Commun. Comput. Phys.
doi: 10.4208/cicp.2014.m414

Vol. **17**, No. 5, pp. 1246-1270
May 2015

A Fast and Rigorously Parallel Surface Voxelization Technique for GPU-Accelerated CFD Simulations

C. F. Janßen[1,*], N. Koliha[1] and T. Rung[1]

[1] *Institute for Fluid Dynamics and Ship Theory, Hamburg University of Technology, Schwarzenbergstraße 95 C, 21073 Hamburg, Germany.*

Received 16 January 2014; Accepted (in revised version) 9 October 2014

Abstract. This paper presents a fast surface voxelization technique for the mapping of tessellated triangular surface meshes to uniform and structured grids that provide a basis for CFD simulations with the lattice Boltzmann method (LBM). The core algorithm is optimized for massively parallel execution on graphics processing units (GPUs) and is based on a unique dissection of the inner body shell. This unique definition necessitates a topology based neighbor search as a preprocessing step, but also enables parallel implementation. More specifically, normal vectors of adjacent triangular tessellations are used to construct half-angles that clearly separate the per-triangle regions. For each triangle, the grid nodes inside the axis-aligned bounding box (AABB) are tested for their distance to the triangle in question and for certain well-defined relative angles. The performance of the presented grid generation procedure is superior to the performance of the GPU-accelerated flow field computations per time step which allows efficient fluid-structure interaction simulations, without noticeable performance loss due to the dynamic grid update.

AMS subject classifications: 51P05, 65Z05, 68W10

Key words: Fast Grid Generation, Parallel Cartesian Mapping, Voxelization, GPU, Lattice Boltzmann, ELBE.

1 Introduction

CFD methods rely on the discretization of the continuous governing equations into discrete, finite approximations, on either Lagrangian grids, Eulerian grids, or in meshless formulations. Lattice Boltzmann methods, as addressed in the present publication, discretize the governing equations on an equidistant Eulerian grid. Grid points of a computational domain used for Lattice Boltzmann CFD implementations can be of a variety of

*Corresponding author. *Email addresses:* `christian.janssen@tuhh.de` (C. Janßen), `nils.koliha@gmail.com` (N. Koliha), `thomas.rung@tuhh.de` (T. Rung)

different kinds. Those points representing a fluid particle ensemble are considered *fluid particles*, whereas other points can represent a slip wall, a no-slip wall, an inflow or even moving velocity boundaries. Since the grid itself is static and does not change over time, it is - apart from proper dynamic boundary conditions at the body surface - the changing character of the points through which a moving body's motion is manifested. Fig. 1 shows fluid particles as black circles and solid body particles as red dots. Black circles overlaid with red dots signify solid body particles from previous time steps, where a lighter shade of red indicates solid body particles from iterations further back in time.

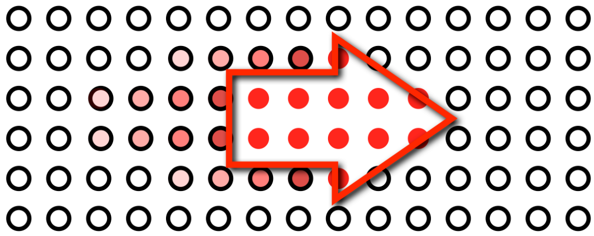

Figure 1: At each time step a moving body (red arrow) has to be mapped to the Cartesian grid. Hues of red indicate the changing grid node domain occupied by the body over time.

The grid generation algorithm that is presented in this paper is designed to be integrated seamlessly into a GPU-based CFD solver, the efficient Lattice Boltzmann environment ELBE [1]. The ELBE solver solves for three-dimensional turbulent free surface flows, including effects of viscosity and turbulent dissipation. Interactions of fluid and structure are considered as well. In transient CFD simulations with fluid-structure interactions, the solid body points have to be updated at each time step. The GPU-based implementation of ELBE allows for very competitive simulation times. Hence, the performance of the grid update algorithm and its implementation is crucial for successful simulations of challenging FSI problems. The main goal of this contribution is to develop a grid generation algorithm, that is

- Real-time capable — to rapidly voxelize the surface of a geometric object into a surface voxel representation in (or near) real time to be included in near-real time CFD simulations.
- Error-free and robust — to minimize the errors and the number of misidentified voxels, a source of potential instabilities in the numerical simulation.
- Efficient and extensible — to be convenient to integrate into a wide range of other CFD applications, e.g. SPH solvers.

After a brief literature review in Section 2, the following sections give further details of the necessary surface mesh preparations, details of the concrete algorithm and a detailed analysis of the performance of the algorithm, including numerical experiments.