Method for 3D printing of highly complex geometries

The first “flat torus” printed in 3D

T. Henocque [a], P. Marin [b]

[a] AIP Primeca Dauphine-Savoie, Grenoble-INP
[b] Grenoble-INP/UJF-Grenoble 1/CNRS, G-SCOP UMR5272

Article Information

Keywords:
Flat torus
3D printing
Cutting methods
Mesh manipulation

Corresponding author:
Thierry Henocque
Tel.:+33 688 449 128
Fax.:+33 476 543 894
e-mail: Thierry.henocque@grenoble-inp.fr
Address:AIP Primeca Dauphine-Savoie, 740 rue de la piscine, Domaine Universitaire, BP 46, 38042 Saint Martin d'Hères Cedex

Abstract

This paper presents a case study based on a complex printing of a mathematical object challenged as a mathematical problem by Kuiper and Nash in the 50s.

The purpose was to build a physical representation of this abstract mathematical object.

The problem was to deal with a large amount of data in highly complex topology with cost and accuracy constraints

We have pointed out several problems relatives to cutting 3D highly waved objects and propose different ways for cutting regarding aesthetic aspect and other constraints.

This paper raises interesting research questions related to data storage structure and manipulation of highly complex geometries.

1 Introduction

In the 1950s, J. Nash [1] and N. Kuiper [2] worked on isometric imbeddings and demonstrated the theoretical existence of an abstract mathematical structure called flat torus. Using the results of Mikhail Gromov [3] on convex integration thergeneral and associated in the Hevea Project [4][5][6], V. Borrelli, S. Jabrane, F. Lazarus and B. Thibert [7] realized for the first time synthetic images (fig.1, fig. 5) representing the object, as announced on the 20th of April 2012 in the press release [8] of the French National Center for Scientific Research.

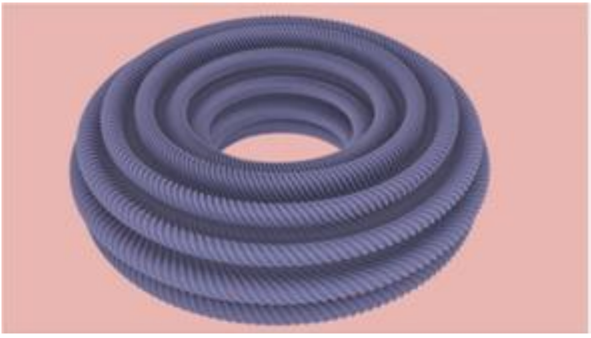

Fig. 1 computed outside view of a flat torus copied from [7]

Francis Lazarus and Boris Thibert came in December on our prototyping platform asking us to print the torus to test the surface roughness that could be reached with a 3D printer and could be visually and by touch detected for different scale factors. One challenging issue was to allow B. Tiber to "show" the shape and roughness of the torus to one of his PhD jury member who is blind.

As they were used to compute 3D objects and were familiar with the 3D file formats, there where no visible problem in printing the object. If ever, we usually use professionals softwares on a well sized computer to repair structural defects on files generated by 3D software.

Nevertheless, all our attempts to modify the model structure, to cut it in smaller parts, to dig inside to lighten the final result, simply failed in the limits of the power of our computer.

This paper describes why we encountered these problems and how we managed to solve them and finally printed the famous torus on a Zcorp ZPrinter® 650 color 3D printer

2 Materials, initial data and expected results

2.1 Printer technology

The printer used is inkjet 3D printing systems, which constructs the part, layer by layer in a plaster powder type, by dumping coloured plots of binder in place where we want to produce the solid material. This technology is suitable for printing this model because it doesn’t need any additional support material and provides a colored result.

The maximum size of the bounding box is 254x381x203 mm3 (10x15x8 inches) with a resolution of 540x600x200 dpi and a minimal plot size, which is a sphere, of 300µm diameter.

The construction speed depends on the size of the part, but is usually close to 1cm high per hour.

The bounding box size of the model is approximately 250x250x80mm.

After printing, we use as post-treatment an infuser which impregnates of wax the part for a better surface quality and a stronger result.

2.2 Data structure and model size

The mathematical theory of the construction of the torus is explained in [7] and very well popularized in [6].

