Research Note
23rd International Meshing Roundtable (IMR23)

Periodic meshes for the CGAL library

Aymeric Pellé*, Monique Teillaud

INRIA Sophia Antipolis - Méditerranée, France

Abstract

This work is motivated by the need for a 3D periodic mesh generator in various fields, such as material engineering or modeling of nano-structures. In this paper, we present a software package answering this need, and which will be made publicly available in the open source CGAL library.

Keywords: periodic mesh, flat torus, Delaunay refinement, software

1. Introduction

CGAL, the Computational Geometry Algorithms Library (`www.cgal.org`), is an open source software library offering robust and efficient implementations of various geometric algorithms. It has a number of industrial and academic users in a wide range of fields, like molecular biology, medical modeling, astrophysics, particle physics, geographic information systems, etc.

The library is a collection of C++ software packages with an homogeneous API. Its sticks to the generic programming paradigm through templates, with a style close to the STL programming style [1]. Robustness is ensured by following the exact geometric computation paradigm, in which decisions are made on the basis of predicate evaluations that are guaranteed to be exact. Combining exact predicates with filtering techniques allows to gather efficiency and exactness.

CGAL offers in particular a package for volume mesh generation in $\mathbb{R}^3$, based on Delaunay refinement [2, 3]; it uses the 3D triangulation package [4]. While that package fulfills the needs of users for many applications, periodic meshes are needed for other fields, such as material engineering and modeling of nano-structures, as confirmed during several multidisciplinary meetings [5–7]. To the best of our knowledge, no publicly accessible software is answering that need.

A package for computing periodic Delaunay triangulations in $\mathbb{R}^3$ was introduced in 2009 in CGAL 3.5 [8]. The software was demonstrated by a video [9] and has already been used by researchers in other research fields, e.g. condensed matter physics [10] and astrophysics [11, 12].

* Corresponding author
E-mail address: Aymeric.Pelle@inria.fr, Monique.Teillaud@inria.fr

After quickly reviewing the two CGAL 3D mesh generation and 3D periodic triangulation packages, we present a new software for 3D periodic mesh generation. We show a sample of results it has produced, both on mathematical surfaces and on real world data. The software will be submitted to the CGAL Editorial Board for integration into a future release of the library, under a GPL licence.

2. CGAL 3D mesh generation

The mesh generation package of CGAL [3] allows users to generate isotropic tetrahedral meshes of 3D bounded domains. Fig. 1 schematizes its design and the main steps of the computation.

The user defines the domain through an oracle, providing the mesh generator with predicates and constructions that will allow it to query the domain. The domain may consist of multiple components, it may also be subdivided in several subdomains. The behavior of the mesh generator can be customized to meet user-defined requirements, for instance in terms of sizing field or quality (criteria on size or angles of mesh elements). The meshing algorithm is based on Delaunay refinement: in a nutshell, facets in a first step, then cells in a second step, are refined until the criteria are satisfied. The mesh is finally optimized in a later step. All details can be found in [2].

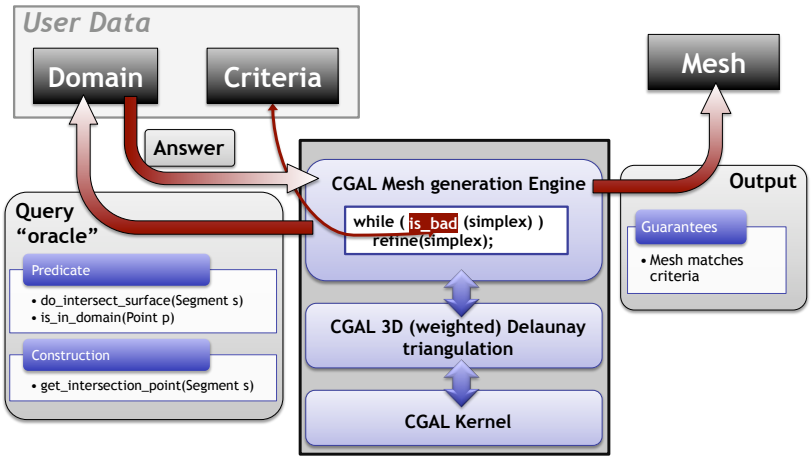

Fig. 1. The CGAL 3D mesh generator (picture taken from [2])

3. CGAL 3D periodic triangulations

Periodic triangulations in $\mathbb{R}^3$ can also be seen as triangulations the 3D flat torus, defined as the quotient space of $\mathbb{R}^3$ under the action of a group generated by three independent translations. The CGAL 3D periodic triangulation package [8] currently allows to handle Delaunay triangulations in the 3D flat torus $\mathbb{R}^3/(\mathbb{Z}^3, +)$, for which the group is generated by three equal length translations along the coordinate axes; its fundamental domain thus is a cube.

