INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2009; **00**:1–6 *Prepared using* nmeauth.cls *[Version: 2002/09/18 v2.02]*

High Quality Surface Remeshing Using Harmonic Maps

J-F Remacle[1], C. Geuzaine[2], G. Compère[1] and E. Marchandise[1]*

[1] *Université catholique de Louvain, Institute of Mechanics, Materials and Civil Engineering (iMMC), Place du Levant 1, 1348 Louvain-la-Neuve, Belgium*
[2] *Université de Liège, Department of Electrical Engineering and Computer Science, Montefiore Institute B28, Grande Traverse 10, 4000 Liège, Belgium*

SUMMARY

In this paper, we present an efficient and robust technique for surface remeshing based on harmonic maps. We show how to ensure a one-to-one mapping for the discrete harmonic map and introduce a cubic representation of the geometry based on curved PN triangles. Topological and geometrical limitations of harmonic maps are also put to the fore and discussed. We show that, with the proposed approach, we are able to recover high quality meshes from both low input STL triangulations and complex surfaces defined by many CAD patches. The overall procedure is implemented in the open-source mesh generator Gmsh [1]. Copyright © 2009 John Wiley & Sons, Ltd.

KEY WORDS: surface remeshing; surface parametrization; STL file format; surface mapping; harmonic map; surface smoothing

1. Introduction

Creating high quality meshes is an essential feature for obtaining accurate and efficient numerical solutions of partial differential equations as it impacts both the accuracy and the efficiency of the numerical method using those meshes [2, 3].

In many cases, surfaces do not have a standard CAD representation and are only known by triangulations such as stereolithography (STL) triangulations. These kinds of surfaces are commonplace in many areas of science and engineering, e.g. in the form of 3D scanned images, terrain data, or medical data obtained from imaging techniques through a segmentation procedure. Such triangulations are often oversampled and/or of poor quality (with triangles exhibiting very small aspect ratios), which makes them unsuited for direct use by numerical methods like finite elements, finite volumes or boundary elements. This is also problematic for the volume mesh since the surface mesh serves as input for the volume meshing algorithms. Improving the mesh quality can then be performed using a remeshing procedure.

Contract/grant sponsor: Fonds National de la Recherche Scientifique, rue d'Egmont 5, 1000 Bruxelles, Belgium
*Correspondence to: emilie.marchandise@uclouvain.be

Received
Revised

In the case of manufactured objects, the surfaces are often designed using a CAD system and described through a constructive solid geometry procedure. Non Uniform Rational B-Splines (NURBS) are commonly used for describing the shape of surfaces. NURBS surfaces are usually nice and smooth so that it is possible to produce high quality surfaces meshes using NURBS as input. However, most surface mesh algorithms mesh model faces individually, which means that points are generated on the bounding edges and that these points will be part of the surface mesh. If thin CAD patches exist in the model they will result in the creation of small distorted triangles with very small angles [4, 5]—even if the bounding edges of these thin patches have no physical significance. As in the case of a poor quality STL triangulation, a remeshing procedure is then also desirable.

There are mainly two approaches for surface remeshing: mesh adaptation strategies [6, 7, 8] and parametrization techniques [9, 10, 11, 12, 13, 14]. Mesh adaptation strategies use local mesh modifications in order both to improve the quality of the input surface mesh and to adapt the mesh to a given mesh size criterion. In parametrization techniques, the input mesh serves as a support for building a continuous parametrization of the surface. (In the case of CAD geometries, the initial mesh can be created using any off the shelf surface mesher for meshing the individual patches.) Surface parametrization techniques originate mainly from the computer graphics community: they have been used extensively for applying textures onto surfaces [15, 16] and have become a very useful and efficient tool for many mesh processing applications [17, 18, 19, 20, 21]. In the context of remeshing procedures, the initial surface is parametrized onto a surface in $\mathcal{R}^2$, the surface is meshed using any standard 2D mesh generation procedure and the new triangulation is then mapped back to the original surface [22, 4].

