SEMI-STRUCTURED MESH GENERATION BASED ON MEDIAL AXIS

Peter Sampl

AVL List GmbH, Graz, Austria. peter.sampl@avl.com

ABSTRACT

A three-dimensional, medial-axis-based mesh generator is described. First the medial axis of the geometry to be meshed is constructed, then quad-dominant meshes on the medial surface are generated and extruded to the boundary at both sides of the medial surface. The resulting single cell layer is subdivided in direction normal to the boundary, yielding columns of hexahedral and three-sided prismatic cells. The resulting volume mesh is orthogonal at the boundary and "semi-structured" between boundary and medial surface. Mixed cell types (tets, pyramids, degenerate hexahedra) may result along the medial surface. The medial axis is computed indirectly from the Voronoi diagram of a dense set of sample-points on the boundary. An advancing front algorithm (paving) is used for meshing the subfaces of the medial surface. Development of the mesh generator has not been fully completed yet. First medium-complexity geometries have been meshed, however, showing moderate meshing times.

Keywords: medial axis, hybrid meshing, semi-structured, hex-dominant

1. INTRODUCTION

The two most popular strategies for automatic hexahedral or hex-dominant mesh generation are (1) meshing first the boundary of the geometry and then advancing the surface mesh towards the interior, or, (2), meshing the volume based on regular grids and projecting or intersecting this grid at the boundary. The first approach faces the problem of properly closing the advancing fronts where they meet in the interior. The second is problematic with respect to mesh quality and proper feature resolution at the boundary. Sweeping or extrusion techniques circumvent these problems but are restricted to the subset of sweepable geometries. Medial axis based meshing provides a way to generate sweepable subvolumes for all kinds of geometries by inserting an internal medial surface.

For mesh generation, input is usually a *boundary representation (b-rep)* of the connected geometric region to be analyzed. The medial axis can be computed from the b-rep. The *medial axis* of a geometric region, also called *medial object* [1] or *shape skeleton* [2, 3], is a well-known object in computational geometry. A first concise definition was given by Blum [4]. The definition of the medial axis is simple: it is the set of all points having more than one nearest point on the boundary of the region. This holds for *n*-dimensional space. The distance from a medial axis point to its nearest boundary points is called the *local feature size* [5]. The local feature size is a measure for the local "diameter" of the geometry. A hypersphere centered at a medial axis point with radius equal to the local feature size is a maximal inscribed hypersphere for the region. Where the boundary is not smooth (i.e. not C^1), e.g. at boundary corners in two-dimensional and boundary edges in three-dimensional space, the medial axis touches the boundary.

The medial axis is closely related to the wider concept of Voronoi diagrams [6, 7]. The Voronoi diagram is a space subdivision that assigns *Voronoi regions* to all geometric objects under consideration. The Voronoi region of a geometric object is the (connected) set of points in space nearer to the object than to any other. The medial axis is, in general, a subset of the boundaries of the Voronoi regions, if the geometric objects are the b-rep elements.

In two dimensions ($\mathbf{R}^2$), the medial axis is a set of curves (*medial edges*) meeting at branching points (*medial vertices*). Figure 1 shows the boundary of a two-dimensional region (a section through the passenger compartment of a car) and the boundary together with the medial axis. The medial axis touches the boundary at convex corners as mentioned above. Figure 2 additionally shows the maximal inscribed circles at the medial vertices. Medial vertices have three or more nearest boundary points.

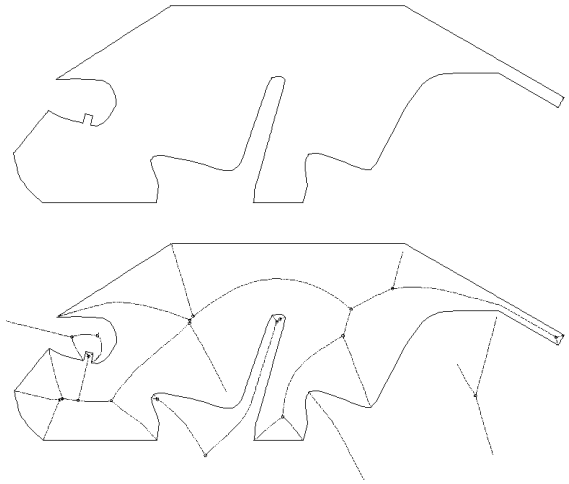

Figure 1: Domain boundary (above), and together with the medial axis (below) in R^2

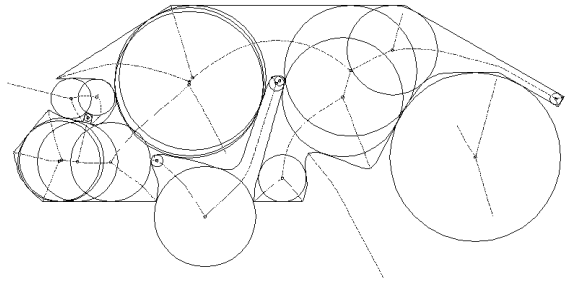

Figure 2: Domain boundary, medial axis and maximal inscribed circles at medial vertices

In three dimensions ($\mathbf{R}^3$), the medial axis is a set of *medial faces* connected along medial edge curves, which again meet at medial vertices. Figure 3 shows the boundary and the interior part of the medial axis of a solid composed of two connected blocks with a square hole. The interior part of the medial axis in $\mathbf{R}^3$ is a connected surface, if the boundary is a non-intersecting two-manifold.

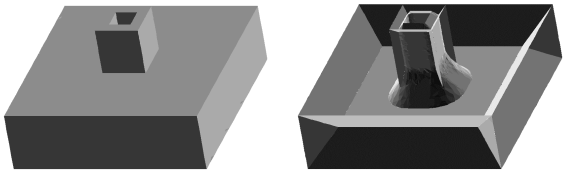

Figure 3: Boundary of a simple region in R^3 (left) and its medial surface (right)

In non-simple cases medial faces may degenerate to medial edges and medial edges to medial vertices. The medial surface of a sphere, e.g., collapses to the sphere center point. The medial surface of a long cylinder collapses, except for the ends of the cylinder, to its axis. Degeneracies generally occur at convexely curved parts of the geometry, if no other surface part is nearer than the curvature-radius.

The medial axis together with the local feature size distribution gives a complete description of the original geometry. This kind of description can be called the *medial-axis-representation (ma-rep)*, in literature it is also referred to as the *medial axis transformation* (*MAT*, see [8]).

Compared to a b-rep, the ma-rep explicitly contains additional information on the region's geometry and topology:

- information on the local feature size,
- information on "opposite" boundary points by relating the nearest boundary points of a medial axis point,
- information on the region's topology by the medial axis topology.

For these reasons, the ma-rep is valuable for a number of applications including feature recognition, dimensional reduction, shape simplification, detail suppression, mesh generation, shape synthesis and various shape interrogation operations (see e.g. [1] and [9]). The medial axis can be used to partition the geometry into subregions that are easier to mesh. This can be done in different ways. Li, Price and Armstrong [10,11,12] describe a subdivision yielding one subregion for each medial vertex, medial edge and medial face. The subregions are subsequently meshed by mid-point-subdivision. Sheffer et al. [13] describe a subdivision approach yielding sweepable subregions based on the embedded Voronoi graph. This graph is very closely related to the medial axis. All-hexahedral meshes can be generated with these methods. However, only meshes of moderate complexity have been presented in [10] and [13]. We use a different subdivision strategy to obtain some kind of sweepable subregions more directly based on the geometry of the medial axis. The basic meshing algorithm in $\mathbf{R}^3$ reads as follows.

