Mesh Generation Using Dynamic Sizing Functions

Florian Rudolf, Yannick Wimmer, Josef Weinbub, Karl Rupp, Siegfried Selberherr
Institute for Microelectronics, Technische Universität Wien
rudolf@iue.tuwien.ac.at, wimmer@iue.tuwien.ac.at,
weinbub@iue.tuwien.ac.at, rupp@iue.tuwien.ac.at,
selberherr@iue.tuwien.ac.at

Abstract

Assuming elements have a proper shape, the mesh element size is the primary parameter for adjusting the quality of the solution of most numerical simulation problems [1]. Smaller elements usually indicate a better resolution but also result in a higher number of elements, affecting the runtime performance of the simulation. Therefore, a good balance between the numerical quality and the runtime performance has to be found. In many simulation scenarios, a-priori information, like boundary conditions or non-uniform material properties, is available which can be used to optimize the mesh regarding the element size. Typically, the desired element size can be specified by global sizes for all elements of a (possibly partitioned) mesh or by scalar fields. While global sizes lack flexibility, scalar fields are difficult to treat due to their non-trivial impact, in particular, when using multiple scalar fields simultaneously. Additionally, some software libraries, like Triangle, support a callback mechanism for externally providing element size criteria in order to steer the mesh generation process. However, this option typically requires recompilation with specific compilation flags and the actual implementation of the element size function. We cope with this problem by extending our meshing software ViennaMesh [2] with a dynamic framework for defining the size of mesh elements with an XML-based configuration. The framework automatically manages the computation of the local element sizes and provides interfaces to mesh generation backends, like Triangle, Tetgen, CGAL, or Netgen. Arbitrary element size functions can be composed from scalar fields, arithmetic operations, and geometric predicates such as the local feature size [3]. We demonstrate the practicability of our approach by presenting a typical mesh generation workflow for microelectronic devices which require heterogeneous element sizes due to very thin layers and non-uniform material properties in the interior of the mesh. The resulting meshes are processed with the spherical harmonics simulator ViennaSHE [4], allowing to evaluate the mesh quality via, for instance, the convergence behavior.

References

1. S. Cheng et al. Delaunay Mesh Generation. CRC Press, 2013.
2. F. Rudolf et al. ViennaMesh. http://viennamesh.sourceforge.net/.
3. J. Ruppert. A Delaunay Refinement Algorithm for Quality 2-Dimensional Mesh Generation. J. Alg., 18(3), 1995.
4. K. Rupp et al. On the Feasibility of Spherical Harmonics Expansions of the Boltzmann Transport Equation for Three-Dimensional Device Geometries. In Proc. of IEEE IEDM, 2011.