This paper proposes a quality remeshing strategy based on *harmonic maps* for the surface parametrization (see [21] for a survey of alternative parametrization techniques). Harmonic maps exhibit several useful properties: (i) they are easy to compute and can be approximated using linear systems, (ii) they are independent of the initial triangulation, (iii) they are indefinitely differentiable on a surface and (iv) they are one-to one for convex mapped regions [21, 23]. Harmonic maps do not preserve angles such as the conformal maps usually used for texture mapping [16, 19] and by some authors for surface remeshing [24]. However, this is not quite an issue in the context of mesh generation. Indeed, we can deal with non-conforming maps as soon as we have access to the metric tensor that allow us to measure both lengths and angles in the parameter plane.

Discrete harmonic maps have first been successfully used for surface remeshing by Eck [22] and Marcum [4]. However, as mentioned by Floater in [25], discrete harmonic maps are in general not guaranteed to be one-to-one. To ensure a one-to-one discrete map, Floater suggested a different edge spring weighting that guarantees an embedding for convex boundaries, also called "convex combination map". We show however in this paper that the quality of the metrics of the convex combination map are not sufficient for generating high quality meshes and suggest to only locally apply a simple geometrical algorithm called "cavity check". Another important but rarely discussed issue regarding harmonic maps concerns the geometrical aspect of the surfaces to be parametrized. By presenting the harmonic maps as the solution of Laplace equations, we show why the harmonic mapping fails for surfaces with large aspect ratio. We then suggest some ways to address this issue.

The aim of the paper is twofold: (i) we first present the harmonic mapping in a comprehensive manner such that it becomes accessible to a wider community than the one of computer

graphics and (ii) we show that even with the known limitations of harmonic maps, they can be used for efficiently generating high quality surface meshes. The paper also deals with implementation. We show a simple way to compute and implement efficiently harmonic maps using linear finite elements with appropriate boundary conditions. We show how to guarantee that the discrete harmonic mapping is one-to-one and well-defined for geometries with large aspect ratios. The remeshing procedure is enhanced by using cubic mapping to smooth the initial triangulation. Finally, different results demonstrate that high-quality unstructured meshes can be efficiently and consistently generated for subsequent numerical simulations. We show that the resulting surfaces meshes that are produced with the new technique have a better quality than standard available remeshing techniques.

All the results presented in the paper were generated using the open-source mesh generator Gmsh [1], where the proposed algorithms can be further studied, tested and enhanced.

2. Parametrization of discrete surfaces

Parametrizing a surface $\mathcal{S}$ is defining a map $\mathbf{u}(\mathbf{x})$

$$\mathbf{x} \in \mathcal{S} \subset \mathcal{R}^3 \mapsto \mathbf{u}(\mathbf{x}) \in \mathcal{S}' \subset \mathcal{R}^2 \tag{1}$$

that transforms continuously a 3D surface $\mathcal{S}$ into a surface $\mathcal{S}'$ embedded in $\mathcal{R}^2$ that has a well known parametrization (see Fig. 1). Such a continuous parametrization exists if the two surfaces $\mathcal{S}$ and $\mathcal{S}'$ have the same topology, that is have the same genus $G(\mathcal{S})$ and the same number of boundaries N_B. The genus $G(\mathcal{S})$ of a surface is the number of handles in the surface. For example, a sphere has a genus $G = 0$ and $N_B = 0$, a disk has $G = 0$ but $N_B = 1$ and a torus has $G = 1$ and $N_B = 0$.

In this work, we consider that the only available representation of a surface $\mathcal{S}$ is a conforming triangular mesh $\mathcal{S}_\mathcal{T}$ in 3D , i.e. the union of a set of triangles T_j that intersect only at common vertices or edges $\mathcal{T} = \{T_1, ..., T_N\}$. Let us consider a triangulated surface $\mathcal{S}$ that has N_V vertices, N_E edges and N_T triangles. The genus $G(\mathcal{S}_\mathcal{T})$ is given through the Euler-Poincaré formula:

$$G(\mathcal{S}_\mathcal{T}) = \frac{-N_V + N_E - N_T + 2 - N_B}{2}. \tag{2}$$

As an example, the left part of Figure 1 shows a triangulated Tutankhamun mask. This triangulated surface is homeomorphic to the unit disk, i.e. they have both a zero genus and one boundary. It is therefore possible, in principle, to find a smooth transformation that maps $\mathcal{S}$ into $\mathcal{S}'$.

