Improving the quality of Delaunay triangulations for the control volume discretization method

Nancy Hitschfeld, María-Cecilia Rivara, Mauricio Palma

Department of Computer Science, University of Chile
Blanco Encalada 2120, Santiago, CHILE

Abstract

In this paper we present the results of applying a new algorithm to improve the quality of Delaunay triangulations for the numerical simulation of semiconductor devices using the control volume discretization method. The resulting triangulations are Delaunay triangulations, whose boundary triangles (triangles with at least one edge on the boundary or on a material interface) do not have obtuse angles opposite to any boundary or interface edges. In addition, the algorithm guarantees that the minimum and maximum angles of the triangles are bounded (minimum angle greater than or equal to 30° and maximum angle less than or equal to 120°), with the exception of a few triangles related with small angles of the boundary geometry.

1. Introduction

The numerical solution of partial differential equations (PDEs) is invaluable in design and optimization in many fields of engineering. The spatial discretization (mesh) of the structure to be simulated, i.e. its subdivision in cells, is key to the accuracy of the computed solution. An appropriate mesh should fulfill several requirements. First, it must provide a reasonable approximation of the geometry to be modeled, in particular of its boundary and internal material interfaces. Second, it is extremely important to accurately approximate all internal quantities relevant to the solution of the PDEs. Third, each cell must fulfill certain geometric constraints imposed by the numerical integration method: if the PDEs are solved with the finite element method, no angle must be smaller than some bound supplied *a priori*. However, if the equations are solved using a control volume discretization method(CVM)[1], the center of the smallest circumcircle that surrounds each boundary element must be inside the region of the element [2, 3]. For two dimensional geometries (2-D) this means that the angle opposite to a boundary edge must be a nonobtuse angle.

The CVM is very popular in the numerical simulation of semiconductor devices. In 2-D, both triangulations and mixed element meshes have been used. A review of previous work on this area can be found in [4, 5]. A more recently approach is the one presented in [6] based on the sphere packing technique [7].

This paper presents a new algorithm to improve the quality of Delaunay triangulations for the control volume discretization method which extends the Lepp-Delaunay algorithm introduced by Rivara in [8]. The algorithm not only eliminates obtuse angles opposite to boundary or interface edges but it guarantees that the minimum and maximum angles of the most of the triangles are bounded (minimum angle greater than or equal to 30° and maximum angle less than or equal to 120°). Then, this kind of meshes are also appropriate for the finite element method and for the combination of both methods.

2. New Algorithm

The construction of the good quality (constrained) Delaunay triangulation consists of: (a) The generation of an initial constrained Delaunay triangulation (which essentially uses the polygon vertices), and b) the use of an Lepp-Delaunay algorithm which improves the quality of the mesh so that the minimum angle is greater than or equal to 30° [8]. The basic Lepp-Delaunay improvement strategy uses the Longest-Edge Propagation Path of the target triangles (to be either refined and/or improved in the mesh) in order to decide which is the best point to be inserted, to produce a good-quality distribution of points. This strategy is repeatedly used until the target triangle is destroyed. An special boundary treatment technique is also used to avoid the insertion of undesirable points in the neighborhood of the boundary.

The step designed to eliminate boundary obtuse triangles (triangles with the obtuse angle opposite to a boundary or an interface edge) of polygonal regions considers three cases: (a) triangles with only one boundary edge which is opposite to an obtuse angle (1-edge boundary obtuse triangle), (b) triangles with two boundary edges and one of them opposite to an obtuse angle (2-edge boundary obtuse triangle), and (c) triangles with three boundary edges. The case (a) is solved by inserting the mid-point at the boundary edge. Since the obtuse angle is smaller than or equal to 120°, the insertion of only one point is required. Some diagonal swapping might be necessary. For the case (b) and since the insertion of a point on the longest-boundary edge keeps the obtuse angle in the new triangle with two boundary edges, we propose to insert two points so that the new boundary triangle with two boundary edges is isosceles. Since the boundary constrained angle (angle defined by the two boundary edges) can be smaller than 30°, the previous strategy can generate a new 1-edge boundary obtuse triangle with an obtuse angle greater than 120° . Then, several edge midpoint insertions might be required to eliminate this new boundary obtuse triangle. A triangle with three boundary edges (case (c)) is a particular case. Depending on the angles of the triangle, one or two isosceles triangles with two boundary edges are generated.

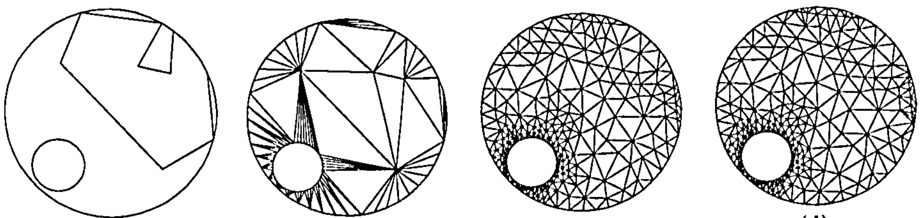
The previous process can also be applied to handle interface obtuse triangles. For obtuse triangles with one interface edge the same strategy as for case (a) is applied. For interface triangles with two or more interface edges adjacent to other triangles of the same type, we also propose the generation of isosceles triangles for the new triangles which keeps two interface edges. A complete description of the algorithm and its properties can be found in [9].

