GENERATION OF A SUITABLE SURFACE OR VOLUME MESH FROM SCAN DATA

Gustavo Lunardon Quilló*[a], Éliton Fontana[a], Luiz Fernando de Lima Luz Jr[a]

*Corresponding author: *Federal University of Paraná – UFPR – PR, gustavoquillo@hotmail.com*
[a] *Dept. of Chemical Engineering, Federal University of Paraná, P.O. Box 19011 Curitiba, PR, Brazil*

A procedure to generate a suitable surface and volume meshes from image sequences or other scan data is presented. The methodology gives preference to readily available free software for the final goal of using the generated mesh for computational fluid dynamics or finite element simulations. The steps involve the extraction of a surface mesh from the image sequence, segmentation, trouble-shooting, treatment, refinement/coarsening, smoothing, translation into a volume mesh and post-processing. The user controls the detail of the final mesh through a series of algorithms included in the suggested software. The methodology is illustrated with a computer microtomography data of a carbonate rock. Finally, the mesh is imported in ANSYS® Fluent to demonstrate that the resulting mesh can be used in simulations.

1 Introduction

Reconstructing a suitable volume mesh from image sequences taken from micro-computed tomography (micro-CT) or magnetic resonance imaging (MRI) is a recurrent challenge for fields that deal with objects that have complex geometry or internal or structure, e.g. porous media, internal organs, microstructured devices. These simulations and models are meant for geological applications, developing new chemical products, surgery planning, diagnostics, additive manufacturing (e.g. 3D printing) and more.

Many commercial software products are able to perform all or several steps described in the procedure, e.g. Amira® [5], Simpleware® [32], Avizo® [5], 3D Doctor® [1], MATLAB® [19]. Their license costs are, however, prohibitively expensive for occasional projects or small research groups so one of the major concerns in this study is to suggest some free alternatives to complete this task. These free software products usually perform just one or two steps of the procedure, however, they are open-source, which gives the opportunity for developers to tailor the code to their specific needs.

The steps involve a series of manipulations that may lower the resolution of the original dataset in order to create a manageable working mesh for the computer fluid dynamics (CFD) simulator, finite element analysis (FEA) or 3D printing. These operations are not mandatory but they are advisable depending on the hardware, time constraints and application. In addition, only relevant features that were of great use in the present work have been described. The information is not exhaustive and the pertinent documentation should be consulted to look for the specifics of each algorithm.

The software products used in the procedure are displayed in Table 1, along with their version and a brief description. Other free software products that may be of general use in image and mesh processing but were not included in Table 1 are FEBio [8], GIBBON code [10], Salome [31], Graphite [12], ImageVis3D [15], Paraview [27], OpenFlipper [25], Freecad [9], Blender [6], AI-FEMesh [13] and pyFormex [28].

Some of the programs mentioned in this work do not offer a directly executable file so the user has to manually compile the code from the source code files in order to produce a .exe file in Windows (e.g. Tetgen [35]). This can be done by making a project for a console application in Codeblocks [7] after downloading and installing MinGW [23]. Note that the flags static libgcc, static libstdc++ and static linking should be checked in the compiler settings before building the project. If the interface of the program is directly in the Windows command prompt (e.g. Tetgen [35]), execute cmd to call the prompt, write cd to change the directory to the .exe file to implement the commands.

