PLOUGH3D
Discrete Implicit Surface Mesher
Technical description

University of Technology of Troyes
12 Rue Marie Curie - CS 42060 - 10004 TROYES CEDEX
houman.borouchaki@utt.fr

1 Introduction

An implicit surface is defined by an analytical equation $F(x, y, z) = C$ where (x, y, z) belongs to a volume domain and C is a constant called "level" and the corresponding surface is a level set. A discrete representation of this surface is the data values of the underlying function, $F(x_i, y_j, z_k)$, at the vertices (x_i, y_j, z_k) of a 3D (Cartesian) grid, $1 \leq i \leq n_i$, $1 \leq j \leq n_j$ and $1 \leq k \leq n_k$ where (n_i, n_j, n_k) are three integers defining the resolution of the grid, also implicitly representing the boundaries and so the (discrete) definition volume of the surface. Such a grid is often called a "voxel" grid.

The PLOUGH3D software generates the triangulation of a given level set of the implicit surface from a voxel grid which can be defined in two different ways:

1. **A 3D grid of real (scalar) values.**
2. **A set of grayscale images.**

The first concerns the values of a given analytical function and the second, images obtained via digitalized scans such as "CT scan" or "MRI scan".

In addition, the software also generates the corresponding level curves in each plane $z = z_k$. Several features are also available for the shape quality optimization, the simplification and the smoothing (roughness removal) of the resulting triangulation.

In the case of a 3D grid, on option via the environment parameters, the software also produces a closed surface of a given thickness around the imposed level. This kind of triangulation is particularly useful for surface 3D printing.

The software includes several processing phases (depending on the user environment parameters, the phases expressed in italics are optional):

1. Analysis of the grid (minimum and maximum values and initialization of the parameter thresholds deduced from these values).
2. Generation of the vertices of the triangulation on the edges of a 3D grid triangulation.
3. Generation of the triangles of the triangulation.
4. *Extraction of the topology of the triangulation.*
5. *Optimization of the shape quality of the triangulation.*
6. *Simplification of the triangulation.*
7. *Extraction of the connected components of the triangulation.*
8. *Low frequency smoothing.*
9. Loading into memory and writing the resulting triangulation.

The software makes maximum use of available CPU resources and most of the phases are performed in parallel (multi-core).

2 Features

Two types of inputs are considered, a grid of real values and a set of images.

The first one is a 3D grid of real values described as follows:

- Three integers (n_i, n_j, n_k) defining the x, y and z dimensions of the grid.
- A list of $n_i * n_j * n_k$ real (scalar) values.

The grid must have as extension the keyword "grd" (format described above), e.g. grid.grd. The second is a set of ASCII images in "pgm" format. The images must have a fixed name followed by an integer number representing its z-scan level, e.g. image.1.pgm, image.2.pgm, etc.

The PLOUGH3D software is driven by the ASCII environment file "plough.env". In this file, the user specifies the software control parameters. Each parameter is identified by a keyword preceding the parameter. The following list describes the keywords with the type of the expected parameter:

verb: real varying from 0 to 1 indicating the verbosity of the program progress. For a value equal to 0 (resp. 1), the program displays the minimum (resp. maximum) information about the program progress. The default value is 1.
grid: integer (0, 1, 2, 3) indicating whether it is images input or a 3D grid:

- 0: a set of images.
- 1: a 3D grid.
- 2: a 3D grid with the extraction of the surface with a given thickness.
- 3: a 3D grid with the extraction of the two "offsets" surfaces according to a given "offset" distance.

The default value is 0.
file: character string indicating the name of the ASCII file (without extension) containing the 3D grid in "grd" format. The default value is "file".
image: character string indicating the common name of the ASCII images (without extension) in "pgm" format. The default value is "image".
ninit: integer indicating the number of the first image (keyword taken into account if **grid** = 0). The default value is 1.
nstep: integer indicating the increment of the images (keyword taken into account if **grid** = 0). The default value is 1.
nimages: integer indicating the number of images (keyword taken into account if **grid** = 0). The default value is 2.
resolution: integer specifying the resolution considered for each image. For a **resolution** = 1 all pixels are taken into account, for a **resolution** = 2 only one pixel out of 2 in row and column is considered. In the same way, for a **resolution** = n only one pixel out of n in row and column is considered. The default value is 1.
level: real corresponding to the level set to extract. The default value is 0.
origin: 3 reals representing the real coordinates of the lower corner of the grid. The default value is (0, 0, 0).
deltax: real indicating the x-discretization step of the grid. The default value is 1.
deltay: real indicating the y-discretization step of the grid. The default value is 1.
deltaz: real indicating the z-discretization step of the grid. The default value is 1.
orientation: integer (1, -1) indicating the orientation of the surface to extract.

- 1: the surface is oriented towards smaller values.
- -1: the surface is oriented towards larger values.

thickness: real indicating the thickness or offset in case of a 3D grid (keyword taken into account if **grid** = 2 or 3). For **grid** = 2, a closed surface with a thickness value indicated by **thickness** is generated around the level surface indicated by **level**. For **grid** = 3, two "offset" surfaces with a distance indicated by **thickness** are generated around the level surface indicated by **level**. The default value is 0.

topology: boolean (0, 1) indicating the activation of the topological structure determination of the extracted surface. Activation is automatically triggered if **optim**, **decim** or **smooth** are activated. The default value is 0 (no activation).
optim: boolean (0, 1) indicating the activation of the shape quality optimization. The optimization is based on edge flipping operations and is triggered if **optim** = 1. The default value is 0.
decim: boolean (0, 1) indicating the activation of the triangulation simplification. If **decim** = 1, the simplification is activated with the threshold defined by the keyword **deps**. In this case, edges with a relative length less than the value indicated by **deps** are eliminated. The default value is 0.
deps: real indicating the relative size limit of the edges to be eliminated (keyword taken into account if **decim** is activated). The absolute length of the edges to be removed is the product of the values given by **deps** and **deltaz**. The value of **deps** is limited to $\sqrt{2}/2$. The default value is 0.1.
smooth: boolean (0, 1) indicating the activation of the high frequency filter. In case of activation (**smooth** = 1), a smoothing procedure to reduce the roughness is performed. The default value is 0.
components: integer indicating the number of connected surface components to be extracted. If **components** = 0, all components are extracted. The components are sorted by the number of elements from the largest to the smallest. The default value is 0.
write: integer (0, 1, 2) indicating the writing in "mesh" format of the resulting triangulation:

- 0: no writing.
- 1: writing in binary mesh format.
- 2: writing in ASCII mesh format.

The default value is 1.
iwrite: pair of integers, bounds of an interval containing the z levels for which a writing of the 2D level curves in mesh format is requested. The default value is (-1 -1).

A minimalist example of an environment file is the following:

```
grid    1
file    schwarz
level   0
```

In this case, the input is a 3D grid named "schwarz.grd" and the level 0 surface must be constructed. As an output, PLOUGH3D produces all the connected components of the resulting triangulation in the binary "mesh" format file, schwarz.plg3D.meshb, as well as the level curves in each z-level plane defined in the input. These curves are given in the binary mesh file schwarz_i_plg3D.meshb.

Several PLOUGH3D modules (procedures in C) are also provided to integrate the software in a given platform. In this sense, the software is used as a library. The general scheme includes the following steps:

1. Initialization of the environment parameters.
2. Reading of the environment parameters.
3. Initialization of the grid (voxels).
4. Reading of the grid and loading in memory.
5. Generating the desired triangulation and storing it in memory.
6. Release of the grid.

Each module takes as input at least one pointer to a **plg** parameter of type *Plugh* (internal to the library) to be declared, including all the structures of the code. All the structures are defined in the header file "plough.h". These modules are listed below.

Initialization of the environment parameters.

PLGInitializeEnv(&plg) ;

The environment parameters are initialized to the values indicated above.

Reading of the environment parameters.

PLGReadEnv(&plg);

The user environment parameters are loaded from the "plough.env" file.

Initialization of the voxel grid.

PLGInitializeGrid(&plg);

The various variables associated with the grid are initialized.

Reading of the grid or images.

PLGReadFile(&plg);

The 3D grid or image data are read and loaded into memory.

Generation of the triangulation.

PLGMesh(&plg, &msh);

The triangulation corresponding to the level indicated by the environment parameter **level** is constructed (also taking into account the other parameters acting on the triangulation) and is loaded into memory in the **msh** variable of type **Imesh**, a structure containing the geometry and topology of the triangulation. This structure is defined in the header file "plough.h". In this structure which consolidates all the data of a triangulation, we find several sets:

- *Ivertex* : set of vertices (the coordinates, the z level and a reference associated with the number of the connected component).
- *Iedge* : set of segments (the connectivity, the z level and a reference associated with the number of the connected component).
- *Ielement* : set of elements (the connectivity, the z level and a reference associated with the number of the connected component).
- *Iridge* : set of border edges (the connectivity and a reference associated with the number of the connected component).

More precisely, these structures are written as:

```
typedef struct ivertex
    {
        float       c[3];
        int         img;
        int         ref;
    } Ivertex;
```

```
typedef struct iedge
      {
            int             v[2];
            int             img;
            int             ref;
      } Iedge;


typedef struct ielement
      {
            int             v[3];
            int             img;
            int             ref;
      } Ielement;

typedef struct iridge
      {
            int             v[2];
            int             ref;
      } Iridge;
```

And the *Imesh* structure containing all of these structures is written as:

```
typedef struct imesh
      {
            int             nv, ne, nr, ncc, nimv, nime;
            Ivertex         *vx;
            Ielement        *ex;
            Iridge          *rx ;
            Ivertex         *imvx;
            Iedge           *imex;
      } Imesh;
```

In that respect, the resulting triangulation is described by:

- The number of vertices *nv*
- The set of vertices *vx*
- The number of elements *ne*
- The set of elements *ex*
- The number of boundary edges *nr*
- The set of boundary edges *rx*
- The number of connected surface components *ncc.*

And the level curves are described by:

- The number of vertices *nimv*
- The set of vertices *imvx*
- The number of segments *nime*
- The set of segments *imex.*

In addition, this triangulation is written in a "mesh" format file whose description in ASCII format is given in the appendix.

Release of the grid.

```
PLGFinalizationGrid(&plg);
```

The grid is unallocated in memory.

The "general" context of use of this library includes the phases:

1. Initialization of the environment parameters.
2. Reading the environment parameters.
3. Initialization of the grid (voxels).
4. Reading of the grid and loading in memory.
5. Loop for creating triangulations for different values of **level.**
   a. Loading of new environment parameters including the new value of **level** for the same grid.
   b. Generating the corresponding triangulation and storing it in memory.
6. Releasing the grid.

In this case, there are several generated triangulations, each of them associated to a given **level**. The loading of the new environment parameters must be performed by the user. In this scheme, the grid data remain in memory as long as a new triangulation is to be generated.

In the following, there are several examples to illustrate the software features.

3 Examples

In this section, several application examples are presented with illustrations. All examples are realized on a laptop computer with an Intel Core i9-8950HK (6 cores and 12 threads).

The first example is the implicit surface defined by:

$$F(x,y,z) = 3(\cos(x) + \cos(y) + \cos(z)) + 30\cos(x)\cos(y)\cos(z)$$

with $(x,y,z) \in [-8\pi, 8\pi]^3$. It is a variant of the (minimal) periodic surface of Schwarz. The input is a grid (256,256,256) of points uniformly distributed in $[-8\pi, 8\pi]^3$. This gives the origin (-8π, -8π, -8π) and a discretization step of 16π/256. A simple program is given in the appendix for the generation of the voxels (F values) of this grid and its storage in the file "schwarz.grd".

We first consider the environment file "plough.env" given by:

```
grid      1
file      schwarz
level     0.0
origin    -12.566371 -12.566371 -12.566371
deltax    0.098560
deltay    0.098560
deltaz    0.098560
```

It is the reconstruction of a triangulation defined by level 0 of F with its data on a 3D grid provided in the file "schwarz.grd", the origin and the discretization steps in x, y and z are defined as above. No additional processing is required (as indicated in the default values of the environment parameters).

The resulting triangulation is composed of 4,997,096 vertices and 9,953,520 triangles and is performed in 0.96 seconds and required 0.588 Gigabytes of memory. Figure 1 shows this (initial) triangulation as well as the trace corresponding to a section and figure 2, an enlargement.