Periodic Delaunay triangulations can be constructed and modified incrementally by inserting and removing vertices. The correctness of the algorithm can be ensured in two ways [13]. The default consists in computing in the 27-sheeted covering space $\mathbb{R}^3/((3\mathbb{Z})^3, +)$, i.e. with 27 copies of each input points, as long as the edges of the triangulation are longer than some value; when all edges are short enough, then the triangulation can safely be converted to $\mathbb{R}^3/(\mathbb{Z}^3, +)$, and all remaining points are inserted without resorting to copies. It was shown that only the first few hundred points are copied, so when the input data set is large, this has a very low impact on the running time. Optionally, the user can choose the heuristic way: it consists in initializing the triangulation with a precomputed triangulation of "dummy" points, chosen in a way to ensure that the above-mentioned condition on edge lengths is always satisfied; then all input points can safely be inserted; the algorithm tries to remove the dummy points at the end of the construction, which is possible if the input point set is sufficiently large and well distributed. Both methods manage to use copies of points only when actually needed, thus allowing for both correctness and efficiency.

4. Periodic mesh in 3D

Our software uses the above two CGAL packages. As the 3D mesh generation package was designed on top of the CGAL 3D triangulation package, several obstacles must be overcome to use a CGAL 3D periodic triangulation instead.

Some of the difficulties occur at the level of the interface between the two existing packages, and require to modify the CGAL source code, for instance:

- Since a vertex may be associated with several points, copies on an input point, accessing the precise point in the domain must be done through `t.point(v)` instead of `v→point()` (here `t` denotes the triangulation and `v` denotes a vertex handle).
- For the same reason, the periodic criteria must be given more information to access points. This is achieved by passing the triangulation to the criteria as an additional template parameter.

Due to lack of space, we omit a few other technical changes to be done.

At a higher level, the semantics of the oracle and of the criteria is modified: they must take periodicity into account. For instance, when evaluating the predicate `do_intersect_surface(Segment s)` for a segment that intersects a facet of the fundamental domain, it may be the case that `s` does not intersect the surface in the domain, but its translated copy that intersects the opposite domain facet does intersect the surface in the domain (see Fig. 2 for an illustration on a simple shape in 2D). So, for such segments, if the first call of the predicate does not find an intersection, it must be called a second time with the appropriate translated image of the segment.

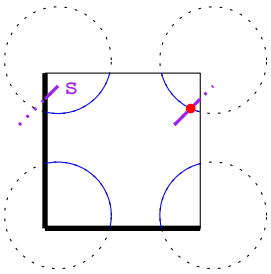

Fig. 2. Periodic oracle

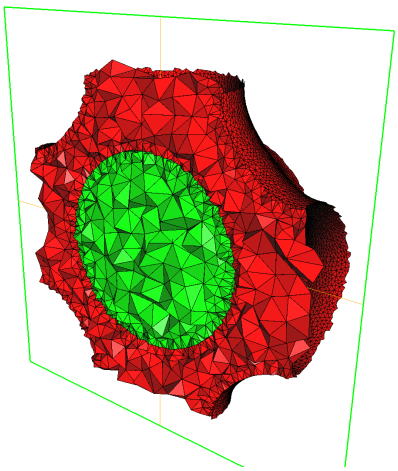

Fig. 3. Meshing subdomains, and using a sizing field

The heuristic option presented in Sec. 3 is chosen for the underlying periodic triangulation, so no point is duplicated during the construction of the mesh, and only the part of the shape that is included in one fundamental domain is meshed.

The periodic mesh generator provides users with the same flexibility as the non-periodic one: it allows to mesh multiple domains and to define shape criteria, as illustrated in Fig. 3 for a triply periodic minimal surface, the Schwarz P surface, containing spheres. (Handling domains with sharp features and optimizations is in progress.)

Fig. 4 to 7 show results obtained on the gyroid, another triply periodic minimal surface. Fig. 8 and Fig. 9 show results obtained on real world examples.

