Generalization of modified octrees for geometric modeling

N. Hitschfeld

Dpto. Ciencias de la Computación, Univ. de Chile
Blanco Encalada 2120, Santiago, CHILE
E-mail: nancy@dcc.uchile.cl

Abstract. This paper discusses several aspects of modified octrees that can be generalized in order to obtain solid representations using less primitive elements than the traditional modified octree. The aspects under study include the use of elements of different type as internal nodes, a general refinement approach and cuboids, pyramids, prisms and tetrahedra as final elements. These concepts can be applied to the generation of mixed elements meshes for different applications. In particular, the new ideas are presented here for the generation of mixed element meshes that satisfy Delaunay condition. Examples are given to compare a new implementation with previous approaches.

1 Introduction

Since the last fifteen years, modified octrees have been used very often in geometric modeling and mesh generation[1, 2]. The modified octree approach works as follows: The 3-D domain is enclosed in a cube, whose octants are repeatedly refined at their edge midpoints until the boundary and internal quantities are sufficiently approximated. Elements with and without edge midpoints are partitioned into tetrahedra. In case of mesh generation, the final elements have to fulfill the requirements imposed by the numerical method.

The modified octree approach has the following drawbacks: (1) The use of only cubes as internal nodes while fitting the device geometry does not allow to stop the refinement as soon as the part of the object intersecting a cube can be represented with a pre-defined set of *well-shaped elements*. (2) The refinement into eight similar elements increases unnecessary the number of nodes for several applications. (3) In case of mesh generation, the most common is to use tetrahedra as final elements. Tetrahedra are required for the finite element method but if some problem can be solved using the *Box method*[3], several element types can be used.

This paper discusses several aspects in the generation of octrees and modified octrees that can be generalized in order to get a final domain representation that contains less basic elements than the former approaches: (1) The domain can be enclosed for a cuboid. A cuboid has rectangular faces. (2) The internal elements (nodes) can belong to a set of *well shaped elements*, such as pyramids, prisms and tetrahedra of rectangular basis, and cuboids. The set of elements that is called *well-shaped* depends on the application. This set has to be closed under the *refinement operator*, i.e, each element can be refined in such a way that all newly generated elements belong to this set. The trees that can handle different element types as internal nodes are called *mixed element trees.* (3) The refinement can be either bisection or what we have called *intersection* based approach. Using the bisection based approach the refinement is always made at the edge midpoints. Using *intersection* based approach the refinement is made at the most convenient edge point. The best point—the one whose associated refinement generates sons with the smallest aspect ratio —is chosen from the available Steiner points (points generated by the refinement of the edge neighbors) and intersection points (points generated by the intersection between the object geometry and the current element). (4) Internal elements can be divided into a different number of elements and into elements of different type. It depends on the type of the internal node and on the refinement direction. For example, if a refinement is required along one, two, or three coordinate axes, cubes, are subdivided into two halves, four quadrants, and eight octants, respectively. (5) The set of final elements are defined by the application. They can be of the same type of the ones used as internal nodes, or of other type. What we kept from the modified octree approach is that the refinement is parallel to the axes of the coordinate system.

2 Characterization of well-shaped meshes

This section introduces the concept of well-shaped mixed element meshes independent of the application.

2.1 Basic algorithm to generate a well-shaped representation

The discussion of this paper is focused on the algorithms that are extensions of modified octrees and generate solid representations refining coarse elements. Independent of the application, the following consecutive steps are used:

1. Generate a macro-mesh that fits the geometry of the modeled device exactly
2. Refine due to variations in some internal values and certain geometrical parameters
3. Generate a proper mesh for the current application
   - make the mesh 1-irregular
   - look for proper tessellations
4. Store the information required by the application

2.2 Macro-elements

A macro-mesh is composed by macro-elements. Macro-elements are used to fit the device geometry.

The following theorem characterizes the set of macro-elements used in the generation of well-shaped mixed element meshes.

Theorem 1. *Let P be a set of polyhedra. P leads to well-shaped meshes if each polyhedron $p \in P$*
(i) fulfills the restrictions imposed by the current application, and
(ii) can be refined in such a way that all newly generated polyhedra also belong to P (P is closed).

