INVITED PAPER *Special Issue on TCAD for Semiconductor Industries*

Mesh Generation for Application in Technology CAD

Peter FLEISCHMANN[†a)], **Wolfgang PYKA**[†], *and* **Siegfried SELBERHERR**[†], *Nonmembers*

SUMMARY After a brief discussion of the demands in meshing for semiconductor process and device simulation, we present a three-dimensional Delaunay refinement technique combined with a modified advancing front algorithm.

key words: *conforming Delaunay triangulation, constrained Delaunay triangulation, Steiner point, advancing front, quality mesh generation, boundary conforming mesh, Schönhardt polyhedron*

1. Introduction

We limit the discussion to fully unstructured tetrahedral mesh generation. It has been widely understood that cartesian based approaches sooner or later need unstructured extensions or hybrid methods to cope with the demands of Technology CAD in three dimensions [1]–[5]. Automatic unstructured hexahedral meshing is far more complicated and might become of interest for TCAD applications once it is fully solved. Currently, provable tetrahedral mesh generation for a well defined but unrestricted input still poses a challenge in three dimensions.

2. Meshing Demands in TCAD

We list four tasks and their main field of application:

- Meshing of comparatively simple (near 90° dihedral angles between input facets) but huge structures composed of a large number of vias and lines. *Simulation of 3D Interconnects.*
- Handling the minimum and possibly faulty information on arbitrary complex structures provided by topography simulation with eventually moving structure boundaries [5]. *Semiconductor process simulation.*
- Resolving highly non-linear quantities with a directional mesh density which is not only limited to the three directions of the cartesian axises (anisotropy). *Semiconductor device simulation.*
- Dealing with very small input angles and an extremely steep grading of the local feature size (inhomogeneous). *Semiconductor process and device simulation.*

Manuscript received October 10, 1998.

[†]The authors are with the Institute for Microelectronics, Technische Universität Wien, Gusshausstrasse 27-29/E360 A-1040 Wien, Austria.

a) E-mail: fleischmann@iue.tuwien.ac.at

Full scale three-dimensional simulation has suffered from the lack of available CAD tools for the various established data formats in the field. In fact, such a framework of tools is itself still a subject to research and development [6].

Aside from existing means to couple and control the simulators (e.g. job farming [7]), powerful geometry processors are missing. Considerable person power is consumed to detect inconsistencies in a tedious manner, and to debug structures that the mesher should, but does not mesh, e.g. [8]. Dimensional reduction by use of the medial axis eliminates unnecessary input features and simplifies the meshing [9]. However, the medial axis is directly derived from the Voronoi Graph which itself needs a robust Delaunay triangulation for computation.

3. Methodologies

The history of methodologies in two-dimensional process and device simulation leads to the observation that an unrestricted — with respect to input structures — and provably terminating *triangulation engine* is required. In addition so called "protection layers" or similar approaches to enforce a certain anisotropy along inversion layers or otherwise highly non-linear interfaces are required [2], [3], [10], [11]. All methods have come to a stage where they more or less rely on a triangulation engine , e.g. [12], to generate a valid tessellation of a set of supporting grid lines and/or grid nodes supplied together with the boundary. Nevertheless, the following questions arise:

- Are Delaunay methods necessary, and if, what kind ?
- Which conclusions with regard to three dimensions can one draw from observing the development in two dimensions ?

3.1 Constrained- and Conforming-Delaunay Triangulation

It is still a common believe that the sole purpose of the Delaunay triangulation is the element quality (maximizing the minimum angle in a triangulation) and the application of its dual Voronoi Graph to the Control Volume Integration method (Box Integration method).

However, most importantly it is a mathematically defined triangulation which enables fast (because direct) and provable triangulation algorithms as opposed to for instance original advancing front methods [13] which generally run much slower. They place the grid nodes and the elements simultaneously to achieve boundary aligned elements (boundary conforming mesh) and need to employ elaborate schemes to test for mesh self-intersections in order to produce a consistent tessellation. An attempt to utilize the efficient and well defined Delaunay triangulation as a "background" mesh for advancing front style mesh generation has been made by [14].

When it comes to incorporating the boundary into the Delaunay triangulation two different concepts exist.

Constrained Delaunay Triangulation, *CDT*
The insertion of additional points on the boundary is not allowed. The initial Delaunay triangulation of the input vertices will not be conform with all input edges. To ensure their presence in the final *CDT* local modifications are performed which remove Delaunay edges and replace them by un-split input edges. The resulting Delaunay triangulation is said to be constrained by the input boundary. Elements near the boundary or interfaces do *not* necessarily satisfy the Delaunay criterion (see below).

Conforming Delaunay Triangulation, *RDT*†
The initial set of vertices is extended with additional "Steiner Points" [15]. (The name "Steiner Point" does not indicate a specific location for a new point, but rather refers to any refinement point.) The Steiner Points are added in such a way so that the triangulation of the extended set of input vertices and refinement points contains the boundary. All elements must satisfy the Delaunay criterion.

The stated criteria can be derived from the definition of the Delaunay triangulation as the dual of the Voronoi Graph.

Property 1 (Delaunay edge): Let D be a finite set of points in a sub-domain Ω^n of the n-dimensional space R^n. Two points d_i and d_j are connected by a Delaunay edge e if and only if there exists a point $x \in \Omega^n$ which is equally close to d_i and d_j and closer to d_i, d_j than to any other $d_k \in D$. The point x is the circumcenter of an n-dimensional sphere which passes through the points d_i, d_j and which contains no other points d_k of D.

$$e_{Delaunay}(d_i, d_j) \Longleftrightarrow \exists x$$

$$x \in \Omega^n \ \wedge$$

$$\|x - d_i\| = \|x - d_j\| \ \wedge$$

$$\forall\, k \neq i, j \ \ \|x - d_i\| < \|x - d_k\|$$

Property 2 (Delaunay triangle): Let D be a finite set of points in a sub-domain Ω^n of the n-dimensional space R^n. Three non-collinear points d_i, d_j and d_k form a Delaunay triangle t if and only if there exists a point $x \in \Omega^n$ which is equally close to d_i, d_j and d_k and closer to d_i, d_j, d_k than to any other $d_m \in D$. The point x is the circumcenter of an n-dimensional sphere which passes through the points d_i, d_j, d_k and which contains no other points d_m of D.

$$t_{Delaunay}(d_i, d_j, d_k) \Longleftrightarrow \exists x$$

$$x \in \Omega^n \ \wedge$$

$$\|x - d_i\| = \|x - d_j\| = \|x - d_k\| \ \wedge$$

$$\forall\, m \neq i, j, k \ \ \|x - d_i\| < \|x - d_m\|$$

It depends on the task at hand and on the used discretization scheme which method, *CDT* or *RDT*, is preferable. Control Volume Integration which is often used in device simulation usually requires closed control volumes at the boundary and interfaces. If the Voronoi boxes are used as the control volumes, the requirement directly translates to the following criterion for the boundary and interfaces.