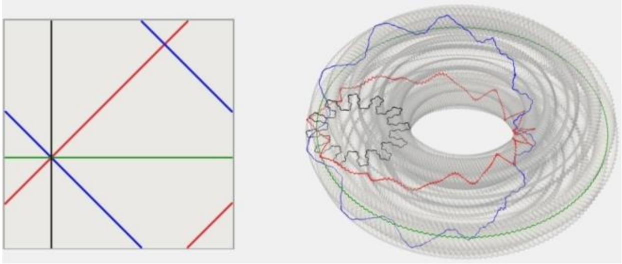

Fig. 2 Isometric embedding of a vertical, a horizontal, a diagonal and an anti-diagonal copied from [6]

As authors of the model wanted to view as much details as possible, they calculated the size of the model to be consistent with our printer definition, taking care of the theoretical number of voxels[1] per inch to maximise the point's number for the best result for their part.

Doing so, they gave us a VRML[2] file composed by 16 million of points and 32 million of triangles, corresponding to the external surface of a flat torus with 3 levels of corrugation. This represents approximately 0.6 gigabytes in binary mode, which does not seem to be impossible to manipulate. In fact, professional software we use to fix models and prepare 3D printing, running on a computer with eight processors and sixteen Giga-Bytes of RAM[3] doesn't have any problems to load the part, but they were just unable to cut or modify it.

When using 3D printers and many other additive manufacturing technologies, it is important to keep in mind that the part price is proportional to the weight. As the final part weight was evaluated at 2.5 kg, it was very important to be able to print a hollow part to save material and price. For testing the surface roughness, we expected to be able to print only a quarter or any portion of the torus. As this requires cutting and keeping only a part of the model, this was not possible with our computer and software.

2.3 The First printed model

To overcome encountered problems, we asked to the authors to provide us a completely hollowed model, with a simple hollow torus inside the model. We also asked them, to reduce the ASCII[4] file size, to reduce the precision from 1/1000mm to 1/100mm, to reduce the number of spaces and tabulations in the VRML file. We just had to reverse the surface's normal of the inside torus, by reverting (in VIM[5]) point's order in the Indexed-Face-Set[6] [11] to make the part really hollowed. The solution used to empty the powder remaining inside the torus was to print it in two separates half torus. This was obtained by printing a first job from the beginning to a layer chosen to minimize the visual impact of cuts (fig 3), and then a second job for the remaining layers to obtain a part that has been glued afterward over the previous part.

This way of doing allows the 3D printer driver to compute the required horizontal sections (layers) of the simplified torus, and doesn't require any other software to separate it. Thus this experience shows that the software is able to compute the plane cross-section of the torus, which result is a polygonal closed contour, and send it to the printer, but is unable to cut the volume, which expected result should be two or more separated volumes.

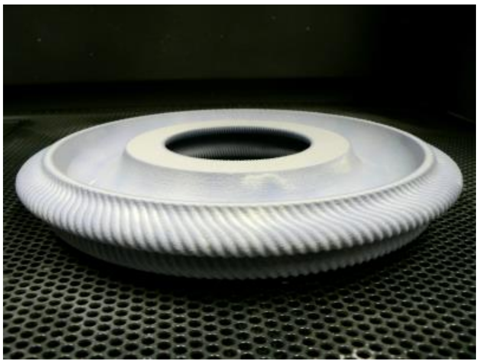

Fig.3 The first 3D printed half flat torus

As result of this cutting and gluing operation, we save 1kg of powder, and so, approximately 1500€ on the price cost of the built.

The result (fig.4) has been exposed the day after the printing during the day of ''Les sciences du numérique à Grenoble''[10] in the Jean Kuntzmann building on 2012 December the 14th.

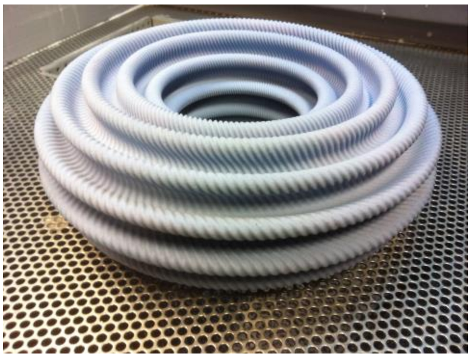

Fig.4 The first 3D printed flat torus

2.4 Printing the internal view

Mathematicians also expected a view of the inner side of the torus surface. As thickening process alters details

[1] *Volumetric pixel*
[2] *Virtual Reality Modeling Language*
[3] *Random Access Memory*

[4] *American Standard Code for Information Interchange*
[5] *VIM : the best and very well nown text editor www.vim.org*
[6] *In VRML normal to the plane is defined by the polygon point order in the coordindex section of IndexedFaceSet geometry*

and thin corrugations, it is not possible to print a single object on which inner and outer surfaces respect the flat torus topology.