1. Compute medial axis
2. Mesh medial edges
3. Mesh medial faces using advancing front schemes starting from edge meshes
4. Project and extrude face meshes onto boundary of geometry at both sides of medial axis
5. Subdivide extruded cell layer normal to boundary
6. Recombine degenerate cells along medial axis if possible

Algorithm 1: Mesh generation based on medial axis

Figure 4 shows the basic idea in $\mathbf{R}^2$. First the medial axis is meshed, resulting in one-dimensional grid lines. In $\mathbf{R}^2$ step 3 in Algorithm 1 is obsolete. All grid points are then projected onto the boundaries on both sides of the medial axis and the medial axis cells (which are line segments in $\mathbf{R}^2$) are extruded. The projection is always to the nearest boundary points. Hence, the projection rays are orthogonal to the boundary. (At concave boundary vertices or edges in $\mathbf{R}^3$ they are orthogonal to an infinitesimally shrinked version of the geometry).

A set of boundary distance values $\{d_1, d_2, ...\}$ is specified globally and the extruded cells are intersected at these

distances. This gives a mesh which is orthogonal to the boundary and consists of structured cell layers stretching from the boundary to the medial axis. Therefore we call the mesh *semi-structured*. The mesh is also locally symmetric to the medial axis. Degenerate cells may arise at the medial axis as shown in the figure, depending on the number of mesh points generated on the corresponding projection rays. In $\mathbf{R}^2$, the degenerate cells are triangles which can always be recombined with their symmetric counterpart on the other side of the medial axis to give quadrilateral cells again.

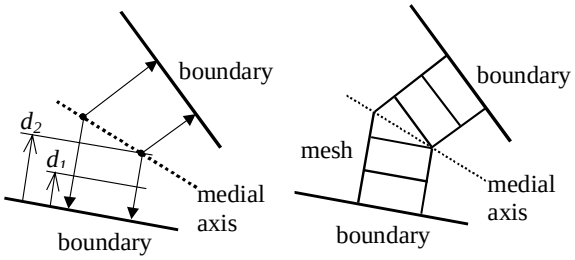

Figure 4: Medial axis based mesh generation

The set of boundary distance values that governs grid spacing need not be global. It can also be made a continuous function over the medial axis. The function should vary smoothly, however, to maintain mesh orthogonality (see e.g. the example mesh shown in Figure 18).

In order to get a hex-dominant mesh in $\mathbf{R}^3$, the medial faces have to be meshed quad-dominantly in step 3. Extruding a triangular face-cell results in a column of three sided prisms, a quad will give a hex-column. Degenerate prisms or hexahedra may arise at the medial axis and at concave boundary edges and vertices. They are recombined to non-degenerate elements if possible in step 6. In general, a hybrid mesh will result. The different steps in the overall algorithm are described in detail below. The algorithm has been implemented in C++.

2. MEDIAL AXIS CONSTRUCTION

As the medial axis is essentially a subset of the Voronoi diagram of the boundary elements of the considered region, usually the Voronoi diagram is computed first and the medial axis is extracted therefrom afterwards. A survey of published algorithms for the construction of the medial axis (and Voronoi diagrams) is given by Sherbrooke et al. [14] and Turkiyyah et al. [2]. In general, *direct* and *indirect* methods can be distinguished. Direct algorithms directly compute the vertices and edges of the Voronoi diagram of the b-rep elements. Indirect algorithms approximate the continuous b-rep elements by a set of *sample points* and compute the Voronoi diagram of the point set, which is much easier than computing the Voronoi diagram of the original continuous boundary elements. Turkiyyah et al. show that the Voronoi vertices converge to medial axis points as the sampling density increases. The approximation of the medial axis is then extracted from the Voronoi diagram. A large number of sample points may have to be generated to obtain the medial axis with reasonable accuracy, however.

Another category of approximating algorithms is based on discretization of the embedding space using quadtrees, octrees, pixels or voxels. Thinning procedures are then used for the generation of the medial axis. This approach is popular in digital image processing, since in this case the region boundary is often available only in pixel format [15]. The space discretization in $\mathbf{R}^3$ for non-trivial regions, however, requires a prohibitive amount of computing time and memory if the correct medial axis topology has to be generated.

2.1. Direct methods

For piecewise linear b-reps in $\mathbf{R}^2$, i.e. polygons, Lee [16] gives an algorithm for computing the Voronoi diagram and medial axis with asymptotic time complexity of $O(n\log n)$, if n is the number of polygon-edges. An algorithm for computing Voronoi diagrams for n curved objects in $\mathbf{R}^2$ in $O(n\log n)$ time is outlined in [17]. Sherbrooke et al. [14] describe a direct algorithm for medial axis construction for arbitrary, non-convex polyhedra with n faces in $\mathbf{R}^3$, but time complexity seems to be of order $O(n^2)$. No direct $\mathbf{R}^3$-$O(n\log n)$-algorithm was found by the author in literature, so indirect construction was selected for the presented meshing algorithm.

2.2. Indirect method

Methods for computing the medial axis based on the Voronoi diagram of a sample point set on the region's boundary were described by Sheehy et al. [8] and Turkiyyah et al. [2]. The described algorithms start from different inputs. Sheehy et al. start from a CAD model and parametric surfaces and require also topological boundary information. Turkiyyah et al. start from an implicit solid modeling (ISM) description of the region (ISM describes the geometric region Ω by specifying a function f: $\mathbf{R}^3 \rightarrow \mathbf{R}$ so that $\Omega = \{x \in \mathbf{R}^3 \mid f(x) < 1\}$). Our target was to accept low-level triangulated b-reps as input (e.g. data in STL-format [18]), so a different implementation was developed. In particular, no topological information on the boundary aside from the connectivity of the b-rep triangles is required.

As mentioned, the sample points have to be placed dense enough on the boundary to obtain a topologically correct medial axis. Figure 5 shows the boundary of the two-dimensional region shown in Figure 1 together with the Voronoi diagram of a dense sample of points on the boundary. The points themselves are not marked in the figure. They are located between the intersections of two neighboring Voronoi edges with the boundary. The extracted medial axis is shown in Figure 1.

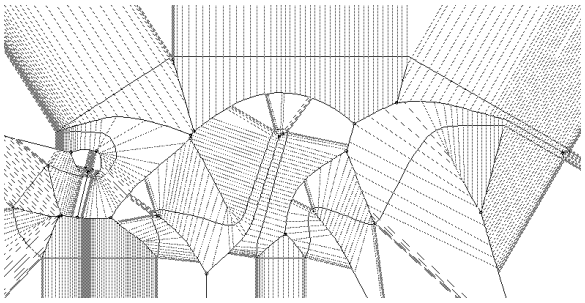

Figure 5: Voronoi diagram of a dense sample of boundary points in R²

The Voronoi diagram of the point set is the dual of the Delaunay triangulation. The major steps in the indirect medial axis construction algorithm are hence Delaunay triangulation and sample point generation. Sample points are generated by splitting edges of the b-rep triangulation according to Rivara's *longest-edge-bisection* algorithm (*LEB*, see [19]). From the Voronoi diagram an approximation of the medial axis is extracted by selecting the medial edges and faces inside the region. Since the obtained medial axis is an approximation, clean-up and geometric correction steps are applied finally. The steps in the construction algorithm are described in the following chapters.

```
start from triangulation of boundary surface
set sample point set = set of triangle vertex points
do{
        insert new sample points into Delaunay
                triangulation
        if (sample is dense enough) break
        generate additional sample points by LEB where
                density too small
}
extract medial surface
clean-up medial axis
geometrically correct medial axis points
```

Algorithm 2: Medial axis construction

2.3. Delaunay triangulation in R^3

The Delaunay triangulation of a point set in $\mathbf{R}^3$ can be considered a settled problem and many different algorithms have been published (see e.g. [20, 21, 22, 23, 24]). The triangulation algorithm we implemented for the construction of the medial axis is a slightly modified version of the randomized incremental flip algorithm by Mücke [23]. It starts from a non-Delaunay triangulation and transforms it to a Delaunay triangulation by a sequence of local transformations, called *flips*. Integer arithmetic is used to avoid round-off errors.