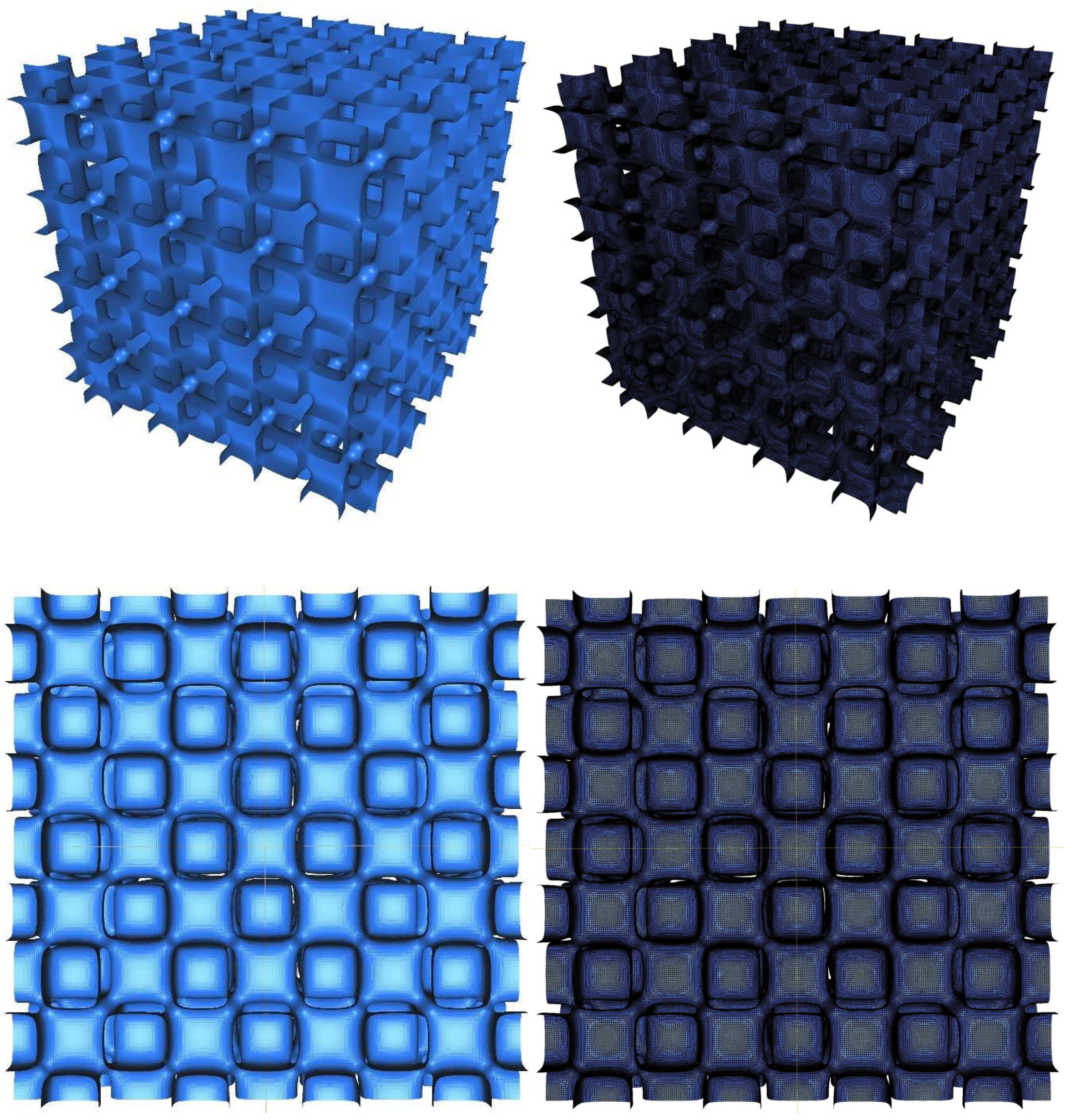


Figure 1: Initial triangulation and a slice.

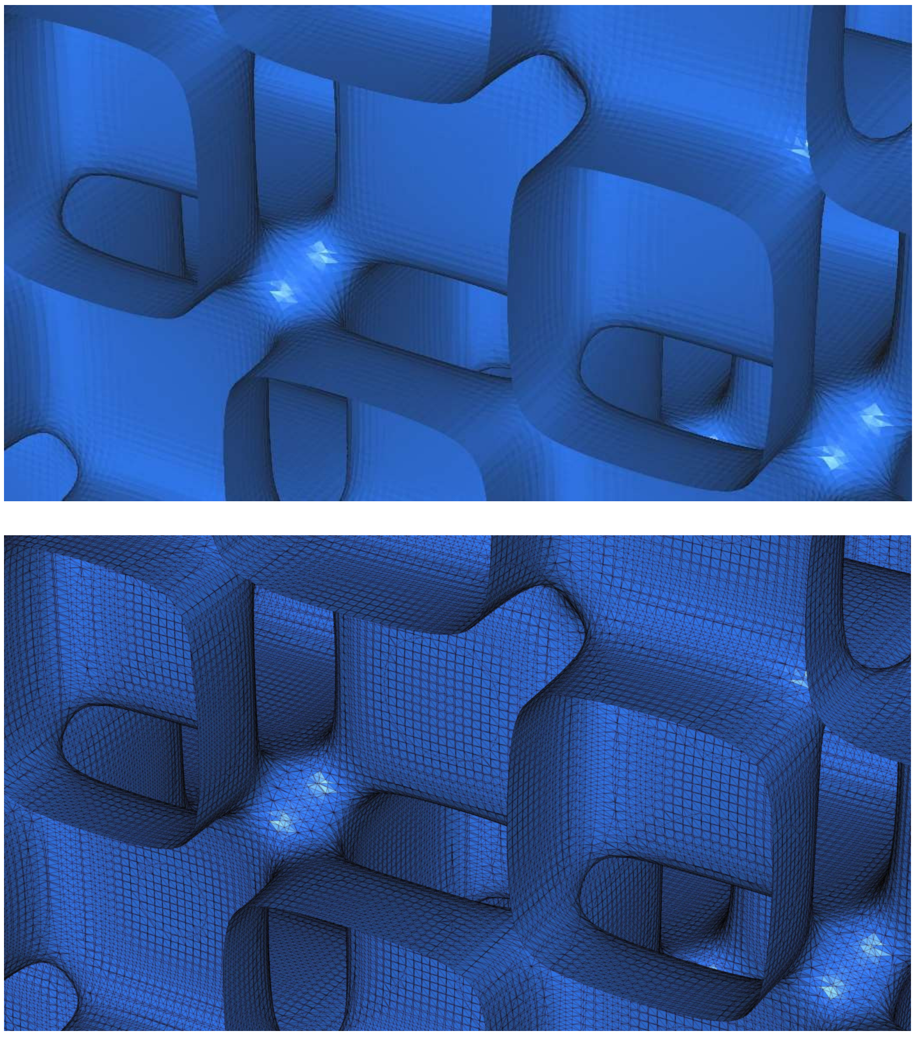


Figure 2: Enlargement of the initial triangulation.

Figure 3 shows the level curves corresponding to the $z = z_{128}$ plane.

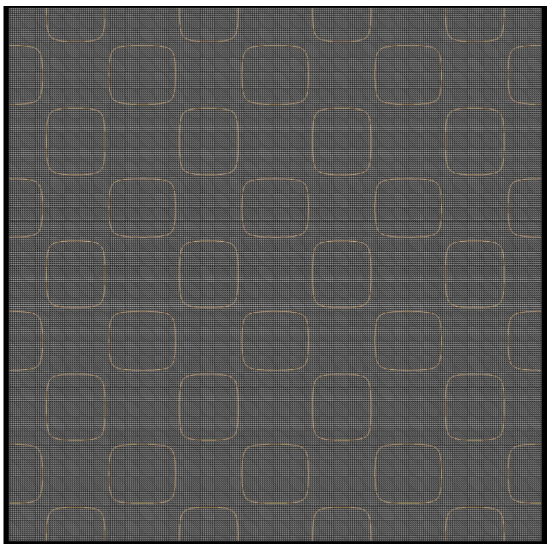


Figure 3 : Contours of the surface on the z=z_128 plane.

Now we are interested in the generation of a quality triangulation and for this we activate the shape quality optimization and the decimation (with a relative threshold length equal to 0.7) of the triangulation via the environment parameters:

```
grid      1
file      schwarz
level     0.0
origin    -12.566371 -12.566371 -12.566371
deltax    0.098560
deltay    0.098560
deltaz    0.098560
optim     1
decim     1
deps      0.7
```

Figures 4 and 5 show respectively the resulting optimized and simplified triangulation and an enlargement of this triangulation. It includes 1,178,862 vertices and 2,339,669 triangles. The construction took 8.57 seconds (due to optimization and simplification). To appreciate the quality of the triangulation, the following diagrams show the histogram of the shape quality of the initial triangulation and that of the optimized and simplified triangulation with the minimum and average values (for a triangle, the measure of the shape quality considered is proportional to the quotient of the area of the triangle by the sum of the squares of the length sides with a maximum value equal to 1 for a regular triangle).

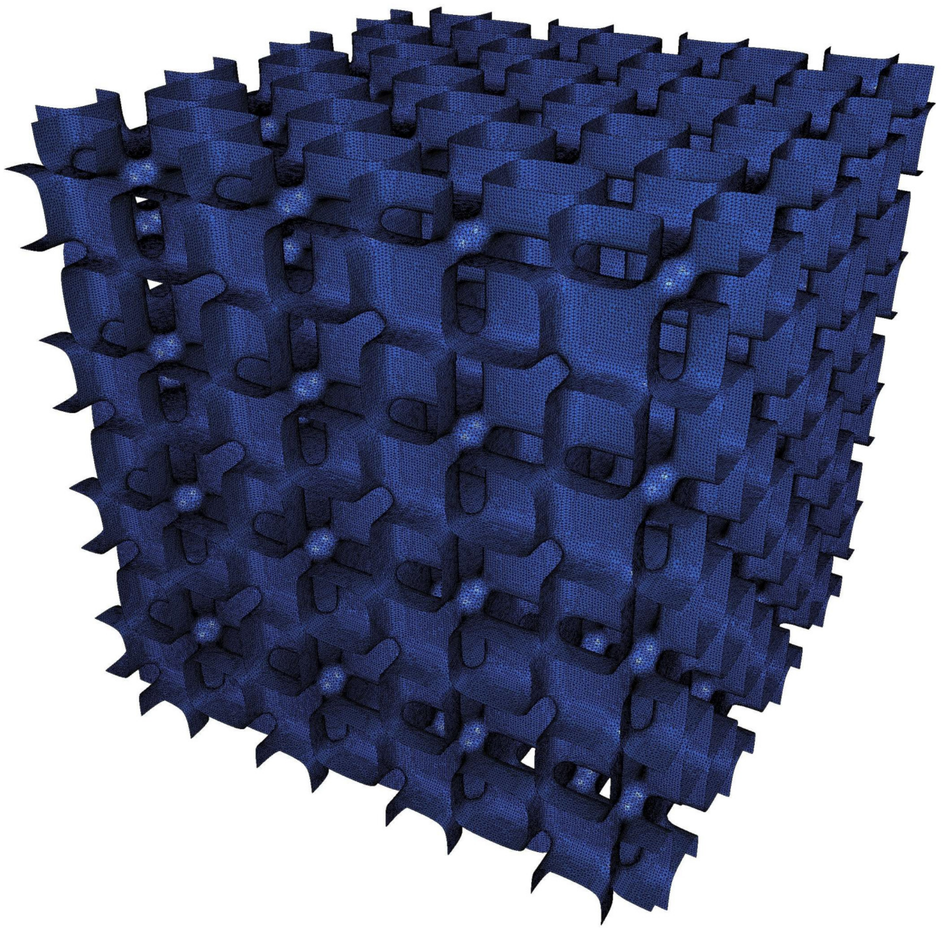


Figure 4: Optimized and simplified triangulation.

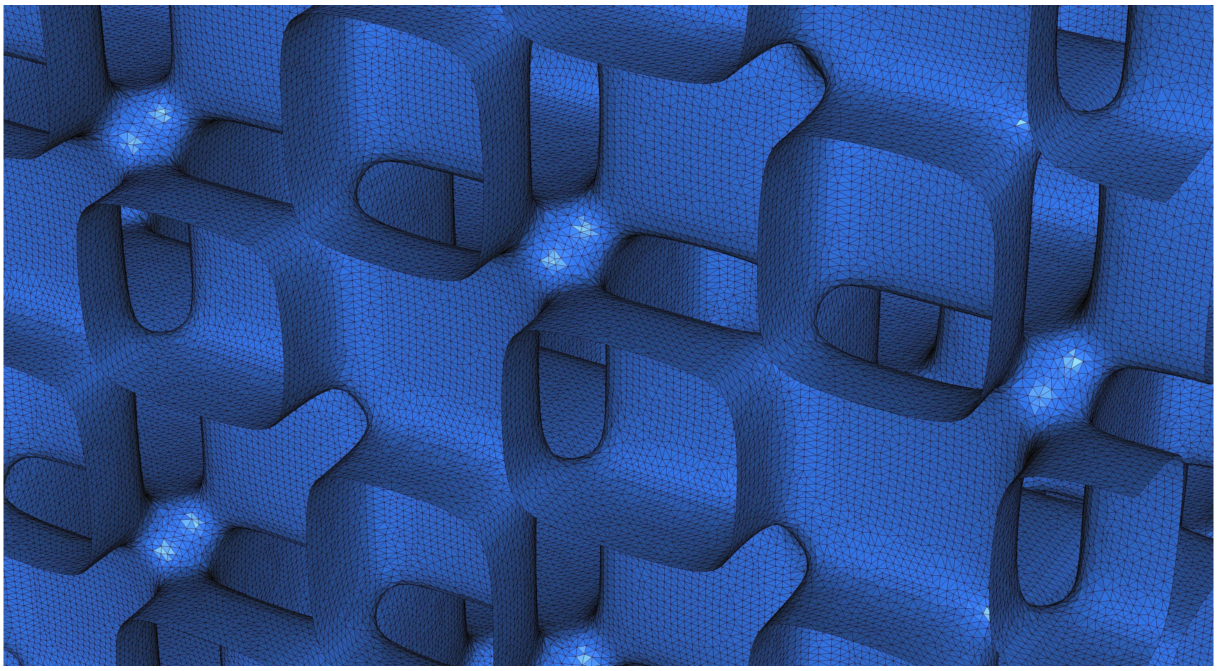


Figure 5: Enlargement of the optimized and simplified triangulation.