Table 1 List of programs and tools to generate a volume mesh from scan data

Name	Version	Description
Tetgen [35]	1.5.0	Free software able to create quality and adaptive tetrahedral meshes for finite volume or finite element methods. Generates constrained Delaunay tetrahedralizations, Voronoi partitions and reports the details of the output mesh. In addition, it has mesh manipulation tools for both refining and coarsening. If used as a standalone program, its output can be visualized in Tetview, also available from the same developer.
ImageJ [14]	1.51j8 with Java 1.8.0_112	Open-source, Java-based software specialized in image processing tools. Its versions and user-written plugins extend its already impressive capabilities by a large amount. Moreover, it can import, read and write many file formats, turning it into an excellent conversion tool. Furthermore, this software is able to remove a selected portion of the image geometrically or using intensity values, alter the intensity values in the image, manipulate the order of images in a stack and more.
iso2mesh [16,29]	2017 (1.8) codenamed Devil Egg	Open-source toolbox for mesh generation and processing in MATLAB or GNU Octave. It is able to create 3D tetrahedral finite element mesh from surfaces (e.g. .stl files), 3D binary and greyscale volumetric images such as MRI or CT scans. It includes CGAL and Tetgen.
ITK-SNAP [17]	3.6.0	Open-source software application specialized in segmentation of image stacks focusing on, but not limited to, 3D medical images. It provides a convenient semi-automatic segmentation tool using active contour methods and can export a surface mesh.
MATLAB [19]	2015b	Versatile commercial software that excels in matrix operations and numerical computing. This software has interface with many other programming languages such as C, C++, Python, Java and Fortran. Uses a high-level programming language and is extensible by a plethora of toolboxes that may be user-written and free (e.g. iso2mesh, stlTools [34]) or developed by Mathworks and sold separately, e.g. Image Processing Toolbox.
Meshlab [21]	64bit v1.3.4BETA	Open-source software for the processing, manipulating and fixing of unstructured 3D meshes, offering a plethora of filters and rendering options. Also displays the quality of the mesh according to a criterion (e.g. area/max side).
Meshmixer [22]	v3.4.35	Free software with powerful mesh generation and manipulation tools. Supports 3D printing and offers a very practical tool for inspecting the mesh for errors and correcting them accordingly, which is useful when correcting bad surface meshes.

2 Method

2.1 Extract information from the sequence of images: generate a surface mesh

The first step is to segment the representation of the image sequence into something meaningful or easier to analyze. This step determines the boundary of the object, which separates what is of interest in the image sequence. For example, the greyscale scan data is processed by selecting appropriate upper and lower thresholds to binarize the image sequence, isolating the part that ultimately will be converted into the surface or volume mesh. Threshold values serve to differ the shades of gray and can be in many scales, e.g. Hounsfield units (a quantitative scale for describing radiodensity) or pixel values (a number between 0 and 255), where each pixel is be represented by 8 bits.

This step is mostly visual, as the segmented part normally appears on the screen of the software and the user can choose a threshold value via trial and error. The thresholding can be done via ImageJ [14], along with other features such as enhance contrast and other image processing tools. There are also variations of ImageJ (e.g. Fiji , BoneJ, Bio7) and the official site offers several plugins that extend this software capabilities. The resulting image can be exported as a Raw Binary file (.raw) to make it readable by ITK-SNAP [17]. ImageJ also provide means to convert the image sequence into other formats, like .tif or .tiff, creating an image stack. Depending on format of the original files (e.g. DICOM .dcm), the segmentation procedure can be performed directly in ITK-SNAP, which has a more user-friendly interface.

The next step in segmentation is carried out in ITK-SNAP, importing the Raw Binary file exported by ImageJ. ITK-SNAP will request the quantity of pixels in each direction of the imported image file, which is available in ImageJ on the top left side of the screen. ITK-SNAP is equipped with a segmentation feature called Active Contour (a.k.a. Snake) [26], which is divided into 3 sub-steps.

The first substep is the presegmentation, which selects of the upper and lower threshold values. ITK-SNAP also allows the user to include the intensity of the neighboring voxels (i.e. 3D pixels) as a criterion in the classification.

For the second sub-step, called bubble growth, the user has to place bubbles in any of the three planes that will grow into the region of interest (ROI). Depending on the complexity of the ROI, adding enough bubbles to capture a good representation of the whole object can be a time-consuming task. In addition, the ITK-SNAP allows a stepwise region of interest segmentation so the user can also circumvent the problem of adding undesired parts into the final ROI, which may happen due to parts that are too similar in terms of pixel intensity. Thus, it may be helpful to add small regions of interest systematically and update the ROI instead of completing the entire segmentation at once.

The third sub-step is the evolution of the bubbles until they hit the boundaries of the object. The user can accept the representation after inspecting if the algorithm could capture a good approximation of its actual shape. If there are regions that the bubbles are taking too much time to reach, the user can go back one sub-step and add more bubbles in the problematic region. Once done, ITK-SNAP can export the obtained ROI as a surface mesh in Stereolithography (.stl) format. This semi-automated routine will also generate a suitable surface mesh that is much easier to process later on than the surface mesh directly extracted from the image sequence, an option that also exists in ImageJ.

2.2 Mesh processing and repairing

The segmentation and surface mesh extraction process explained in ITK-SNAP produces a better surface mesh because this eliminates most of undesired details of the object. The resulting surface mesh, however, must be processed to reduce the number of elements and fix possible issues (e.g. duplicated or non-manifold elements, zero area faces, intersecting faces, holes). The aim is to obtain a workable, manifold mesh without defects and has just enough detail (elements) for the final task.