The time complexity of the overall algorithm for inserting n points is of $O(n^2)$, since up to n flips may have to be performed for the insertion of one point. By randomizing the insertion order of the points the expected running time of $O(n \log n)$ can almost be guaranteed and was experimentally verified with our implementation (Figure 6).

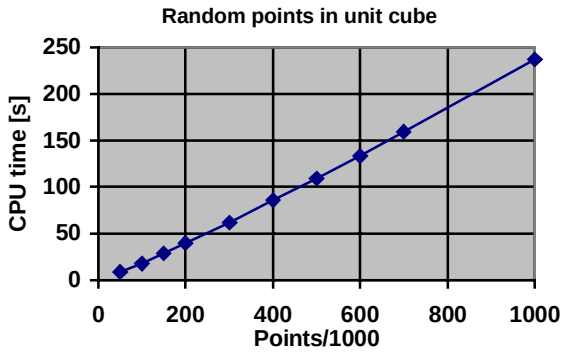

Figure 6: Performance of the R^3 Delaunay triangulation algorithm (Pentium III 600MHz)

In the tests up to 10^6 random points within a unit cube were inserted (yielding $6.75 \cdot 10^6$ tets). Robustness in highly degenerate situations was checked by inserting large numbers of points on a sphere and on a uniform grid.

2.4. Dense-sample-criterion

In Algorithm 2 additional sample points have to be generated while the topology of the medial axis approximation is not correct. Correctness is not easy to define. We relate it, heuristically, to the existence of a homeomorphism between the boundary surface of the original geometry and the *surface Delaunay triangles* (see [5]). The surface Delaunay triangles are the triangle faces in the Delaunay triangulation whose dual Voronoi edges intersect the original surface (the dual edge is the line between the circumcenters of the two tets adjacent to the triangle face). Homeomorphism means, roughly, that the surface Delaunay triangles form a surface topologically equivalent to the original surface. This requirement is more relaxed than that of a constrained Delaunay triangulation (see e.g. [25]), as required for the medial axis construction in [8].

Amenta et al. [5] proof that such a homeomorphism exists if the sample point spacing is smaller than one tenth of the local feature size. We found that for our purpose it is sufficient to have a spacing smaller than half the local feature size. For the sample-density-check in Algorithm 2 we associate the local feature size to the circumradii of the Delaunay-tets and the sample point spacing to the longest edge lengths of the b-rep triangles. If a b-rep triangle's longest edge length is larger than half the circumradius of any tet sharing a point with the triangle it is marked for refinement.

This criterion can never be met at (sharp) convex edges of the original boundary surface, since the medial axis touches the boundary there and the local feature size converges to zero. To stop refinement there, an alternative criterion is defined, based on an auxiliary decomposition of the boundary surface. At the beginning of the construction algorithm the surface is divided into *patches* by greedily grouping adjacent b-rep triangles with their normal vectors all within a prescribed angular tolerance. This is a purely geometric decomposition and does not require any additional topological information on CAD surfaces. Figure 7 shows the boundaries of the

obtained patches for a slotted, block-like object and a tolerance of 18 degrees. Exploiting this decomposition, refinement is stopped also at Delaunay-tets with all their four vertices located on patches that share at least one point. The tet vertices then are necessarily located around a single boundary surface vertex or a single boundary edge. In both cases, the medial axis topology is already represented correctly in the computed approximation without requiring further sample points.

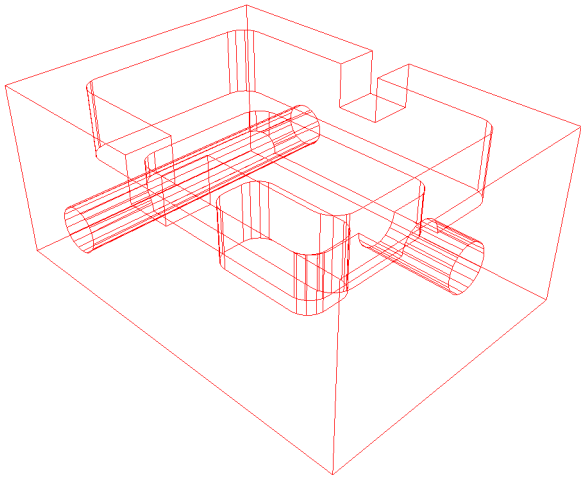

Figure 7: Surface patch-decomposition for a slotted block object

2.5. Sample point generation

Additional sample points are always generated by splitting edges in the b-rep triangulation. For efficiency, the edge splitting has to be designed so that with a minimal amount of additional sample points the length scales of as many triangles as possible fall below the allowed maximum. Obviously, filling a surface with equilateral triangles minimizes global maximal point spacing. Rivara [19] shows that by *longest edge bisection* (LEB) quasi-equilateral triangles tend to be produced. She proves that the smallest angle α of any triangle obtained by LEB satisfies $\alpha \geq \alpha_0/2$, if α_0 is the smallest angle in the initial triangulation. Additionally, if two adjacent b-rep triangles are (approximately) coplanar and non-Delaunay after projection into $\mathbf{R}^2$, the shared edge is flipped, if possible. This again improves the quality of the b-rep triangles after edge splitting. Figure 8 and Figure 9 show the initial b-rep triangulation of the slotted block object and the triangulation after termination of the refinement procedure, respectively.

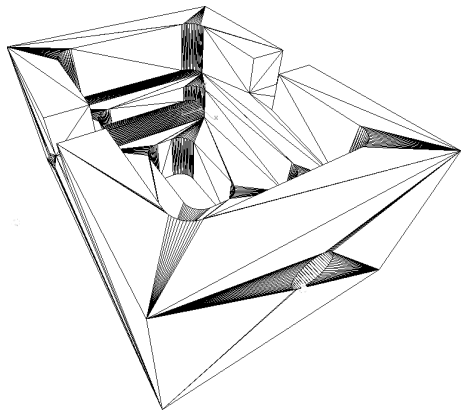

Figure 8: Initial b-rep triangulation of slotted block (STL-data, 658 vertices)

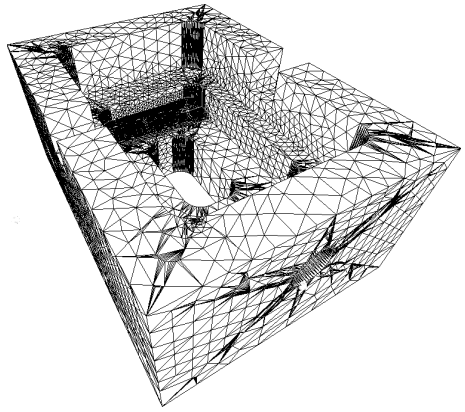

Figure 9: b-rep triangulation after termination of the sample point generation process (13356 vertices)

2.6. Medial axis extraction

After sample point generation and Delaunay triangulation, the medial axis is extracted from the dual Voronoi diagram by eliminating Voronoi edges and faces intersecting the geometry. Since we are interested in the internal part of the medial axis only, we also discard Voronoi entities outside the geometry. For that purpose, all Voronoi vertices (corresponding to the circumcenters of the Delaunay-tets) inside the geometry are determined. We always orientate the boundary surface so that normals point outside. A Voronoi vertex is then considered inside if the boundary normals at all its nearest boundary points point away from the vertex. The vertices of the Delaunay-tets dual to the Voronoi vertices converge to the closest boundary points with increasing sample density. The boundary points closest to a Voronoi vertex thus can be found by a local search on the boundary surface starting from the vertices of the dual tet.