Histogram of shape quality q: initial triangulation

```
minimum : 1.673e-05
moyenne  : 6.837e-01
qualité                  nombre d’éléments      pourcentage
0.0 <= q < 0.1              401410                 4.03   **
0.1 <= q < 0.2              473542                 4.76   **
0.2 <= q < 0.3              472873                 4.75   **
0.3 <= q < 0.4              508798                 5.11   ***
0.4 <= q < 0.5              524215                 5.27   ***
0.5 <= q < 0.6              568447                 5.71   ***
0.6 <= q < 0.7              730414                 7.34   ****
0.7 <= q < 0.8             1131256                11.37   ******
0.8 <= q < 0.9             3629657                36.47   ******************
0.9 <= q < 1.0             1512908                15.20   ********
```

Histogram of shape quality q: optimized and simplified triangulation

```
minimum : 5.204e-01
moyenne  : 8.973e-01
qualité                  nombre d’éléments      pourcentage
0.0 <= q < 0.1                   0                 0.00
0.1 <= q < 0.2                   0                 0.00
0.2 <= q < 0.3                   0                 0.00
0.3 <= q < 0.4                   0                 0.00
0.4 <= q < 0.5                   0                 0.00
0.5 <= q < 0.6                 199                 0.01
0.6 <= q < 0.7                8427                 0.36
0.7 <= q < 0.8              105299                 4.50   **
0.8 <= q < 0.9             1149293                49.12   *************************
0.9 <= q < 1.0             1076451                46.01   ***********************
```

To eliminate the roughness (inherent via a 3D reconstruction from a voxel grid) of this last triangulation, smoothing (without shrinkage) is activated:

```
grid      1
file      schwarz
level     0.0
origin    -12.566371 -12.566371 -12.566371
deltax    0.098560
deltay    0.098560
deltaz    0.098560
optim     1
decim     1
deps      0.7
smooth    1
```

Figures 6 and 7 show the resulting triangulation and an enlargement of the latter (the smoothing phase is carried out in 2.94 seconds). Likewise, to appreciate this triangulation, we find the histogram of the resulting shape quality as well as the diagram of the roughness (with the minimum, maximum and average values) before and after smoothing. This diagram shows for each vertex the maximum angular difference between the unit normal at this vertex and the unit normals of the incident elements (the normal at each vertex is determined from a weighted average of the unit normals of the incident elements).

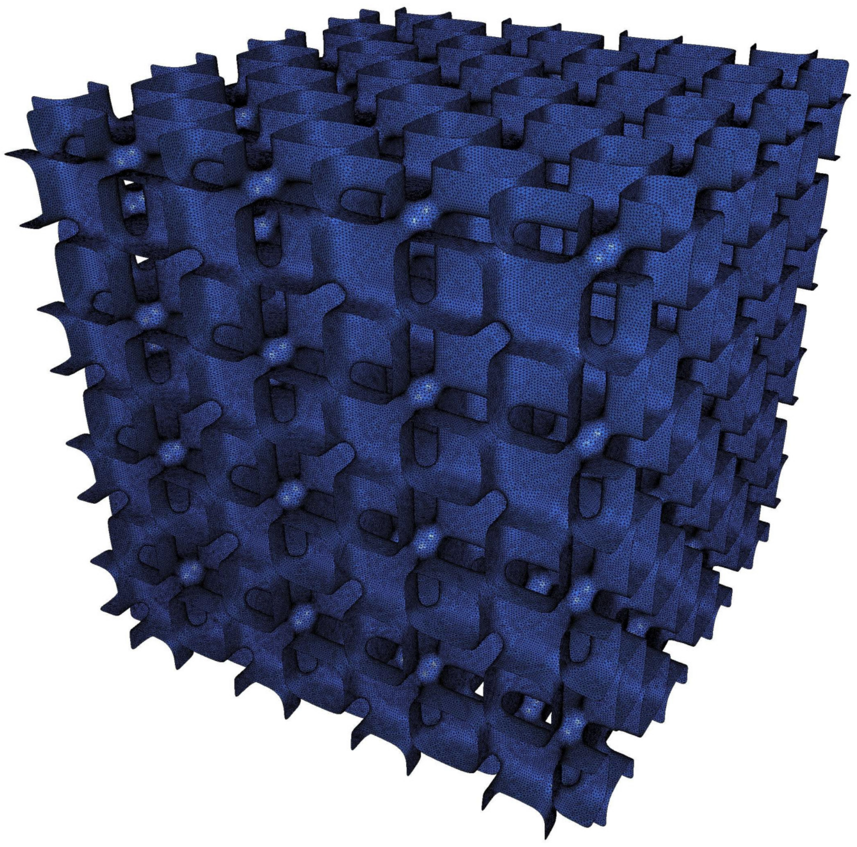


Figure 6: Optimized, simplified and smoothed triangulation.

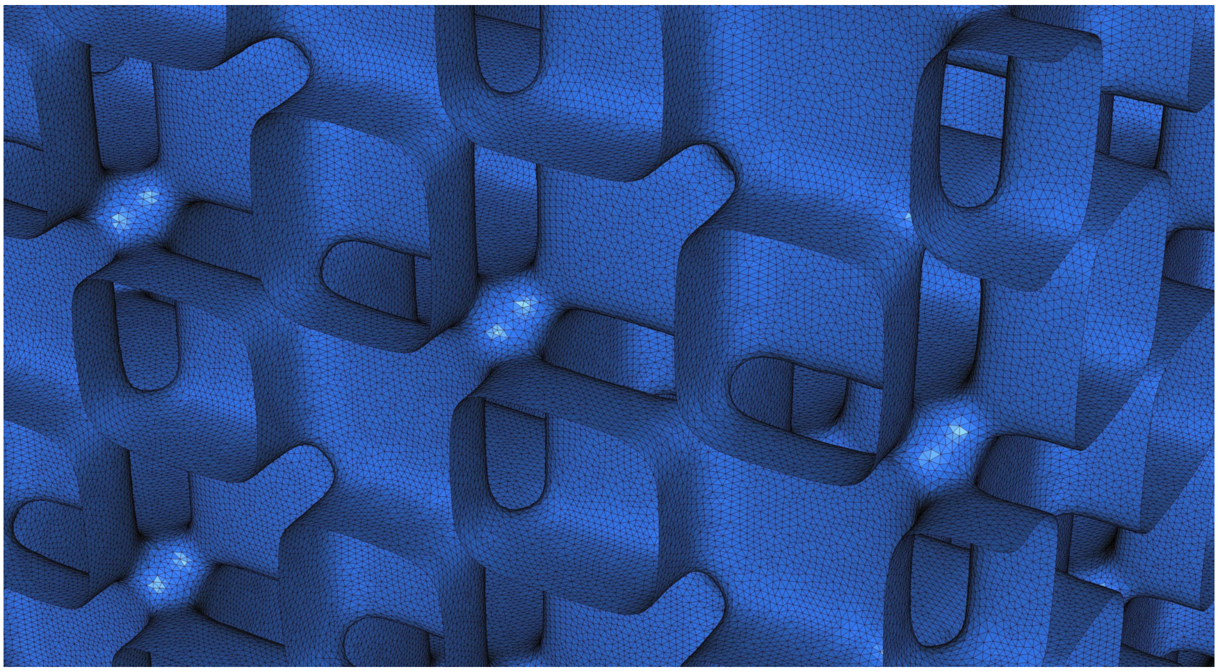


Figure 7: Enlargement of the optimized, simplified and smoothed triangulation.

Histogram of shape quality q: optimized, simplified and smoothed triangulation

```
minimum : 2.646e-01
moyenne  : 9.527e-01
qualité              nombre d'éléments   pourcentage
0.0 <= q < 0.1                    0          0.00
0.1 <= q < 0.2                    0          0.00
0.2 <= q < 0.3                   14          0.00
0.3 <= q < 0.4                    0          0.00
0.4 <= q < 0.5                    0          0.00
0.5 <= q < 0.6                   47          0.00
0.6 <= q < 0.7                 1340          0.06
0.7 <= q < 0.8                22234          0.95
0.8 <= q < 0.9               235902         10.08  *****
0.9 <= q < 1.0              2080132         88.91  ********************************************
```

Roughness histogram r: optimized and simplified triangulation

```
minimum  :  0 degré
maximum : 42 degrés
moyenne   :   5 degrés
rugosité             nombre d'éléments   pourcentage
 0 <=  r <   9              1790718         76.54  **************************************
 9 <=  r <  18               491234         21.00  **********
18 <= r <  27                 56009          2.39  *
27 <= r <  36                  1691          0.07
36 <= r <  45                    17          0.00
45 <= r <  54                     0          0.00
54 <= r <  63                     0          0.00
63 <= r <  72                     0          0.00
72 <= r <  81                     0          0.00
81 <= r <  90                     0          0.00
```

Roughness histogram r: optimized, simplified and smoothed triangulation

```
minimum  :  0 degré
maximum : 34 degrés
moyenne   :  4 degrés
rugosité              nombre d'éléments   pourcentage
 0 <=  r <   9              1921186             82.11   *****************************************
 9 <=  r <  18               399511             17.08   *********
18 <= r <  27                 18787              0.80
27 <= r <  36                   185              0.01
36 <= r <  45                     0              0.00
45 <= r <  54                     0              0.00
54 <= r <  63                     0              0.00
63 <= r <  72                     0              0.00
72 <= r <  81                     0              0.00
81 <= r <  90                     0              0.00
```

We now consider the two possible variants with **grid** = 2 and **grid** = 3. The first is a closed surface of given thickness around the specified level. Here, thickness represents an absolute value. Consider the following environment parameters:

```
grid          2
file          schwarz
level         0.0
thickness     0.1
origin        -12.566371 -12.566371 -12.566371
deltax        0.098560
deltay        0.098560
deltaz        0.098560
optim         1
decim         1
deps          0.7
smooth        1
```

This is a surface centered at the level 0 with thickness 0.1 and closed boundaries, the considered level surface has boundary edges as shown in Figure 8 (edges in red).

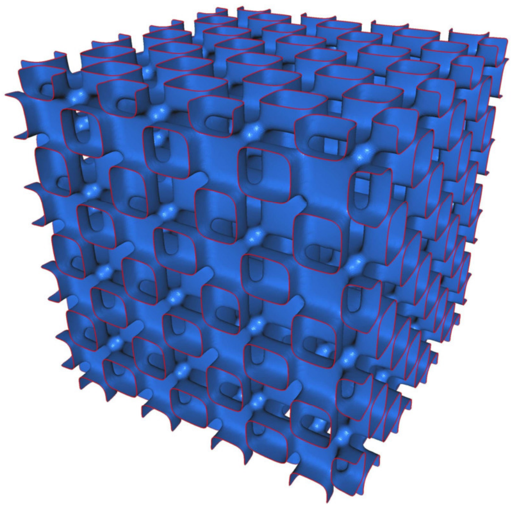
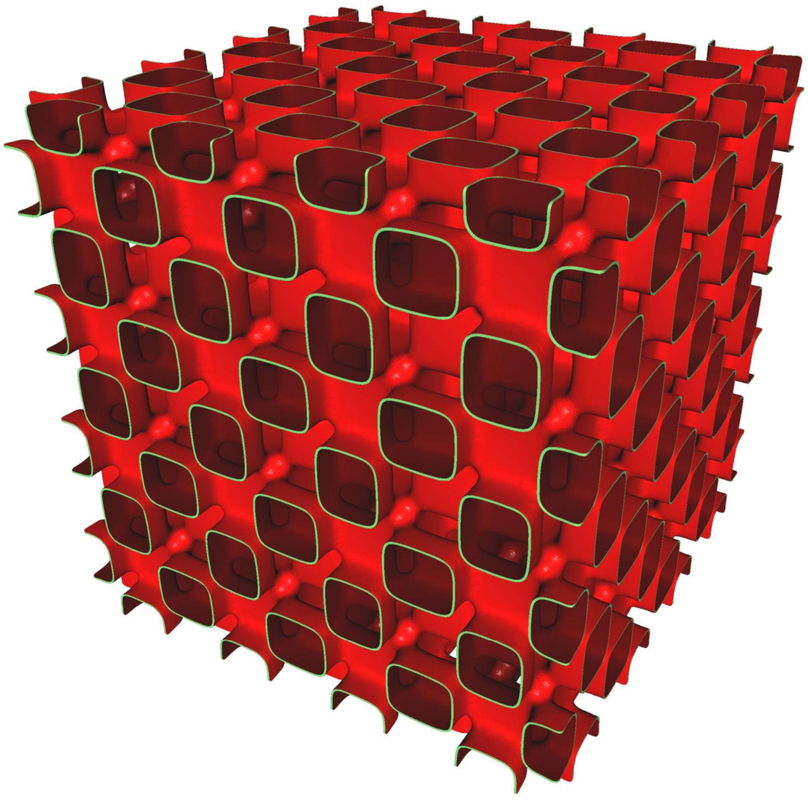


Figure 8: Boundary edges in red of the level 0 surface.

Figures 9 and 10 show the surface and the resulting triangulation with the inner part in dark red, the outer part in red and the closed boundary edges in green and an enlargement of it (the execution time being identical to that for **grid** = 1).