The parametrization we look for is discrete: each vertex V_i, $i = 1, \ldots, N_V$ of the triangulation has two sets of coordinates: the 3D coordinates $\mathbf{x}_i = (x_i, y_i, z_i) \in \mathcal{S}$ and the parametric coordinates $\mathbf{u}_i = (u_i, v_i) \in \mathcal{S}'$. Each triangle has also two representations, one in the 3D space and one in the 2D parametric space. Consider triangle T_j with its three vertices V_1, V_2 and V_3. The parametrization is one-to-one if and only if triangles do not overlap in the parametric space. Note that the notion of triangle overlapping is only well defined in a 2D space.

This triangle can itself be parametrized using for example barycentric coordinates, i.e. standard finite element linear shape functions (see Figure 2):

$$\mathbf{x}(\xi) = (1 - \xi - \eta)\mathbf{x}_1 + \xi\mathbf{x}_2 + \eta\mathbf{x}_3. \tag{3}$$

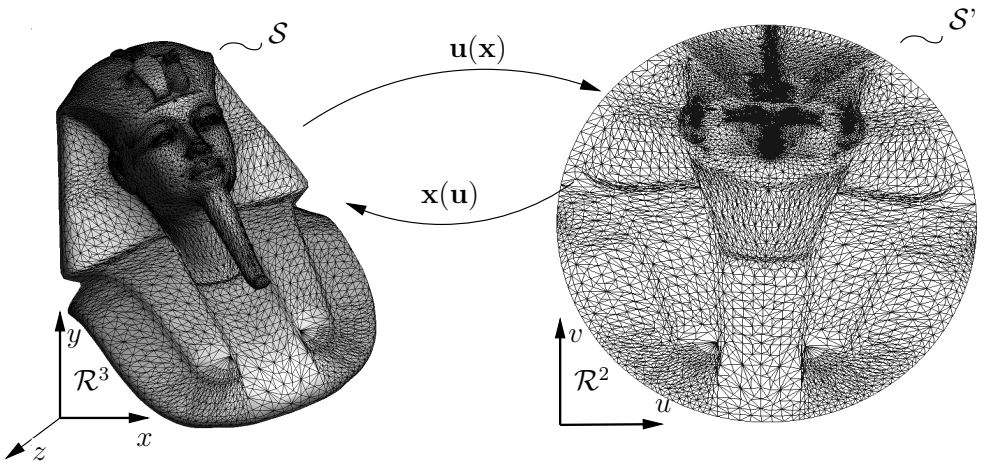


Figure 1. Parametrization $\mathbf{x}(\mathbf{u})$ and inverse parametrization $\mathbf{u}(\mathbf{x})$ of the Tutankhamun mask that associates every point of the surface $\mathcal{S} \subset \mathcal{R}^3$ with a point of the surface $\mathcal{S}' \subset \mathcal{R}^2$.

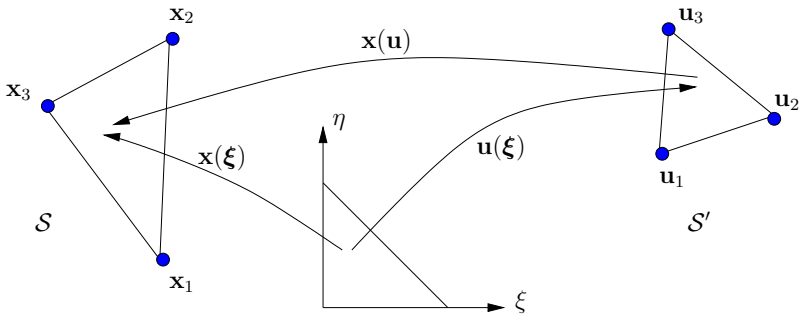


Figure 2. Unit triangle in local coordinates and the maps $\mathbf{x}(\xi), \mathbf{u}(\xi)$ and $\mathbf{x}(\mathbf{u})$.