Internal Voronoi vertices correspond to medial axis points. All Voronoi edges between two internal Voronoi vertices are part of the medial axis, as well all Voronoi faces surrounded by internal Voronoi edges. A medial axis line segment is topologically a medial edge segment if three medial axis face

segments are adjacent. If only one medial axis face segment is adjacent, it is a medial axis boundary line segment. A medial axis point is a medial vertex if there are four incident medial edge segments. In degenerate situations medial axis points may collapse and medial axis line segments may be collinear.

2.7. Medial axis clean-up

The Delaunay triangulation can contain very flat tets at curved boundary faces of the geometry. The circumcenters of these tets are far off the medial axis. Turkiyyah et al. [3] call these tets *spurious*. They result from the approximation of the boundary surface by a set of points and can be avoided only by prohibitive high sampling densities. Fortunately, it is possible to remove these tets without doing harm to the medial axis approximation. The best criterion to filter out erroneous medial axis points proved to be the *has-only-one-closest-boundary-point-test*. Since, by definition, a medial axis point has more than one closest point on the boundary, medial axis points with (nearly) collapsing closest boundary points are discarded.

2.8. Geometric correction

The circumcenters of the Delaunay-tets converge to medial axis points in the refinement process. As described, the process is terminated when the medial axis topology is correctly determined by the approximation. The positions of the computed medial axis points are, however, not yet exactly on the medial surface (see Figure 10; Figure 3 shows the corrected surface). The position of the computed points is therefore corrected based on the obtained medial axis topology. For each computed medial axis point $x \in \mathbf{R}^3$ the closest boundary points $p_k \in \mathbf{R}^3$, $k=1,...,m$ can be determined by a local search as above (m is always 4 since we start from the four vertices of the dual tet). The position of the medial axis point x is then iteratively corrected so that it becomes as equidistant as possible to all points p_k. Accordingly, a scalar function f to be minimized is formulated, with the normalized vectors $n_k \in \mathbf{R}^3$ pointing from p_k to x:

$$f(x) = \sum_k^m \left(d_k(x) - \bar{d}(x)\right)^2 = \min \qquad (1)$$
$$d_k(x) = (x - p_k)n_k$$
$$\bar{d}(x) = \frac{1}{m}\sum_k^m d_k(x)$$

where d_k denotes the distance of x to the boundary point p_k and $\bar{d}$ the averaged boundary distance. Differentiation with respect to the components of x gives a linear equation system. After solving the system and updating x all boundary points p_k are updated again by a local search on the boundary surface for the points closest to x, starting from the previous p_k. This scheme proved to be efficient and converged robustly to the correct position on the medial axis for all test cases.

Along edges of the boundary surface the boundary of the medial axis theoretically touches the boundary. The position of the corresponding medial axis boundary points has to be corrected by setting $\bar{d}(x) \equiv 0$ in (1). These points can be detected exploiting the patch-decomposition. All the vertices of the tet dual to such a medial axis point are located on patches that are adjacent to each other and whose surface normals deviate by more than a specified "sharp-edge"-threshold angle.

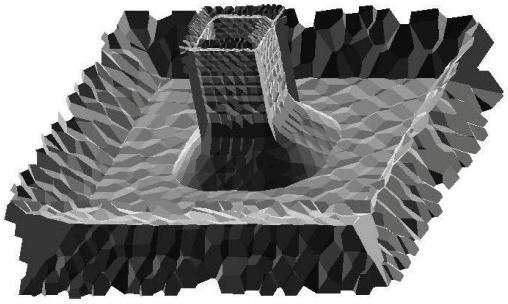

Figure 10: Medial surface without geometric correction

2.9. Medial axis example

Figure 11 shows the medial surface computed for the slotted block object. The different subfaces of the medial surface are shaded differently. The required CPU time was to one third consumed by the Delaunay triangulation. The rest was needed for checking the medial axis correctness, determining the interior Voronoi elements and correcting the medial surface geometry.

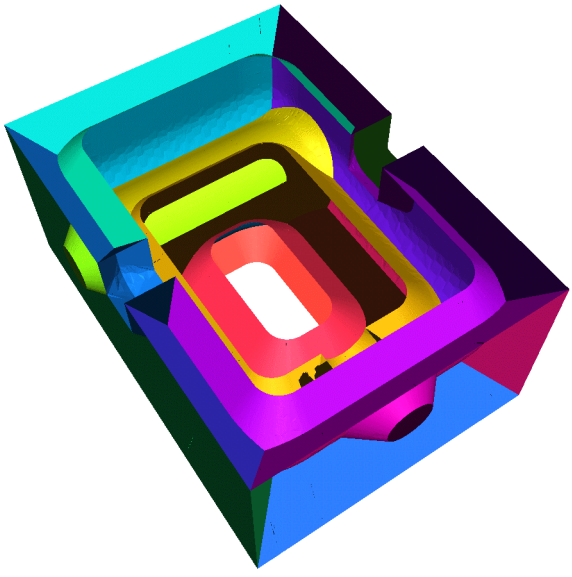

Figure 11: Medial surface of slotted block object (30s on 600MHz-Pentium III)

3. MESHING BASED ON MEDIAL AXIS

As mentioned, the basic idea of the described meshing approach is to first mesh the medial surface with quadrilateral or triangular elements and then to project and extrude this mesh onto the boundary on both sides of the medial surface. The extruded mesh is then further subdivided in direction normal to the boundary at prescribed boundary distances, leading to columns of hexahedral or prismatic cells (in non-degenerate cases). This way the problem of three-dimensional mesh generation is reduced to two-dimensional mesh generation on the medial surface, with subsequent simple projection and subdivision steps.

In the current version of this mesh generator, one global function is used to determine grid spacing. A geometric series $\{ d_k = d_1 (q^k-1) / (q-1) \}$ of boundary distances is selected for subdividing the mesh normal to the boundary of the geometry. (Of course any other ordered set of distance values could be used in general.) The series is specified by its first value, d_1, and the quotient q. The first value determines the thickness of the first cell layer adjacent to the boundary, the quotient, which is typically selected from the interval [1, 1.5], determines the growth of thickness of cell layers towards the interior of the geometry. If necessary, d_1 and q can be made continuous functions over the medial axis to allow for local variations of grid spacing. This subdivision normal to the boundary enforces mesh points on medial edges and mesh lines on medial faces at these prescribed boundary distances as well. Figure 12 shows the medial edges and lines of constant boundary distance on the medial surface for a given set $\{d_k\}$ for the slotted block object. To optimize mesh orthogonality, mesh lines normal to the above lines on the medial faces will also be inserted. However, large parts of the medial edges and faces remain without such constraints and have to be meshed with general surface meshing algorithms.