3. Results and conclusions

The following two examples illustrate the practical behavior of the algorithm. Table 1 presents a summary of the geometrical information of the meshes generated for the examples shown in Fig. 1 and Fig. 2. The left value of the columns with two values was computed considering only the triangles that are improved using the Lepp-algorithm, and the right one was computed considering the triangles that are

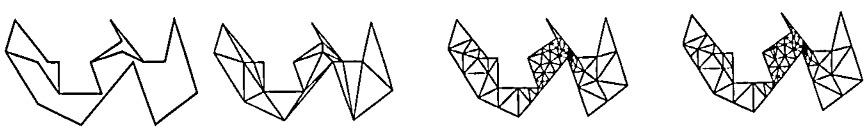
	Example 1			Example 2		
	Delaunay	Lepp-Del	Final mesh	Delaunay	Lepp-Del.	Final mesh
vertices	100	272	291	19	65	77
triangles	104	434	463	18	80	94
min. angle	0.84	30.06-14.99	12.40-14.99	5.19	30.34-10.30	17.91-10.30
aver. min. angle	15.73	43.15-16.87	42.39-16.87	25.91	43.34-20.52	41.88-20.52
max. angle	172.49	115.17-129.32	126.82-82.50	168.69	112.61-116.56	113.62-90.00
aver. max. angle	111.87	79.80-124.87	80.49-82.56	105.85	80.29-100.77	80.78-86.32
b-obtuse triangles	9	8	0	10	6	0

Table 1: Empirical information of the examples 1 and 2



(a) (b) (c) (d)

Figure 1: Example 1 (a) Geometry of the domain (b) Constrained boundary Delaunay triangulation (c) Triangulation after applying the Lepp-Delaunay algorithm (d) Triangulation after applying the strategy to eliminate boundary obtuse angles



(a) (b) (c) (d)

Figure 2: Example 2 (a) Geometry of the domain (b) Constrained boundary Delaunay triangulation (c) Triangulation after applying the Lepp-Delaunay algorithm (d) Triangulation after applying the strategy to eliminate boundary obtuse angles

boundary obtuse triangles with two boundary or interface edges whose constrained angle (angle defined by two boundary or interface edges) is less than 30° . In practice, the 2-dimensional triangulations obtained after applying the Lepp-Delaunay strategy are size-optimal [10]. The number of points inserted after the elimination of boundary obtuse angles is $O(N\_bot)$, where N_obt is the number of boundary obtuse triangles.

Finally, it should be pointed out that the final mesh (without boundary and interface obtuse triangles) is a Delaunay triangulation also for the triangles lying at the boundary and material interfaces (not a constrained Delaunay trianglulation).

4. Acknowledgment

The first author thanks to Norbert Strecker for a valuable interaction about the subject. This work was supported by Fondecyt project No 1960735 and Fondap AN-1 project.

References

[1] R. E. Bank, D. J. Rose, and W. Fichtner. Numerical methods for semiconductor device simulation. *IEEE Trans. on El. Dev.*, ED-30(9):1031–1041, 1983.

[2] M. R. Pinto. *Comprehensive Semiconductor Device Simulation for Silicon ULSI.* PhD thesis, Stanford University, 1990.

[3] N. Hitschfeld, P. Conti, and W. Fichtner. Mixed Elements Trees: A Generalization of Modified Octrees for the Generation of Meshes for the Simulation of Complex 3-D Semiconductor Devices. *IEEE Trans. on CAD/ICAS*, 12:1714–1725, November 1993.

[4] N. Hitschfeld. *Grid Generation for Three-dimensional Non-Rectangular Semiconductor Devices.* PhD thesis, ETH Zürich, Series in Microelectronics, Vol. 21, 1993. PhD thesis published by Hartung-Gorre Verlag, Konstanz, Germany.

[5] G. Garretón, L. Villablanca, N. Strecker, and W. Fichtner. A new approach for 2-d mesh generation for complex device structures. In *NUPAD V - Technical Digest*, Honolulu, USA, June 1994.

[6] Gary L. Miller, Dafna Talmor, Shang-Hua Teng, Noel Walkington, and Han Wang. Control volume meshes using sphere packing: generation, refinement and coarsening. In *Proceedings of the 5th International meshing Roundtable*, pages 47–61, Pittsburgh, Pennsylvania, 1996.

[7] Jim Ruppert Marshall Bern, Scott Mitchell. Linear nonobtusetriangulation of polygons. In *Proc. 10th annu. ACM sympos. computational geometry*, pages 231–241, St.Louis, 1994.

[8] M.C. Rivara. New longest-edge algorithms for the refinement and/or improvement of unstructured triangulations. *International journal for numerical methodsin Engineering*, 40:3313–3324, 1997.

[9] N. Hitschfeld and M.C. Rivara. Lepp-delaunay algorithm for quality non-obtuse boundary delaunay triangulations. *Department of Computer Science, U. de Chile*, 1998.

[10] J Ruppert. A delaunay refinement algorithm for quality 2-d mesh generation. *Journal of algorithms*, 18:548–585, 1995.