Property 3 (smallest sphere): Let D be a finite set of points in n-dimensional space R^n and let t be an $(n-1)$-dimensional boundary simplex. t and its n linear independent points $d_{t,i}$ define an infinite number of n-dimensional spheres S_{t,r_c} where each sphere contains the points $d_{t,i}$ on its perimeter. All points d_k in D are associated with a Voronoi box V_k. The smallest sphere $S_{t,r_{\min}}$ contains no other points of D if the Voronoi box V_m does not intersect t for all m with $d_m \notin d_{t,i}$.

In two dimensions the smallest sphere criterion is also sufficient to guarantee closed control volumes. In three dimensions an empty smallest sphere (equatorial sphere of a boundary triangle) might not guarantee a closed control volume. It can be shown that an additionally applied smallest sphere criterion to the edges of a boundary triangle is in conjunction stronger and suffices.

Property 4 (smallest sphere, edges and triangle): Let D be a finite set of points in three-dimensional space R^3 and let t be a boundary triangle defined by three boundary edges $e_{t,i}$. Each $e_{t,i}$ with its two linear independent points $d_{e,j}$ defines a smallest three-dimensional sphere $S_{e,r_{\min}}$ where each sphere passes through the points $d_{e,j}$. t defines a fourth smallest sphere $S_{t,r_{\min}}$ (equatorial sphere) which passes through the points $d_{t,l}$ of the triangle. All points d_k in D are

†Although "CDT" is a common term in literature, we are not aware of any abbreviation for the Conforming Delaunay triangulation. For the sake of brevity we will call it the "RDT" as in Refining Delaunay Triangulation.

associated with a Voronoi box V_k. No Voronoi box V_m intersects t for all m with $d_m \notin d_{t,l}$ if and only if the four spheres $S_{e,r_{\min}}, S_{t,r\min}$ contain no other points of D.

The proof essentially boils down to showing that the volume covered by the four smallest spheres (one triangle and three edges) "swallows" any sphere $S_{x,R}$ defined by a circumcenter x and radius R, where x is located anywhere on the triangle and R is such that the sphere $S_{x,R}$ does not contain any of the three vertices of the triangle.

The key message is that Property 3 is stronger than the Delaunay criterion (Prop. 1 and Prop. 2). If no Voronoi box associated with an internal grid node intersects the boundary, the simplices forming the boundary must be Delaunay simplices. In other words: If one chooses — for the sake of efficiency — a Delaunay triangulation to utilize its inherently given Voronoi boxes as control volumes, and if the discretization scheme relies on closed control volumes, the boundary elements must be enforced to satisfy the stronger criteria. This can be achieved by using *RDT* where refinement follows the stronger rules given above.

On the other hand, if the discretization scheme does not rely on Delaunay elements at the boundary and interfaces, it might be desirable to avoid any refinement on the boundary. This is important for the first of the following approaches to achieve a desired anisotropy.

(i) The triangulation engine is fed with grid lines to construct protection layers. For some applications it is crucial that these grid lines are not split which destroys the desired anisotropy. Employing *CDT* guarantees that, but one must be aware that the Delaunay criterion for elements near those constraints is not necessarily fulfilled.

(ii) The triangulation engine is supplied with a set of grid nodes only. Then, one must be aware that the resulting connectivity after triangulation might not match the expected grid lines.

(iii) One again supplies the triangulation engine with grid lines but employs *RDT*. This will result in higher refinement but enforces the grid lines and the Delaunay property.

In three dimensions *CDT* does not exist a priori. This follows directly from the existence of the so called "Twisted Prism" or Schönhardt Polyhedron which cannot be tetrahedralized without inserting additional vertices [16] (Fig. 1). No matter which tetrahedron is chosen, it must intersect the boundary. A condition under which *CDT* exists for the constraining input facets is given in [17]. The key question is *how to insert those vertices* to satisfy the condition enabling *CDT* without further refinement or to perform *RDT* with a guaranteed bound on refinement. Very interesting work for input facets which form dihedral angles of no less than 90°

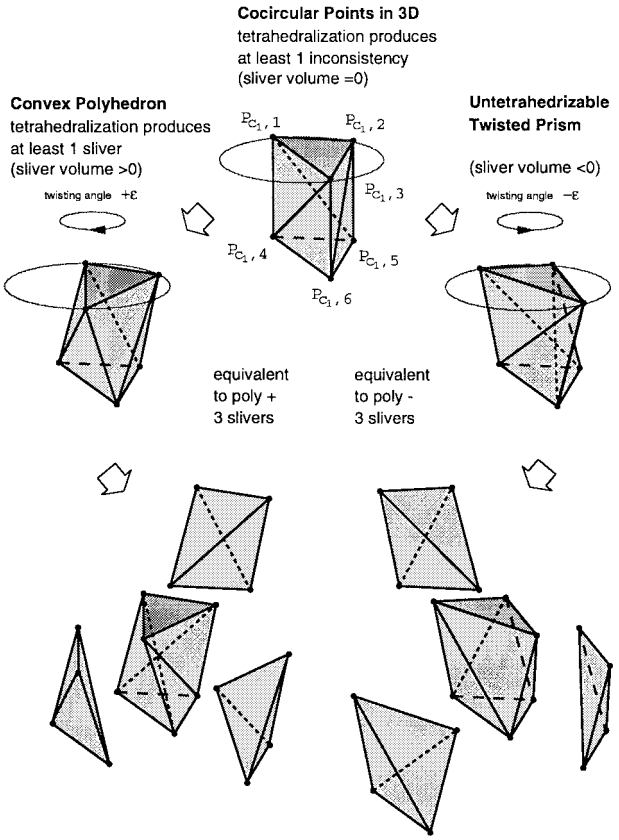


Fig. 1 The Schönhardt polyhedron in relation to slivers and co-spherical point sets.

can be found in [18] and with application to the control volume integration in [19]. However it remains an outstanding problem to optimize the refinement method and to prove its bounds for arbitrary input angles. In Sect. 4 we present a special refining scheme designed for sharp angles often exhibited by semiconductor devices, which, however, at present is still a heuristic technique.

