], shellings of cell complexes from topology, graph algorithms on planar graphs, and non-linear optimization methods for the embedding phase. Some main features are worth to be pointed out.

- The key idea is to separate the mesh generation process into a combinatorial part where an abstract complex of hexahedra is constructed from a surface mesh and a geometric embedding part where the exact positions of the mesh nodes are determined.
- Hexahedral mesh quality benefits from our surface mesh generator which yields very regular meshes (in contrast to paving or advancing front based methods).
- Successive elimination of dual cycles corresponds to the meshing of complete layers of hexahedra (sheets) in each step. Cycle selection can use global geometric information to find good elimination orders. Our strategy avoids all kind of combinatorial degeneracies rigorously. Problems with knives or wedges as known from the whisker weaving or plastering algorithms do not occur.
- The very core of our algorithm, and thus the computationally most expensive part, can be performed in parallel for each subdomain. Only the first three steps of our meshing procedure need global access to the complete set of data.

As mentioned, we have currently only implemented a first prototype for our algorithm and our experiences are still limited. First results are encouraging, but more testing has to be done to refine several parameters of our algorithm, for example the cycle selection rules. We would also like to improve our strategies for cycle insertion for those cases where the cycle elimination process gets stuck.

Self-intersections of dual cycles are a major source of difficulties within the mesh generation process. We developed a consistent strategy to remove all self-intersections, relying on the duplication of all dual cycles and a surface mesh modification. Note that this strategy is feasible if the mesh density requirements are not too strict. Problems may occur if the CAD input model contains very short edges. We are convinced that self-intersections should be avoided in the preprocessing part as far as possible. Future work should concentrate on how

to minimize the number of self-intersections of dual cycles in surface meshing algorithms in combination with rules for the subdivision into topological balls. Theoretical work should try to extend our characterization of classes of shellable surfaces.

Acknowledgments

The author wishes to thank G. Krause for providing us with the finite element preprocessor ISAGEN (which we used for our illustrations) and instances from the automobile industry.

References

[1] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis, *Graph drawing: Algorithms for the visualization of graphs*, Prentice Hall, 1999.

[2] M. Bern and D. Eppstein, *Mesh generation and optimal triangulation*, Computing in Euclidean Geometry, 2nd Edition (D.-Z. Du and F. Hwang, eds.), World Scientific, Singapore, 1995, pp. 47–123.

[3] M. Bern and P. Plassmann, *Mesh generation*, chapter 6 of Handbook of Computational Geometry (J. Sack and J. Urrutia, eds.), North-Holland, Elsevier Science, 2000, pp. 291–332.

[4] T. D. Blacker and R. J. Meyers, *Seams and wedges in plastering: A 3D hexahedral mesh generation algorithm*, Engineering with Computers **9** (1993), 83–93.

[5] T. D. Blacker and M. B. Stephenson, *Paving: A new approach to automated quadrilateral mesh generation*, Int. J. Numer. Methods in Eng. **32** (1991), 811–847.

[6] N. A. Calvo and S. R. Idelsohn, *All-hexahedral element meshing by generating the dual mesh*, Computational Mechanics: New Trends and Applications (S. Idelsohn, E. Oñate, and E. Dvorkin, eds.), CIMNE, Barcelona, Spain, 1998.

[7] S. A. Canann, *Plastering: A new approach to automated, 3d hexahedral mesh generation*, Am. Inst. Aeronautics and Astronautics, Reston, VA (1992).

[8] W. A. Cook and W. R. Oakes, *Mapping methods for generating three-dimensional meshes*, Computers in Mechanical Engineering, CIME Research Supplement (1982), 67–72.

[9] G. Danaraj and V. Klee, *A representation of 2-dimensional pseudomanifolds and its use in the design of a linear-time shelling algorithm*, Annals of Discrete Mathematics **2** (1978), 53–63.

[10] D. Eppstein, *Linear complexity hexahedral mesh generation*, Computational Geometry — Theory and Applications **12** (1999), 3–16.

[11] N. T. Folwell and S. A. Mitchell, *Reliable whisker weaving via curve contraction*, Engineering with Computers **15** (1999), 292–302.

[12] L. A. Freitag and P. M. Knupp, *Tetrahedral element shape optimization via the jacobian determinant and condition number*, Proceedings of the 8th International Meshing Roundtable, South Lake Tahoe, CA, Sandia National Laboratories, Albuquerque, USA, 1999, pp. 247–258.