Any corrections must be applied iteratively during each mesh reconstruction (e.g. mesh reducing, subdividing, smoothing). Mesh simplifying makes editing faster and decreases the size of the file at the cost of the quality of the surface, but also may create new problems during the operation. Thus, it is paramount to look for mesh errors at least after every major operation. Furthermore, any filters may slightly deform the topology of the mesh so it is necessary to judge if the change in geometry is acceptable.

Meshlab [21] and Meshmixer [22] are free tools to for editing, cleaning, healing, inspecting and converting meshes (e.g. tri to quad). The surface mesh should be imported in both of these programs for remeshing and repairing any issues that may arise, complementing the shortcomings of each software product. Note that mesh reduction can be smarter than simply choosing a triangle budget or percentage reduction if the user opts for an algorithm like "max deviation" or "adaptive", which allows the algorithm to reduce more faces where detail is less important (e.g. flat surfaces) instead of reducing the quality of the surface uniformly. On the contrary, if the mesh is too coarse, algorithms like Butterfly subdivision in Meshlab may be convenient. In addition, enabling the wireframe view facilities troubleshooting and aids the decision-making process.

Meshmixer is a powerhouse in terms of mesh manipulation tools. It can import and export surface meshes in formats other than .stl (e.g. .obj, .3mf, .ply and .vrml), some of which can preserve color data. It can also scale the surface mesh in the desired units (e.g. milimeters), as .stl files do not store this information. Another feature includes a managing tools to delete separate shells (e.g. closed surfaces), e.g. to eliminate pieces of the object that are floating in midair. Furthermore, it is trivial to inspect the mesh for errors (e.g. holes) and correct them accordingly, if the problems are not critical. If they are, the user can try to remesh the whole model according to another criteria (e.g. mesh density and regularity) or use the reduce feature for decimating the triangles and inspect the mesh again for errors. Lastly, another robust option that can correct badly formed .stl files is the "make solid" function using a low offset distance, what also leads to a compromise with the quality of the mesh, especially around sharp edges.

An important way to reduce the number of elements in Meshlab is the Quadric Edge Collapse Decimation, where the user can choose the output number of faces along with options to preserve topology of the mesh. Moreover, the user can check the mesh volume, area and if it is watertight by computing geometric measures. Meshlab also provides visual information on the triangle quality, i.e. using the violet-side of the spectrum to color well-shaped triangles (nearly equilateral) and using red to stress triangles with very small or large angles (distorted faces). If the mesh has too badly shaped triangles, smoothing algorithms such as the MLS Projection, Laplacian Smoothing and Taubin Smoothing may mitigate the problem. It is important to notice that Meshlab does not have an "undo" feature so the user must save the file periodically and use the preview option, when offered. If there are issues that could not be solved either in Meshmixer or Meshlab, the user may try the Netfabb Cloud [24] to repair the file.

At this point, the surface mesh would be ready for 3D printing using, for example, a slicer like Simplify 3D [33] or Meshmixer itself.

2.3 Conversion from surface mesh to NURB

If the intended use of the mesh is finite element analysis, one can import the surface mesh as a .stl file into the commercial software Rhinoceros® [30] (e.g. Rhino 6) for a reconstruction process triggered by the command *MeshToNurb* in Rhino.

Mesh and NURB (Non-Uniform Rational B-Splines) are different ways to represent 3D objects [20], as a surface mesh is a series of facets, i.e. the surface is discretized into smaller elements. Contrastingly, NURBS are mathematical representations of the surfaces, so they can generate complex free form surfaces that are smooth *per se*.

The command *MeshToNurb* will translate a faceted mesh to a degree 1 NURBS structure by creating one NURBS surface for each mesh face in the original object and assemble the result to form a polysurface [20]. NURBS files are larger and have some caveats regarding their usefulness but it allows saving the resulting solid in any of the available 3D solid formats (e.g. sat) to export it to ANSYS Multiphysics [4]. A complete example of a 3D reconstruction of a liver and FEA using NURBS can be found in [18].

2.4 Conversion from surface mesh to volume mesh

The process of generating an acceptable surface mesh is just a step towards creating a suitable mesh for CFD, as the surface mesh only carries information about the boundaries of the object and the CFD software cannot discretize space in terms of finite volumes, where the discretized form of important equations may be applied (e.g. Navier-Stokes, Darcy). Thus, a CFD simulation often needs a volume mesh.