Proof: Condition (i) guarantees that the macro-elements fulfill the restrictions imposed by the current application. Condition (ii) guarantees that for each element generated through the refinement process it will be possible to fulfill condition (i).

2.3 Different element refinement approaches

The most common way of refinement is bisecting an element, i.e., each element edge is bisected. This method is easy to analyze and implement but it does not allow flexibility in choosing the most appropriate refinement point. In the following, the refinement approach that allows to chose the refinement point is called *intersection-based* approach. (The bisection-based approach is a particular case of the intersection-based approach.)

2.4 Elements with Steiner points

Irregular macro-elements are elements with edges split at least once. The point splitting an edge is called a *Steiner point*. Irregular elements appear between coarse and fine regions after the density requirements are satisfied. In order to complete the mesh using as few elements as possible, the tessellation of irregular elements is necessary. 1-irregular elements are elements with edges split at most once.

Definition 2. Let p be an irregular convex polyhedron. The tessellation t of p is well-shaped if and only if t satisfies the conditions of the current application.

2.5 Final elements

Final elements are the elements that compose the final mesh. In the current version of the algorithm, we are not considering that final elements will require further refinement. For this reason, the set of final elements can include more basic elements than the ones included in the set of macro-elements used to fit the device geometry.

The set of final elements are defined by the application. They can be of the same type of the ones used as internal nodes, or of other type.

3 Applications

The next sections show applications of mixed elements meshes. A complete description is given for mixed elements meshes that satisfy the Delaunay condition.

3.1 Delaunay meshes

Definition 3. A tessellation T of a set of points S is a *Delaunay tessellation* if there exists point-free circumsphere for each tessellation element.

We use the term *Delaunay tessellation* and not *Delaunay triangulation* because our meshes include other element types than tetrahedra.

Macro-elements

The following theorem characterizes the set macro-elements used in the generation of Delaunay mixed element meshes.

Theorem 4. *Let P be a set of polyhedra. P leads to Delaunay meshes if each polyhedron $p \in P$*
(i) has co-circular vertices. They define the point-free circumsphere, and
(ii) can be refined in such a way that all newly generated polyhedra also belong to P (P is closed).

Our set of macro-elements is composed of rectangular pyramids, rectangular prisms, bricks, rectangular tetrahedron and its complement inside a cuboid (Fig. 1). They are elements that satisfy Theorem 3 and can be properly refined as will be shown in the next section.

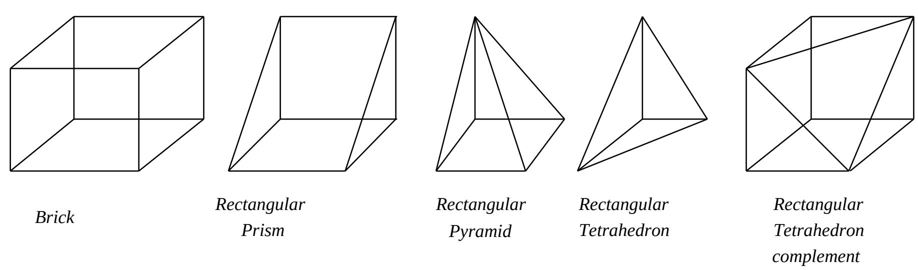


Fig. 1. Set of elements used to fit the device geometry

Element refinement

Since the bisection-based approach is a particular case of the intersection-based approach, the current section only shows the refinement for each macro-element under the general refinement approach.

Using an intersection based approach, elements are rarely split at the edge midpoints. The refinement location along the edges is determined using the current aspect ratio and the location of the current Steiner-points.

Refinement of bricks

Bricks can be split into two halves, four quarters or eight octants as before but edges are not necessary bisected. Figure 2 shows the different ways to split a brick using arbitrary refinement points. The only restriction is that parallel edges have to be split at the same relative position from their endpoints in order to generate bricks and not general polyhedra.

