A METHOD FOR GENERATING STRUCTURALLY ALIGNED GRIDS USING A LEVEL SET APPROACH

ALIREZA SHEIKHOLESLAMI, CLEMENS HEITZINGER, AND SIEGFRIED SELBERHERR

Institute for Microelectronics, TU Wien
Gußhausstraße 27–29/360
Vienna, Austria
Email: sheikholeslami@iue.tuwien.ac.at

Abstract: We describe a technique to generate structurally aligned triangular grids. The main advantage of this method is the adjustable propagating speed of the front in different parts of the simulation domain in order to achieve different densities of triangles in each part of the simulation domain. This feature is usually needed in semiconductor device simulation. Other advantages of this technique are twofold: firstly, the grid can be very well adapted to the structures, and secondly, the grid elements fulfill desirable requirements like Delaunay triangulation and the minimum angle criterion. The technique is based on viewing the boundary of the simulation domain as a front which is propagated structurally at different speeds. A smooth propagation is achieved by the level set method by viewing the front as the zero level set of a higher dimensional function whose equation of motion is described by a partial differential equation.

KEYWORDS: Grid generation, Delaunay triangulation, level set method, semiconductor device simulation.

INTRODUCTION

We describe a method to generate structurally aligned triangular grids and illustrate it in two examples. We use the level set method to propagate the boundary of the simulation domain as a front by viewing it as the zero level set of a higher dimensional function with an adjustable speed depending on how fine the triangular grid should be. The equation of motion of this higher dimensional function is given by a partial differential equation, which is approximated by techniques borrowed from the numerical solution of hyperbolic conservation laws which guarantee that the correct entropy satisfying solution will be produced. The evolving front is thus a hypersurface, e.g., a curve in two space dimensions and a surface in three space dimensions. The resulting algorithm can be used to generate two and three dimensional grids around complex bodies containing sharp corners and significant variations in curvatures. We use this technique to generate different grids around a variety of shapes for different device structures.

The most important advantage of this method is the adjustable propagating speed of the front which provides an automatic way for generating grids with different densities of grid cells in particular parts of its domain. The history of two-dimensional process and device simulation leads to the observation that a stable triangulation engine is one of the most important prerequisites for simulation purposes. In the second part of our algorithm the final grid elements are produced using the TRIANGLE program [Fang and Piegl 1993, Shewchuk 1996]. Furthermore, thereby grids are very well adapted to the structures and are of high quality because we can enforce minimum angle criterion which guarantees that the triangles have angles which are equal or greater than a certain minimum angle and therefor we can well control the shape of the triangles.

Although the level set method has been used for generating structurally aligned grids [Sethian 1994], the method presented there cannot generate anisotropic grids and no condition concerning the quality of the grid, e.g., minimum angles, can be enforced.

The outline of this paper is as follows. Firstly, the basic ideas of the level set method are shortly explained. Secondly, the grid generation algorithm as a combination of the level set method and triangulation is presented. Thirdly, an algorithm for equalizing the length of segments is presented. Finally, examples for two simple initial structures and a real device structure are given.

Proceedings 17th European Simulation Multiconference
(c) SCS Europe BVBA, 2003 ISBN 3-936150-25-7