3.2 Motivation

It is apparent that given a robust triangulation engine one gains many ways in comprising powerful application specific meshing schemes in three dimensions. The extension of Delaunay methods to three dimensions is not straightforward. Nevertheless, at present we are not aware of competitive tessellation algorithms with regard to complex input structures and performance to serve as triangulation engine. Pure octree based solutions combined with tetrahedral templates are hardly used for todays TCAD applications. Considering the variety of existing approaches in three dimensions one can observe that recently most methods have evolved to become hybrid methods and require some sort of tetrahedralization [1], [4], [20], [21]. Often this is achieved by local transformations [22]. For moving boundary situations level set methods have gained great importance

[4], [23], [24]. They require a mesh to define the magnitude which describes the moving boundary. Also, an Euler grid is often needed at some stage and must be derived from the Lagrangian boundary representation by means of a tetrahedralization algorithm.

The anisotropic Delaunay triangulation [25] is worthwhile to mention. Through a linear transformation a circle can be transformed to an ellipse. Applying the Delaunay triangulation in the transformed space yields an anisotropic mesh. The difficulties lie in the grading of the mesh. If the transformation is changing over the domain to adapt to a locally desired mesh density, the anisotropic Delaunay triangulation looses its excellent property to guarantee a valid tessellation. Conformal mapping techniques might overcome their computation difficulties, but it remains an open question as to how efficiently they can be applied to three dimensions.

The two stage concept of linking a fixed node set (triangulation) and placing nodes independently seems to be most stable and offers a high flexibility for mesh generation. If no initial set of nodes is constructed (e.g. in an advancing front style manner) or available (input nodes), the internal nodes are generated by refining and updating the tetrahedralization. This allows for high quality meshes with controllable grading. An example for the success of this approach in two dimensions can be found in [12]. The advantage is that the mere placing of nodes is easy, because one cannot generate inconsistencies and no constraints have to be followed. Nodes located exterior of the structures are permitted, as well as on the boundary. It is the job of the triangulation to yield the valid tessellation for any point set.

We distinguish two Delaunay mesh generation methods.

Convex Hull and Segmentation
: is commonly used where a Delaunay triangulation algorithm is applied only to the point set and is followed by a refinement to recover those input facets not existent in the triangulation. A post segmentation step is necessary to carve the desired mesh out of the entire mesh of the convex hull.

Modified Advancing Front
: has not been used widely, probably due to its original difficulties with degenerate Delaunay cases [26]. In Sect. 5 we present our version which overcomes these problems [27]. The advantage is that the tetrahedralization of the convex hull is avoided, hence also making the segmentation step unnecessary. The refinement of input facets to guarantee their presence in the resulting Delaunay triangulation precedes the actual tetrahedralization.

For the first method an incremental Delaunay algorithm [28] is often used. Meshing the entire convex hull costs performance especially for extremely non-convex domains. In three dimensions it is not trivial to recover missing input facets when the triangulation of the facets is not unique (degenerate cases).

The second method which appears to originate from [29] offers some advantages. It possesses the principle of "locality": Undesired parts of the domain are never meshed or touched upon. This enables nice opportunities for real time visualization. The algorithm can be parallelized [30]. It is equally well suited for local mesh adaptations as the incremental algorithm. To the best of our knowledge there seems to be only one other fairly new reference which can handle degenerate cases by a variational approach [31]. The problem with applying perturbations to the location of the points to avoid degenerate cases is that the tetrahedralization yields a much higher number of slivers in a strict sense: not any element of bad quality, but rather an element formed by four nearly coplanar points on a sphere. The relation of slivers with degenerate cases can be seen in Fig. 1 and is further explained in [32]. A post-processing step to remove slivers becomes stringent.

Note that any Delaunay method will have to employ some refinement scheme in three dimensions (RDT) if it aims to generate a tetrahedralization containing the boundary.

4. Delaunay Surface Refinement

Given a set of constraining input facets (polygonal representation) a triangular surface representation is derived by splitting all polygons into triangles. The polygons may be non-convex. The constraining input facets do not have to form a closed surface. A refinement algorithm is then applied resulting in a triangulation which satisfies Properties 1, 2, and if desired Properties 3 and 4. An important operation is defined to reduce the number of refinement points.

Definition 1 (flip-able triangle): Two triangles t_1, t_2 which are in conjunction convex and which share one common edge e_c are said to be flipped if they are replaced by two triangles $\tilde{t}_1, \tilde{t}_2$ forming the same outline but sharing a different edge $\tilde{e}_c \neq e_c$. This operation is extended to non-planar triangles: If t_1, t_2 sharing an edge form a dihedral angle $\alpha = 180° + \epsilon$ where $|\epsilon| \leq \epsilon_{acc}$ and ϵ_{acc} is a user controlled accuracy, and if the equatorial sphere of t_1 contains a vertex of t_2 which does not belong to t_1, the triangle t_1 may be flipped as well and will be called flip-able triangle.

We introduce the term *flip saturation*: A triangle may not satisfy the empty smallest sphere criterion and may not be flip-able. The triangle may become flip-able after a certain number of flip operations applied on other triangles (Fig. 2). This situation is said to be "unsaturated." It is desired to reach the state of flip saturation before unnecessary refinement occurs. We achieve this by employing the recursive triangle flip.

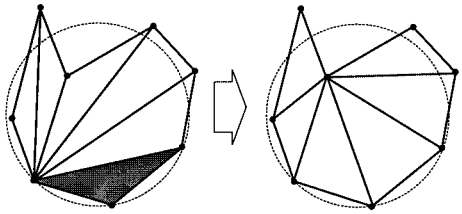


Fig. 2 A triangle that is at first not flip-able and the case when flip saturation has been reached.

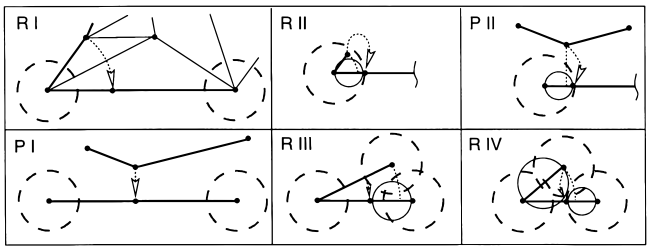


Fig. 3 Refinement types for structural edges.

Recursive Triangle Flip

Two triangles t_1, t_2 are flipped. The resulting triangles $\tilde{t_1}$ and $\tilde{t_2}$ are each checked if they are flip-able. If any of the two triangles $\tilde{t_1}, \tilde{t_2}$ is flip-able with a triangle t_i where $i \notin 1, 2$ it will be flipped as well. Repeat for the flipped triangle $\tilde{t_i}$ until no further flip operations are possible.

This operation is performed once on each triangle during a linear scan over all existing triangles. The procedure during which no refinement is allowed is only necessary at the start to ensure flip saturation. Afterwards, during refinement the flip saturation is maintained after insertion of a point by applying the recursive flip locally on the affected triangle only.