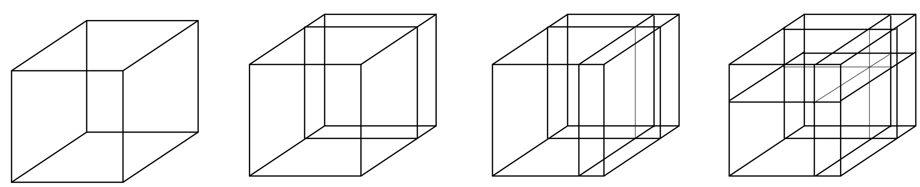


Fig. 2. Bricks refined in one, two, or three directions generate two, four, and eight bricks, respectively

Refinement of prisms

Rectangular prisms can be partitioned in the same way as before, but their triangular faces impose an additional restriction: the refinement point at the

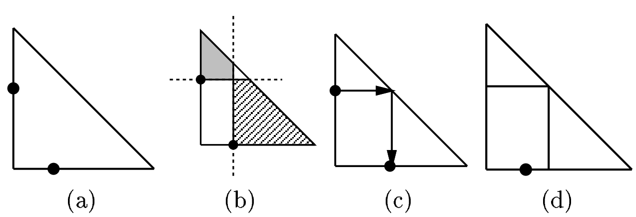


Fig. 3. Refinement of a triangular face

diagonal edge determines the refinement points for other edges as shown in Fig. 3. The triangular face (a) shows a triangular face with two possible refinement points. In the triangular face (b), a vertical and a horizontal dotted line illustrate how this face would be refined if both points were used: four new faces would be generated and the two shaded faces would not belong to any of the macro-elements. edge defines the location of the vertical line as shown in the triangular face (c). Triangular face (d) depicts the final status. Figure 4 shows the whole prism refinement depending on the direction(s) required.

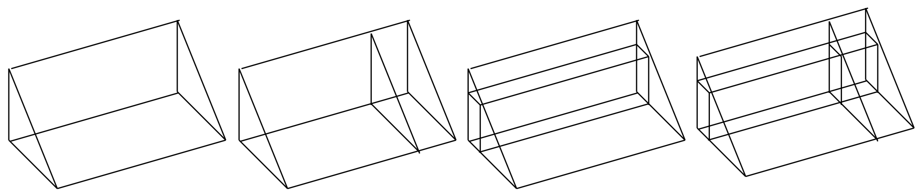


Fig. 4. Prism refinement in one, two, or three directions generates two prisms, one brick plus two prisms, and two bricks plus four prisms, respectively

Refinement of pyramids

The refinement of a rectangular pyramid in a general position is governed by the restrictions impose by its triangular faces. The intersection point that generates subelements with the smallest aspect ratio is chosen. Figure 5 shows its partition using the refinement point p. Pyramid (a) shows the main parameters: the point p, the edge e (where p is located), and the main diagonal d. Let f be the plane with normal vector in direction of e and passing through the point p. The point c corresponds to the intersection point between the plane f and the edge d. The coordinate values of c define the refinement points in the other directions (pyramid (b)). The result is shown in pyramid (c).

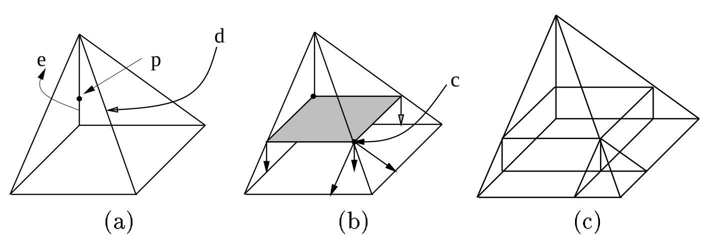


Fig. 5. Pyramid refinement in three directions generates one brick, two prisms and two pyramids

Refinement of the tetrahedron and its complement

Figure 6 shows the refinement of the rectangular tetrahedron and its complement. The rectangular tetrahedron is refined into three similar elements and a rectangular tetrahedron complement (left). The rectangular rectangular tetrahedron complement is refined into three similar elements, four bricks and one rectangular tetrahedron (right).

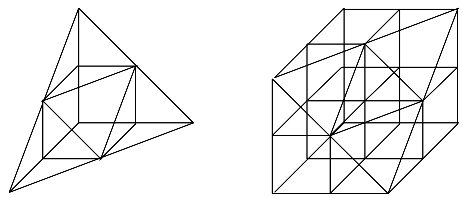


Fig. 6. Refinement of the tetrahedron and its complement

1-irregular elements

Irregular macro-elements appear after refining the grid elements according to given density requirements. Irregular macro-elements, i.e., elements with edges bisected more than once are made 1-irregular looking for well-shaped tessellations. The next definition characterizes such an element.