The most practical tools to obtain a volume mesh from a surface mesh are Tetgen and the MATLAB® [19]/GNU Octave [11] toolbox iso2mesh [16]. Using iso2mesh the user can specify the detail of the final unstructured tetrahedral mesh via the command *s2m* (a.k.a. *surf2mesh*, short for surface to mesh) by choosing the maximum volume of each tetrahedral, the algorithm used for the conversion and the ratio of elements that the user desires to keep in the final mesh. The toolbox iso2mesh also offers some commands to fix degeneracies and mesh problems, e.g. *meshcheckrepair*, which is capable of eliminating duplicated elements, holes and self-intersecting faces.

Other advanced commands include all options Tetgen or GCAL offer, e.g. ISO2MESH_TETGENOPT='-A -q1.0a5 -C', which refines the mesh to improve quality and checks the final mesh for consistency. The user can verify the quality of the final mesh by the command *meshquality*. Finally, the volume mesh is saved in ABAQUS format (*saveabaqus*), which can be imported into the commercial ANSYS ICEM® 17.2 and export the mesh (.mesh file) to ANSYS® Fluent [2]. Another option is saving the mesh as DXF, which can be imported by Rhino3D (*savedxf*).

An additional toolbox, e.g. stlTools [34], is needed for reading the .stl file and extract its information. This toolbox is available in the MATLAB Central File Exhange. Toolboxes like stlTools and iso2mesh can be installed in MATLAB® permanently using the command *pathtool* or temporaly by the command *addpath*.

It is important to notice that iso2mesh and ANSYS® Fluent use different metrics to evaluate the quality of the final mesh. This means further processing my be required to correct eventual mesh problems, e.g. near degenerated elements. ANSYS® ICEM 17.2 [3] offer many possibilities to smooth the mesh acording to a user-defined quality parameter and and repair the volume mesh globally.

3 Results and discussion

The procedure explained in the previous sections was applied to a micro-CT scan performed by LAMIR-UFPR (Laboratório de Minerais e Rochas) of a carbonate rock mainly made of calcite, extracted from the region of Pamukkale (Turkey). The composition of the rock simplified the selection an appropriate threshold for presegmentation in ITK-SNAP, after saving the 10 initial layers of the scan as a Raw Binary File in ImageJ. Due to its complex shape, more than a hundred bubbles were needed during the Active Contour algorithm but it produced a good representation of the rock using 4.5 million faces. Direct extraction of the surface mesh using ImageJ would lead to 6.5 million faces. The first layer of the scan is shown in Figure 1.

The resulting mesh was imported into Meshmixer for reducing, remeshing and repairing and eliminating small shells (e.g. << 1% of total faces), producing a mesh with only one hundred thousand faces. The mesh was then exported to Meshlab, where the Laplacian Smooth and Taubin Smooth were applied. The mesh was verified for its quality using the area/max side as a metric, resulting in the mesh displayed in Figure 2. Next, the mesh was read by stlTools and iso2mesh was used to convert the surface mesh into a tetrahedral volume mesh using all the commands aforementioned.

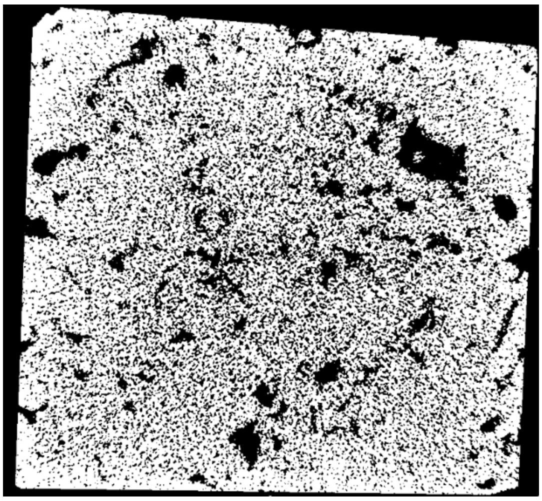

Figure 1 First layer of the micro-CT scan after thresholding.