Similarly, we can also parametrize the triangle in $\mathcal{S}'$:

$$\mathbf{u}(\xi) = (1 - \xi - \eta)\mathbf{u}_1 + \xi\mathbf{u}_2 + \eta\mathbf{u}_3. \tag{4}$$

In order to compute the mapping $\mathbf{x}(\mathbf{u})$, we first invert (4):

$$\mathbf{u} - \mathbf{u}_1 = \underbrace{\begin{bmatrix} u_2 - u_1 & u_3 - u_1 \\ v_2 - v_1 & v_3 - v_1 \end{bmatrix}}_{\mathbf{u}_{,\xi}} \underbrace{\begin{pmatrix} \xi \\ \eta \end{pmatrix}}_{\xi} \tag{5}$$

which gives

$$\xi(\mathbf{u}) = (\mathbf{u}_{,\xi})^{-1} (\mathbf{u} - \mathbf{u}_1) = \left(\xi_{,\mathbf{u}}\right) (\mathbf{u} - \mathbf{u}_1). \tag{6}$$

The discrete mapping $\mathbf{x}(\mathbf{u})$ can therefore be computed in three steps:

1. Find the unique triangle T_j of the parametric space $\mathcal{S}'$ that contains point $\mathbf{u}$;
2. Compute local coordinates $\xi = (\xi, \eta)$ of point $\mathbf{u}$ inside triangle T_j using Equation (6);
3. Use Equation (3) to compute the mapping $\mathbf{x}(\mathbf{u}) = \mathbf{x}(\xi(\mathbf{u}))$.

Mesh generation procedures usually not only require the mapping $\mathbf{x}(\mathbf{u})$ but also its derivatives $\mathbf{x}_{,\mathbf{u}}$. We have

$$\mathbf{x}_{,\mathbf{u}} = \mathbf{x}_{,\xi}\ \xi_{,\mathbf{u}} = \begin{bmatrix} x_2 - x_1 & x_3 - x_1 \\ y_2 - y_1 & y_3 - y_1 \\ z_2 - z_1 & z_3 - z_1 \end{bmatrix} \frac{\begin{bmatrix} v_3 - v_1 & -(u_3 - u_1) \\ -(v_2 - v_1) & u_2 - u_1 \end{bmatrix}}{(u_2 - u_1)(v_3 - v_1) - (v_2 - v_1)(u_3 - u_1)}. \tag{7}$$

The metric tensor (or first fundamental form)

$$\mathbf{M} = \mathbf{x}_{,\mathbf{u}}^T \mathbf{x}_{,\mathbf{u}} \tag{8}$$

then allows to compute lengths, angles and areas. Consider one curve $\mathcal{C}$ drawn on the parametric space $\mathcal{S}'$. Its length is

$$l_{\mathcal{C}} = \int_{\mathcal{C}} dl = \int_{\mathcal{C}} \sqrt{\mathbf{dx}^2} = \int_{\mathcal{C}} \sqrt{(\mathbf{x}_{,\mathbf{u}}\mathbf{du})^2} = \int_{\mathcal{C}} \sqrt{\mathbf{du}^T \mathbf{M}(\mathbf{u})\mathbf{du}}. \tag{9}$$

The practical case for mesh generation is when $\mathcal{C}$ is a mesh edge of the parametric space going from point $\mathbf{u}_1$ to point $\mathbf{u}_2$. Calling $\mathbf{e} = \mathbf{u}_2 - \mathbf{u}_1$, its parametrization is

$$\mathcal{C} = \{\mathbf{u} \in \mathcal{S}' \mid \mathbf{u} = \mathbf{u}_1 + t\ \mathbf{e},\ t \in [0, 1]\}.$$