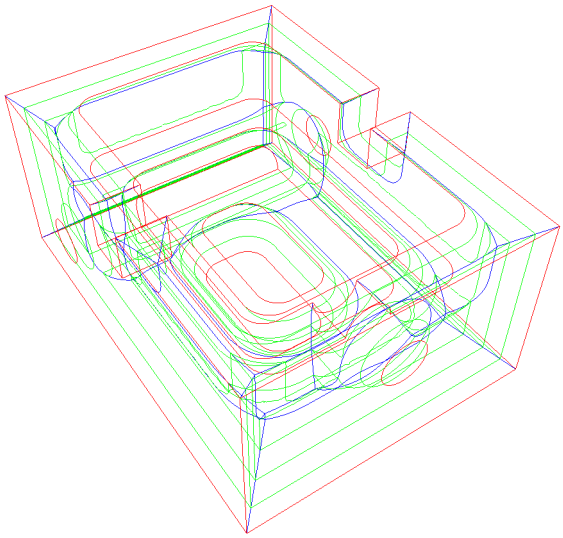

Figure 12: Medial edges and lines of constant boundary distance on medial surface for slotted block object

Grid spacing tangential to the boundary can be derived also from the geometric series by requiring the cells to be isotropic at the medial axis. According to that, the tangential cell size at a medial axis point is set to $(d_{k+1} - d_k)$, if d_k is the value in the series closest to the boundary distance d of the medial axis point.

3.1. Meshing the medial edges

The medial axis construction module provides access to the medial edges in polyline representation. For each polyline vertex the boundary distance (which is also the local feature size) is known. As mentioned, mesh points have to be inserted on the medial edges at the specified boundary distances to obtain cell layers parallel to the boundary. The medial vertices have also to be inserted as mesh points in order to get coherent meshes. Intermediate mesh points are generated on the medial edges so that the distances between the mesh points respect the derived tangential grid spacing. The resulting edge mesh for a given series of boundary distance values for the slotted block example is shown in Figure 13.

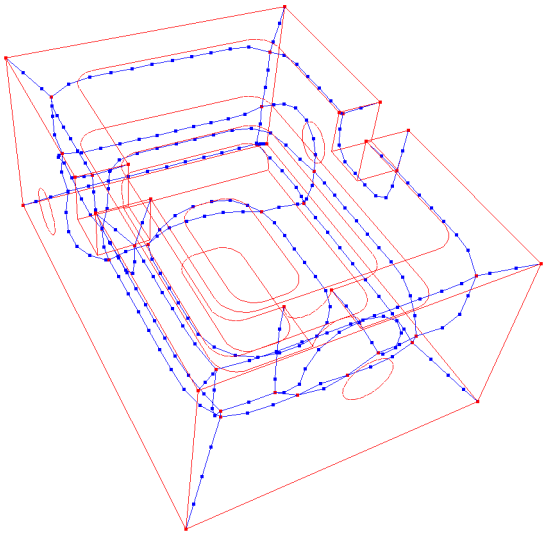

Figure 13: Mesh points generated on the medial edges of the slotted block object

3.2. Meshing the medial faces

The medial faces are meshed starting from the created edge meshes. These define the initial fronts for an advancing front algorithm. Advancing front algorithms have been developed for plane and curved surfaces, generating triangular or quadrilateral meshes [see e.g. 26, 27, 28, 29]. We use a “paving” scheme generating quad-dominant meshes which is closely based on [26, 28]. The basic algorithm is described below (Algorithm 3).

```
initialize list of front segments from edge meshes
while (there are front segments in list) {
        select front segment from list according
                to priority criteria
        if (a new cell can be generated adjacent to
                segment) {
                add cell to list of cells
                connect it with already existing
                        adjacent cells
                update front
        } else {
                reduce priority of segment
        }
}
perform topological clean-up
smooth face mesh
```

Algorithm 3: Basic advancing front surface meshing algorithm

Due to the approximative character of the medial axis construction and the involved correction steps the obtained medial surface triangulation can be of poor quality and might even contain "folded" triangles with normals inconsistent with neighboring triangles. A better geometric representation of the surface is needed for meshing and can be obtained based on the closest boundary points. Normals vectors and projections on the medial surface can then be computed directly from the geometry of the boundary surface (see Figure 14).

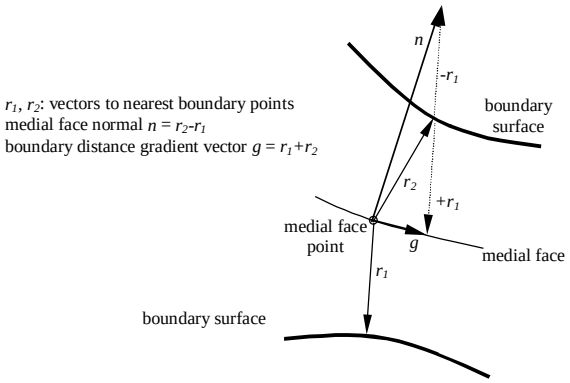

Figure 14: Medial surface normal from closest boundary points

In Algorithm 3 the priority is defined so that segments nearer to the initial front and segments including smaller angles with neighbor segments are processed first.

The fundamental step in the algorithm is the creation of a cell at a selected front segment. It is first tried to generate a cell with edges in the direction of constant boundary distance (see Figure 12) and normal to that. This is tried only if the local gradient of the boundary distance function on the medial surface exceeds a prescribed threshold. Otherwise the cell is generated as in standard paving algorithms [26, 28]. An octree is used to keep track of the position in $\mathbf{R}^3$ of the active front segments to guarantee efficiency of the necessary intersection checks.

Special strategies are used at the final state of the paving process to close (topologically) short loops of front segments. Loops of three or four segments are closed by inserting a triangular or, respectively, quadrilateral cell. For loops of five or six segments all valid closures with quadrilaterals and, in the case of five segment, one triangle, are examined. For six segments also closures with three quadrilaterals, involving the insertion of one additional interior point, are considered. The closure configuration giving the best cell qualities is selected.

After the mesh is generated, topological cleanup operations as shown in Figure 15 are applied where appropriate.

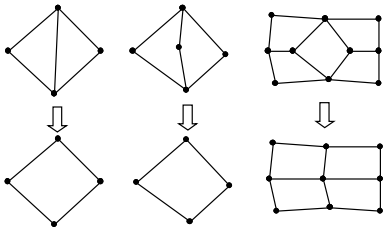

Figure 15: Topological cleanup operations

The final step is smoothing the surface mesh. Simple Laplacian smoothing is applied by shifting a vertex to the averaged position of the centers of the adjacent cells. Only interior vertices are moved.

The algorithm involves only computations of O(n log$n$) time complexity for n generated cells and hence can be regarded as sufficiently efficient. For an example of a meshed medial surface see Figure 20.

3.3. Projection, subdivision and recombination

After all the medial sub-faces are meshed, the remaining operation is the projection and extrusion of these meshes onto the boundary surface and the subdivision of the resulting single cell layer in direction normal to the boundary. For all medial surface mesh points their nearest boundary points are used as projection points. Mesh points are generated on the projection rays at the specified boundary distances $\{d_k\}$ and the volume cells for all medial surface cells are assembled starting from the boundary surface (compare Figure 4). This gives hexahedral cells for quadrilateral medial surface cells and three-sided prismatic cells for triangular surface cells. Figure 16 shows the medial surface mesh for an L-shaped block together with the projection rays and cells extruded from a view selected medial surface cells. If the resulting number of mesh points on the projection rays is not the same for all vertices of a medial surface cell, degenerate cells (prisms, pyramids, tets etc.) will remain at the medial surface. Figure 17 shows such a configuration, where prisms result on both sides of a medial surface quad. Obviously, degenerate cells arise in pairs symmetric to the medial surface. Such cell pairs can often be recombined to hexahedral cells (analogous to the $\mathbf{R}^2$-case, as shown in Figure 4). At medial edge segments often three adjacent tet-pairs arise that can be united to one hex cell.