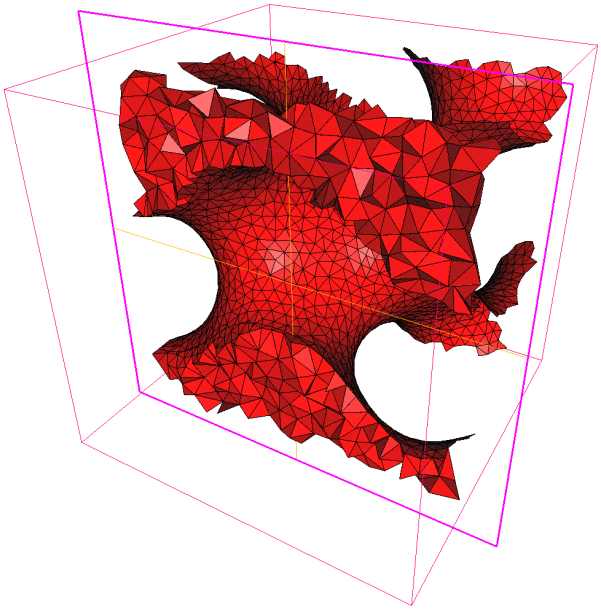

Fig. 4. The mesh that is actually computed (only the interior tetrahedra are shown)

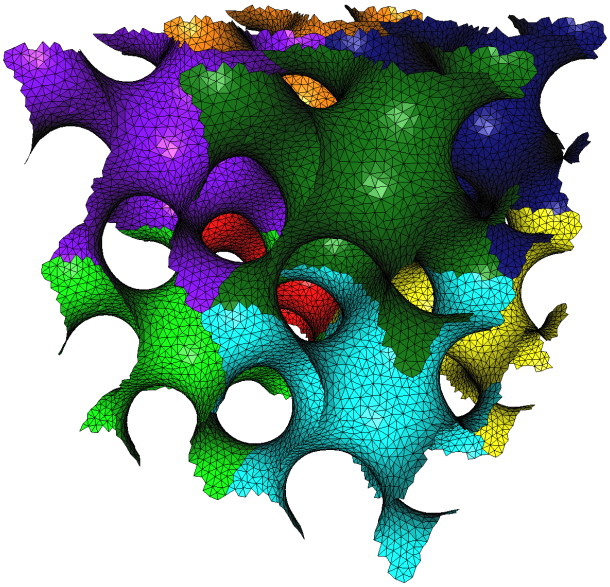

Fig. 5. Visualization of eight copies of the surface facets of the mesh

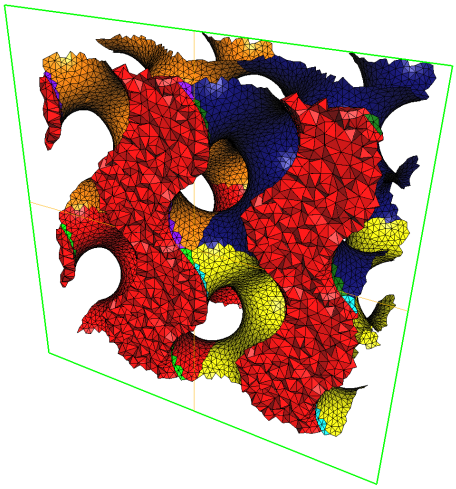

Fig. 6. Visualization of eight copies of the part of the mesh elements included inside the shape (in red on front: cells cut by a plane)

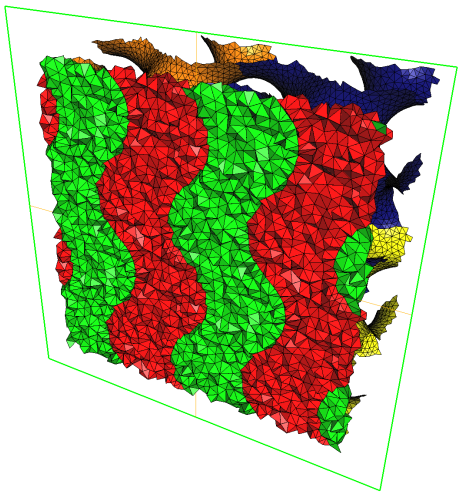

Fig. 7. Visualization of eight copies of all mesh elements (on front: interior of the shape in red, exterior in green)

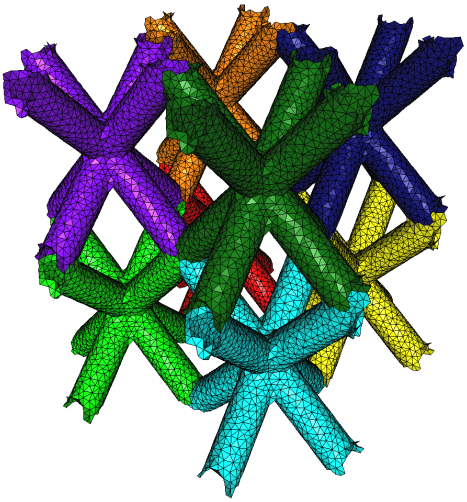

Fig. 8. Material used for bone scaffolding, eight copies are shown (data courtesy of M. Moesen, U. K. Leuwen)