We now define structural and flip-able edges.

Definition 2 (flip-able and structural edges): If two triangles t_1, t_2 sharing an edge e_c form a dihedral angle $\alpha = 180° + \gamma$ where $|\gamma| > \epsilon_{acc}$ and ϵ_{acc} is a user controlled accuracy, the edge e_c is called a structural edge or *S-edge*. If $|\gamma| \leq \epsilon_{acc}$ the edge e_c is called flip-able edge. If a triangle t_3 possesses an edge e_{open} with no existent adjacent triangle, the edge e_{open} will be called a structural edge as well.

Note that two triangles need not be flip-able if they share a flip-able edge (e.g. coplanar triangles with a non-convex outline).

All *S-edges* are detected and stored in a special data structure for further processing. The empty smallest sphere criterion will be enforced for *S-edges* by a novel refinement scheme we have developed (Fig. 3). The idea is to avoid refinement "feedback" where an inserted point causes further point insertions for other *S-edges*, which is especially important where incident *S-edges* span angles of less than 90°. The *S-edges* are drawn in bold, the dashed circles illustrate the minimal edge length up to which refinement is permitted, and the solid circles symbolize empty sphere tests. There are two possible types how a refinement point is derived from a "disturbing point" (a point which infringes the empty sphere criterion).

Type P **Insertion** An orthogonal projection of the disturbing point onto the *S-edge* is used.

Type R **Insertion** The disturbing point is rotated onto the *S-edge*. The rotation axis passes through the point at which the two *S-edges* are incident.

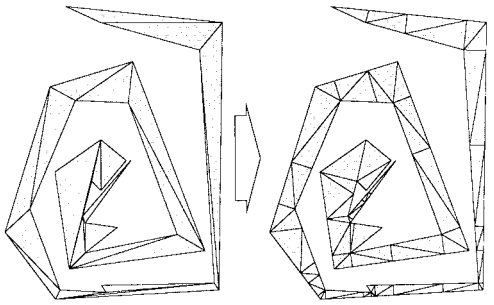


Fig. 4 Refining structural edges for the trivial case of a planar polygon.

If more than one disturbing point exists, a best candidate is selected by comparing the relative distance and choosing the closest. The simple cases are *P I and R I*. In *P II, R II, and R IV* the minimal edge length is limited and the inserted point receives an offset. If the ratio between the length of the two *S-edges* is close to one, the most complex situation case *R III* evolves. An offset as in case *R IV* would not produce a good result and the sphere test depicted by the left solid circle in case *R IV* could fail due to the inserted point. An overall improved situation results from an intended first order feedback where actually two points will be inserted in two consecutive steps *R III* + *R I*. The circle for the sphere test which causes the insertion in the first place is omitted in all sketches. Once the location of the new point has been determined and prior to its insertion, sphere tests are performed for some of the new edges to avoid feedback (solid circles).

Note that the algorithm was designed for three dimensions in spite of the two-dimensional sketches. Figure 4 shows the result for the trivial case of a planar polygon. The *S-edges* (outline) satisfy the empty smallest sphere criterion after refinement.

The triangles are processed after the *S-edges*. If desirable, the refinement can be reduced by omitting the empty smallest sphere criterion for triangles (Property 3). The weaker Delaunay Property 2 cannot be checked easily, because the number of spheres to test can theoretically be infinite: A single sphere of unspecified size is required to be empty. Hence, the in-sphere

test would have to be repeated for different sizes until an empty sphere has been found. We implemented an equivalent criterion which needs at most two in-sphere tests. It uses a metric λ described in Sect. 5.

Property 5 (double sphere): Let D be a finite set of points in three-dimensional space R^3 and let t be a boundary triangle with a non-empty smallest sphere $S_{\min}$. The point $d_{k,\lambda_{\min}}$ is contained in $S_{\min}$ and minimizes a metric $\lambda = H\vec{M}_k \cdot \vec{n}$. The triangle t is a Delaunay triangle if and only if the sphere S_{double} defined by t and $d_{k,\lambda_{\min}}$ contains no other point in D.

A key idea for refinement with provable bounds originates in [33] and can be found in [15], [34]–[36]. The circumcenter of the triangle becomes the location of the new point to be inserted. Hence, the triangle which will actually be split can be a different triangle than the "bad" triangle which causes the refinement. The bad triangle will not survive, because the new point at the circumcenter violates its sphere criterion. This maintains a good spacing between all points and is the key to the proven bound.

We have extended this method for non-planar surfaces. The location of the new point is derived by orthogonal projection of the circumcenter onto the surface. It is thereby checked if the refinement point really violates the sphere criterion for the bad triangle. A point location is necessary to find the triangle containing the projection. For the sake of efficiency the projection is not orthogonal to the unknown triangle which we search for. Instead it is orthogonal to the bad triangle. Hence, the search can be concluded in the two-dimensional plane spanned by the bad triangle onto which neighboring triangles will be projected.

Locate Triangle

Starting with a triangle $t = t_{bad}$ a point H_{proj} in the vicinity is searched for (H usually is the circumcenter of t_{bad}, hence $H \equiv H_{proj}$). Determine one edge e of t which faces H_{proj} and is a flip-able edge (Def. 2). Follow edge e by finding the adjacent triangle t_{adj}. Repeat for $t = t_{adj}$ until the projection of triangle t contains H_{proj} or an *S-edge* has been encountered.

This algorithm can efficiently search for arbitrary points H (not just circumcenters) as long as the point is known to be in the vicinity. If the triangle is not extremely obtuse, H_{proj} will be contained in the triangle itself or in an adjacent triangle. Then, no iterations or only a few will be required. If an *S-edge* is encountered the triangle will not be refined. Instead the *S-edge* will be refined as described above. This is important for several reasons: Insertion of H_{proj} is usually followed by triangle flip operations which eliminate the bad triangle. Hence, the edge to follow must be flip-able. Also, projecting the circumcenter does not make sense for surface areas with sharp dihedral angles.