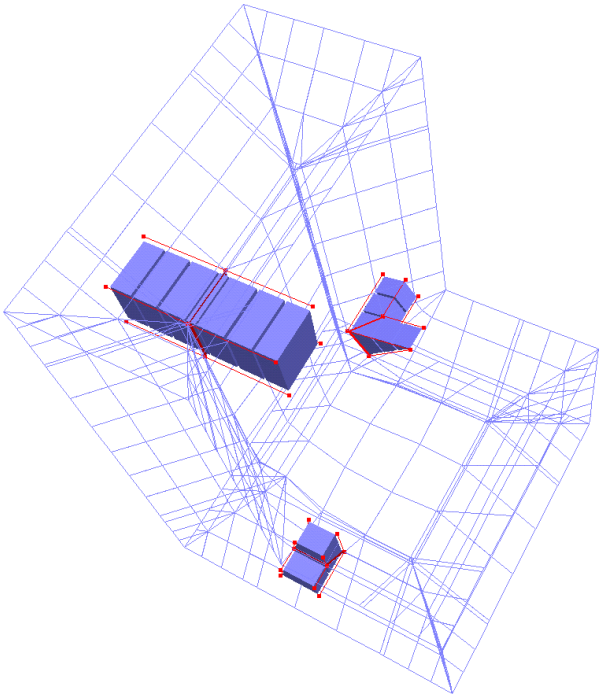

Figure 16: L-shaped block with medial surface mesh and some selected cell-extrusions

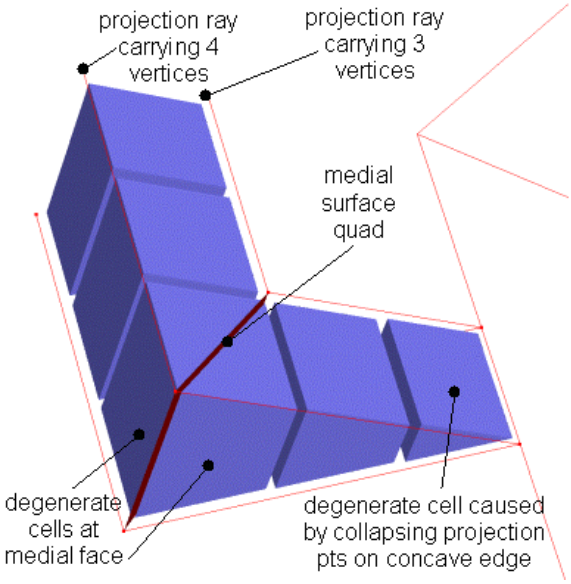

Figure 17: Degenerate cells at medial surface and extrusion towards concave boundary edge

Additional degenerate cells will remain along concave edges and vertices of the boundary where two or more vertices of a medial face mesh element are projected on to the same boundary point. In Figure 17 projection on one side of the medial surface is to the concave edge of the L-block, resulting in a degenerate hex at the edge. Projecting two opposite medial quad edges to two points on the concave edge is even necessary in this case. If the projection in Figure 17 was to four different points on the concave edge, the cell generated at the edge would contain a quadrilateral face with four collinear vertices. This still may happen in our current implementation and the problem how to avoid this in all cases has not been solved yet.

By the fact that a medial face always has two sides for projection, it is guaranteed that at least two cell layers are generated across any section of the geometry, no matter how small the local feature size is. Thus all geometric features are resolved properly also if a coarse spacing is prescribed.

A major open issue is the handling of degenerate (collapsing) medial faces. The surface meshes would reduce to an edge mesh or even a single vertex in this case. The projection is no more to two sides of a medial surface. An arbitrary number of nearest boundary points and hence possible projections exists. Projections would have to be chosen at certain angular intervals round the degenerate medial edge or vertex. The main difficulty is to detect such collapsing medial faces properly, however. If done geometrically, it has to be distinguished between nearly collapsing faces and degenerate faces that do not exactly collapse due to numerical noise. If detected, the angular projections could be chosen or, which is probably simpler, degenerate faces could be eliminated by adding them to the boundary surface itself and updating the medial axis. For a sphere, e.g., the medial surface degenerates to its center point. One would have to insert the center point as part of the boundary surface. The resulting medial surface would then be a sphere between the original sphere and the center, for which a proper surface mesh can then be generated and extruded. The cell-recombination step is not fully implemented yet. Without recombination, the L-block mesh e.g. contains 1036 hexes, 1072 prisms, 240 tets, 8 pyramids and 8 hexes with one edge collapsed to a vertex.

4. RESULTS AND CONCLUSIONS

In $\mathbf{R}^2$ the meshing process is much simpler, since only medial edges have to be meshed before projection. One example mesh is shown in Figure 18 and Figure 19. The $\mathbf{R}^2$ version is part of the meshing package FAME [30, 31, 32] since 1995.

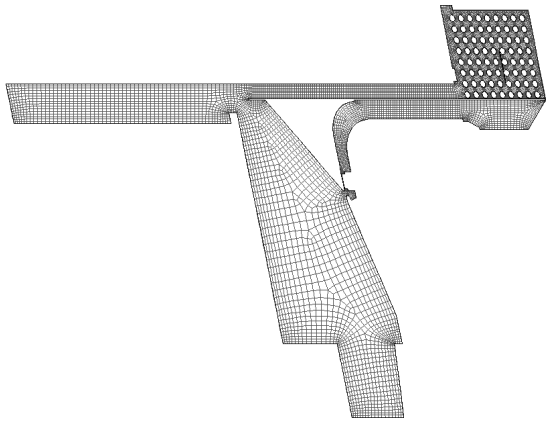

Figure 18: Heat exchanger in R^2, 15898 quads / 10s

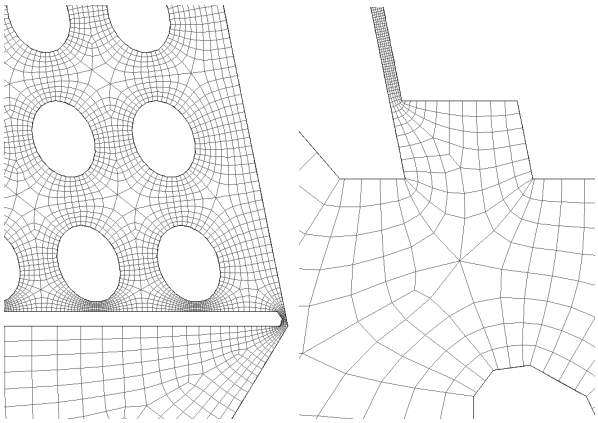

Figure 19: Heat exchanger, details

Interactive input is restricted to entering the two parameter specifying the series of boundary distances and the number of smoother-iterations. In the $\mathbf{R}^2$-version, an additional global parameter can be specified to produce meshes selectively finer at regions with small local feature size. This was applied in the shown example. The semi-structuredness and boundary-orthogonality of the mesh are evident. At concave corners triangular cells appear, which actually are quads with two collapsing vertices. These cells turned out to be unproblematic for our CFD solvers, however.

The $\mathbf{R}^3$-version has not been completed and integrated in FAME yet. Only a few first examples can be shown, together with meshing times (Figure 20 - Figure 26). Relatively coarse meshes were generated and the shown examples are more related to structural analysis. The mesh generator, however, is intended to be used for viscous CFD analysis mainly. Much thinner, anisotropic cell layers at the boundary would be generated for such applications. No final smoothing was applied to the example meshes. Meshing times are considered satisfactory. Two thirds of the meshing time are consumed for the medial axis construction which is independent of cell numbers, so times increase only sublinearly with mesh size.

For complex cases and without the not yet implemented cell recombination the mesh can not yet be called hex-dominant really. The mesh shown in Figure 21 e.g. contains 4141 hexahedra, 3917 prisms, 544 tets, 110 pyramids and 86 elements with 7 vertices (hexes with one collapsed edge).