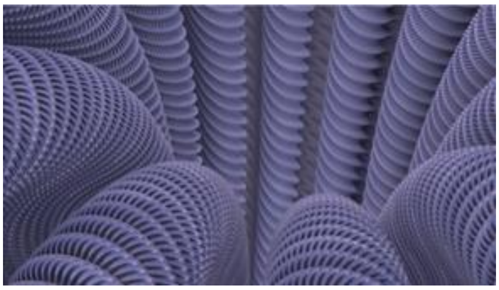

Fig. 5 computed inside view of a flat torus copied from [7]

One of the challenges to print the internal view was to be able to thicken the given surface by a skin as thin as possible to reduce the cost and to cut the model in a nice cutting plane to get an artistic inside view of the torus. A second challenge was to rebuild from the cut surfaces and their borders a single and well defined volume (according to geometric modelling criteria of closed boundary, manifold and not self-intersecting) that could be printed by additive manufacturing.

If we look to the dataset and to the structure (fig.5), which is a smooth fractal, we can enumerate several problems which can be responsible of the commercial software inability to cut and transform it.

The following sections describe problems encountered and solutions used to thicken and to cut the part properly.

As the dataset was given as a 2D table of points ordered along the green horizontal line (fig.2), the first operation was to compute the transposed matrix to make the dataset ordered along the black vertical line (fig.2). This new dataset has many advantages like giving the ability of analysing a portion of the data without reading and loading all the dataset.

3 Thicken the part

The main idea was to thicken the part in a graphical way unambiguous about the interesting surface. That means that, we don’t want people to feel that the external surface has to be considered.

The simplest way to do that should be to add a standard torus outside the surface with a diameter calculated to be as close as possible of the inside surface. According to the material used by the printer, this distance must be at least of 2 mm.

This solution is not the cheapest one, so we preferred a solution with a skin close to a constant distance of the surface, with as few points as possible not to increase to much the part weight and complexity.

3.1 Thicken a torus

Thicken a 3D volume consists on analysing intersections of multiple plans and, in many cases, would result on increasing the number of facets.

Most of time, thicken a volume can be realised by thicken 2D layers following the z-axis[7] of the sliced volume. This technic can only be a problem when the slope variation is greater than the added skin. It can be considered like evidence that it can be compensated with taking care of the position of the surface on adjacent layers, but is often overlooked (fig.6) in slicers used by 3D printers.

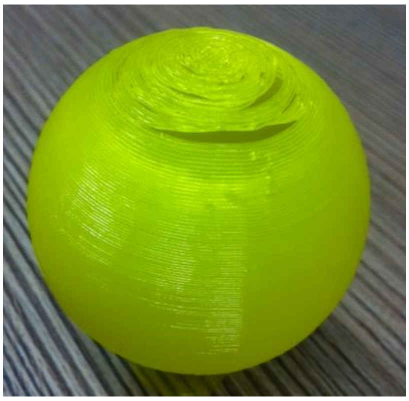

Fig. 6 aberration in slicing volume with thin surface

For a torus, even if it is very wavy like ours, the best way to thicken it, layer by layer, is to consider each successive layer as successive plane rotating around its center, perpendicular to its median plan.

If we have a look on the torus (fig.4), we can be sure that the perimeter generated by thickening a single layer will be approximately the same for all the layers. Consequence for calculation is to transform an $O(n^{i+1})$ algorithm in an $O(n^{i})$ one.

3.1.1 Thicken optimised algorithm used.

As we know many things on the data set and the 3D structure, we used optimisations that could be exploited in many case with highly complex geometry.

First, as the part is a torus, only one layer, i.e. one black line (fig.2) of the dataset is used to compute a thicken layer and this thickened layer is repeated by rotation around the principal axis of the torus.

2D Thickening algorithm consists on creating a line which is for each point at a constant distance of the original one (fig.7). In a 2D discrete space consisting of line segments, the first step is to calculate all the segments (red) of thicken distance away from the surface (purple) to thicken.