In this special case, the length of a straight edge in the parameter space is computed as

$$l_{\mathcal{C}} = \int_0^1 \sqrt{\mathbf{e}^T \mathbf{M}(\mathbf{u}_1 + t\ \mathbf{e})\ \mathbf{e}}\ dt. \tag{10}$$

3. Harmonic maps with appropriate boundary conditions

As illustrated in Fig.1, we have chosen to map our 3D surfaces $\mathcal{S}$ onto a unit disk $\mathcal{S}'$. Therefore we require genus zero surfaces that have at least one boundary that will be mapped on the unit disk. We compute coordinates u and v separately as solutions of the following two Laplace problems:

$$\begin{aligned} \nabla^2 u = 0, \quad \nabla^2 v = 0 \quad &\text{on} \quad \mathcal{S}, \\ u = \bar{u}(\mathbf{x}), \quad v = \bar{v}(\mathbf{x}) \quad &\text{on} \quad \partial\mathcal{S}_1, \\ \partial_n u = 0, \quad \partial_n v = 0 \quad &\text{on} \quad \partial\mathcal{S}/\partial\mathcal{S}_1. \end{aligned} \tag{11}$$

Then, we have to supply functions $\bar{u}(\mathbf{x})$ and $\bar{v}(\mathbf{x})$ that map $\partial\mathcal{S}_1$ onto the unit circle. For that, we choose arbitrarily a starting vertex V_s (see Figure 3) and we compute l_i that is the distance from V_S to V_i along $\partial\mathcal{S}_1$. If L is the total length of $\partial\mathcal{S}_1$, the following boundary conditions

$$u(\mathbf{x}_i) = \cos(2\pi l_i/L), \quad v(\mathbf{x}_i) = \sin(2\pi l_i/L) \tag{12}$$

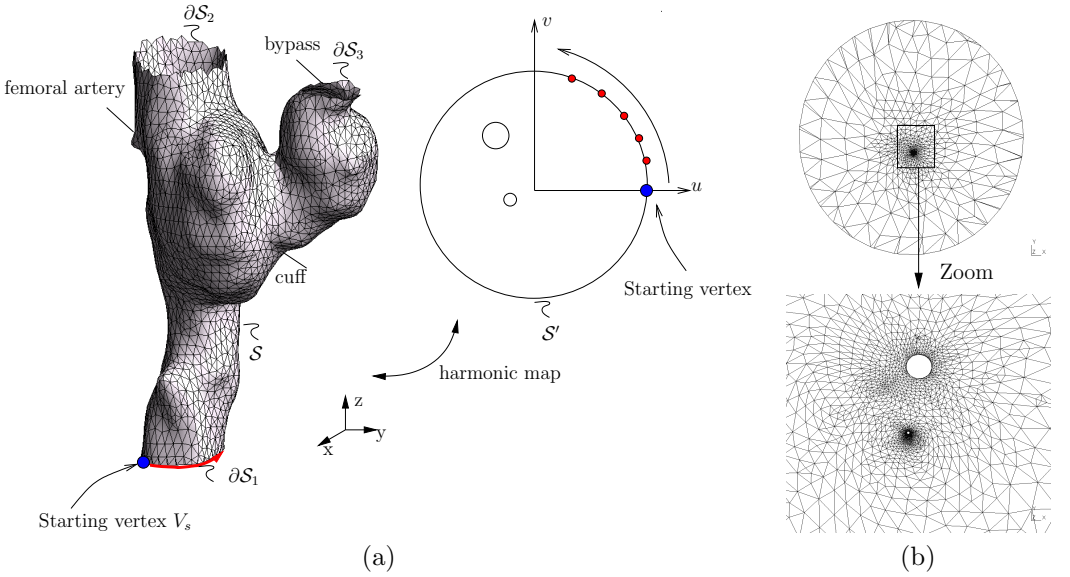


Figure 3. (a) STL triangulation and its map onto the unit disk and (b) the mapped mesh on the unit disk.

map $\partial\mathcal{S}_1$ onto the unit circle.