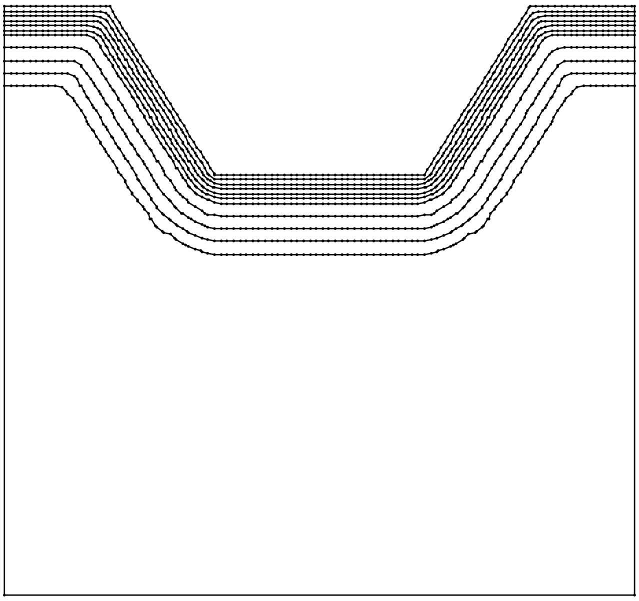

Figure 1: The extracted boundaries at 10 time steps.

THE LEVEL SET METHOD

The level set method [Sethian 1999] provides means for describing boundaries, i.e., curves, surfaces or hypersurfaces in arbitrary dimension, and their evolution in time, which is caused by forces or fluxes normal to the surface. The basic idea is to view the curve or surface in question at a certain time t as the zero level set (with respect to the space variables) of a certain function $u(t, \mathbf{x})$, the so called level set function. Thus the initial surface is the set $\{\mathbf{x} \mid u(0, \mathbf{x}) = 0\}$.

Each point on the surface is moved with a certain speed normal to the surface which determines the time evolution of the surface. The speed function $F(t, \mathbf{x})$ generally depends on the time and space variables and we assume for now that it is defined on the whole simulation domain and for the time interval considered.

The surface at a later time t_1 shall also be considered as the zero level set of the function $u(t, \mathbf{x})$, namely $\{\mathbf{x} \mid u(t_1, \mathbf{x}) = 0\}$. This leads to the level set equation

$$\begin{aligned} &u_t + F(t, \mathbf{x}) \|\nabla_{\mathbf{x}} u\| = 0, \\ &u(0, \mathbf{x}) \quad \text{given} \end{aligned}$$

in the unknown variable u, where $u(0, \mathbf{x})$ determines the initial surface.

Having solved this equation the zero level set of the solution is the seeked curve or surface at all later times.

Although in the numerical application the level set function is eventually calculated on a grid, the resolution achieved is in fact much higher than the resolution of the grid, and hence higher than the resolution achieved using a cellular format on a grid of same size.

In summary, first the initial level set grid is calculated as the signed distance function from a given initial surface. Then the speed function values on the whole grid are used to update the level set grid in a finite difference or finite element scheme. Usually the values of the speed function are not determined on the whole domain by the physical models and therefore have to extrapolated suitably from the values provided on the boundary, i.e., the zero level set. A fast and efficient level set algorithm combining extending the speed function and narrow banding was presented in [Heitzinger et al. 2002, Heitzinger and Selberherr 2002]. There a surface coarsening algorithm similar to the one used in this work was described as well.

GENERATING THE LEVEL SET STRUCTURED TRIANGULATED GRID

Our basic philosophy is to advance the front through the simulation domain using different speed functions. Throughout this section we restrict ourselves to two-dimensional grids. At discrete chosen time intervals, zero level set functions are constructed using a boundary extraction algorithm. In our example we have assumed a constant speed for the first 6 time steps and 8/3 times this speed for the next 4 time steps. This is shown in Fig. 1. We can see that the whole simulation domain is now divided into three different parts according to three different grid resolutions depending on the application. An arbitrary number of segments and speed functions can be used if desired.

Based on the edges constructed in the first step the grid generator TRIANGLE is used to obtain a Delaunay triangulation. In this example we demanded that the produced triangles have no angles smaller than 20 degrees. Requiring minimum angles is important since it enables a priori error estimates and estimates of the order of convergence [Knabner and Angermann 2000].

Fig. 2 shows the triangulated simulation domain. Because of different lengths of the segments which are obtained by each boundary extraction, we can clearly see that this triangulation contains triangles which are too small. An enlargement of this undesirable situation is shown in Fig. 3. We introduce an algorithm for overcoming this problem in the next section.