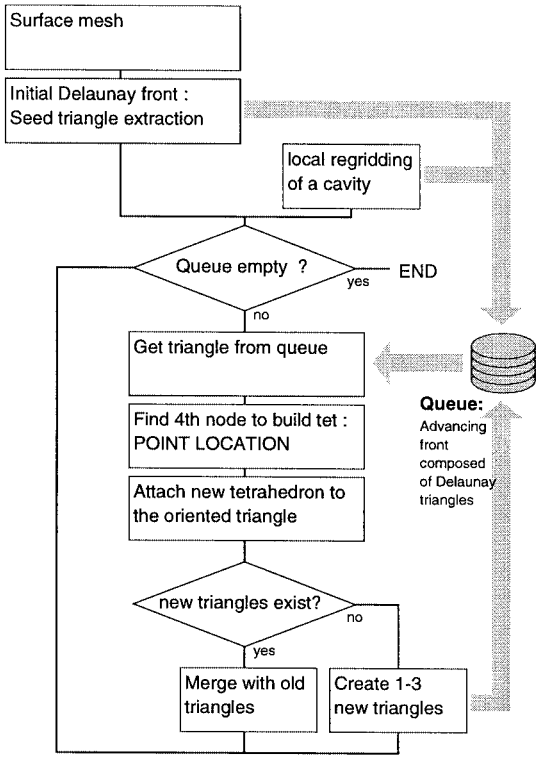


Fig. 5 Modified advancing front algorithm.

Originally, this type of refinement was only applied to Delaunay triangles with an empty circumsphere to improve quality angle criteria. When the sphere is not empty, points might be inserted extremely close to each other. However, in our case we refine *non-Delaunay* triangles to enforce their Delaunay property. This can be done successfully, provided the state of flip saturation is maintained at all times.

It is worthwhile to mention that the applied properties and sphere tests during triangle refinement are extended with min-angle or max-area criteria to increase quality and refinement.

5. Modified Advancing Front Algorithm

Figure 5 shows the flow diagram of the modified advancing front algorithm. From the generated surface triangulation Delaunay triangles are extracted to form an oriented initial front. These triangles can be imagined as seeds which are inserted into a queue to "grow" tetrahedra. At the start, the algorithm needs a non-empty queue. It does not require the queue to hold all surface triangles. One triangle per enclosed segment is sufficient. The surface triangulation has two purposes:

1. It provides the initial front for the advancing front algorithm to start with. One triangle per segment is enough, because a single triangle forms a front as well.
2. It provides a border for the advancing front algo-

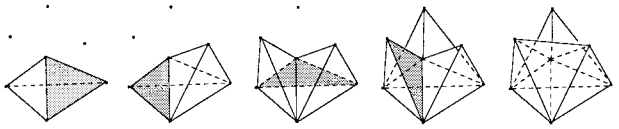


Fig. 6 The triangle to which the next tetrahedron is attached is grey.

rithm which cannot be passed.

The triangles of the initial front and all later generated triangles of the advancing front have a well defined orientation depending on the order of their vertices. They "face" the half-space to which their normal vector points. Given a seed triangle (taken from the queue) a tetrahedron is attached which contains a fourth point that has a positive distance to the triangle relative to the normal vector. In other words, the tetrahedron will only be attached to that side of the triangle which faces the half-space to which the normal vector points. In this way, one can distinguish a "front side" and a "back side" of each triangle.

Repeatedly attaching tetrahedra to the front sides of the triangles of the queue, removing them from the queue when they have been processed, and inserting newly generated triangles into the queue leads to a growth process of tetrahedra (Fig. 6). Note, that the triangles of the queue form the advancing front at all times. It advances when a new tetrahedron (attached to a triangle which is removed from the queue) results in new triangles which are inserted into the queue. Generally, a created tetrahedron can produce any number between 0 and 3 new triangles. At the start of the tetrahedralization process with the given seed triangles each created tetrahedron will more likely produce 3 new triangles and the queue will increase rapidly. Later on, the advancing front will merge with itself or parts of the surface triangulation. A tetrahedron consists of n new triangles, $(3-n)$ previously generated triangles, and the triangle to which it is attached. The $(3-n)$ previously generated triangles must have been previously inserted into the queue or belong to the initial surface triangulation. They are part of the advancing front. When they are encountered during the creation of a new tetrahedron, they are removed from the queue and the advancing front is stopped. The front cannot pass through the boundary or itself [32]. In such cases the creation of the tetrahedron results in a decrease of the size of the queue. When the queue is empty and all its triangles have merged with each other, the tetrahedralization process is finished.

In the described manner a segment of the input domain is "filled" with tetrahedra, if there exists at least one seed triangle which has a normal vector that points into the volume of the segment.

Given the fixed grid node distribution and an oriented triangle only one grid node will complement the triangle to form a valid Delaunay tetrahedron to be attached to the triangle's front side. A local region is defined by a sphere with circumcenter M containing the circumcircle of the triangle with circumcenter H and a normal distance λ. The unit length normal vector of the triangle is $\vec{n}$.

$$\lambda := \vec{HM} \cdot \vec{n}$$

Each point P_i in the local region defines a sphere with circumcenter M_i and λ_i. The point P_j with minimal $\lambda_{j,\min}$ will be chosen and hence the attached tetrahedron must satisfy the Delaunay property.

The running time of the algorithm depends heavily on the point location method. The sphere defining the search region always fits into a cube parallel to the bounding box. Therefore, we implemented a fast point bucket octree which provides a search function for such rectangular regions. The octree search and the minimizing of λ (λ-criterion) introduces the logarithm into the overall time complexity $O(n \log n)$ where n is the number of tetrahedra. The performance for random point clouds is given in Table 1.

Table 1 Running time on an HP 9000-735/100.

quantity of ...			CPU time
points	tetrahedra	triangles	(in sec)
103	535	1098	0.2
503	3016	6112	1.4
703	4323	8739	2.1
1003	6268	12635	3.3
1503	9494	19107	5.1
2003	12713	25557	6.8
2503	16076	32300	8.8
3003	19394	38967	11.1
4003	25969	52140	15.6
5003	32691	65605	19.7
10003	65927	132160	46.5
20003	132854	266133	92.0
30003	199613	399756	145.0
40003	266899	534405	207.0

Degenerate cases where a compound of points $P_{c,i}$ has identical λ_c result from non-unique Delaunay triangulations. A specially enhanced algorithm which we have developed copes with such cases and is described in [27].

6. Application Examples

Dependable mesh generation and robustness under finite precision arithmetics is a must for practical applications. Often, floating point filters are integrated to control round off errors [1], [37]. Such filters usually have a great effect on the runtime of an algorithm. Exact arithmetics cannot be achieved easily for Delaunay methods which require in-sphere tests and the calculation of square roots. A different approach to ensure a robust and consistent implementation of the modified advancing front algorithm is described in [27].

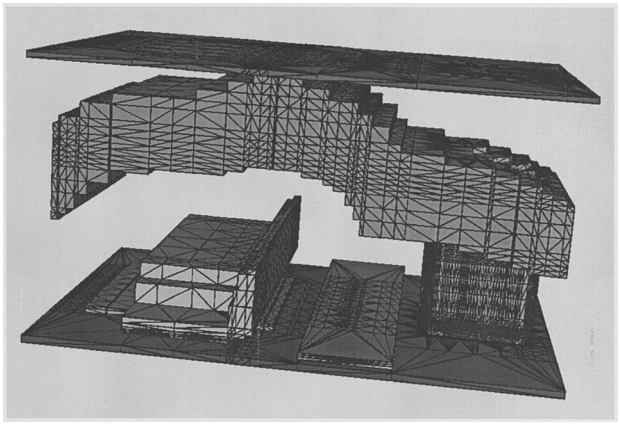


Fig. 7 Layered mesh with 71986 elements.