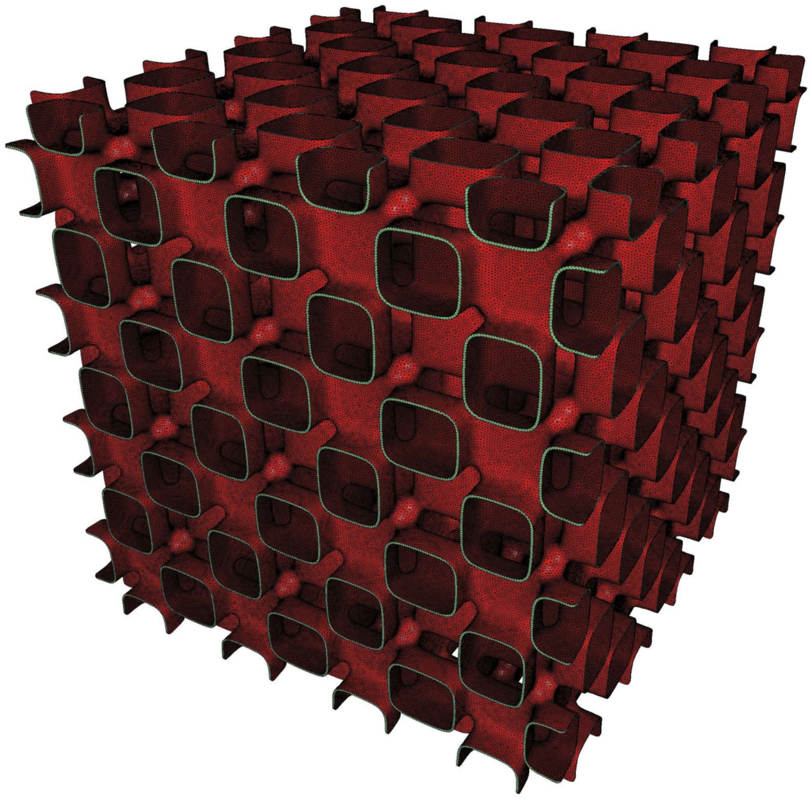


Figure 9: Closed surface of thickness 0.1 centered at level 0.

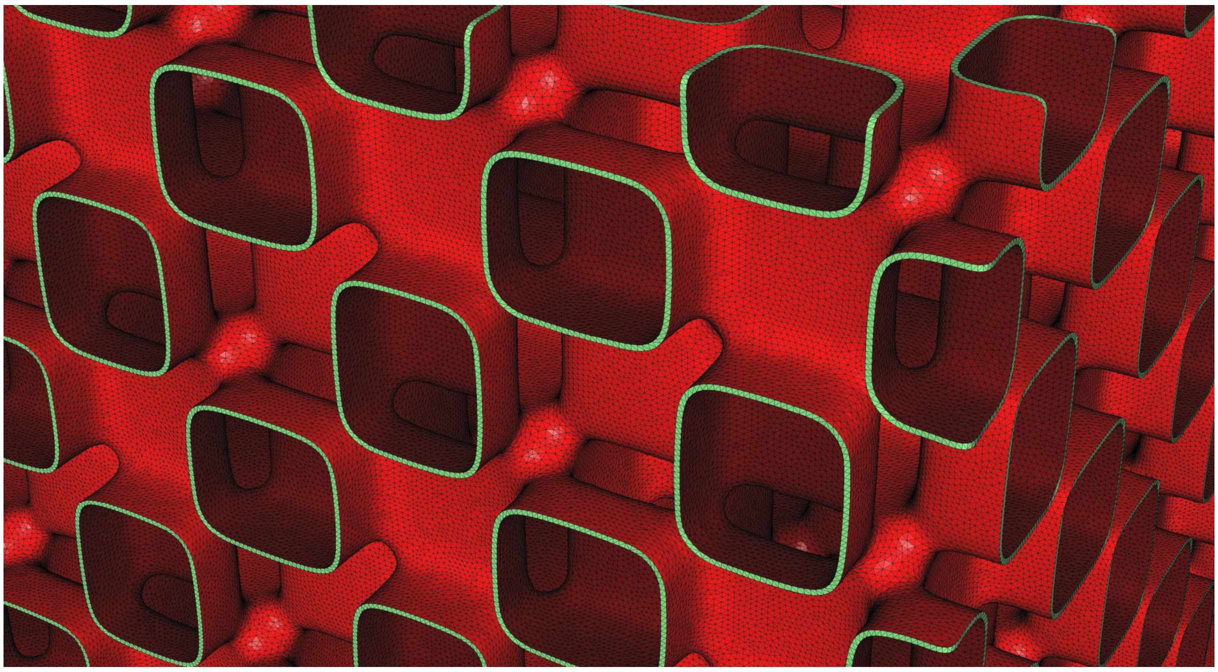


Figure 10: Enlargement of the closed surface of thickness 0.1 centered at level 0.

Considering the specification **grid** = 3, two level surfaces defined by $F = \boldsymbol{level} + \boldsymbol{thickness}/2$ and $F = \boldsymbol{level} - \boldsymbol{thickness}/2$ are generated, these are two offset surfaces on either side of the specified level surface. In this case, thickness is a value relative to the levels of F. Figures 11 and 12 show the corresponding triangulation and an enlargement of this triangulation for the value of thickness = 0.5 (triangulation generated in 22.7 seconds and using 0.502 Giga-bytes in memory).

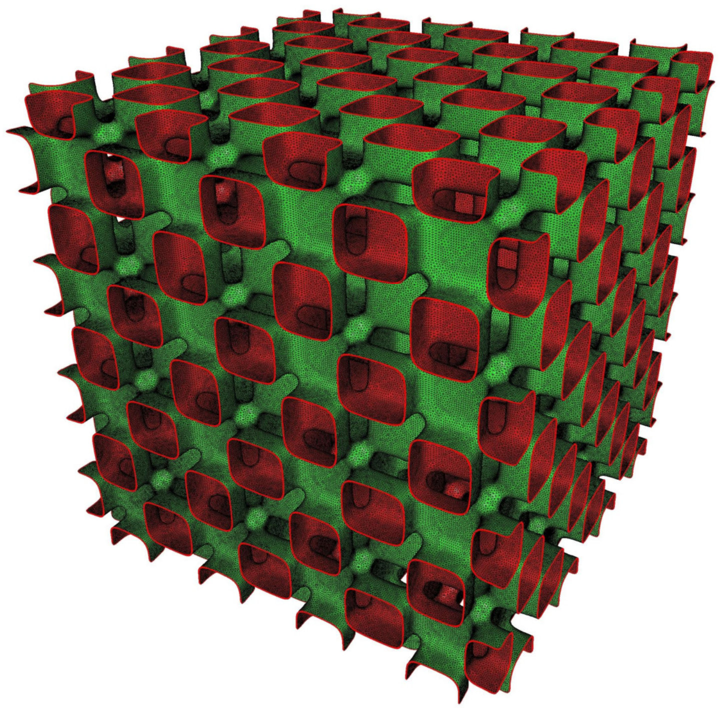


Figure 11: Surfaces of levels -0.25 and 0.25 (centered at 0).

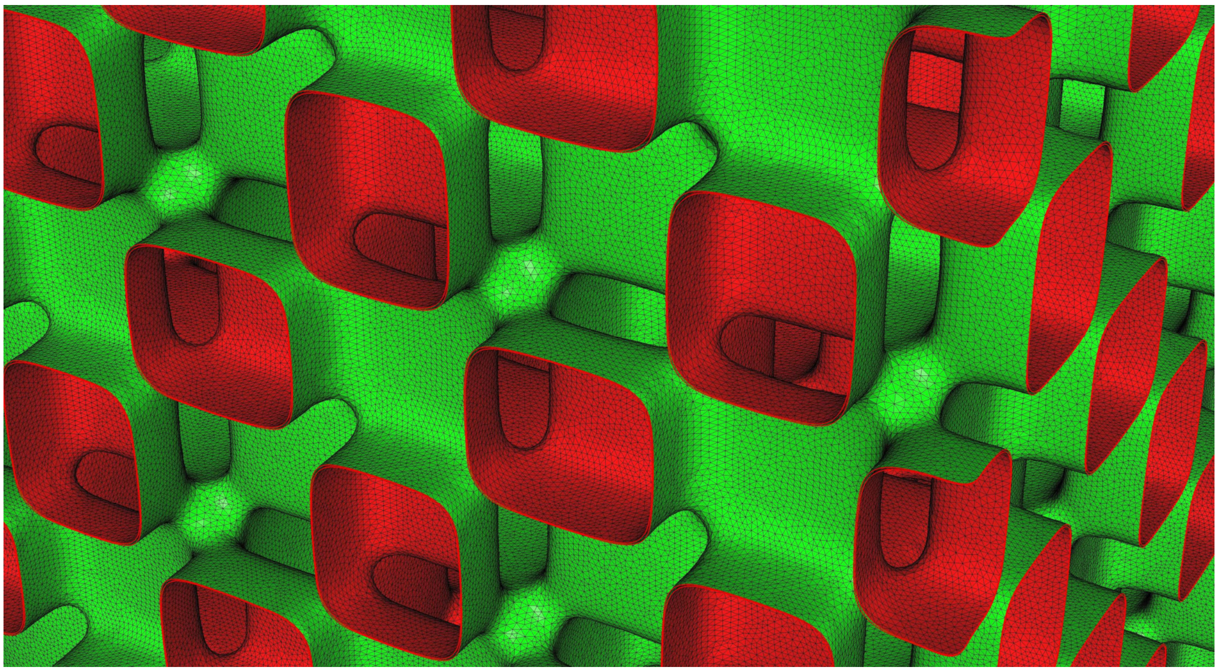


Figure 12: Enlargement of the surfaces of levels -0.25 and 0.25 (centered at 0).

To go further with the first example, we consider different values of **level** with **grid** = 2 and **thickness** = 0.1 by activating the environment parameters **optim**, **decim** with **deps** = 0.7 and **smooth**. Figures 13 to 17 show respectively the triangulations obtained for different values of **level** (the colors representing the different surface components and, in particular, the different connected components).

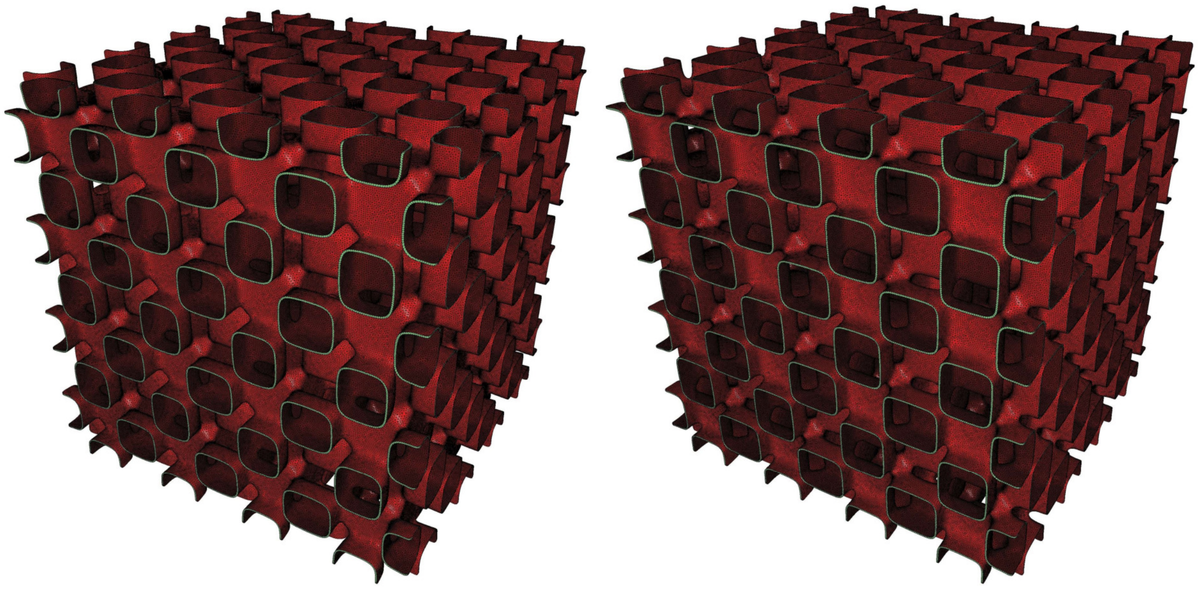


Figure 13: Surfaces of levels $F = -1$and $F = 1$.

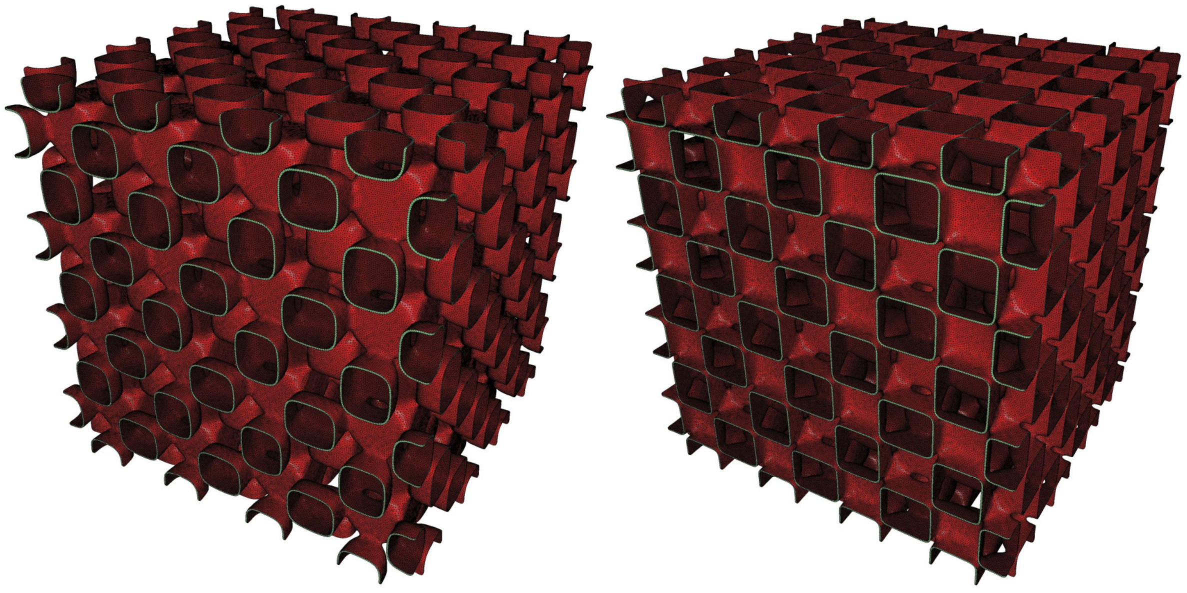


Figure 14: Surfaces of levels $F = -2$ and $F = 2$.