Definition 5. Let l be a 1-irregular macro-element. l is a well-shaped if no Voronoi point of l lies outside its convex hull (in this case, the 1-irregular macro-element itself).

The previous definition can be fulfilled by finding a Delaunay tessellation of the 1-irregular macro-element that satisfies the following theorem:

Theorem 6. *Let $S \subset \mathbb{R}^n$, $n \leq 3$ be a set of points, C the convex hull of S, and T a Delaunay tessellation of S. Then no Voronoi point of S lies outside C if and only if for each face f_{ijk} in 3-D of T on the surface of C, the circumsphere of f_{ijk} with the center in the middle of f_{ijk} is point-free.*

The proof of this theorem can be found in [4]

The final mesh is generated only after checking that each 1-irregular element fulfills Definition 5. Theorem 6 imposes the restrictions to the tessellation of each 1-irregular macro-element: the local tessellation must be Delaunay and there exist a point-free circumsphere for each face on the surface of a 1-irregular macro-element. The last restriction guarantees that exist a point-free circumsphere for neighboring elements.

Final elements

The current set of final elements is shown in Fig. 7. A final element is any one whose vertices are co-circular. This set of elements solves around 80% of the 1-irregular configurations for a cuboid [5].

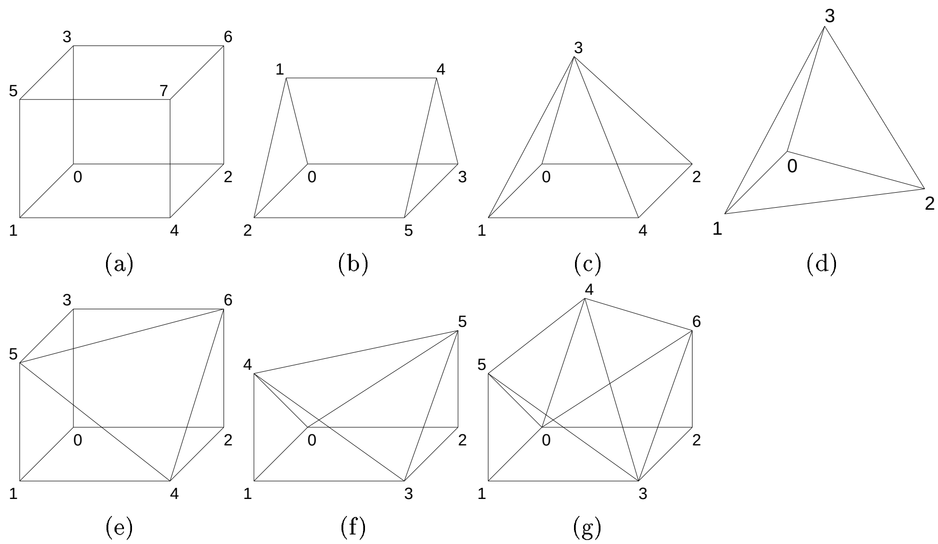


Fig. 7. Set of final elements

3.2 Meshes for Box-method

The meshes required by the control volume discretization method are a subset of Delaunay tessellations. Definition 7 characterizes these meshes:

Definition 7. A Delaunay tessellation of a set of points S is adequate for the control volume discretization method if the corresponding Voronoi diagram fulfills the following conditions: (a) No Voronoi point is outside the boundary of the tessellation (b) Each Voronoi face intersects its Delaunay edge

The previous Definition implies that elements lying at the boundary or material interfaces have additional restrictions. Compared to those inside the material, no Voronoi point (center of the element) must lie outside the faces at the border or at a material interface. The number of macro-elements is then reduced to rectangular prisms, rectangular pyramids and cuboids. They contain the center of the circumsphere that surrounds them. More details are given in in [5].

3.3 Solid modeling

Mixed element trees can be used as a solid representation model. It belongs to the methods based on cell decomposition. Depending on the kind of the application it is also possible to use incomplete mixed element meshes.

A complete comparison with the most used representation model would require the revision of the properties of the representation models.

4 Description of different implementations

4.1 Mixed element trees

The trees that can handled different element types as internal nodes are called *mixed element trees.*

Definition 8. Let T be a tree. T is a *mixed element tree* if
(i) each node (internal or leaf) is a polyhedron
(ii) each internal node is labeled with the axes across which the node is refined.