6.1 Capacitance Extraction Using SAP

The finite element package SAP (Smart Analysis Program, [38]) can be used for electro-thermal interconnect simulation. It contains a layer-based preprocessor to generate three-dimensional meshes from cross-sections. This limited approach forces a uniform element size along one coordinate axis. The lateral mesh density must be identical for all cross-sections [5]. Figure 7 shows a typical mesh generated from a solid model based on layers.

6.2 Topography Simulation Linked to AMIGOS

The structures evolving after etching and deposition steps are more complex than can be handled with methods such as the layer-based method mentioned above. The output of a topography simulator is often highly refined to match the required resolution for the manipulation of the structure. After applying the surface Delaunay adaptation the modified advancing front algorithm produces a mesh for AMIGOS [39].

One application is the high pressure CVD of Tungsten used for a Ti/TiN/W plug fill process. The geometry results from an initial low pressure deposition of a TiN barrier layer into the via. This PVD process is determined by ballistic transport of the sputtered Ti particles. The finite element model used for the following high pressure CVD process is calculated on the mesh and assumes that W is reduced from WF_6 using H_2 and forming HF as by-product. The three gas species diffuse in the feature and the reduction takes place at the bottom and the side walls of the feature. Depending on the diffusion coefficients and the reaction rates a steady state of the gas distribution is reached leading to a depletion of WF_6 in the feature and to a characteristic non uniformity in the deposition rates.

Figure 8 shows the mesh of a cylindrical via and the WF_6 concentration. A cross-section of the mesh is

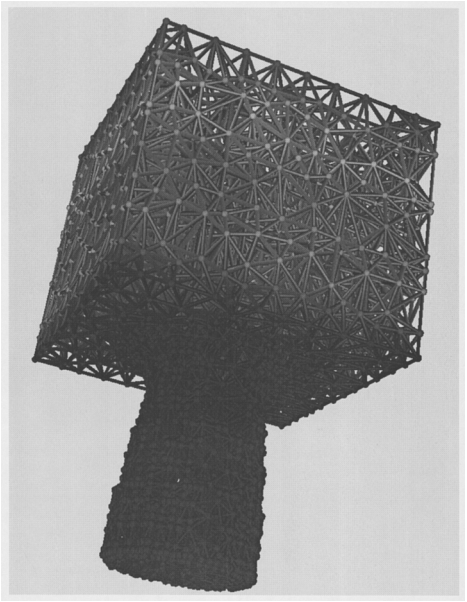


Fig. 8 Cylindrical via, mesh with 7324 elements.

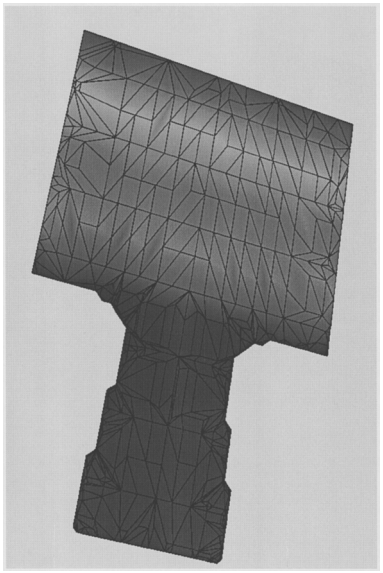


Fig. 9 Cross-section of cylindrical via, mesh with uniform grid nodes (7324 elements).

depicted in Fig. 9. A different mesh with a highly refined region near the boundary and in the interior of the via was generated by constructing a non-uniform grid node distribution derived from the boundary vertices (Fig. 10). The same model was calculated on a damascene structure and the resulting WF_6 concentration is shown in Fig. 11.

6.3 NMOS Transistor

The presented example in 0.18μm technology includes a thin oxide layer of 30 nm (Fig. 12). Devices with high ratios between local feature sizes pose a challenge to

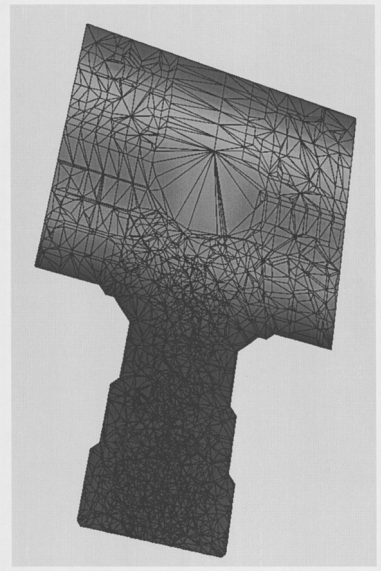


Fig. 10 Cross-section of cylindrical via, mesh with non-uniform grid nodes (75720 elements).

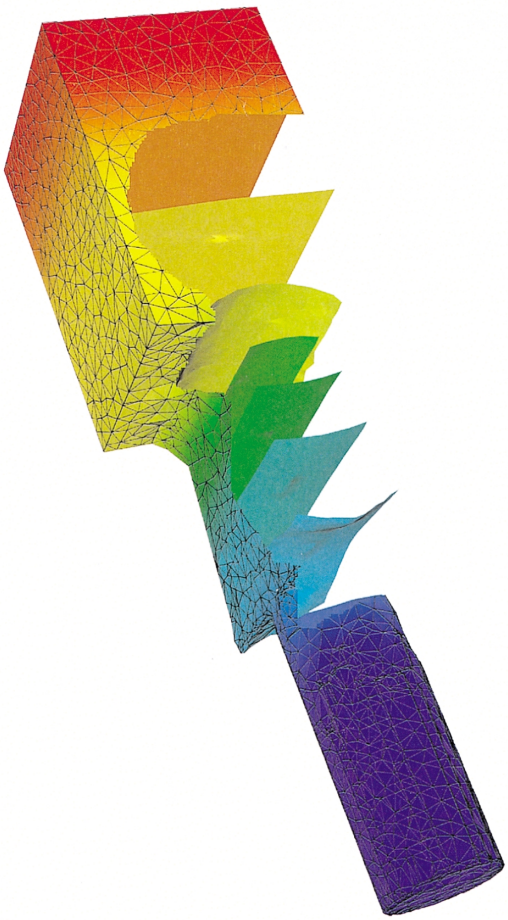


Fig. 11 Iso surfaces of the WF_6 concentration in a damascene structure, mesh with 18148 elements.