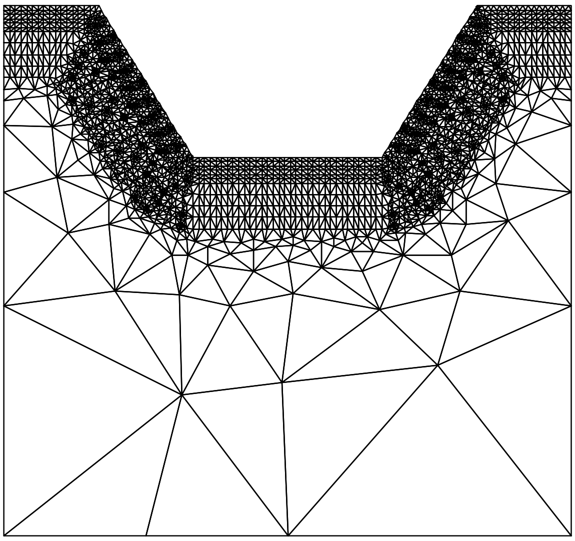

Figure 2: The triangulated grid without using the segment length equalizer.

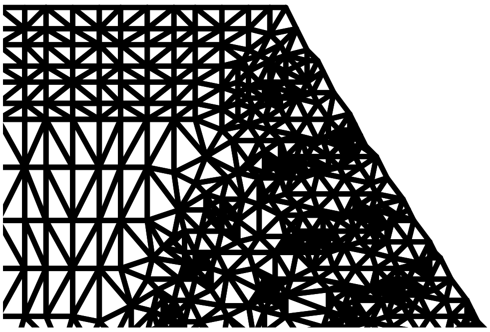

Figure 3: A part of the above grid on a larger scale.

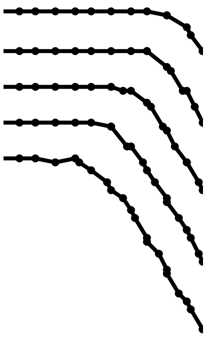

Figure 4: The last five steps of advancing the front is shown partly on a larger scale. The varying lengths of the segments are shown clearly.

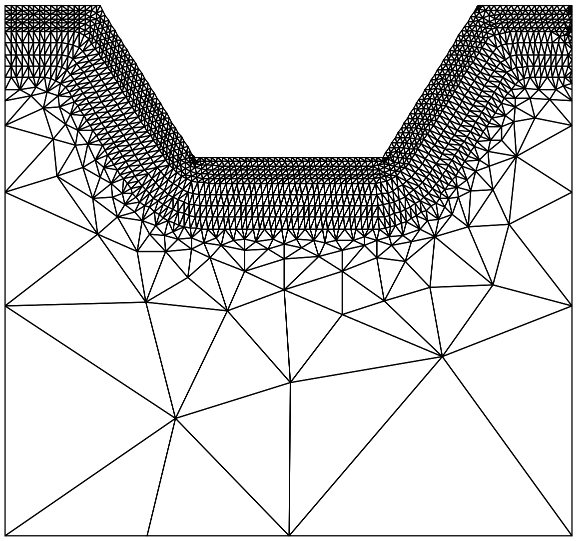

Figure 5: The triangulated grid is caused using the segment length equalizer.

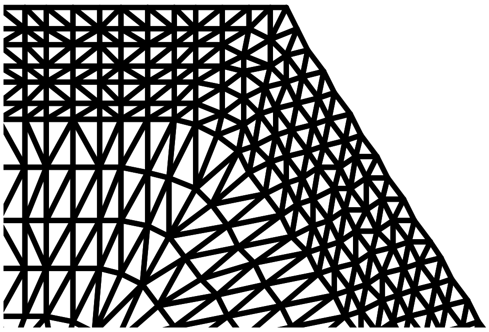

Figure 6: A part of the above grid on a larger scale.