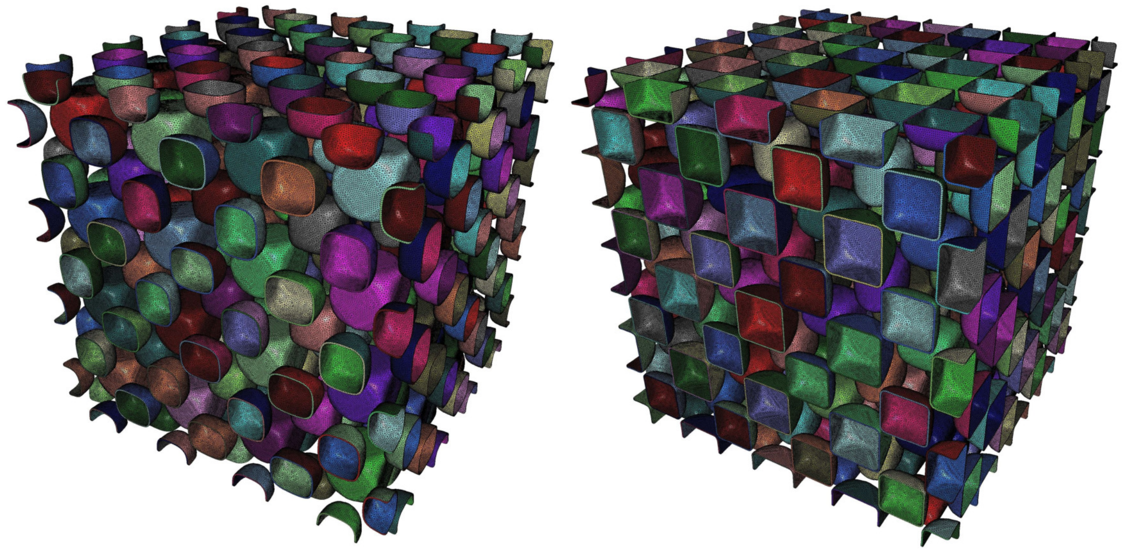


Figure 15: Surfaces of levels $F = -3$and $F = 3$.

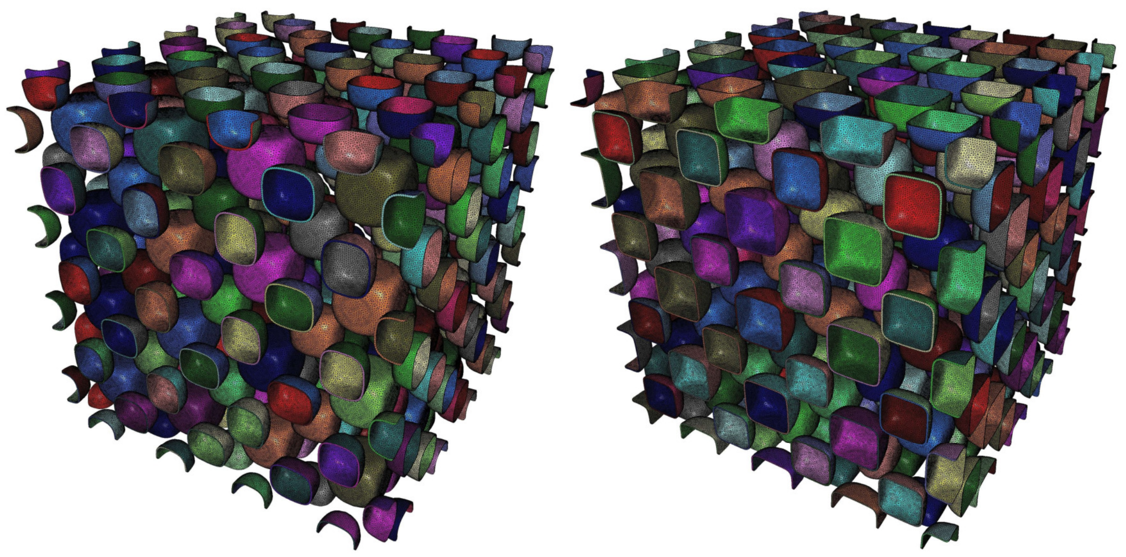


Figure 16: Surfaces of levels $F = -4$and $F = 4$.

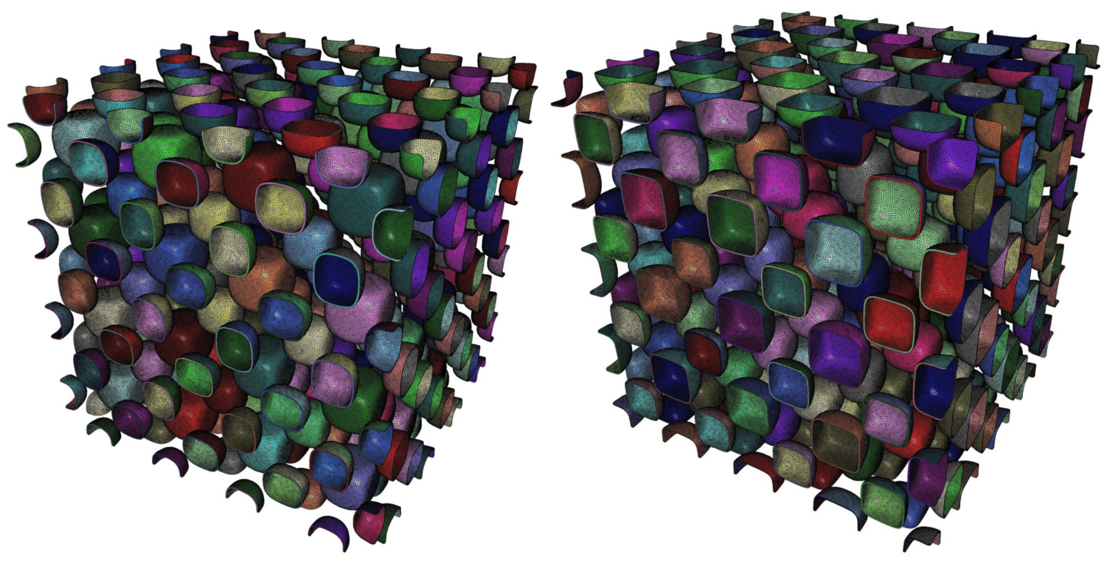


Figure 17: Surfaces of levels $F = -5$ and $F = 5$.

The second example concerns the 3D reconstruction from a set of MRI images. Each image is considered as a 2D grid of values, here gray levels. In each image, a given gray level is representative of a particular organ. The goal is then to reconstruct the surface (the discrete 3D model) corresponding to a given organ (a gray level value). The considered example is made of 158 images of resolution (512,512) with gray levels varying from 0 (black) to 255 (white). In this example, the distance between two consecutive z-level planes is equal to 2 (assuming that the distance between two consecutive pixels is 1). The first image is numbered 10 and the increment of the images is equal to 1. The common name of the images being "Axial", we want to build the 3D model corresponding to the level **level** = 70 by activating all the environment parameters (especially **decim** with **deps** = 0.7). The corresponding 3D model can have several connected components; we are interested in the extraction of the first 3 principal components (remember that the components are sorted in decreasing order of the number of elements). The corresponding environment file is then written as:

```
grid        0
image       Axial
nimages     158
ninit       10
nstep       1
level       70
deltax      1.0
deltay      1.0
deltaz      2.0
optim       1
decim       1
deps        0.7
smooth      1
components  3
```

Unlike a 3D grid, here **grid** = 0 (it is a set of images), the common name is given by **image** = Axial, the first image is numbered **ninit** = 10, the increment of images is **nstep** = 1 and there are **nimages** = 158 images. The values for **deltax**, **deltay** and **deltaz** are set as shown above. Finally, we are interested in **components** = 3 principal components.

Figure 18 shows some images of this set of MRI images.

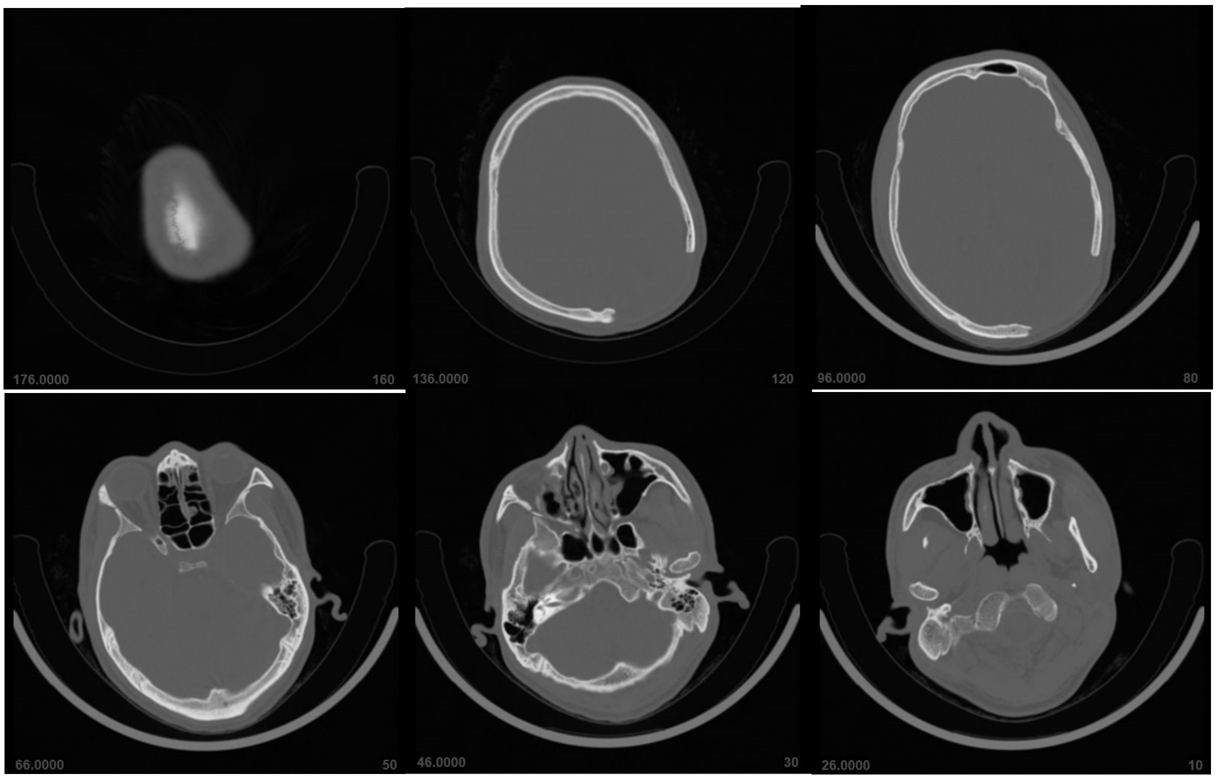


Figure 18: Some MRI images in grayscale.

The resulting triangulation (after the optimization and decimation phases) has 227,931 vertices, 453,795 triangles, and 9,286 boundary edges. The reconstruction was completed in 6.9 seconds and required 0.658 Giga-bytes in memory. Figures 19 and 20 show the three principal components of the resulting surface and an enlargement of these surfaces (each color representing a connected surface component).

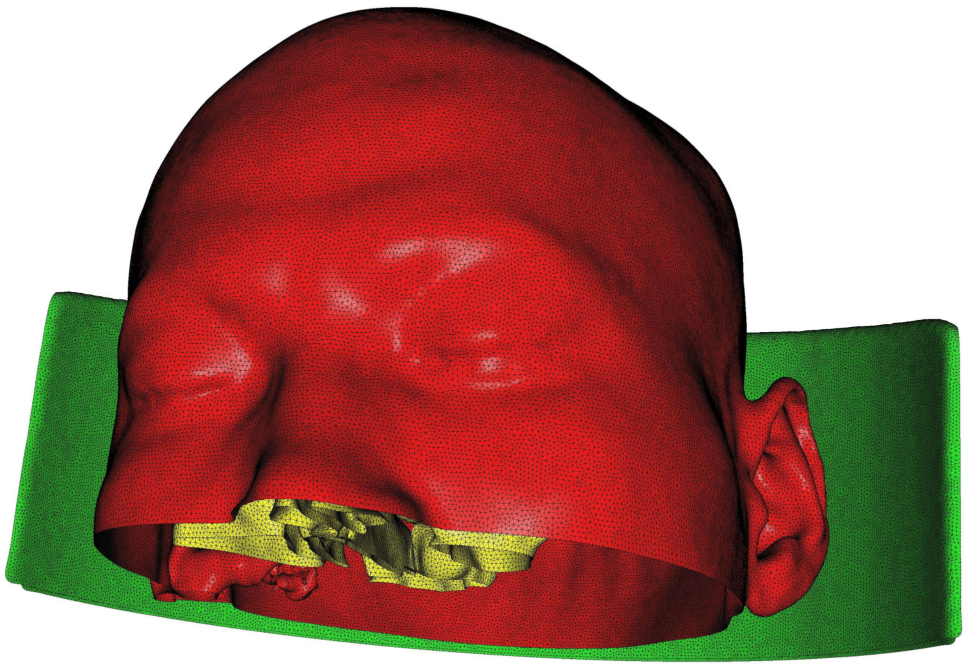


Figure 19: 3D model corresponding to gray level 70.