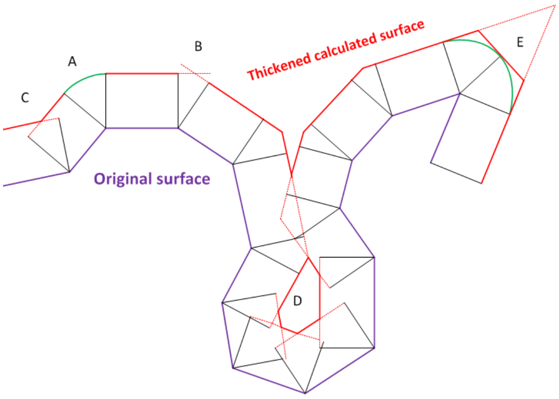

Fig.7 description of thickening algorithm

Points of reference used in the case of convex corners (A: fig.7) are usually the intersection point (B: fig.7) of the two adjacent segments, exactly like for concave angle (C: fig.7).

For important convex angle (E: fig.7), it is necessary to compute an additional point. The arbitrary chosen limit to consider the angle is ‘’important’’ was set to 90°.

[7] *Z-axis is usually the vertical axes*

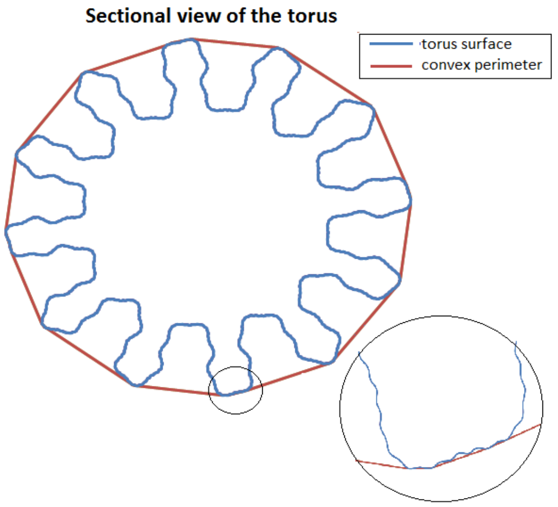

Fig.8 sectional view of the torus surface and its convex perimeter

Considering the cross-sectional view of the torus and the representation of its convex perimeter (fig.8), six facilitation were used to optimize and limit the amount of memory necessary to compute the thicken surface in the particular case of the flat torus.

1) Data-set is organized in a well-structured squared grid so a cross section corresponds to a line in it.
2) A point belonging to the convex perimeter is always on a convex angle and can't interact with any other segment when thickening adjacent segments.
3) A concave zone is always surrounded by 2 points belonging to the convex perimeter, and 2 different concave zones can't interact together.
4) The perimeter is divided into twelve areas (fig.8), so it is not necessary to use all points to calculate the next point of the convex perimeter.
5) Small concave areas bounded by two close points of the convex perimeter (zoom: fig.8) can be ignored.
6) Holes located inside the calculated surface (D: fig.7) are simply ignored.

So, the thickening algorithm used started by calculating the convex perimeter using Jarvis-walk algorithm [12] taking into consideration the topology of the data-set and the forth facilitation, limiting for each point, the search to the next point to 1/12 of following points in sectional view.

After that, for each concave area, we simply ignore all points if the distance between the two end points of the concave region is less than twice the added thickness.

Then, for big concave zones, segments are stored in an indexed array, and for each calculated segment, we look for the nearest intersection point with all other calculated following segments. The intersection point found become the starting point of the intersected segment, which is set to be the next one to study.

With all these facilitations, this algorithm calculates a thicken surface for a torus of 16 million of points and generates the VRML associated file in less than 1 second. Of course, this algorithm is really dedicated to regular torus, and, as it is based on convex perimeter, only valid to increase volume outside the part.

4 Cutting into separate parts

The main problem of this operation seems to be related to the presence of a large number of corrugations. We can represent as example the result of cutting a ''one level of corrugation'' torus (fig.9), and notice that the result is not two but five separate parts.