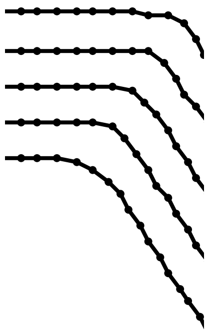

Figure 7: The last five steps of advancing the front is shown partly on a larger scale after equalizing the lengths of the segments. The length of the segments are not more different.

THE SEGMENT LENGTH EQUALIZER

To find the origin of this problem we briefly describe the boundary extraction algorithm which uses an interpolation method to find the points of the boundary and represents these as a list of segments with different lengths. Fig. 4 shows a part of the last five steps of advancing the front on a larger scale to show more clearly the varying lengths of the segments. The segments may become arbitrarily small and are the cause of the areas of dense triangles. To overcome this problem we need to ensure that all segments of the boundary have about equal lengths.

We start the algorithm by choosing a certain common length d for all segments. In our example we chose the minimum value of the vertical or horizontal distance between the points of our original rectangular grid which is used in the level set step. The first point of the extracted boundary stays without any changes but to find the second point we have to discern two cases. The first one is that the distance between the second and first point of the originally extracted boundary is equal or greater than our d and in the second one this distance is smaller than d. In the first case we compute the second point of the new boundary in this manner that we get a point which fulfills two restrictions: first, the caused segment must be along the first segment of the originally extracted boundary and second, the length of the new segment must be equal to d. In this case the new segment is a part of the old segment but the length of the new segment is equal or smaller than the old one. In the second case we compute the second point of the new boundary along the next segment of the origin boundary and like the first case fulfilling the length requirement. In this case the new segment is parallel to the second segment of the origin boundary and the length of the new segment is greater than the old one. These steps are iterated until we reach the boundary of the domain. Fig. 7 and Fig. 5 show the resulting segments with the enlargement and triangulated grid after equalizing the lengths of the segments. Furthermore in Fig. 6 a part of Fig. 5 is shown on a larger scale. In Fig. 8, Fig. 9 and Fig. 10 we show a simulation domain with a rectangular advancing front as another example and the resulting grid also with the enlargement.

GRID GENERATION FOR A REAL DEVICE STRUCTURE

Fig. 11 shows the device structure of a trench gate UMOS transistor. This device is useful for power switching at high voltages [Bulucea and Rossen 1991, Shenai 1992, Dharmawardana and Amaratunga 1998]. Trench gate UMOS transistors also provide advantages because of their geometric layout, i.e., because their inversion and accumulation channel regions are perpendicular to the wafer surface. Hence they enable to maximize the ratio of cell perimeter to area and thus increase packing density. An analytical model for a typical trench gate UMOS transistor is given in [Dharmawardana and Amaratunga 2000].

The model is derived using the charge control analysis of the channel and drain drift regions and gradual channel approximation is assumed to be valid in modeling the channel region. The shape of the different junctions is obtained by the doping concentration profile which is modeled with a Gaussian distribution.

For the grid generation we used four boundaries which follow the three junctions. At the $n^+ - p$ junction we used three boundaries in each direction of the initial boundary which follow the junction with a distance of 0.02μm between any two adjacent boundaries.

At the $p - n$ junction we used one boundary above and below the initial boundary and a distance of 0.02μm. At the $n - n^+$ junction in the lower part of the device we took into account two boundaries with a distance of 0.5μm going downwards from the initial boundary following the junction. For the last prescribed edges we started at the tight hand side of the p region and moved to the left using three boundaries at a distance of 0.005 μm.

Finally, we applied the TRIANGLE program requiring a minimum angle of 25^o with the prescribed edges as input. The grid produced is shown in Fig. 12, and it resolves very finely the junction areas as demanded.