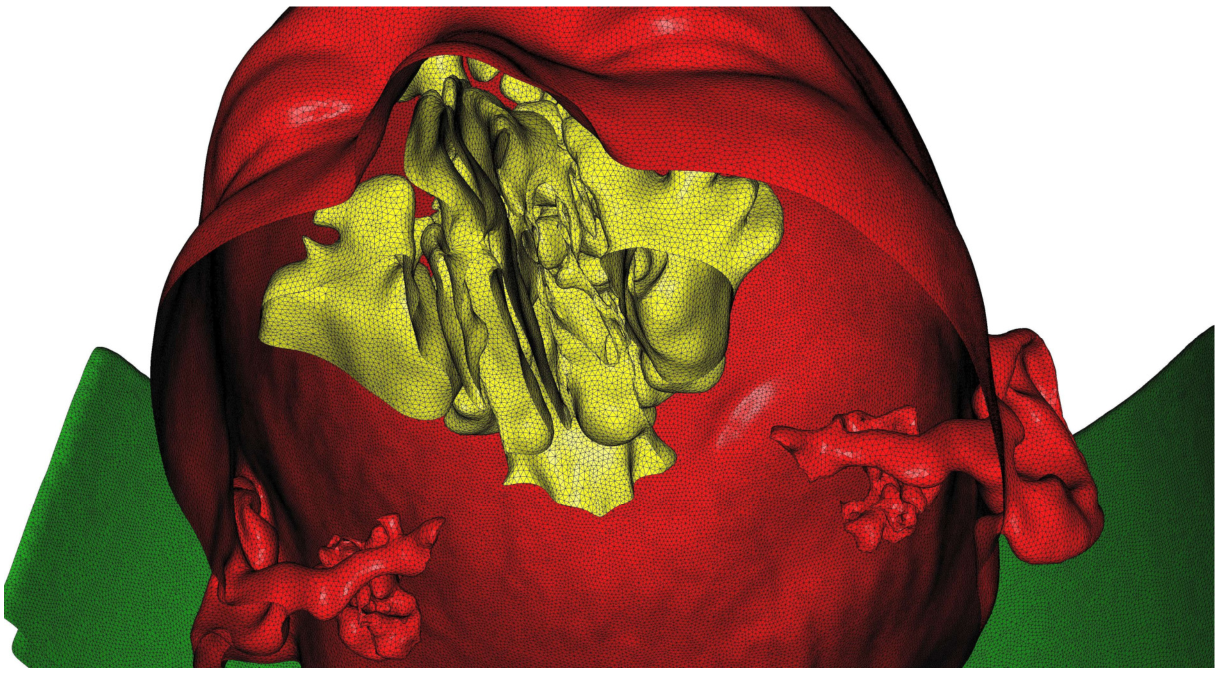


Figure 20: Enlargement of the 3D model corresponding to the gray level 70.

Figure 21 shows the level curves (corresponding to level **level** = 70) in each plane of level z.

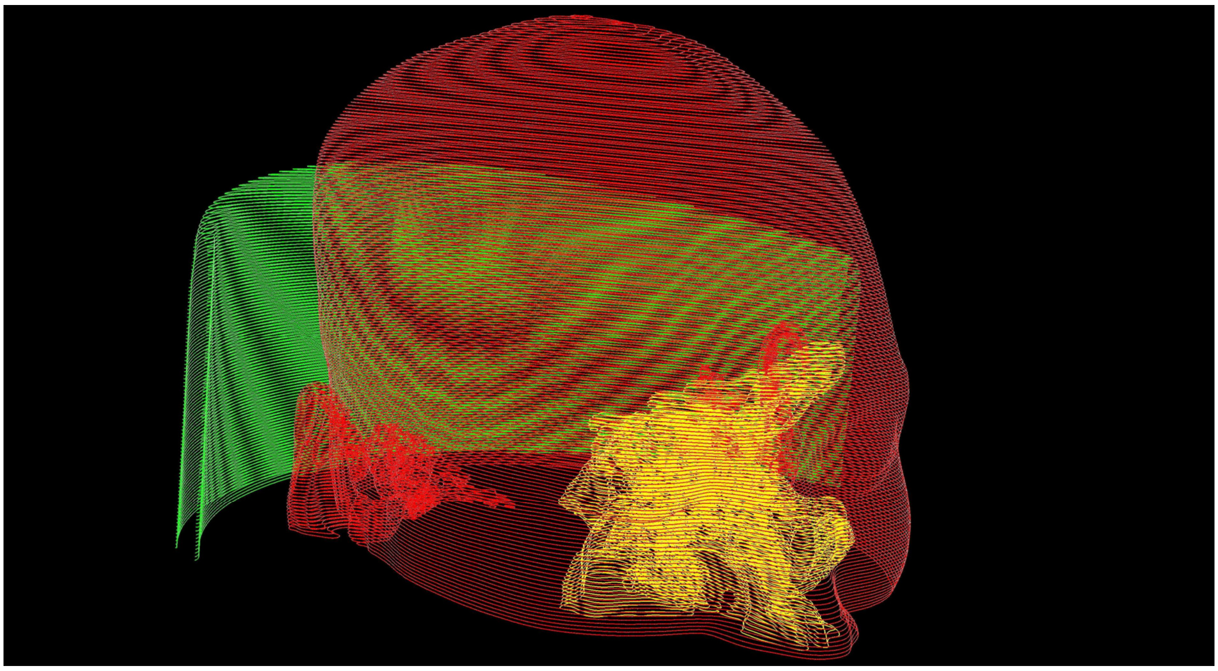


Figure 21: Level curves in z-level planes.

The level 70 surface is made up of 45 connected components, here we are only interested in the 3 principal connected components. Indeed, a given gray level can correspond to several organs or entities, in particular artefacts in the image. A preliminary study must be established to identify the components associated with a given organ, the organ that one wishes to model. In the absence of this information, it can be assumed that the identification could be established from the principal components.

The third example also concerns MRI images. This is a more complex model with 899 images of resolution (512,512). For this model we are interested in the gray level **level** = 100 and the 16 principal connected components. The corresponding environment file is:

```
image       TRAUMA
nimages     899
ninit        1
nstep        1
level       100
deltax       1
deltay       1
deltaz       1
optim        1
decim        1
deps         0.7
components   16
smooth       1
```

Figure 22 shows some MRI images of this model. Figure 23 shows the corresponding 3D model, containing 3,344,714 vertices and 6,705,955 triangles and 14,996 boundary edges, produced in 48.68

seconds and occupying 3,886 Giga-bytes in memory. Figures 24, 25 and 26 show enlargements of the resulting triangulation. Finally, Figure 27 illustrates an enlargement of the level curves in each plane of level z.

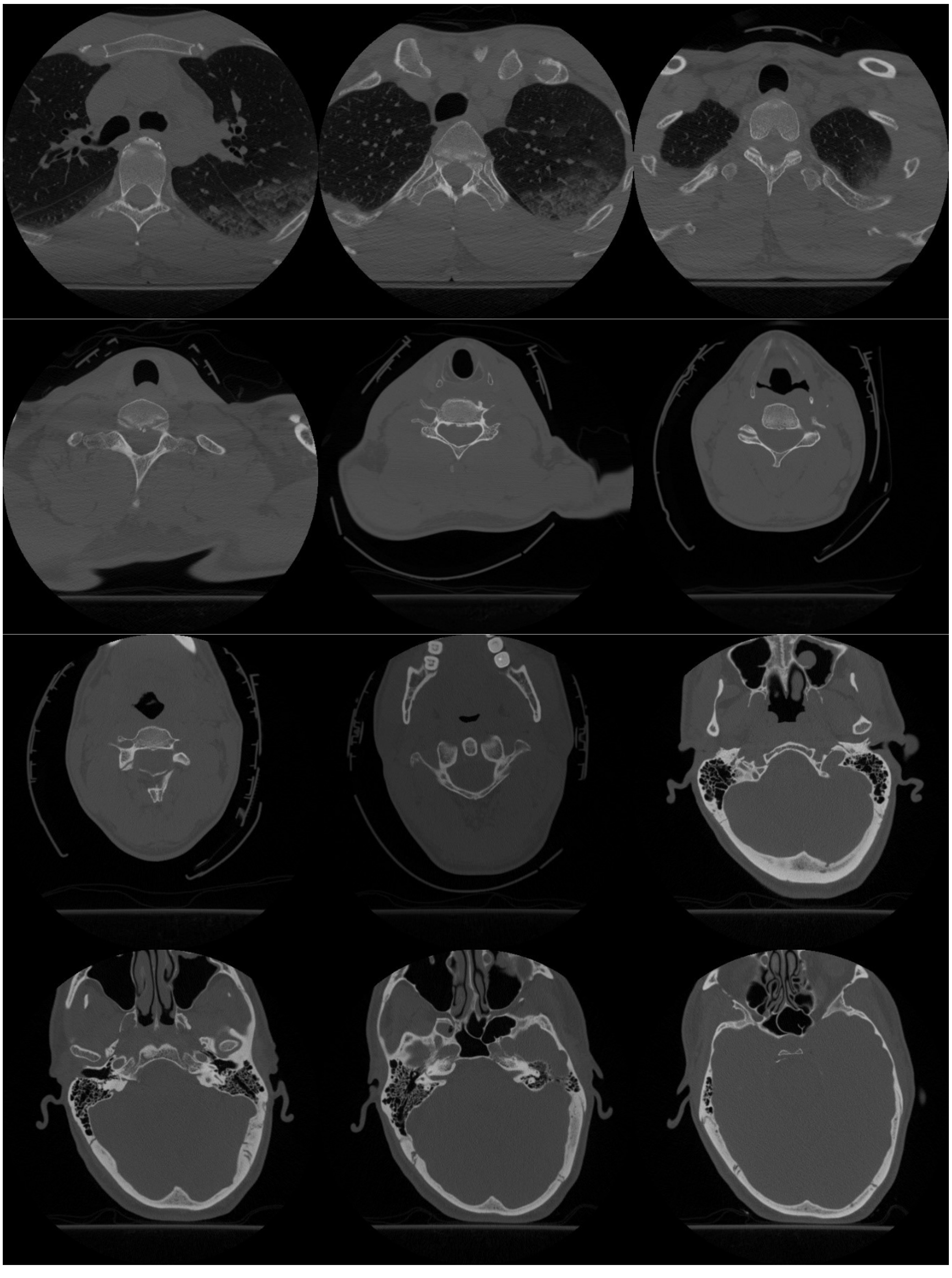


Figure 22: Some MRI images of the model.

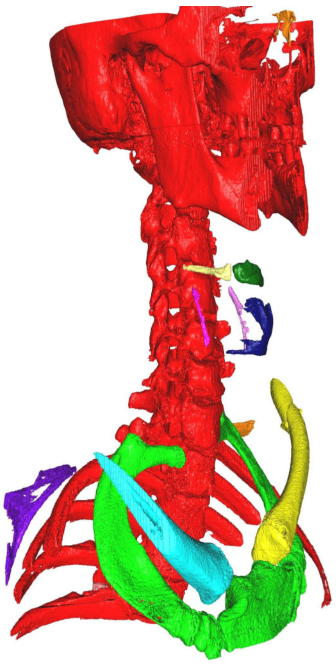


Figure 23: 3D model corresponding to gray level 100.

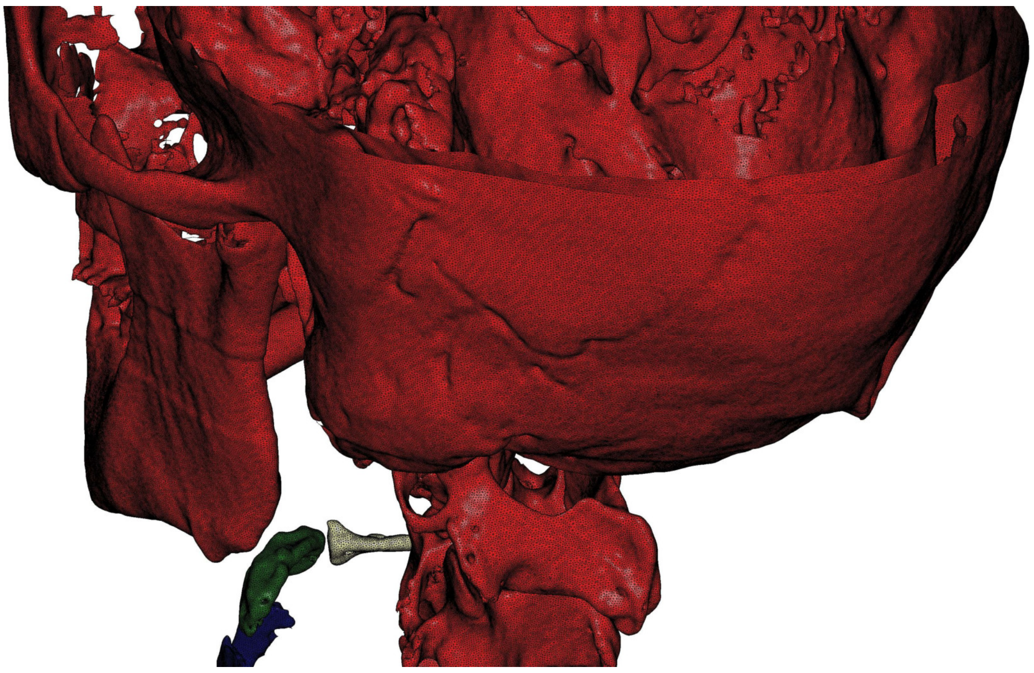


Figure 24: Enlargement of the 3D model corresponding to gray level 100.