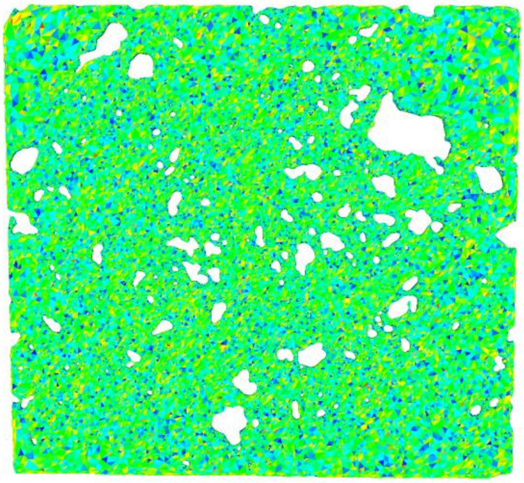

Figure 2 Snapshot from Meshlab showing the quality (area/max side) of each face.

Finally, the volume mesh was post-processed by ANSYS® ICEM to smooth elements according to Quality, Min Angle and Skew, resulting in the mesh shown in Figure 3, where the minimum orthogonal quality is 0.2, according to ANSYS Fluent. The bottom part of the mesh was labeled as a different region so as to facilitate the application of contour conditions. Lastly, the mesh was exported to ANSYS® Fluent for a simple simulation of heat transfer, shown in Figure 4. The initial condition for the bottom part was 500 K whereas upper part has an initial temperature of 300 K. Additionally, the upper part is exposed to a convection coefficient of 10 W m^{-2} K^{-1}. The energy under-relaxation factor was set to 0.5 to improve the robustness of the solution.

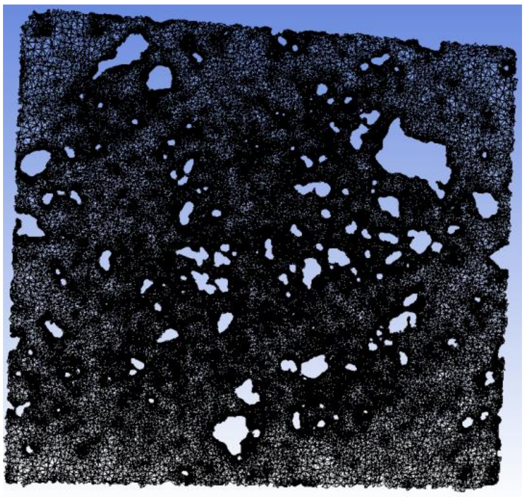

Figure 3 Snapshot from ANSYS ICEM of the repaired volume mesh.

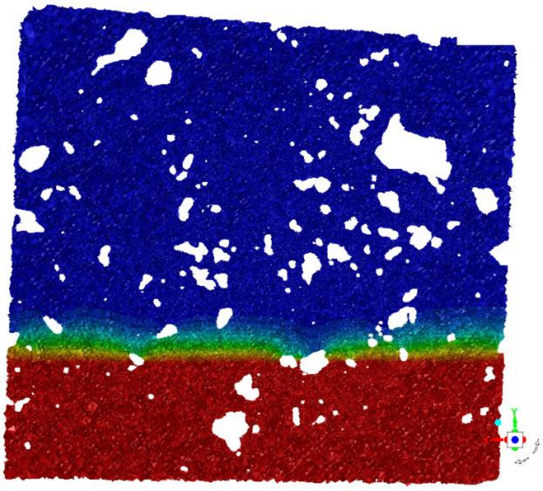

Figure 4 Snapshot from ANSYS Fluent of the heat transfer simulation.

4 Conclusions

A detailed procedure for converting scan data into a feasible volume mesh for CFD simulations was presented. The intermediate steps show a clear connection with technologies as additive manufacturing and finite element analysis. The CFD simulation of the calcite rock included only heat transfer but it was completed without any warnings, indicating that the final mesh is indeed suitable for simulations using finite volumes.

Instead of following the present methodology, the licenses for commercial software would imply in major costs for a small businesses. Information regarding this topic is difficult to retrieve and compile into a comprehensive guide that could suit all needs, so some assertiveness is recommended to judge the advantages and the drawbacks of each filter in terms of, for instance, preserving the topology or sharp edges in the model.

Acknowledgements

We thank the financial support of CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), grant proc. 480989/2013-3 and the research scholarship by CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior). In addition, the authors thank the developers for creating sharing these free and powerful tools, online instructors and all who provided enlightenment during this research.