CONCLUSION

A technique for generating structurally aligned triangulated grids using the level set method was described and implemented in two dimensions. In contrast to previously generated structurally aligned grids based on the level set method [Sethian 1994] the anisotropy of the grids and their quality can be controlled. The simulation domain can be divided into parts with different resolutions using adjustable speeds for advancing the front through the simulation domain with level set method. This adjustable grid resolution is essential in semiconductor device simulation where high resolutions are required in certain parts of the simulation domain. Furthermore the grid can very well adapted to different structures. Finally enforcing the minimum angle criterion is important for the numerical behavior of the subsequent finite element calculations and ensures high quality grids. At the same time, the diameter of the triangles may vary over several orders of magnitude within one simulation domain (cf. Fig. 5, Fig. 9, and Fig. 12). Our technique enables to produce triangulated grids for each form of semiconductor device structure with demanded resolution at different junctions (cf. Fig. 12).

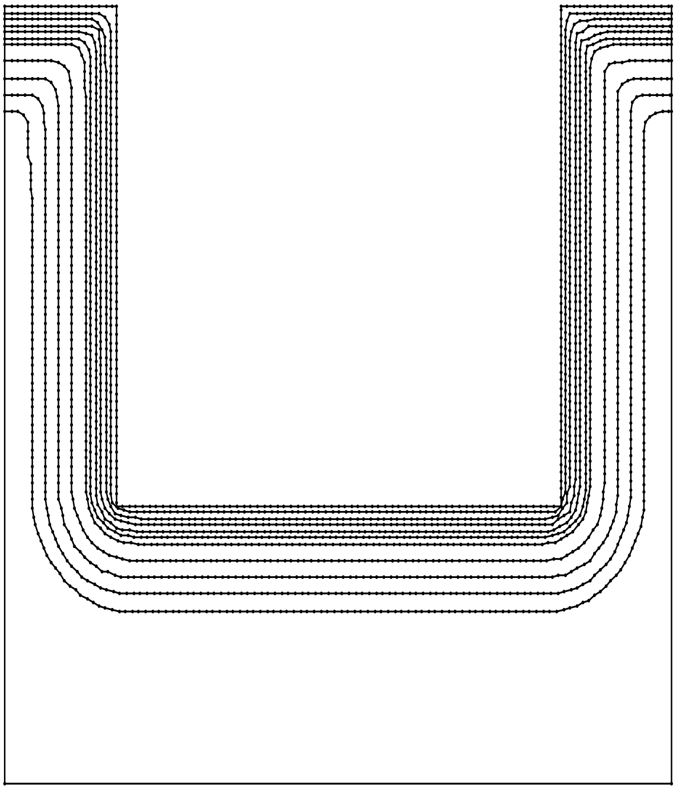

Figure 8: The advancing rectangular front after 10 time steps. As same as Fig. 5 the ratio of the speed in the first 6 steps to the last 4 steps is 3/8.