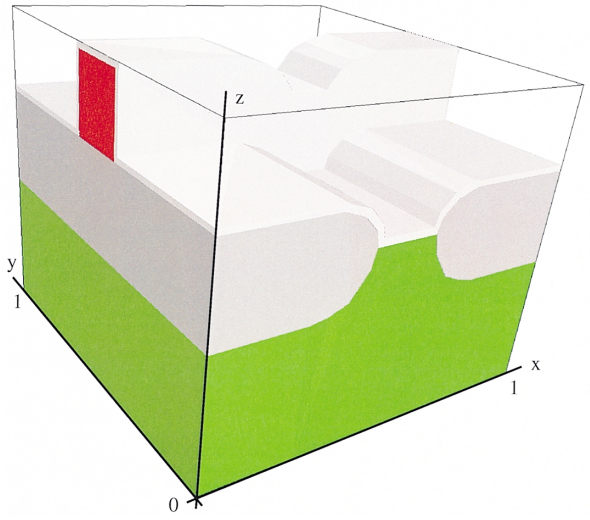


Fig. 12 NMOS Transistor with a thin oxide layer.

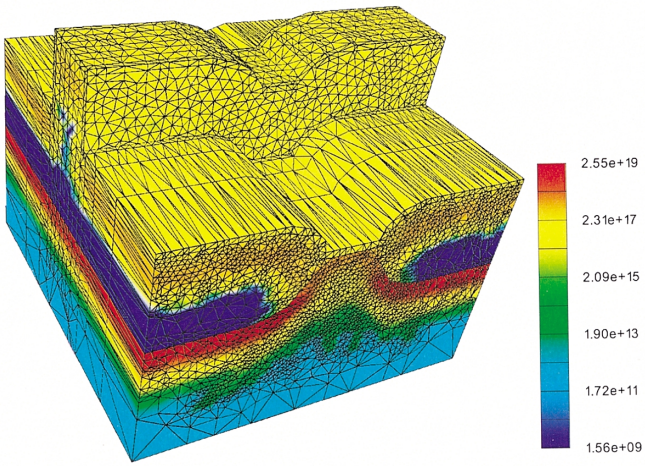


Fig. 13 Boron implantation profile and mesh with 134374 elements.

most existing methods. The difficulty lies in the grading of the mesh density. Isotropic grading results in a mesh with an extreme number of elements especially in three dimensions. Simplified meshing concepts like cartesian-, octree-, or layer-based methods are able to produce acceptable anisotropic grading as long as the e.g. thin layer is planar and parallel to the coordinate system. Their limitations with respect to complex structures are recently overcome with employing fully unstructured methods. The device depicted in Fig. 13 was meshed using the modified advancing front algorithm.

7. Conclusion

We briefly discussed common methodologies for mesh generation in TCAD. The importance of the Delaunay triangulation as a fast and well defined method to tessellate the domain was explained. We mentioned im-

portant properties related to Delaunay methods. Some advantages of the modified advancing front algorithm in comparison to the standard Watson algorithm [28] were given. We also presented a novel three-dimensional Delaunay refinement technique which is important for all boundary consistent Delaunay triangulation methods.

Future work will have to improve the handling of non-planar thin layers and of arbitrary anisotropic density requirements with fully unstructured mesh generation techniques.

Acknowledgement

Part of this work is supported by Christian Doppler Forschungsgesellschaft, Vienna, Austria, and by the European Community ESPRIT project 24038 PROMPT-II.

References

[1] G. Garretón, L. Villablanca, N. Strecker, and W. Fichtner, "A hybrid approach for building 2D and 3D conforming Delaunay meshes suitable for process and device simulation," in Simulation of Semiconductor Processes and Devices, eds. K. De Meyer and S. Biesemans, pp. 185–188, Springer, Wien, New York, 1998.

[2] V. Axelrad, "Grid quality and its influence on accuracy and convergence in device simulation," IEEE 0278-0070, 1998.

[3] V. Moroz, S. Motzny, and K. Lilja, "A boundary conforming mesh generation algorithm for simulation of devices with complex geometry," Proc. Simulation of Semiconductor Processes and Devices, pp.293–295, 1997.

[4] K. Lilja, "A 3D mesh generation method for the simulation of semiconductor processes and devices," Proc. MSM, Santa Clara, 1998.

[5] P. Fleischmann, R. Sabelka, A. Stach, R. Strasser, and S. Selberherr, "Grid generation for three-dimensional process and device simulation," Proc. Simulation of Semiconductor Processes and Devices, pp.161–166, 1996.

[6] N. Kotani, "TCAD in Selete," in Simulation of Semiconductor Processes and Devices, eds. K. De Meyer and S. Biesemans, pp.3–7, Springer, Wien, New York, 1998.

[7] R. Strasser and S. Selberherr, "Parallel and distributed TCAD simulations using dynamic load balancing," in Simulation of Semiconductor Processes and Devices, eds. K. De Meyer and S. Biesemans, pp.89–92, Springer, Wien, New York, 1998.

[8] G. Butlin and C. Stops, "CAD data repair," Proc. 5th International Meshing Roundtable, Sandia National Labs, pp.7–12, Pittsburgh, 1996.

[9] R.J. Donaghy, W. McCune, S.J. Bridgett, C.G. Armstrong, D.J. Robinson, and R.M. McKeag, "Dimensional reduction of analysis models," Proc. 5th International Meshing Roundtable, Sandia National Labs, pp.307–320, Pittsburgh, 1996.

[10] T. Syo, Y. Akiyama, S. Kumashiro, I. Yokota, and S. Asada, "A triangular mesh with the interface protection layer suitable for the diffusion simulation," Proc. Simulation of Semiconductor Processes and Devices, pp.173–174, 1996.

[11] M. Rottinger, "Generation of triangular grids in MINIMOS-NT," Annual Review, Institute for Microelectronics, Vienna, 1997.

[12] J.R. Shewchuk, "Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator," First Workshop on Applied Computational Geometry, ACM, pp.124–133, Philadelphia, 1996.

[13] R. Löhner and P. Parilch, "Three-dimensional grid generation by the advancing front method," Int. J. Numer. Meths. Fluids., vol.8, pp.1135–1149, 1988.

[14] P.J. Frey, H. Borouchaki, and P.L.George, "Delaunay Tetrahedralization using an advancing-front approach," Proc. 5th International Meshing Roundtable, Sandia National Labs, pp.31–43, Pittsburgh, 1996.

[15] M. Bern and D. Eppstein, "Mesh generation and optimal triangulation," in Computing in Euclidean Geometry, eds. F.K. Hwang and D.-Z. Du, pp.201–204, World Scientific, 1992.