[13] J. E. Hopcroft and R. E. Tarjan, *Dividing a graph into triconnected components*, SIAM J. of Computing **2** (1973), 135–158.

[14] P. M. Knupp, *Matrix norms & the condition number: A general framework to improve mesh quality via node-movement*, Proceedings of the 8th International Meshing Roundtable, South Lake Tahoe, CA, Sandia National Laboratories, Albuquerque, USA, 1999, pp. 13–22.

[15] P. M. Knupp, *Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities, part II — a framework for volume mesh optimization*, Int. J. Numer. Methods in Eng. **48** (2000), 1165–1185.

[16] T. S. Li, R. M. McKeag, and C.G. Armstrong, *Hexahedral meshing using midpoint subdivision and integer programming*, Computer Methods in Applied Mechanics and Engineering **124** (1995), 171–193.

[17] W. Min, *Generating hexahedron-dominant mesh based on shrinking-mapping method*, Proceedings of the 6th International Meshing Roundtable, Park City, Utah, Sandia National Laboratories, Albuquerque, USA, 1997, pp. 171–182.

[18] S. A. Mitchell, *A characterization of the quadrilateral meshes of a surface which admit a compatible hexahedral mesh of the enclosed volume*, Proceedings of the 13th Annual Symposium on Theoretical Aspects of Computer Science (STACS'96), Lecture Notes in Computer Science 1046, Springer, 1996, pp. 465–476.

[19] R. H. Möhring and M. Müller-Hannemann, *Complexity and modeling aspects of mesh refinement into quadrilaterals*, Algorithmica **26** (2000), 148–171.

[20] R. H. Möhring, M. Müller-Hannemann, and K. Weihe, *Mesh refinement via bidirected flows: Modeling, complexity, and computational results*, Journal of the ACM **44** (1997), 395–426.

[21] M. Müller-Hannemann, *Hexahedral mesh generation by successive dual cycle elimination*, Engineering with Computers **15** (1999), 269–279.

[22] M. Müller-Hannemann, *High quality quadrilateral surface meshing without template restrictions: A new approach based on network flow techniques*, International Journal of Computational Geometry & Applications **10** (2000), 285–307.

[23] P. Murdoch, *The spatial twist continuum: A dual representation of the all hexahedral finite element mesh*, Ph.D. thesis, Mechanical Engineering, Brigham Young University, Utah, 1995.

[24] P. Murdoch, S. E. Benzley, T. D. Blacker, and S. A. Mitchell, *The spatial twist continuum: a connectivity based method for representing and constructing all-hexahedral finite element meshes*, Finite Elements in Analysis and Design **28** (1997), 137–149.

[25] T. Nishizeki and N. Chiba, *Planar Graphs: Theory and Algorithms*, Annals of Discrete Mathematics, vol. 32, North–Holland, 1988.

[26] S. Owen, *Meshing research corner*, http://www.andrew.cmu.edu/user/sowen/mesh.html.

[27] M. A. Price and C. G. Armstrong, *Hexahedral mesh generation by medial surface subdivision: Part II, solids with flat and concave edges*, Int. J. Numer. Methods in Eng. **40** (1997), 111–136.

[28] M. A. Price, C. G. Armstrong, and M. A. Sabin, *Hexahedral mesh generation by medial surface subdivision: Part I, solids with convex edges*, Int. J. Numer. Methods in Eng. **38** (1995), 3335–3359.

[29] R. Schneiders, *Information on finite element mesh generation*, available online at http://www-users.informatik.rwth-aachen.de/~roberts/meshgeneration.html.

[30] R. Schneiders, *Open problem*, available online at http://www-users.informatik.rwth-aachen.de/~roberts/open.html, 1995.

[31] R. Schneiders, *A grid-based algorithm for the generation of hexahedral element meshes*, Engineering with Computers **12** (1996), 168–177.

[32] R. Schneiders, R. Schindler, and F. Weiler, *Octree-based generation of hexahedral element meshes*, Proceedings of the 5th International Meshing Roundtable, Sandia National Laboratories, Albuquerque, USA, 1996, pp. 205–215.

[33] G. C. Shephard, *Combinatorial properties of associated zonotopes*, Canadian Journal of Mathematics **26** (1974), 302–321.

[34] T. J. Tautges, T. Blacker, and S. A. Mitchell, *The whisker weaving algorithm: A connectivity-based method for constructing all-hexahedral finite element meshes*, Int. J. Numer. Methods in Eng. **39** (1996), 3327–3349.