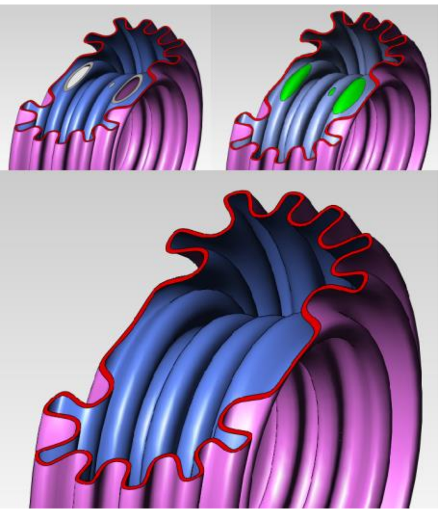

Fig.9 Cutting a corrugation with a cutting plane

The main portion just above the cutting plane, not shown in this figure, the main portion under the cutting plane (top left of fig.9) and three small portions, (in green in top right of fig.9), which are located over the cutting plane.

If we consider this problem applied on the flat torus shown in (fig.5) which has three levels of corrugation, it is not five, but potentially an infinite and at least a very large number of separate parts.

Cutting software's are probably not optimized for this problem and, as each separate part is in fact just a surface, it must be closed along its flat face, usually with a Delaunay triangulation[13], to be transformed in a volume.

Many other problems, described by C. FOREST in his PhD thesis [9], are related to the fact that cutting complex parts can generate topological aberrations, like generating null surfaces when several polygon's nodes lies on the cutting plane surface.

A very particular situation is when a single point or a single segment in the cutting plane is the unique contact point between two convex areas, like if one of the green fragments (fig.9) was attached to the main portion by only one point or one segment. In this situation, it is impossible de determine if the cutting result will be two or three parts. The printed result will be two or three parts depending on the bleed-compensation parameter that has been set up in the 3D printer driver. (See Appendix for details)

4.1 Aesthetic and functional consideration

Due to the presence of many corrugations, we can consider several different kinds of cut depending on the result we want to obtain, the reason the cut is for, and the position of potential fragments relative to cutting plan.

If we have to cut the part to get a planar surface to stabilize the object on a support, we need to eliminate the green fragments whatever their relative position to the cutting plane, because the final result have to be a plan, and we don't need to print isolated fragments.

If we have to cut the part to show a particular structure inside a model, we need to keep attached fragments which are located over the cutting plane, and would result on an unsightly result, but we need to eliminate fragments which are located under the cutting plane and would result on isolated fragments which don't need to be printed.

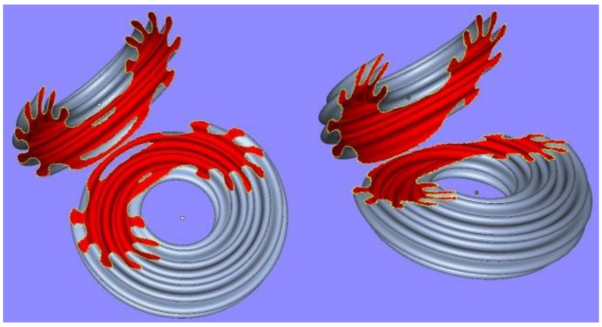

Fig.10 Two different methods of cutting

The choice between those solutions will often depend on the aesthetic of the expected result. It can also depend, in simulation [9] for example, on the process used to cut.

4.2 Algorithm used for the cut

Standard cutting algorithms are based on the determination of the list of polygons which straddle the plane and need to be cut into at least two polygons. Then, all polygons which lie entirely above the cutting plane are suppressed or moved to be separate from the initial part.

In case we want to keep little fragments located over the cutting plane attached to the main portion of the torus which lies below it, we need to determine if a cut polygon belongs to the external perimeter of the part or not.

A way to find the external perimeter is to start from a point belonging to the convex perimeter of the part on the cutting plane and to follow the intersection line between the cutting plane and the object until being back to the starting point.

We took advantage of the structure of the dataset which is in fact based on a two dimensional array of quadrilateral, simply divided into two triangles in the VRML file. This array came from the construction of the torus which is derived from the deformation by isometric embedding of a flat square. This means that points are organised in the array as a continuous structured grid, like in the VTK[8] STRUCTURED_GRID type [14].

As it is a flat torus, it is possible to navigate into the array with no limits re-entering by a side when getting out by the opposite one.

There are many algorithmic advantages of this continuous data structure. In particular, it is possible to walk around the cut perimeter (fig.11) walking step by step through quadrilaterals from a starting point using intersection between cutting plane and other segments of the quadrilateral.