Theorem 9. *Let T be a mixed element tree. T leads to well-shaped meshes by construction if and only if*
(i) each internal node is one of the macro-elements described in the previous section,
(ii) each leaf is an irregular macro-element satisfying Definition 5 or a macro-element without Steiner points (regular element), and
(iii) each 1-irregular leaf can be tessellated into the final elements.

This representation allows the generation of well-shaped meshes by construction because it permits the implementation of the concepts introduced in the previous section in a natural way.

4.2 Mixed element trees using a bisection based approach: Ω_{mebi}

The main algorithm looks as follows:

1. Generate a mesh that fits the geometry of the modeled device exactly. This initial mesh consists of cuboids, rectangular prisms, rectangular pyramids, rectangular tetrahedra and its complement inside a cuboid. The mesh is handled as a forest where each element is the root of a tree.
2. Refine due to variations of some internal values and certain geometrical parameters. In order to fit physical and other geometrical parameters, an irregular mesh is generated by refining each element independently of the others.
3. Generate a proper Delaunay mesh. A finite element mesh is obtained after tessellating all the irregular elements into tetrahedra, pyramids, prisms and cuboids.

The most serious problem of Ω_{mebi} [6] originates from the algorithm that fits the original device geometry. This algorithm generates an initial (tensor product based) mesh that is a complete partition of the device (i.e. mesh elements have no Steiner points). A complete partition is required in order to be able to use a bisection based approach. During the generation of the initial mesh, small geometry features are propagated (by inserting planes) to the boundaries of the device. Therefore, the initial mesh contains a high number of unnecessary elements with a very bad aspect ratio. In addition, the repetitive generation of new points due to intersections between the inserted lines (planes in 3-D) and some boundary or material interfaces makes it impossible to fit several device geometries.

4.3 Mixed elements trees using an intersection based approach: Ω_{mein}

The same consecutive steps are used to generate a mesh using an *intersection*-based approach but each step is focused in a different way.

The geometry is fitted by refining the mesh elements at the best possible point. The best point—the one whose associated refinement generates sons with the smallest aspect ratio—is chosen from the available Steiner points and intersection points. Elements are bisected for generated sons with bad aspect ratio.

After the geometry is completely fitted, the irregular macro mesh is further refined until the density requirements are fulfilled. Elements are partitioned in the required direction according to the best located Steiner point.

The mesh is made 1-irregular before looking for proper tessellations. Subsequently, the algorithm checks the *splittable* condition, i.e., a condition that guarantees the existence of a proper tessellation. If an element is non splittable, proper points are inserted by looking inside the problematic element.

Once all elements are splittable, each local tessellation is computed using an algorithm to compute Delaunay tessellations inside 1-irregular macro-elements [5]. The set of final elements was shown in Fig. 7.

4.4 Comparison

The implementation of a general refinement approach is quite more difficult than the bisection based approach. The main difficulties are in:

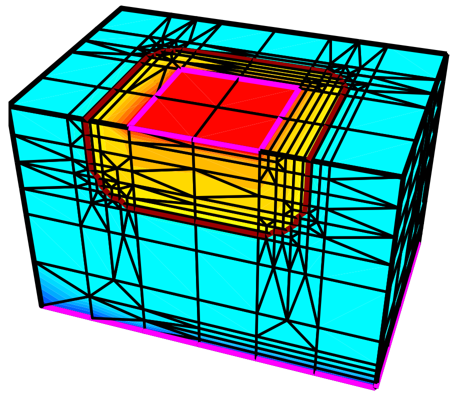
- The algorithms and data structures to consistently keep the geometrical information generated at each refinement step [7].
- The tessellation of elements with edge midpoints can be solved pattern-wise. The tessellation of elements with Steiner points at arbitrary position must be solved using an algorithm that recognize the basic elements [5].
- The propagation of points among the neighbors must be controlled very carefully.

5 Comparison using examples

The following examples are used to analyze empirically the influence of the aspects that have been generalized from modified octrees in the generation of mixed element meshes.