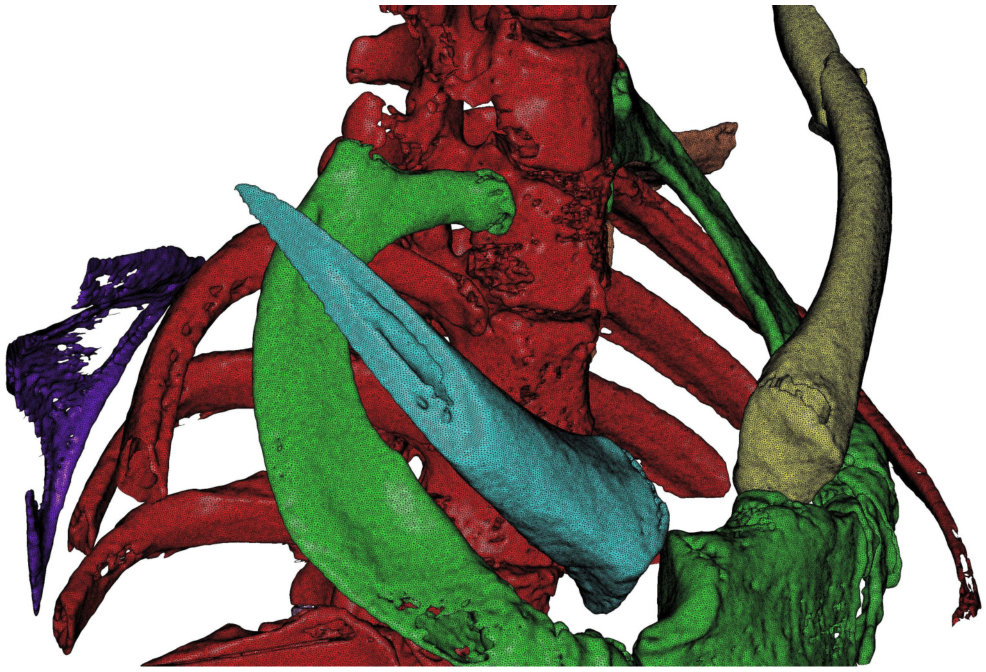


Figure 25: Enlargement of the 3D model corresponding to gray level 100.

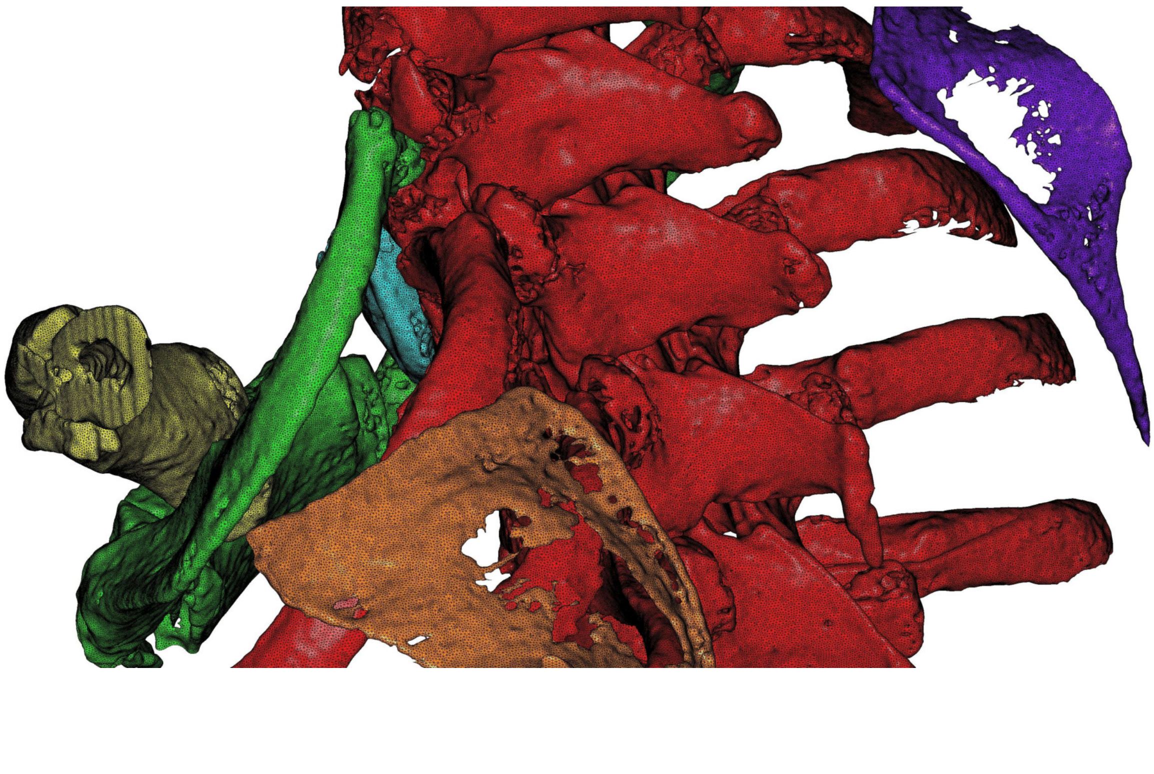


Figure 26: Enlargement of the 3D model corresponding to gray level 100.

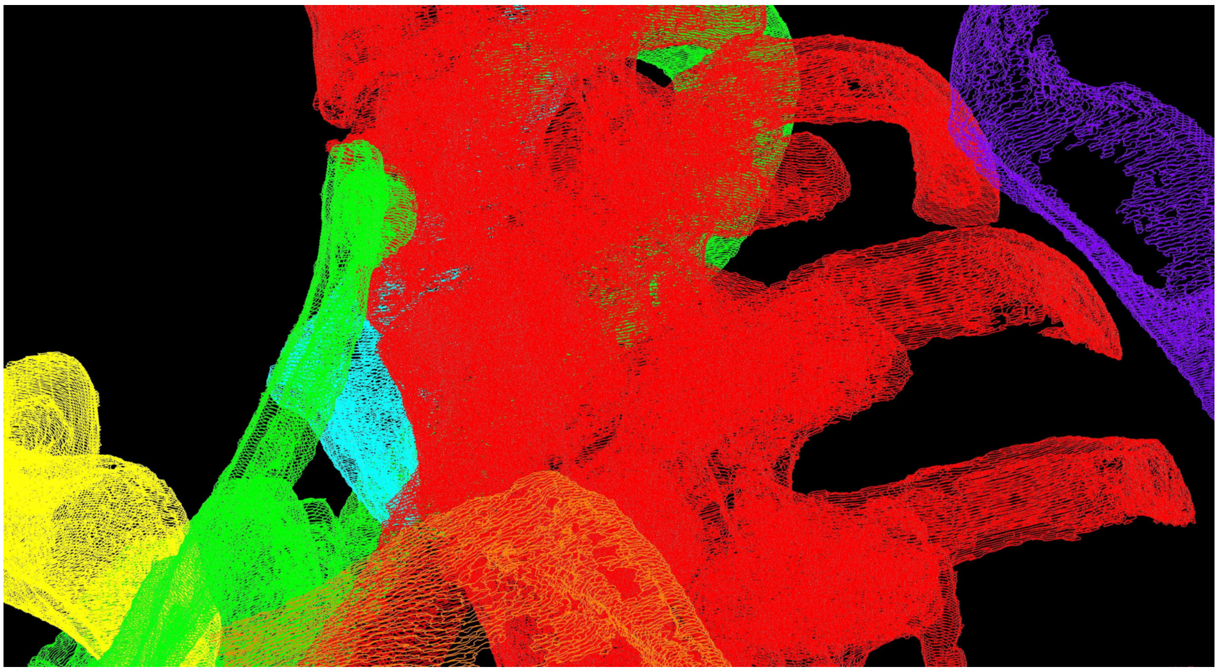


Figure 27: Enlargement of level curves in z-level planes.

The last example also concerns a set of images with only two extreme gray levels 0 (black) and 255 (white). These are MRI images of a metallic foam (Figure 28). In this case, the desired level is 255. For this example, in order to avoid the jagged steps, inherent to the discrete model, the smoothing (the elimination of roughness) is necessary. Figures 29 and 30 show the resulting triangulation without smoothing (and other treatments) and Figures 31 and 32 show the resulting triangulation after the treatments (optimization and decimation) and in particular the smoothing.

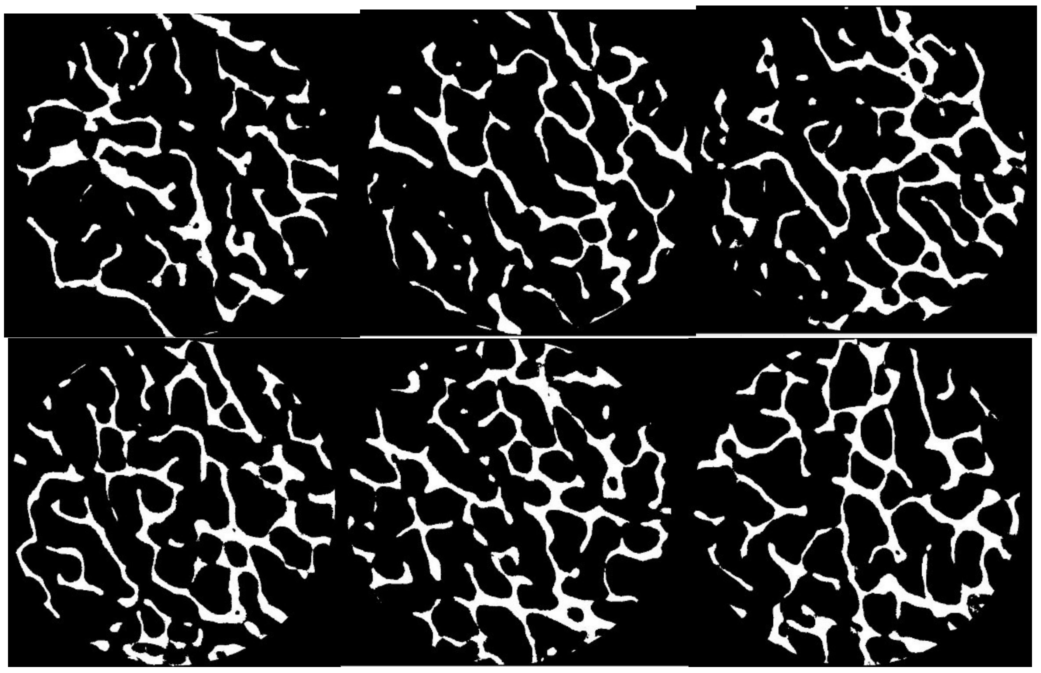


Figure 28: Some MRI images of the metal foam.

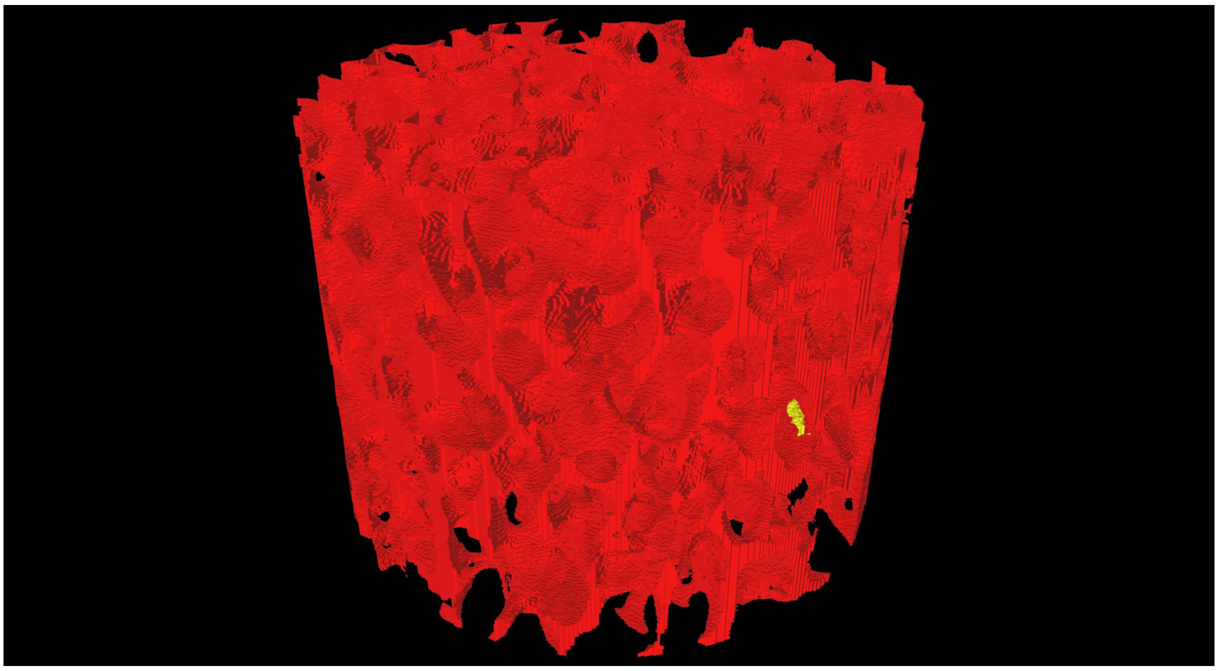


Figure 29: Initial triangulation of the foam.