In this figure, the intersections of the grid constitute the dataset points. The green line represents the points from the convex perimeter that are furthest from the geometrical center of the torus, and the red line represents the walk along the intersection line between the torus and something else.

The first step of the algorithm is to walk along the green line to fine an intersection point with the cutting plane.

The second step is to walk along the red line searching for each polygon the next one along the cutting plane.

The third step is to suppress all the polygons that are on the side of the line being above the plane.

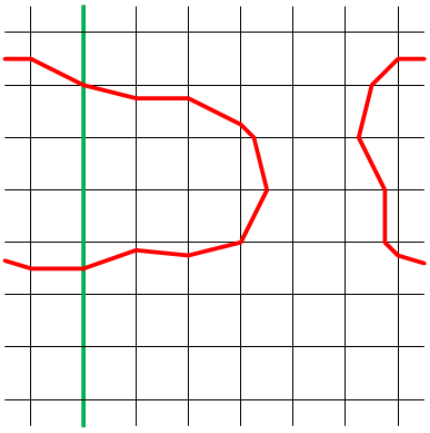

Fig.11 Walking along the cut perimeter

4.3 Volume reconstruction.

After all, we have three shapes:

- The surface generated by thickening
- The original torus surface for the area which is not concerned by the cutting process
- The plane surfaces generated by the cut.

Le last part of the process is to reassemble the shapes and to close the holes left by cutting.

VRML description of part allows to duplicate points, and also allows to group shapes with their own indexed point set so reassembling is not a real problem.

The language can support any size of polygon, so, it is possible to fill holes by a big polygon including all points involved in the cut perimeter. It is not necessary to compute triangulation for these planar surfaces, as all VRML viewers will compute it when loading the file.

4.4 Debugging method

Debugging and calculation time are still problematic due to the size of the dataset. The solution which consists on working on smaller dataset is often insufficient because the number of particular and exceptional situations increases with the size of the dataset.

To monitor debug information, I used graphical representations (fig.12) (fig.13) of polygon status modeled on the dataset structure in which each pixel color represents a value of the status.

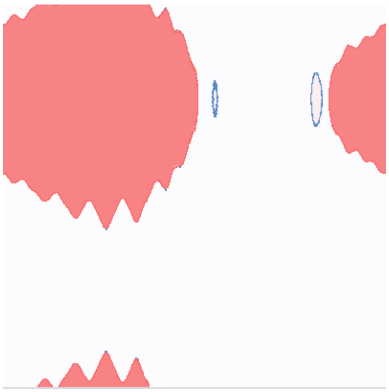

Fig.12 flat view of a cut of the torus

The flat view (fig.12) of the cut corresponds exactly to the cuts realized with the two different methods (fig.10). The big red area corresponds to the principal cut and the two blue surrounded pink areas correspond to the two big fragments (fig.10) visible on the left side of the picture.

[8] The Visualization Toolkit

The focus on a small area on (fig.13) show us very small fragments as blue surrounded pink areas, which correspond to polygons that are located above the cutting plane but must be preserved and a small pink area into the red one, which corresponds to polygons that are located below the plane, but must be deleted to prevent the generation of independent fragments.

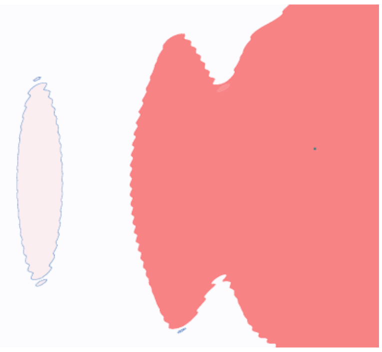

Fig.13 focus on a small area of a cut

4.5 Results and Perspectives

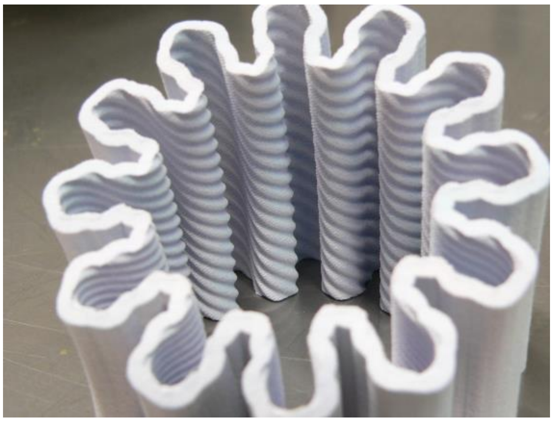

Fig.14 first print of an inside view of the torus

The cutting algorithm is fully functional and gives real satisfaction, but VRML files must be perfect because additive manufacturing slicers can't fix it due to the dataset size and complexity.