References

[1] 3D Doctor. http://www.ablesw.com/3d-doctor/ (accessed 7 January 2018)
[2] ANSYS Fluent 19.0. ANSYS Inc., Southpointe, Pennsylvania, United States.
[3] ANSYS ICEM 17.2. ANSYS Inc., Southpointe, Pennsylvania, United States.
[4] ANSYS Multiphysics 10.0 ANSYS Inc., Southpointe, Pennsylvania, United States.
[5] Amira 6.4; Avizo 9.4. www.fei.com/software/amira-avizo/ (accessed 9 January 2018).
[6] Blender v2.79. www.blender.org/ (accessed 17 February 2018).
[7] Codeblocks. http://www.codeblocks.org/ (accessed 10 February 2018).
[8] FEBio 2.3 Software suite. https://febio.org/ (accessed 5 February 2018).
[9] FreeCAD 0.16. J. Riegel. https://www.freecadweb.org/ (accessed 2 February 2018).
[10] GIBBON Code. https://www.gibboncode.org/ (accessed 4 February 2018).

[11] GNU Octave. https://www.gnu.org/software/octave/ (accessed 20 February 2018).
[12] Graphite 3.0 Alpha. ALICE. http://alice.loria.fr/index.php?option=com_content&view=article&id=22 (accessed 8 February 2018).
[13] IA-FEMesh version 1.0. The University of Iowa. http://www.ccad.uiowa.edu/MIMX/projects/IA-FEMesh (accessed 6 February 2018).
[14] ImageJ 1.51j8. Image Processing and Analysis in Java. National Institutes of Health. https://imagej.nih.gov/ij/ (accessed 2 February 2018).
[15] ImageVis 3D 3.1.0. NIH/NIGMS Center for Integrative Biomedical Computing. http://www.sci.utah.edu/software/imagevis3d.html (accessed 14 February 2018).
[16] iso2mesh 2017 1.8 (Devil Egg) – toolbox for MATLAB/GNU Octave. http://iso2mesh.sourceforge.net (accessed 9 February 2018).
[17] ITK-SNAP 3.6.0. http://www.itksnap.org/ (accessed 13 February 2018).
[18] Kirana, K. P. Reconstructing Solid Model from 2D Scanned Images of Biological Organs for Finite Element Simulation. https://knolskiranakumarap.wordpress.com/2011/07/03/reconstructing-solid-model-from-2d-3rc2kfwq179j2-5/ (accessed 1 February 2018).
[19] MATLAB 2015b. The MathWorks, Inc., Natick, Massachusetts, United States.
[20] McNeel Wiki. The MeshToNurb Command. https://wiki.mcneel.com/rhino/meshtonurb (accessed 10 February 2018).
[21] Meshlab 64bit v1.3.4BETA. The open source system for processing and editing 3D triangular meshes. Cignoni, P; Ranzuglia, G; Visual Computing Lab; ISTI-CNR. http://www.meshlab.net (accessed 5 February 2018).
[22] Meshmixer v3.4.35. https://www.meshmixer.com/ (accessed 5 February 2018).
[23] MinGW. https://sourceforge.net/projects/mingw/ (accessed 11 February 2018).
[24] Netfabb Online Service. service.netfabb.com/login.php (accessed 9 February 2018).
[25] OpenFlipper-3.1. http://www.openflipper.org/ (accessed 4 February 2018).
[26] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. Neuroimage. 2006 Jul 1; 31(3):1116-28.
[27] ParaView SC17-5.5. https://www.paraview.org/ (accessed 9 February 2018).
[28] pyFormex 1.0. http://www.nongnu.org/pyformex/ (accessed 6 January 2018).
[29] Qianqian Fang and David Boas, "Tetrahedral mesh generation from volumetric binary and gray-scale images," Proceedings of IEEE International Symposium on Biomedical Imaging 2009, pp. 1142-1145, 2009.
[30] Rhinoceros: design, model, present, analyze, realize. Robert McNeel & Associates. https://www.rhino3d.com (accessed 13 February 2018).
[31] Salome v8.3.0 .The Open Source Integration Platform for Numerical Simulation. OPEN CASCADE. http://www.salome-platform.org/user-section/documentation/current-release (accessed 10 February 2018).
[32] Simpleware. www.synopsys.com/simpleware.html (accessed 8 February 2018).
[33] Simplify 3D. https://www.simplify3d.com/ (accessed 18 February 2018).
[34] stlTools version 1.1 – Toolbox for MATLAB. Pau Micó. https://www.mathworks.com/matlabcentral/fileexchange/51200-stltools (accessed 10 February 2018).
[35] Tetgen 1.5. wias-berlin.de/software/tetgen/ (accessed 1 February 2018).