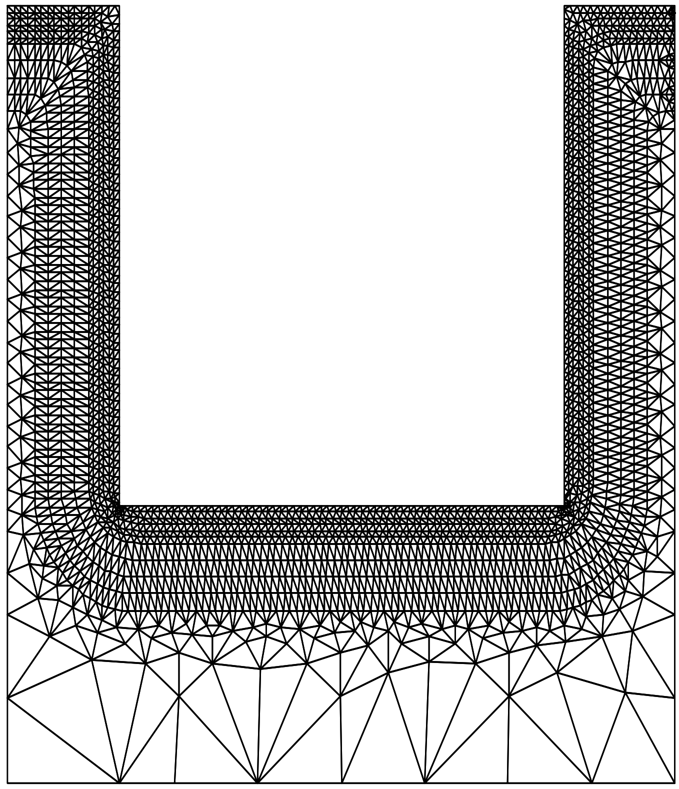

Figure 9: The triangulated grid of simulation domain in Fig. 8.

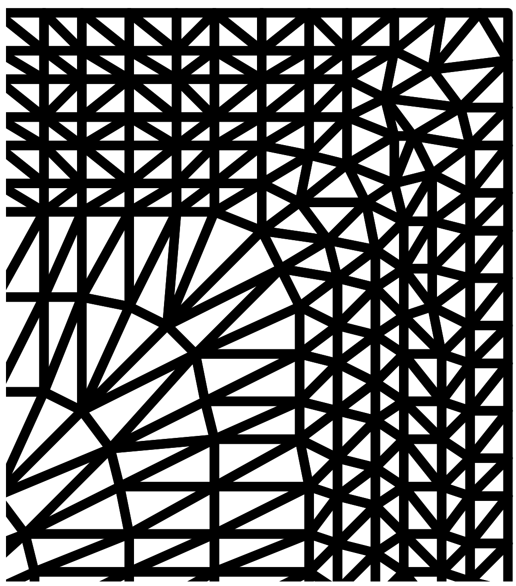

Figure 10: Fig. 9 is shown partly on a larger scale.

REFERENCES

Bulucea, C. and R. Rossen, 1991, "Trench DMOS Transistor Technology for High Current (100A Range) Switching". *Solid-State Electron.*, 34(5):493–507.

Dharmawardana, K. and G. Amaratunga, June 1998, "Analytical Model for High Current Density Trench Gate MOSFET". In *Proc. of the 10th International Symposium on Power Semiconductor Devices and ICs (ISPSD 1998)*, pages 351–354, Kyoto, Japan.

Dharmawardana, K. and G. Amaratunga, December 2000, "Modeling of High Current Density Trench Gate MOSFET". *IEEE Trans.Electron Devices*, 47(12):2420–2428.

Fang, T.P. and A. Piegl, 1993, "Delaunay Triangulation Using a Uniform Grid". *IEEE Computer Graphics and Applications*, pages 36–46.

Heitzinger, C.; J. Fugger; O. Häberlen; and S. Selberherr, September 2002, "On Increasing the Accuracy of Simulations of Deposition and Etching Processing Using Radiosity and the Level Set Method". In *European Solid-State Device Research Conference (ESSDERC 2002)*, pages 347–350, Florence, Italy.

Heitzinger, C. and S. Selberherr, June 2002, "On the Topography Simulation of Memory Cell Trenches for Semiconductor Manu-