The cut algorithm based on walking on a structured grid corresponding to a topological bijection of a 3D model is potentially a very effective solution to solve identification of intersections between multiple volumes.

5 Conclusion

Reflexion on the visual result and aesthetic considerations has revealed a diversity in cutting methods closer to reality than the usual duality based on the position relative to a plane. As 3D printing processes aim to afford nice objects, they need to be provided with suitable tools to cut objects in different maner.

As a reasearch project, it could be considered that many objects are homeomorphic to a torus. If all these objects can be described as a 2D matrix considered as an approximation of the geometrical position of each point of the volume surface, it could be a very important evolution and give a considerable benefit for the optimisation of algorithms working on mesh transformation of highly complex geometries.

Appendix

Bleed compensation in plaster 3D printing process.

During the printing process, the printer adds a point of binder in the powder in all the places that have been identified as solid material. As the point is big regarding to the printer definition, around 300µ diameter regarding to 42µ for 600dpi, the printer has to shift is trajectory by the radius of the point inside the part, to give a part with true dimension.

This compensation mechanism is exactly the same with all the printing and machining process. Usually, the printer is calibrated according to the material used.

In case of inkjet printer with a post-treatment, the post-treatment is not known by the printer, so, you have to calibrate the bleed compensation regarding to the post-treatment you use.

This compensation mechanism can be ignored, resulting on a bigger but stronger result, on which badly attached fragment will be glued by the process.

Acknowledgement

Thanks for Philippe, Alain, Pierre-Marie and Christofer for their help during analysing the problem and during the printing process.

Special thanks for Francis Lazarus and Boris Thibert for their patience and for the generation of all the datasets we ask for.

References

[1] Nash J (1954) C1-isometric imbeddings. Ann Math 60:383-396.

[2] Kuiper N (1955) On C1-isometric imbeddings.Indag Math 17:545-556.

[3] Gromov M (1970) A topological technique for the construction of solutions of differential equations and inequalities. Proc Intl Cong Math 2:221-225

[4] http://math.univ-lyon1.fr/~borrelli/Hevea.html

[5] Vincent Borrelli, « Gnash, un tore plat ! » — Images des Mathématiques, CNRS, 2012.
URL : http://images.math.cnrs.fr/Gnash-un-tore-plat.html

[6] Projet Hévéa : le dossier (Les tores plats visualisés !) www.gipsa-lab.fr/~francis.lazarus/Hevea/Presse/

[7] Flat tori in three dimensional space and convex integration - Vincent Borrelli, Saïd Jabrane, Francis Lazarus et Boris Thibert, Proceedings of the National Academy of Sciences of the United States of America (PNAS), avril 2012

[8] Mathématiques : première image d'un tore plat en 3D, Communiqué de presse du CNRS du 20 avril 2012 www2.cnrs.fr/presse/communique/2583.htm

[9] Clément FOREST, Simulation de chirurgie par coelioscopie : contribution à l'étude de la découpe volumique, au retour d'effort et à la modélisation des vaisseaux sanguins. Thèse de Doctorat de l'école Polytechnique de Paris (www-sop.inria.fr/asclepios/Publications/Forest/These_forest.pdf)

[10] Mi^{2}S Informatique-Mathématique-Automatique-Signal : *Les sciences du numérique à Grenoble - 14 décembre 2012*. mi2s.imag.fr/pm/les-sciences-du-numerique-a-grenoble-14-decembre-2012

[11] Jed Hartman & Josie Werecke *The VRML handbook : Building Moving Worlds on the Web*

[12] Jarvis, R. A. (1973). "On the identification of the convex hull of a finite set of points in the plane". Information Processing Letters 2: 18–21

[13] B. Delaunay (1934) " Sur la sphère vide. A la mémoire de Georges Voronoï " : *Bulletin de l'Académie des Sciences de l'URSS. Classe des sciences mathématiques et na, 1934:6, 793–800*

[14] VTK File Formats from “The VTK User’s Guide.”(2010) 19.3:469-482 ISBN-13: 978-1-930934-23-8