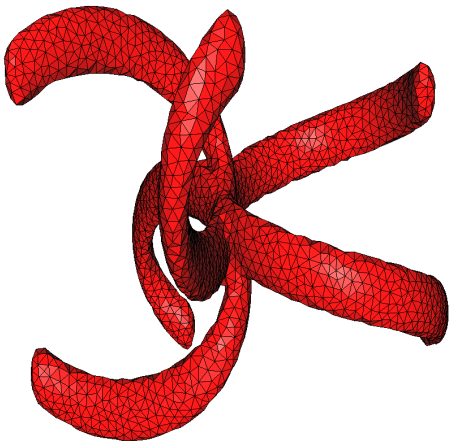

Fig. 9. Photonic crystal (data courtesy of M. Blome, Zuse Institut Berlin)

Acknowledgments

The authors wish to thank Mikhail Bogdanov for his work on a first prototype implementation.

References

[1] E. Fogel, M. Teillaud, Generic programming and the CGAL library, in: J.-D. Boissonnat, M. Teillaud (Eds.), Effective Computational Geometry for Curves and Surfaces, Springer-Verlag, Mathematics and Visualization, 2006, pp. 313–320. URL: http://hal.inria.fr/hal-01053388.

[2] C. Jamin, P. Alliez, M. Yvinec, J.-D. Boissonnat, CGALmesh: a Generic Framework for Delaunay Mesh Generation, Research Report 8256, INRIA, 2013. URL: http://hal.inria.fr/hal-00796052.

[3] P. Alliez, L. Rineau, S. Tayeb, J. Tournois, M. Yvinec, 3D mesh generation, in: CGAL User and Reference Manual, CGAL Editorial Board, 2014. URL: http://doc.cgal.org/4.4/Manual/packages.html#PkgMesh_3Summary.

[4] S. Pion, M. Teillaud, 3D triangulations, in: CGAL User and Reference Manual, CGAL Editorial Board, 2014. URL: http://doc.cgal.org/4.4/Manual/packages.html#PkgTriangulation3Summary.

[5] CGAL prospective workshop on geometric computing in periodic spaces, 2008. URL: http://www.cgal.org/Events/PeriodicSpacesWorkshop/.

[6] Subdivide and tile: Triangulating spaces for understanding the world, 2009. URL: http://www.lorentzcenter.nl/lc/web/2009/357/info.php3?wsid=357.

[7] Workshop on geometric structures with symmetry and periodicity, 2014. URL: http://www-sop.inria.fr/geometrica/events/wocg14-symmetry_and_periodicity/.

[8] M. Caroli, M. Teillaud, 3D periodic triangulations, in: CGAL User and Reference Manual, CGAL Editorial Board, 2014. URL: http://doc.cgal.org/latest/Manual/packages.html#PkgPeriodic3Triangulation3Summary.

[9] M. Caroli, M. Teillaud, Video: On the computation of 3D periodic triangulations, in: Proc. 24th Annual Symposium on Computational Geometry, 2008, pp. 222–223. URL: http://www.computational-geometry.org/SoCG-videos/socg08video/.

[10] V. Ogarko, S. Luding, Hierarchical multi-scale modeling, URL: http://www.utwente.nl/ctw/msm/.

[11] T. Sousbie, The persistent cosmic web and its filament structure I: Theory and implementation, Monthly Notices of the Royal Astronomical Society 414 (2011) 350–383. Also in arXiv:1009.4015.

[12] R. van de Weygaert, G. Vegter, H. Edelsbrunner, B. J. Jones, P. Pranav, C. Park, W. A. Hellwing, B. Eldering, N. Kruithof, E. P. Bos, J. Hidding, J. Feldbrugge, E. ten Have, M. van Engelen, M. Caroli, M. Teillaud, Alpha, Betti and the megaparsec universe: on the homology and topology of the cosmic web, in: Transactions on Computational Science XIV, volume 6970 of *Lecture Notes in Computer Science*, Springer-Verlag, 2011, pp. 60–101. URL: http://www.springerlink.com/content/334357373166n902/.

[13] M. Caroli, M. Teillaud, Computing 3d periodic triangulations, in: European Symposium on Algorithms, volume 5757 of *Lecture Notes in Computer Science*, 2009, pp. 37–48. URL: http://www.springerlink.com/content/55415144316j214g/fulltext.pdf.