5.1 Complete meshes

Figure 8 shows two meshes for a diode; the left one was generated using Ω_{mebi}, and the right one was generated using Ω_{mein}. Ω_{mein} fulfilled the density requirements specified by the user using one half of the elements required by Ω_{mebi}. The exact number of elements of each type used in the final mesh is given in Table 5.1.



(a) (b)

Fig. 8. A final mesh for a diode (a) using Ω_{mebi}, (b) using Ω_{mein}

The main reasons for the strong reduction in the number of elements are: (1) Ω_{mein} uses a larger set of final elements than Ω_{mebi}, (2) Ω_{mein} uses an algorithm to compute the tessellation of 1-irregular configurations while Ω_{mebi} uses a template-based solution. The template-based solution only includes the most common 1-irregular configurations that can be tessellated using cuboids, prisms, pyramids and tetrahedra.

	Cuboid	Prism	Pyramid	Tetr.	Tetr. Compl.	Deformed Prism	DBC	Total
Ω_{mebi}	2628	1682	404	153	0	0	0	4867
Ω_{mein}	692	588	575	478	93	1	0	2427

5.2 Incomplete meshes

Figure 9 shows the same two views of the silicon part of a ECL bipolar transistor. The device geometry is fitted in Fig. 9(a) using Ω_{mebi} approach and in Fig. 9(b) using Ω_{mein}. The partial mesh shown in Fig. 9(b) strongly reduces the number of unnecessary mesh points and elements. In addition, the intersection-based approach avoids the propagation of mesh planes to the whole device allowing the fitting of more complicated geometries because small changes in the geometry can be fitted locally. The aspect ratio of the elements is controlled according to user parameters.

Empirically, the algorithm to fit the device geometry using an intersection-based approach is much faster. For example, Ω_{mein} is four times faster than Ω_{mebi} for the example shown in Fig. 9.

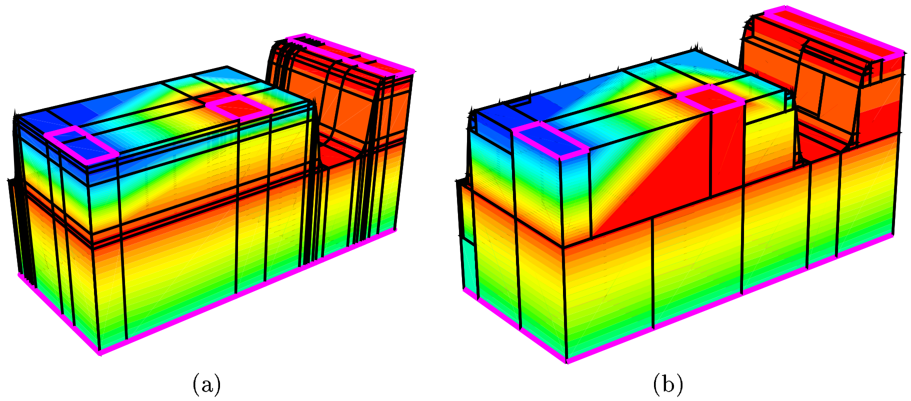


Fig. 9. Fitting the device geometry for the bipolar transistor (a) 2365 pts (b) 881 pts

Figure 10 shows the next step in the generation of a proper mesh. The density requirements specified by the user are fulfilled in both meshs. However, the partial mesh generated using the intersection-based approach (Fig. 10(b)) needs fewer mesh points and elements. The main reason is that a better macro-mesh is generated to fit the device geometry.

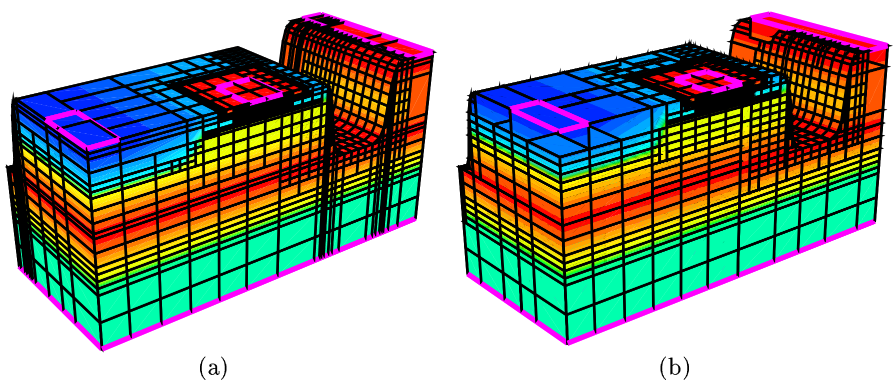


Fig. 10. Achieving the desired mesh density for the bipolar transistor (a) 10,376pts (b) 7,097pts