[35] T. J. Tautges and S. A. Mitchell, *Whisker weaving: Invalid connectivity resolution and primal construction algorithm*, Proceedings of the 4th International Meshing Roundtable, Sandia National Laboratories, Albuquerque, USA, 1995, pp. 115–127.

[36] W. Thurston, *Hexahedral decomposition of polyhedra*, Posting to sci.math., 25 Oct., 1993, available online at http://www.ics.uci.edu/~eppstein/gina/Thurston-hexahedra.html.

[37] P. M. Tuchinsky and B. W. Clark, *The "HexTet" hex-dominant automesher: An interim progress report*, Proceedings of the 6th International Meshing Roundtable, Park City, Utah, Sandia National Laboratories, Albuquerque, USA, 1997, pp. 183–193.

[38] W. T. Tutte, *Convex representations of graphs*, Proceedings London Mathematical Society **10** (1960), 304–320.

[39] W. T. Tutte, *How to draw a graph*, Proceedings London Mathematical Society **13** (1963), 743–768.

[40] D. R. White, L. Mingwu, S. E. Benzley, and G. D. Sjaardema, *Automated hexahedral mesh generation by virtual decomposition*, Proceedings of the 4th International Meshing Roundtable, Sandia National Laboratories, Albuquerque, USA, 1995, pp. 165–176.

[41] G. M. Ziegler, *Lectures on polytopes*, revised ed., Graduate Texts in Mathematics, vol. 152, Springer-Verlag, New York, 1998.

[42] G. M. Ziegler, *Shelling polyhedral 3-balls and 4-polytopes*, Discrete & Computational Geometry **19** (1998), 159–174.

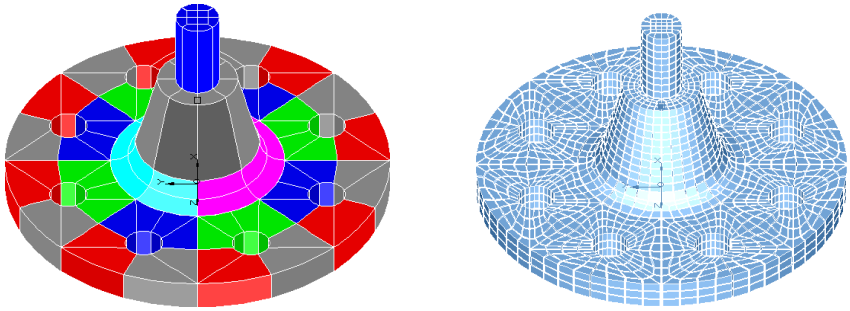

Figure 20: The model of a shaft with a flange which has been decomposed into simpler subdomains where different colors correspond to the subdomains (left), and the hexahedral mesh constructed by our algorithm (right).

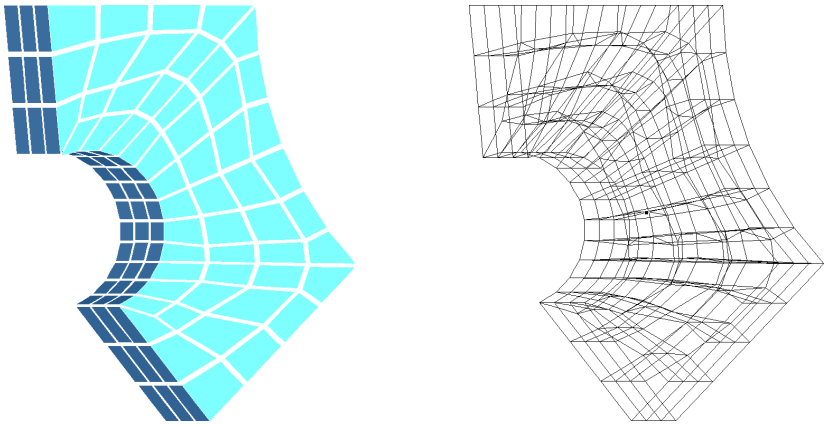

Figure 21: Hexahedral mesh for a part of the model in Fig. 20.