[16] E. Schönhardt, "Über die Zerlegung von Dreieckspolyedern in Tetraeder," Mathematische Annalen, vol.98, pp.309–312, 1928.

[17] J.R. Shewchuk, "A condition guaranteeing the existence of higher-dimensional constrained Delaunay triangulations," Proc. 14th Annual Symposium on Computational Geometry, pp.76–85, 1998.

[18] J.R. Shewchuk, "Tetrahedral mesh generation by Delaunay refinement," Proc. 14th Annual Symposium on Computational Geometry, pp.86–95, 1998.

[19] G.L. Miller, D. Talmor, S. Teng, N. Walkington, and H. Wang, "Control volume meshes using sphere packing," Proc. 5th International Meshing Roundtable, Sandia National Labs, Pittsburgh, pp.47–62, 1996.

[20] Y. Kallinderis, A. Khawaja, and H. McMorris, "Hybrid prismatic/tetrahedral grids for turbomaschinery applications," Proc. 6th International Meshing Roundtable, Sandia National Labs, Park City, pp.21–31, 1997.

[21] K. Tanaka, A. Notsu, and H. Matsumoto, "A new approach to mesh generation for complex 3D semiconductor device structures," Proc. Simulation of Semiconductor Processes and Devices, pp.167–168, 1996.

[22] B. Joe, "Construction of three-dimensional improved-quality triangulations using local transformations," SIAM: J. Sci. Comput., vol.10, no.6, pp.1292–1307, 1995.

[23] D. Adalsteinson and J.A. Sethian, "A fast level set method for propagating interfaces," J. Comp. Phys., vol.118, pp.269–277, 1995.

[24] T. Chen, D.W. Yergeau, and R.W. Dutton, "A common mesh implementation for both static and moving boundary process simulations," in Simulation of Semiconductor Processes and Devices, eds. K. De Meyer and S. Biesemans, pp.101–104, Springer, Wien, New York, 1998.

[25] F.J. Bossen and P.S. Heckbert, "A pliant method for anisotropic mesh generation," Proc. 5th International Meshing Roundtable, Sandia National Labs, Pittsburgh, pp.63–74, 1996.

[26] T. Fang and L.A. Piegl, "Delaunay triangulation in three dimensions," IEEE Comput. Graphics & Appl., vol.15, no.3, pp.62–69, 1995.

[27] P. Fleischmann and S. Selberherr, "Three-dimensional Delaunay mesh generation using a modified advancing front approach," Proc. 6th International Meshing Roundtable, Sandia National Labs, Park City, pp.267–276, 1997.

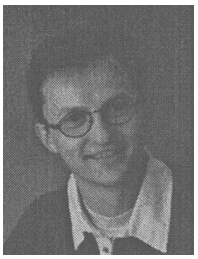
[28] D.F. Watson, "Computing the n-dimensional Delaunay tesselation with application to Voronoi polytopes," Computer Journal, vol.24, no.2, pp.167–172, 1981.

[29] F. Hermeline, "Triangulation automatique d'un polyèdre en dimension N," RAIRO Analyse Numérique, vol.16, no.3, pp.211–242, 1982.

[30] P. Cignoni, C. Montani, R. Perego, and R. Scopigno, "Parallel 3D Delaunay triangulation," in EUROGRAPHICS, vol.12, eds. R.J. Hubbold and R. Juan, p.C-129, Eurographics Association, Blackwell Publishers, 1993.

[31] P. Krysl and M. Ortiz, "Generation of tetrahedral finite element meshes: Variational Delaunay approach," Proc. 7th International Meshing Roundtable, Sandia National Labs, Dearborn, pp.272–284, 1998.
[32] P. Fleischmann and S. Selberherr, "Fully unstructured Delaunay mesh generation using a modified advancing front approach for applications in technology CAD," IEEE Trans.Semiconductor Technology Modeling & Simulation, http://www.ieee.org/journal/tcad/accepted/, 1997.
[33] L.P. Chew, "Guaranteed-quality triangular meshes," Tech. Rep. TR-89-983, Cornell University, 1989.
[34] L.P. Chew, "Guaranteed-quality Delaunay meshing in 3D," Proc. 13th Annual Symposium on Computational Geometry, ACM, pp.391–393, 1997.
[35] J. Ruppert, "A Delaunay refinement algorithm for quality 2-dimensional mesh generation," Journal of Algorithms, vol.18, pp.548–585, 1995.
[36] P. Fleischmann and S. Selberherr, "A new approach to fully unstructured three-dimensional Delaunay mesh generation with improved element quality," Proc. Simulation of Semiconductor Processes and Devices, pp.129–130, 1996.
[37] J.R. Shewchuk, "Adaptive precision floating-point arithmetic and fast robust geometric predicates," Discrete & Computational Geometry, vol.18, no.3, pp.305–363, 1997.
[38] R. Sabelka, R. Martins, and S. Selberherr, "Accurate layout-based interconnect analysis," in Simulation of Semiconductor Processes and Devices, eds. K. De Meyer and S. Biesemans pp.336–339, Springer, Wien, New York, 1998.
[39] M. Radi, E. Leitner, E. Hollensteiner, and S. Selberherr, "AMIGOS: Analytical model interface & general object-oriented solver," Proc. Simulation of Semiconductor Processes and Devices, pp.331–334, 1997.

Peter Fleischmann was born in Kabul, Afghanistan, in 1969. He studied electrical engineering at the Technical University of Vienna, where he received the degree of 'Diplomingenieur' in 1994. He joined the 'Institut für Mikroelektronik' in December 1994. In December 1997 he was with NEC in Sagamihara, Japan. He is currently working for his doctoral degree. His research interests include mesh generation as well as algorithms and data structures in computational geometry.



Wolfgang Pyka was born in Innsbruck, Austria, in 1970. He studied material science at the 'Montanuniversität Leoben', where he received the degree of 'Diplomingenieur' in June 1996. In December 1996 he joined the 'Institut für Mikroelektronik', where he is currently working for his doctoral degree. His work is focused on simulation and modeling of etching and deposition processes and on algorithms for topographic simulations.

Siegfried Selberherr was born in Klosterneuburg, Austria, in 1955. He received the degree of 'Diplomingenieur' in electrical engineering and the doctoral degree in technical sciences from the Technical University of Vienna in 1978 and 1981, respectively. Since that time he has been with the Technical University of Vienna as professor. Dr. Selberherr has been holding the 'venia docendi' on 'Computer-Aided Design' since 1984. He has been the head of the 'Institut für Mikroelektronik' since 1988. His current topics are modeling and simulation of problems for microelectronics engineering.