Edited by H. Ryssel, P. Pichler - September 1995

Mesh Generation for 3D Process Simulation and the Moving Boundary Problem

S. Bozek, B. Baccus, V. Senez and Z.Z. Wang

IEMN Département ISEN,
41, Boulevard Vauban, 59046 Lille Cedex, FRANCE

Abstract

This paper presents the concepts of a mesh generation technique for 3D process simulation involving structure deformation. One of the main problems is the displacement of boundaries leading to a (complete) remeshing of the structure, large cpu times and complexity of the algorithms. Our approach, based on Delaunay criterion, tetrahedral elements and triangular faces, allows local remeshings of the structure.

1. Introduction

Physical phenomena applied to a structure can be divided in two sets : the ones which do not modify the shape of the structure and those which induce moving boundaries. Dopant diffusion belongs to the first set while oxidation or silicidation belong to the second one. In numerical simulation, moving boundaries induce severe constraints on the mesh generation [1]. The very large number of nodes and elements needed for realistic 3D process simulations prevents the use of a strategy that resorts to complete remeshing at each time step, due to the cpu time. As a result, it is desirable a) to define the parts of the structure that really need to be meshed and b) to investigate algorithms based on local mesh updates. Let us focus our attention on the oxidation phase (fig. 1).

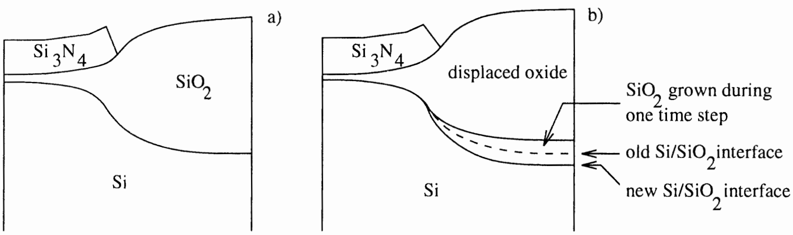

Figure 1: Schematic representation of the mechanical problem for local oxidation (LOCOS) with *a*) the initial structure, *b*) the structure after oxidation.

After one time step, the new oxide displaces the initial oxide layer upwards and the Si/SiO_2 interface downwards. If we assume that the displaced oxide is always well triangulated, the only part of the oxide to mesh is the new narrow band which appeared (fig. 1b). As may be seen on fig. 2, there is only a local zone of the silicon layer which is affected by Si/SiO_2 interface displacement. This thin band, in comparison with the Si layer, also needs to be remeshed.

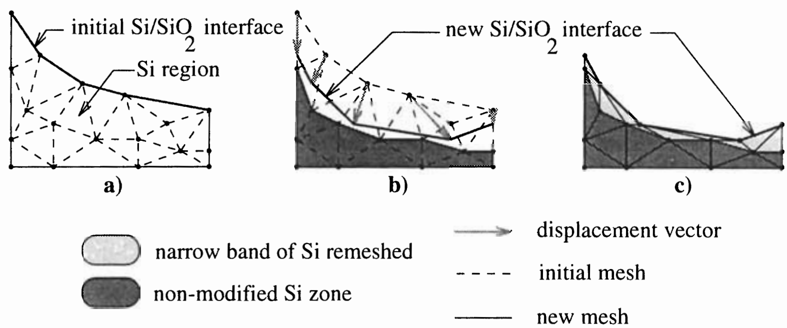

Figure 2: Displacement of the Si/SiO_2 interface and local remeshing of the Si layer.

As a result, this approach requires especially the meshing of narrow bands at each time step, with the following basic requirements : a) complete respect of the faces describing the hull of the region and b) no insertion or deletion of boundary nodes. While efficient methods have been reported for 3D device simulation (e.g. in [2]), they usually rely on octree, tensor-product or intersection-based algorithms, which are generaly not compatible with these requirements. In this paper, we present the basic concepts designed for this purpose and based on tetrahedral mesh generation.

2. Basic principles

In this second part, we briefly present the concepts that lead to the generation of a mesh from a set of given points [3, 4]. In our case, those points belongs to the hull of a local zone to mesh. First, the main stages are presented. Then, the local mesh modification produced by a node insertion is detailed.

Let P be the set of boundary points of R^3 and F the set of triangular faces describing the hull of the region. The algorithm can be divided in three steps. At first, a set T of tetrahedra is calculated, defining the convex hull of P. Next, the external elements are removed from T via F, in order to restore the original hull. At last, internal nodes are inserted into the mesh to fit physical needs and geometrical quality. Let's focus our attention on the first stage. Let T_j be the Delaunay-triangulated polytope (convex polyhedron) built with P_j, the j first nodes of P, and F_j the external faces of T_j. T_{j+1} is derivated from T_j by a local remeshing using a) p_{j+1}, b) the set of tetrahedra of T_j with their circumscribed sphere including p_{j+1}, and c) the elements of F_j which define a plane that strictly separates p_{j+1} and T_j (that means that p_{j+1} is on one side of the plane and all the P_i points, $i \leq j$, are on the other side or in this plane). More precisely, three cases can be considered (fig. 3). In the first case, the inserted node p is in the meshed polytope T_j (fig. 3a). Local remeshing is performed by deleting the tetrahedra with their circumscribed sphere including p, and by creating new elements with the faces of the hull defined by those elements and p. In the second case, p is out of any circumscribed sphere. Then, new tetrahedra are created with p and the separating external faces of the polytope T_j. In the third case, p is out of T_j but inside some circumscribed spheres. The mesh is updated with the non-separating faces of the hull defined by the tetrahedra with their circumscribed sphere that includes p, and the separating external faces of T_j.