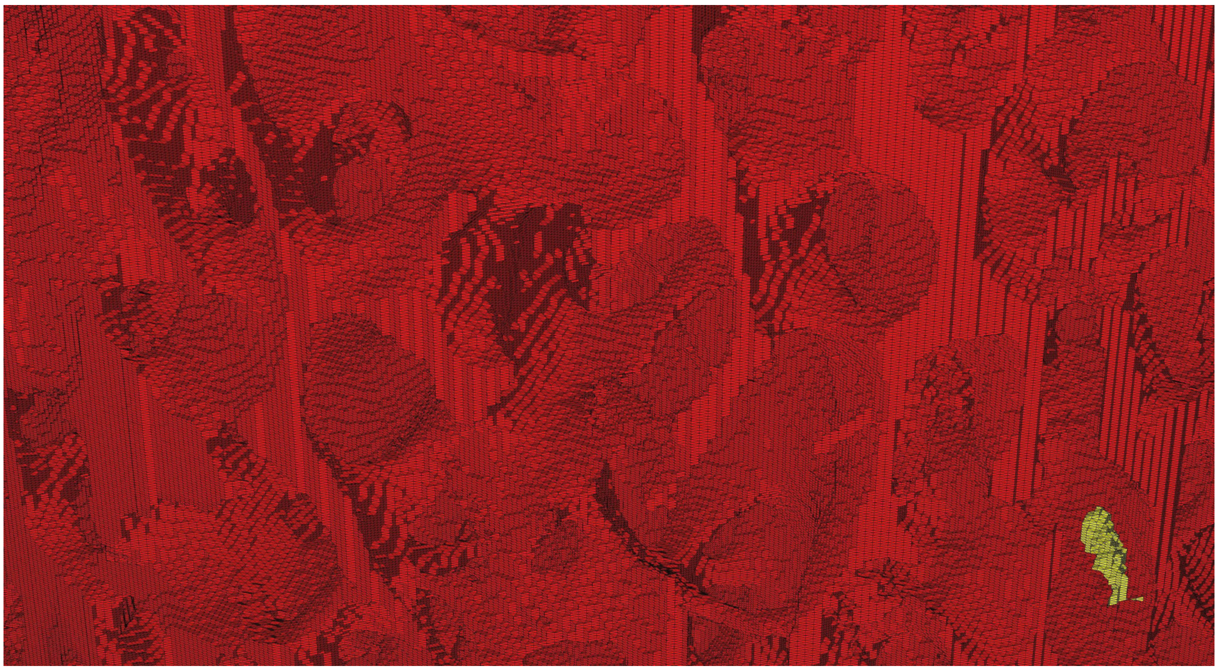


Figure 30: Enlargement of the initial triangulation of the foam.

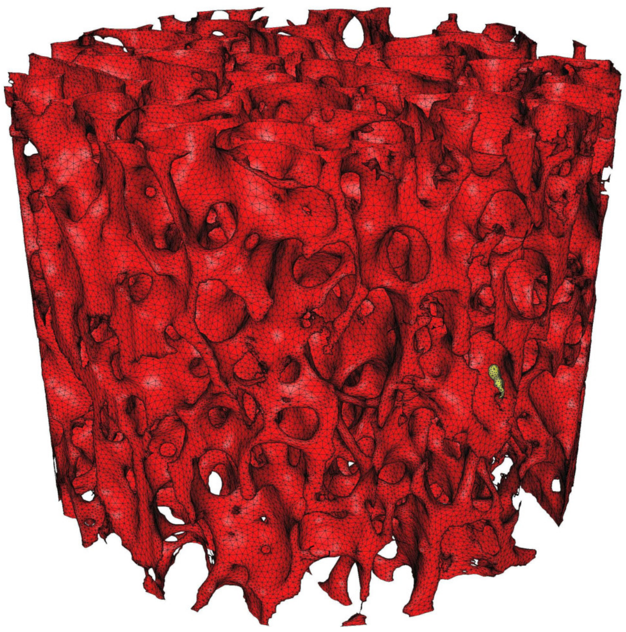


Figure 31: Optimized, decimated and smoothed triangulation of the foam.

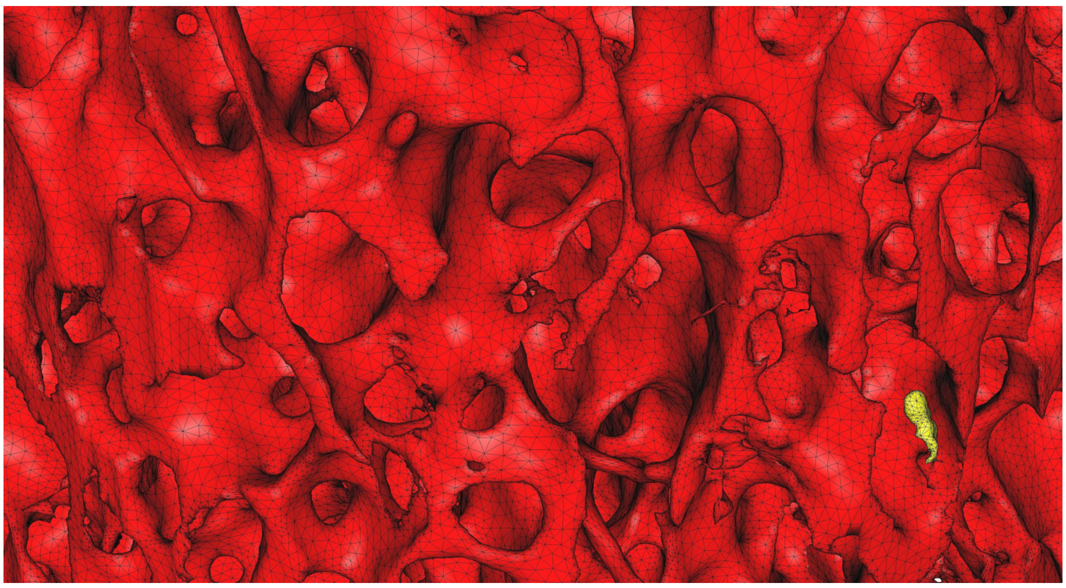


Figure 32: Enlargement of the optimized, decimated and smoothed triangulation of the foam.

Several other examples illustrating complex implicit surfaces are given in the appendix.

4 Perspectives

The next version of the software will include the following features:

- Upstream processing of images, in particular for the elimination of singular level curves.
- Extraction of a surface corresponding to an interval of levels.
- Generation of tetrahedral volume meshes of the model (with optimization and simplification options).

Annex

An ASCII file in *mesh* format is a file describing a given triangulation. In this file, the different components of a triangulation are identified by keywords followed by the description of the corresponding component. The first line of the file specifies the version of the mesh format via the **MeshVersionFormatted** keyword followed by an integer (1, 2, or 3). The following list describes in order only the keywords and the associated components taken into account by PLOUGH3D:

1. **Dimension**. An integer (2 or 3) representing the dimension of the vertices. Here only dimension 3 is considered (triangulated surface).
2. **Vertices**. An integer representing the number of vertices and a list of triplets of real numbers and an associated integer. Each triplet of real numbers represents the coordinates of a vertex, and the associated integer the reference of the vertex. The numbering of vertices starting at 1 is implicit.
3. **Triangles**. An integer representing the number of triangles and a list of integer triples and an associated integer. Each triplet of integers represents the numbers of the vertices of a triangle, and the associated integer the reference of the triangle.
4. **Edges**. An integer representing the number of edges and a list of pairs of integers and an associated integer. Each pair of integers represents the numbers of the vertices of an edge, and the associated integer the reference of the edge.
5. **Ridges**. An integer representing the number of sharp edges and a list of integers representing the numbers of edges (in the edge list) that are sharp.

The last line of the file consists of the single **End** keyword. The components associated with the keywords **Edges**, **Ridges** representing geometric characteristics are optional and are indicated if they are known. An example of a mesh file is given below:

```
MeshVersionFormatted 2
Dimension
3
Vertices
76677
3.716360 0.000000 2.343387 0
4.126565 0.000000 0.642027 0
3.454971 0.000000 2.169877 0
…
Triangles
35450
254 3070 3081 0
3073 202 3074 0
3078 3077 255 0
…
Edges
4245
460 3452 0
3452 3453 0
3453 3454 0
…
Ridges
4245
1
2
3
…
End
```

```c
/********** grid.c **********/

#include        <stdio.h>
#include        <stdlib.h>
#include        <stddef.h>
#include         <string.h>
#include        <math.h>

#define             DPI          3.14159265359

int main(int argc, char **argv)
{
  FILE          *file;
  char          data[256] = "schwarz.grd";
  int           nx, ny, nz;
  double        xmin, xmax, ymin, ymax, zmin, zmax;
  double        x, y, z, dx, dy, dz, f, deltamin, deltamax;
  int           i, j, k;

  deltamin = - 8 * DPI / 2;
  deltamax =   8 * DPI / 2;
  xmin = deltamin;
  xmax = deltamax;
  ymin = deltamin;
  ymax = deltamax;
  zmin = deltamin;
  zmax = deltamax ;
  nx = 256;
  ny = 256;
  nz = 256;
  dx = (xmax - xmin) / (nx-1);
  dy = (ymax - ymin) / (ny-1);
  dz = (zmax - zmin) / (nz-1);
  if ((file = fopen(data, "w")) == NULL)
  {
    printf("   error --> file %s not found\n", data);
    exit(1);
  }
  (void)printf("origin : %f %f %f\n", (float)xmin, (float)ymin, (float)zmin);
  (void)printf("delta  : %f %f %f\n", (float)dx, (float)dy, (float)dz);
  (void)fprintf(file, "%d %d %d\n", nx, ny, nz);
  for (k=0; k<nz; k++)
  {
    z = zmin + k * dz;
    for (j=0; j<ny; j++)
    {
      y = ymin + j * dy;
      for (i=0; i<nx; i++)
      {
        x = xmin + i * dx;
        f = 3*(cos(x)+cos(y)+cos(z)) + 30*(cos(x)*cos(y)*cos(z));
        (void)fprintf(file, "%f ", (float)f);
        if ((i+1) % 10 == 0)
          (void)fprintf(file, "\n");
      }
      (void)fprintf(file, "\n");
    }
  }
  fclose(file);
  return(0);
}
```

The following illustrations are 3D reconstructions of surfaces defined by an implicit equation $F(x,y,z)=0$. For each of these surfaces, a 3D grid, of resolution (256,256,256), of values of F is generated in $[-k\pi,k\pi]^3$ with k an integer. For the generation of 3D models (triangulations), the optimization, simplification and smoothing environment settings are enabled. As an indication, the implicit equations considered are:

- **Schwarz** surface and variants

1. $\cos(x)+\cos(y)+\cos(z)$
2. $\cos(rx)+\cos(ry)+\cos(rz)$ avec $r=1+0.15(x^2+y^2+z^2)$
3. $\cos(rx)+\cos(ry)+\cos(rz)$ avec $r=\frac{100}{1+x^2+y^2+z^2}$
4. $\cos(rx)+\cos(ry)+\cos(rz)$ avec $r=2+0.3\cos(x)\cos(y)\cos(z)$
5. $\cos(0.7x^2)+\cos(0.7y^2)+\cos(0.7z^2)$
6. $\sin(x)\sin(y)\sin(z)+\sin(x)\cos(y)\cos(z)+\cos(x)\sin(y)\cos(z)+\cos(x)\cos(y)\sin(z)$
7. $\sin(rx)\sin(ry)\sin(rz)+\sin(rx)\cos(ry)\cos(rz)+\cos(rx)\sin(ry)\cos(rz)+\cos(rx)\cos(ry)\sin(rz)$ avec $r=1+0.15(x^2+y^2+z^2)$

- **Neovius** and variants

8. $3(\cos(x)+\cos(y)+\cos(z))+4\cos(x)\cos(y)\cos(z)$
9. $3(\cos(rx)+\cos(ry)+\cos(rz))+30\cos(rx)\cos(ry)\cos(rz)$ avec $r=1+0.15(x^2+y^2+z^2)$

- **Gyroid** and variants

10. $\sin(x)\cos(y)+\sin(y)\cos(z)+\sin(z)\cos(x)$
11. $\sin(rx)\cos(ry)+\sin(ry)\cos(rz)+\sin(rz)\cos(rx)$ avec $r=1+0.15(x^2+y^2+z^2)$

- **Fisher** surface and variants

12. $\cos(2x)+\cos(2y)+\cos(2z)+2(\sin(3x)\sin(2y)\cos(z)+\cos(x)\sin(3y)\sin(2z)+$ $\sin(2x)\cos(y)\sin(3z)+\sin(2x)\cos(3y)\sin(z)+\sin(x)\sin(2y)\cos(3z)+\cos(3x)\sin(y)\sin(2z))$
13. $\cos(2rx)+\cos(2ry)+\cos(2rz)+2(\sin(3rx)\sin(2ry)\cos(rz)+\cos(rx)\sin(3ry)\sin(2rz)+$ $\sin(2rx)\cos(ry)\sin(3rz)+\sin(2rx)\cos(3ry)\sin(rz)+\sin(rx)\sin(2ry)\cos(3rz)+$ $\cos(3rx)\sin(ry)\sin(2rz))$ avec $r=1+0.15(x^2+y^2+z^2)$

- **Prime-D**

14. $\sin(x)\sin(y)\sin(z)+\cos(x)\cos(y)\cos(z)-\cos(2x)\cos(2y)-\cos(2y)\cos(2z)-\cos(2z)\cos(2x)$

- Other surface

15. $x\cos(y)\cos(z)+y\cos(x)\cos(z)+z\cos(x)\cos(y)$