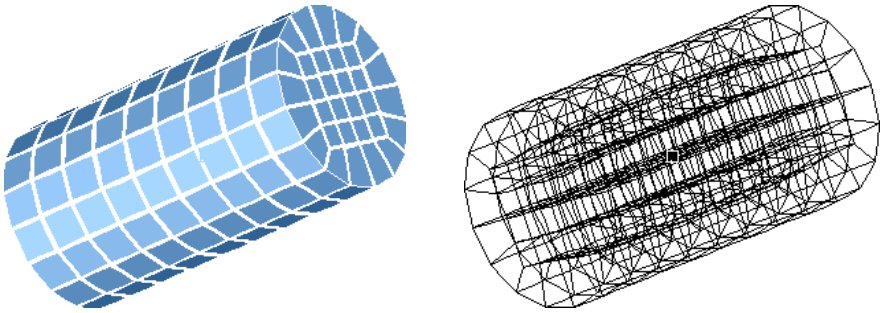

Figure 22: Hexahedral mesh for a cylinder (part of the model in Fig. 20). The cycle elimination scheme first selects parallel cycles corresponding to eight layers, and uses than a double elimination to remove the 16 hexahedra forming a torus on the last layer.

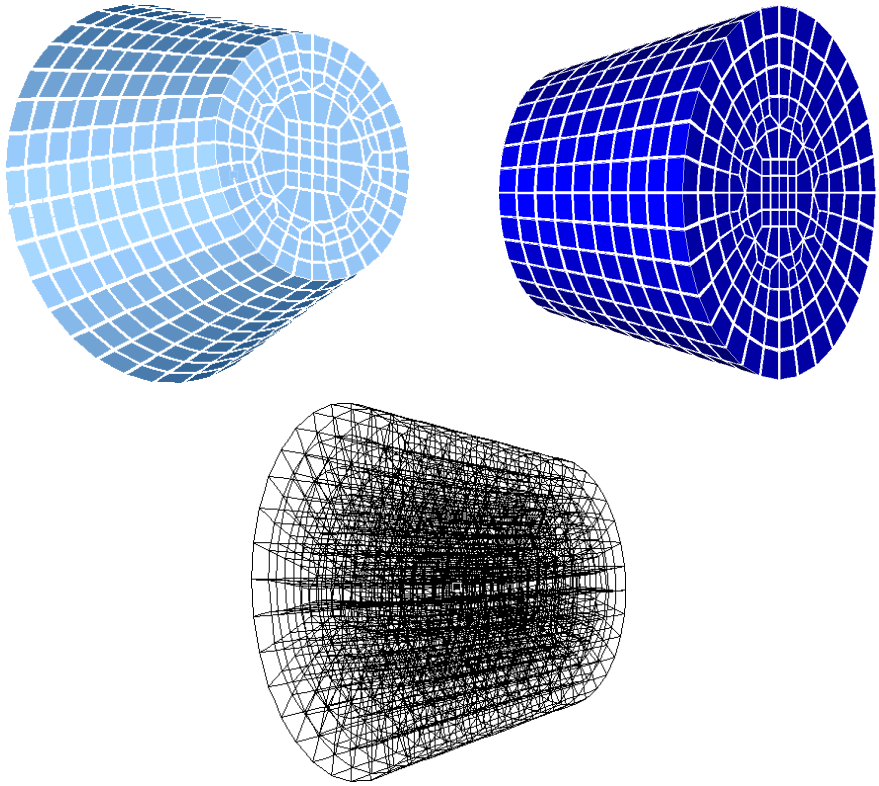

Figure 23: Hexahedral mesh for a truncated circular cone (also part of the model in Fig. 20). The meshing is non-trivial because the base cycles of the cone have slightly different surface meshes.

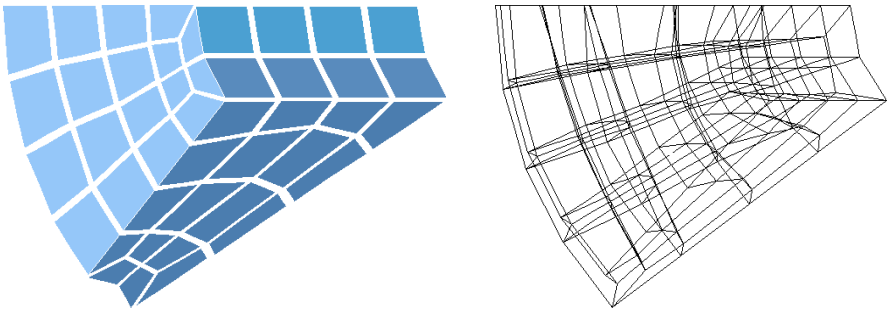

Figure 24: Hexahedral mesh for another part of the model in Fig. 20.