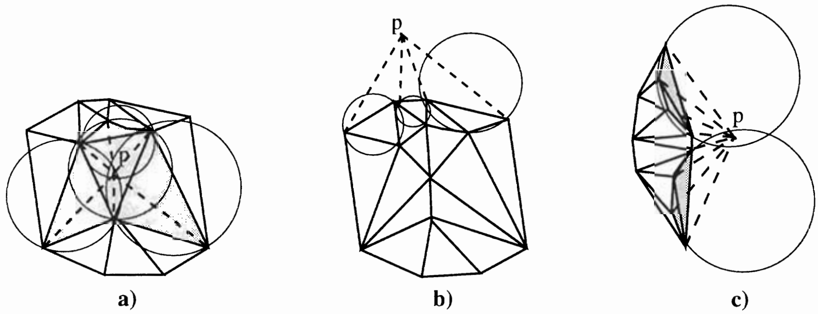

Figure 3: Different derivations from T_j to T_{j+1} using the inserted node p, after [3].

The internal node insertion is only a sub-case of the convex hull calculation, i.e. the case where p_{j+1} is inside T_j.

3. Implementation

The mesh generator has been implemented in C++ language. Although the C++ code is generaly less efficient at run-time than fortran code, the main advantages of an oriented-object approach consist in security of code and quickness to develop or modify part of program. Special attention has been devoted to the implementation of the algorithms, in particular the different cases of fig. 3. Indeed, it is very sensitive to rounding errors which can produce erroneous results.

4. Application

Hereafter, we give two first results issued from the above method. The figure 4 shows the mesh generated for a narrow band as can be obtained from the strategy reported in fig. 1-2. Despite the slow variation of the slopes, no points are inserted on the boundary, as would have generally been the case with strategies used for 3D device simulation. As can be seen, the external tetrahedra have been removed and no internal nodes have been introduced. The region contains 152 nodes, 309 tetrahedra and 300 faces.

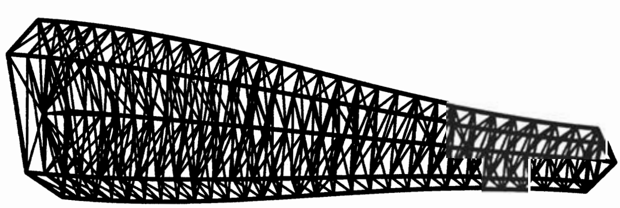

Figure 4: 3D mesh of a band as arising from the oxidation step.

The figure 5 is the result of the refinement of an initial cubic region with node insertion in accordance with the doping profile variation. The decrease in element quality is minored by the use of the local remeshing technique reported in fig. 3a. This example contains 33,793 tetrahedra, 5912 nodes and 1422 faces.

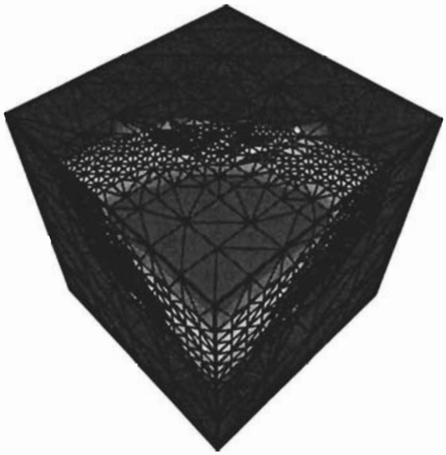

Figure 5: Arsenic contours and 3D meshing during a diffusion step, including a refinement procedure based on node insertion.

5. Conclusion

A strategy aimed in limiting the zones to be remeshed during the oxidation steps has been presented. A concept of mesh generation has been summarized and first results have been given.

6. Acknowledgements

This work is part of PROMPT (JESSI project BT8B) and was funded as ESPRIT project 8150. The authors would like to thanks ISE-AG for the use of PICASSO. Helpful discussions with P.L. George are also gratefully acknowledged.

References

[1] M.E. Law, "Grid generation for three-dimensional non-rectangular semiconductor devices", In *Proc. of SISDEP 5 Conf.*, pp. 1-8, 1993.

[2] N. Hitschfeld, *Challenges to Achieving Accurate Three-Dimensional Process Simulation* , PhD thesis, ETH Zurich, 1993. Published by Hartung-Gorre Verlag, Konstanz, Germany.

[3] P.L. George, F. Hermeline, "Delaunay's mesh of a convex polyhedron in dimension d. Application to arbitrary polyhedra", *Int. J. Num. Meth. Eng.*, vol. 33, pp. 975-995, 1992.

[4] P.L. George, F. Hecht, and E. Saltel, "Automatic 3D mesh generation with prescribed meshed boundaries", *IEEE Trans. Magn.*, vol. 26, no. 2, pp 771-774, 1990.