The ratios are even worse for the other two examples: 14984 hexes, 28133 prisms, 11009 pyramids, 26675 tets and 37536 other degenerate hex elements for the piston example (Figure 22) and 22732 hexes, 28104 prisms, 23361 pyramids, 55478 tets and 84199 (!) other degenerate elements for the flange example (Figure 25). The high number of degenerate elements is to a large extent caused by the missing strategy for meshing degenerate medial surface parts (see e.g. the rounded parts on the piston top surface in Figure 22, where an extremely unstructured mesh is generated by the current implementation). These meshes are clearly not useful yet.

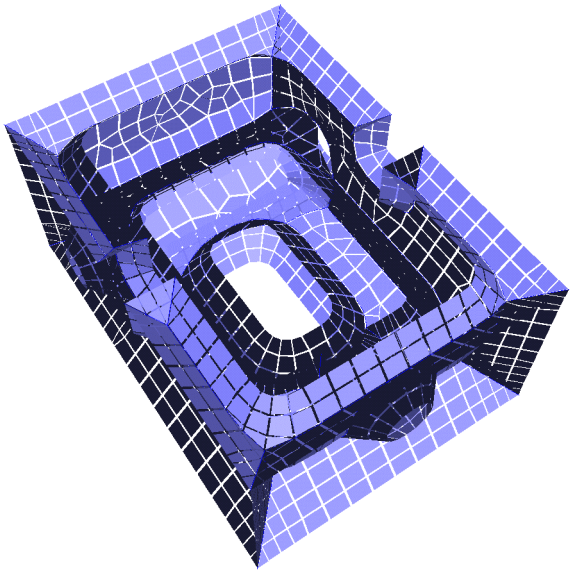

Figure 20: Medial surface mesh for slotted block

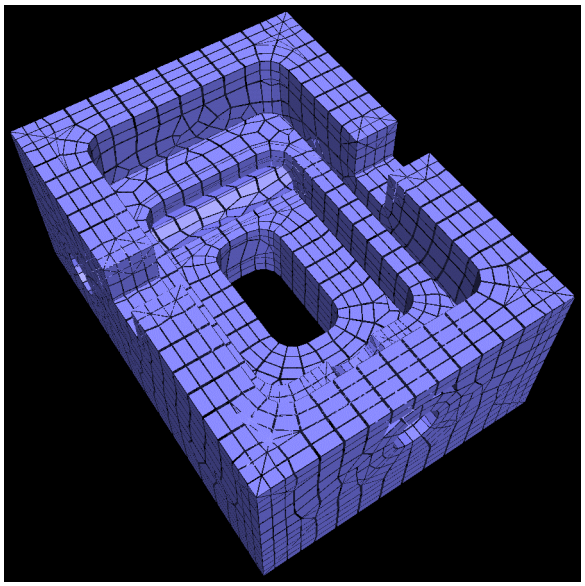

Figure 21: Volume mesh after projection and subdivision, 8798 cells / 38s

Figure 23, however, shows in a plane cut the semi-structuredness of the resulting meshes and the orthogonality at the boundary. As mentioned, problems are also still encountered at concave edges of the boundary surface (see e.g. Figure 24).

A further development step will be to allow for local variation of cell layer thickness, specified as a continuous scalar function on the medial surface. A lot of work still lays ahead for making our approach really useful.