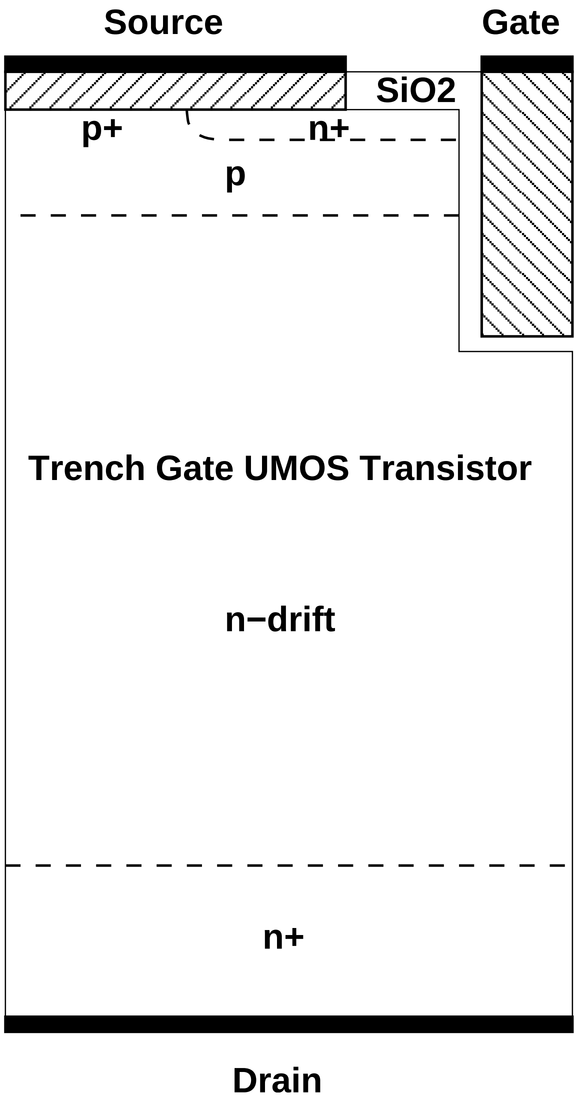

Figure 11: Structure of TMOSFET. The half cell pitch of the device is 2.5μm and its n drift length is about 9.5μm.

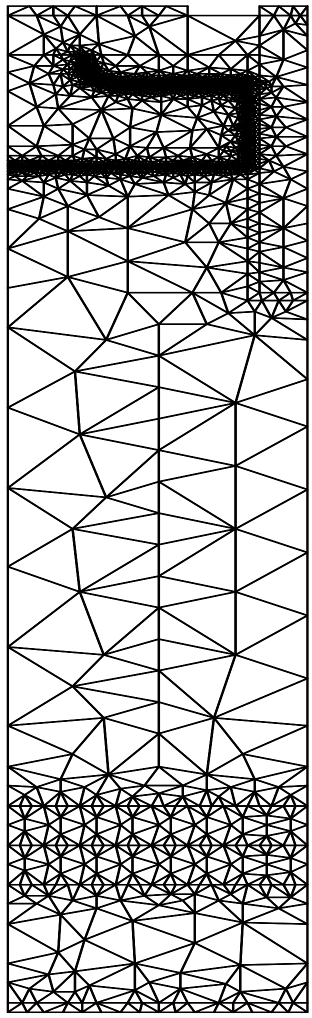

Figure 12: The grid generated for the device in Fig. 11.

facturing Deposition Processes Using the Level Set Method". In *16th European Simulation Multiconference (ESM 2002): Modelling and Simulation*, pages 653–660, Darmstadt, Germany.

Knabner, P. and L. Angermann, 2000, *Numerik partieller Differentialgleichungen*. Springer, Berlin.

Sethian, J.A., 1994, "Curvature Flow and Entropy Conditions Applied to Grid Generation". *J.Comput.Phys.*, pages 440–454.

Sethian, J.A., 1999, *Level Set Methods and Fast Marching Methods*. Cambridge University Press, Cambridge.

Shenai, K., 1992, "Optimized Trench MOSFET Technologies for Power Devices". *IEEE Trans.Electron Devices*, 39(6):1435–1443.

Shewchuk, J.R., May 1996, "Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator". *First Workshop on Applied Computational Geometry (Philadelphia, Pennsylvania)*, pages 124–133. http://www-2.cs.cmu.edu/ quake/tripaper/triangle0.html.

AUTHOR BIOGRAPHY

AlIREZA SHEIKHOLESLAMI was born in Babol, Iran, in 1971. He studied electrical engineering at the University of Science and Technology in Tehran and "Technische Universität Wien", where he received the degree of "Diplom-Ingenieur" in 2002. He joined the "Institut für Mikroelektronik" in April 2002, where he is currently working on his doctoral degree. His scientific interest is focused on process simulation for semiconductor manufacturing.