The intersection based approach shows clear advantages with respect to the bisection based in the first steps of the mesh generation process. Unfortunately, it is still not possible to compare the final mesh for this example. Previous tests have shown that the current implementation of the next steps will generate more elements than Ω_{mebi}. The problem is that the current strategy to stop the propagation of points among the neighbors while doing the mesh 1-irregular is too simple. It does not use the neighboring information. This is a very important aspect that is currently studied in order to reduce the number of elements in the final meshes in examples with complex geometries.

6 Conclusions

This paper presents several extensions to modified octrees that have being implemented using mixed element trees. Ω_{mebi} generates mixed elements meshes using bisection based approach. In comparison to modified octrees, it uses several elements types to represent the device geometry, elements are only refined in the required direction and uses cuboids, prisms, pyramids and tetrahedra as final elements. This allows to represent exactly more complex geometries and using less final than mesh generators based on the modified octree approach. Ω_{mein} introduces the refinement at any position, the design of an algorithm to automatically compute Delaunay tessellations for 1-irregular macro-elements, and a larger set of final elements than Ω_{mebi}. The test examples have shown that the fitting of the device geometry and the fulfilling of the density requirements are done more efficiently in Ω_{mein} than in Ω_{mebi}. The last steps of the mesh generation process need still more testing to give confident conclusions. For the objects with simple geometry like the diode, Ω_{mein} shows a better performance and reduces strongly the number of elements of the final mesh. For complicated geometries, like the bipolar transistor, new strategies have to be included to stop the propagation of points among the neighboring elements.

7 Current work

The generation of mixed element meshes using an intersection based approach are still under study. The complete mesh generation process is being improved in the following parts:

- the algorithm that controls the propagation of points among the neighbors
- the algorithm that generates automatically tessellations for 1-irregular macro-elements can include more final elements than the ones presented here.
- the robustness of the complete process.

The generation of more test examples is necessary to give complete comparison.

8 Acknowledgments

This work was partially supported by FONDECYT project No. 1940323 in 1996.

References

1. M. A. Yerry and S. Shephard, "Automatic Three-dimensional Mesh Generation by the Modified-Octree Technique," *Int. J. Numer. Methods Eng.*, vol. 20, pp. 1965–1990, 1984.
2. M. S. Shephard and M. K. Georges, "Automatic Three Dimesional Generation by the Finite Octree Technique," in *International Journal for Numerical Methods in Engineering*, vol. 32, pp. 709–749, 1991.
3. R. E. Bank, D. J. Rose, and W. Fichtner, "Numerical methods for semiconductor device simulation," *IEEE Trans. on El. Dev.*, vol. ED-30, no. 9, pp. 1031–1041, 1983.
4. N. Hitschfeld and W. Fichtner, "3-D Grid Generator for Semiconductor Devices using a fully flexible Refinement Approach," in *Int. Conf. on Semiconductor Devices and Processes, pub. in Simulation of Semiconductor Devices and Processes*, vol. 5, pp. 413–416, Springer-Verlag, 1993.
5. N. Hitschfeld and R. Farías, "1-irregular element tessellation in mixed element meshes for the control volume discretization method," in *Proceedings of the 5th International Meshing Roundtable*, pp. 195–204, Pittsburgh, Pennsylvania, U.S.A., October 10-11, 1996.
6. N. Hitschfeld, S. Müller, and W. Fichtner, "Generation of 3-d Delaunay Meshes for Complex Geometries using Iterative Refinement," *Ifip Transactions. Algorithms, Software, Architecture. Information Processing 92.*, vol. I, pp. 388–394, 1992.
7. N. Hitschfeld, "Algorithms and data structures for handling a very flexible refinement approach," in *Proceedings of the 4th International Meshing Roundtable*, pp. 265–276, Sandia National laboratories. Albuquerque, October 1995.

This article was processed using the LaTeX macro package with LLNCS style