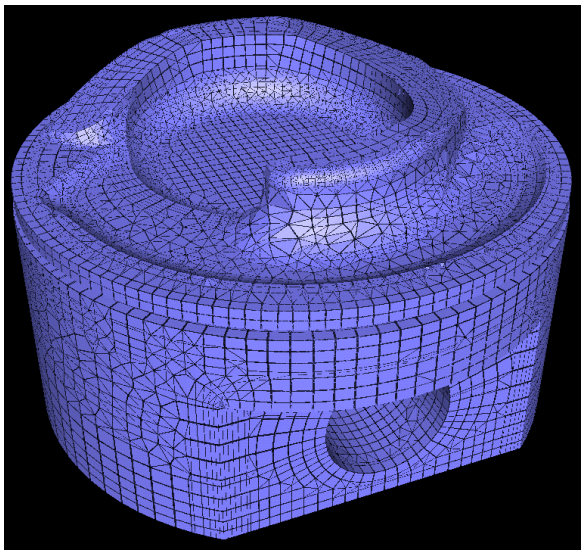

Figure 22: Piston: 118337 cells / 228s

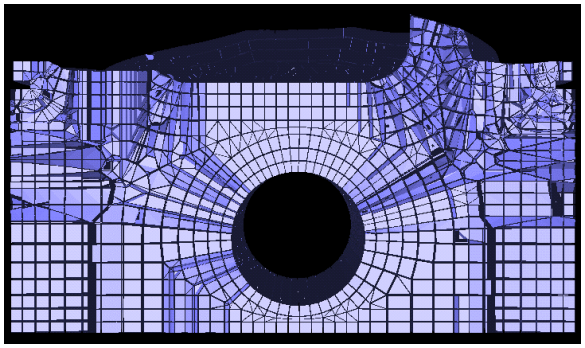

Figure 23: Piston, cut

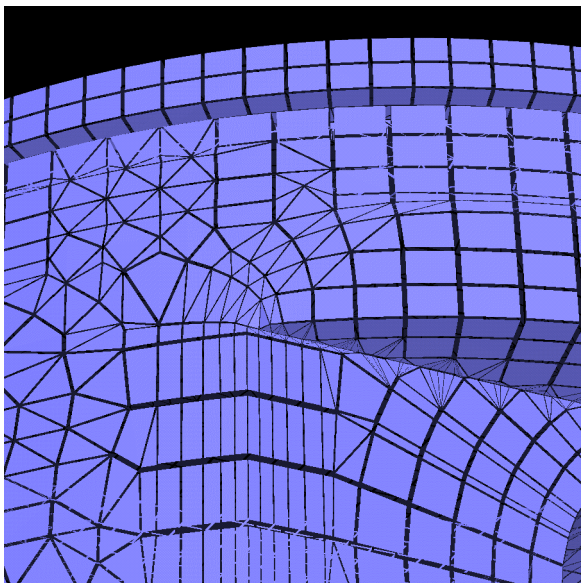

Figure 24: Piston, detail

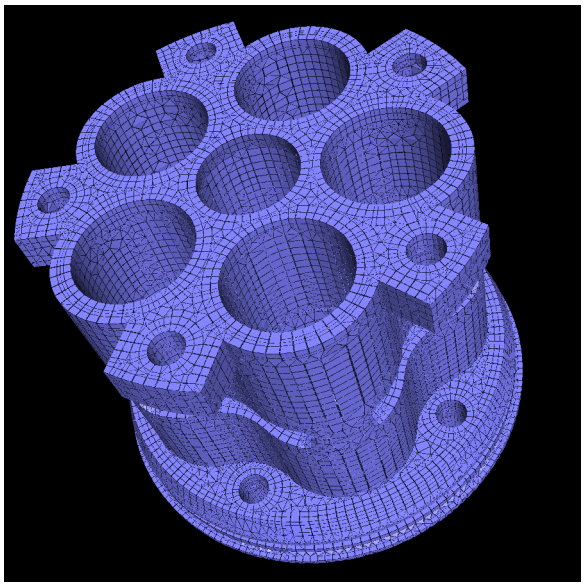

Figure 25: Flange, 213874 cells / 1137s

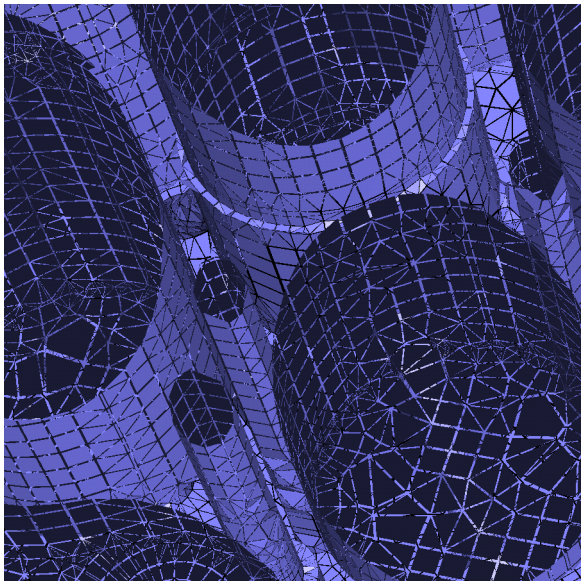

Figure 26: Flange, view from inside

ACKNOWLEDGEMENTS

The described meshing algorithms have been developed and implemented at AVL in the framework of the industrial R&D project AMeGOS (http://www.avl.com/amegos). AMeGOS is partly funded by the European Community under the High Performance Computing and Networking (HPCN) Area 6 Programme (ESPRIT IV), contract 28195.

REFERENCES

[1] Armstrong C.G., Robinson D.J., McKeag R.M., Bridgett S.J., Donaghy R.J., McGleenan C.A., Medials for meshing and more, Proc. 4th Int. Meshing Roundtable '95, Albuquerque NM, 277-288, Sandia National Laboratories, 1995.

[2] Bookstein, F.L., The Line-Skeleton, Academic Press, Computer graphics and images processing 11, 123-137, 1979.

[3] Turkiyyah G.M., Storti D.W., Ganter M., Chen H., Vimawala M., An accelerated triangulation method for computing the skeletons of free-form solid models, Computer-Aided Design, Vol. 29, No. 1, 5-19,Elsevier, 1997.

[4] Blum, H., A transformation for extracting new descriptors of shape, Models for the Perception of Speech and Visual Form (W. Wathen-Dunn, ed.), Cambridge MA, MIT Press, 1967.

[5] Amenta N., Bern M., Surface reconstruction by Voronoi filtering, Proc. 14th ACM Symposium on Computational Geometry, June 1998.

[6] O'Rourke, J., Computational geometry in C, Cambridge University Press, 1993.

[7] Aurenhammer, F., Voronoi diagrams: a survey of a fundamental geometric data structure. ACM Comput. Surv., 23, 345-405, 1991.

[8] Sheehy, D.J., Armstrong, C.G., Robinson, D.J., Computing the medial surface of a solid from a domain Delaunay triangulation, Proc. ACM/IEEE Symp. on Solid Modeling and Applications, Salt Lake City, Utah, May, 1995.

[9] Donaghy R.J., McCune W., Bridgett S.J., Armstrong C.G., Robinson D.J., McKeag R.M., Dimensional reduction of analysis models, Proc. 5th Int. Meshing Roundtable '96, Pittsburgh PA, 307-320, Sandia National Laboratories, 1996.

[10] Li T.S., Meshing and substructuring of 3D stress analysis models, PhD thesis, Dept. of Computer Science, The Queen's University of Belfast, 1994.

[11] Price M.A., Armstrong C.G., Hexahedral mesh generation by medial surface subdivision: part I, solids with convex edges, Int. J. Num. Meth. Eng., 38, 3335-3359, Wiley 1995.

[12] Price M.A., Armstrong C.G., Hexahedral mesh generation by medial surface subdivision: part II, solids with flat and concave edges, Int. J. Num. Meth. Eng., 40, 111-136, Wiley 1997.

[13] Sheffer A., Etzion M., Rappoport A., Bercovier M., Hexahedral mesh generation using the embedded Voronoi graph, Proc. 7th Int. Meshing Roundtable '98, Dearborn, 347-364, Sandia National Laboratories, 1998.

[14] Sherbrooke E.C., Patrikalakis N.M., Brisson E., Computation of the medial axis transform of 3-d polyhedra, Proc. 3rd ACM Symposium on Solid Modeling and Applications, ed. C.M. Hoffmann and J. Rossignac, 187-199, 1995.

[15] Ogniewicz, R. and Ilg, M., Voronoi skeletons: Theory and applications, Proc. CVPR'92, 63-69, Champaign, Illinois, June 1992.

[16] Lee, D.T., Medial axis transformation of a planar shape, IEEE Trans. PAMI 4, 363-369, 1982.

[17] Alt H., Schwarzkopf O., The Voronoi diagram of curved objects, Proc. 11th ACM Symposium on Computational Geometry, 89-97, June 1995.

[18] STL-format, http://www.ennex.com/fabbers/StL.shtml, technical source: StereoLithography Interface Specification, 3D Systems Inc., October 1989.

[19] Rivara, M.-C., New mathematical tools for refinement and/or improvement of unstructured triangulations, Proc. 5th Int. Meshing Roundtable '96, Pittsburgh PA, 77-86, Sandia National Laboratories, 1996.

[20] Fortune, S., A sweepline algorithm for Voronoi diagrams, Algorith. 2, 153-174, 1987.

[21] Bowyer, A., Computing Dirichlet tesselations, The Computer Journal, 24(2), 162-166, 1981.

[22] Watson, D.F., Computing the n-dimensional Delaunay tesselation with application to Voronoi polytopes, The Computer Journal, 24(2), 167-172, 1981.

[23] Mücke, E.P., Shapes and implementations in three-dimensional geometry, PhD thesis, Report UIUCDCS-R-93-1836, Dept. of Comp.Sci., Univ. of Illinois, Urbana-Champaign, 1993.

[24] Weatherill, N.P. and Hassan, O., Efficient three-dimensional grid generation using the Delaunay triangulation, Proc. First European Computational Fluid Dynamics Conference, Brussels, Sept. 1992, Ed. C. Hirsch and W. Kordulla, pub. Elsevier, 1992.

[25] Pebay, P.P., Frey, P.J., A priori Delaunay-conformity, Proc. 7th Int. Meshing Roundtable '98, Dearborn, 321-333, Sandia National Laboratories, 1998.

[26] Blacker, T.D., Stephenson, M.B., Paving: a new approach to automated quadrilateral mesh generation, Int. J. Numer. Meth. Engng., 32, 811-847, 1991.

[27] Borouchaki, H., Frey, P., Adaptive triangular-quadrilateral mesh generation, Int. J. Numer. Meth. Engng., 41, 915-934, 1998.

[28] Cass, R.J., Benzley, S.E., Meyers R.J., Blacker, T.D., Generalized 3-D paving: an automated quadrilateral surface mesh generation algorithm, Int. J. Numer. Meth. Engng.,39, 1475-1489, 1996.

[29] Owen, S.J., Staten, M.L., Canann, S.A., Saigal, S., Advancing front quadrilateral meshing using triangle transformations, Proc. 7th Int. Meshing Roundtable '98, Dearborn MG, 409-428, Sandia National Laboratories, 1998.

[30] AVL-AST FAME product description, at http://www.avl.com, AVL Graz, 2000.

[31] Wieser, K., Ennemoser A., Plimon, A., Automatic mesh generation - the key to successful integration of CFD in the development process", MTZ Worldwide, May 1998.

[32] Putz, N., Rebhandl, H., Sampl, P., FAME: Flexible Automatic Meshing Environment, 4th World Conference in Applied Fluid Dynamics (WUA-CFD), June 7 to 11, 1998, Convention Center Konzerthaus Freiburg, Freiburg i. Br., Germany, 1998.