Figure 3 shows both an initial triangular mesh of $\mathcal{S}$ and its map onto the unit disk. The surface $\mathcal{S}$ results from the segmentation of an anastomosis site in the lower limbs, more precisely a bypass of an occluded femoral artery. The unit disk $\mathcal{S}'$ contains two holes that correspond to the boundaries of the femoral artery $\partial\mathcal{S}_2$ and the saphenous vein $\partial\mathcal{S}_3$.

At the continuous level, such a mapping can be proven to be one-to-one, provided that surface $\mathcal{S}'$ is convex. This result is called the Radò-Kneser-Choquet (RKC) theorem [26, 27]. This result strongly depends on the fact that the solution of the Laplace equation obeys a strong maximum principle: $u(\mathbf{x})$ attains its maximum on the boundary $\partial\mathcal{S}$ of the domain. This means that there exists only one single iso-curve $u = u_0$ in $\mathcal{S}$ and that this iso-curve goes continuously from one point of the boundary to another. If another iso-curve $u = u_0$ existed, it should be closed inside $\mathcal{S}$, violating the maximum principle. This is also true for the iso-curve $v = v_0$.

Consider the surface $\mathcal{S}$ of Figure 4a with one single boundary ($N_B = 1$). Surface $\mathcal{S}'$ is convex if any vertical line $u = u_0$ intersects $\partial\mathcal{S}'$ at most two times. This is also true for any horizontal line $v = v_0$. This means that any coordinate $u = u_0$ ($u_0 \in]-1, 1[$) appears exactly two times on the boundary $\partial\mathcal{S}$. The two points of $\partial\mathcal{S}$ for which $u = u_0$ are designated as V_A and V_B while the two points for which $v = v_0$ are designated as V_C and V_D. Note that those points appear interleaved while running through $\partial\mathcal{S}$ (V_A appears either after V_D or after V_C but never after V_B). This means that there exists one point in $\mathcal{S}$ for which $u = u_0$ and $v = v_0$.

Now consider the case where $N_B = 2$ (Figure 4b). Here, zero Neumann boundary conditions are applied to the inner boundary of $\mathcal{S}$. Those are equivalent to the resolution of a Laplace problem on the whole domain while defining a small diffusivity inside the hole (Figure 4c).

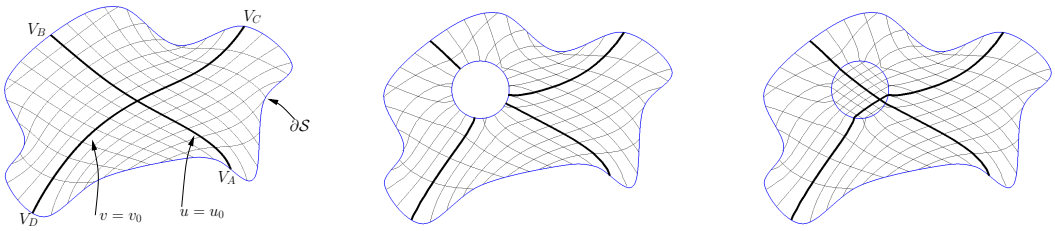


Figure 4. Iso-values of coordinates u and v on a surface $\mathcal{S}$ that are computed as solutions of the Laplace equation on $\mathcal{S}$ with boundary conditions that map $\partial\mathcal{S}$ on the unit circle. a) Dirichlet boundary conditions are imposed on the outer boundary of $\mathcal{S}$ for two configurations: b) $\mathcal{S}$ excludes the interior disk and zero Neumann boundary conditions are applied on the inner circular boundary and c) $\mathcal{S}$ includes the interior disk, where a small diffusion coefficient is used.

This second problem obeys the same maximum principle as the one with constant diffusivity, which means that the mapping remains one-to-one even when considering holes in the domain.

Of course, any other convex planar surface can serve as $\mathcal{S}'$. In our implementation, we have tried ellipses and rectangles. Yet, no significative difference was observed while changing the definition of the parametric domain.

3.1. Discrete harmonic maps with linear finite elements

It is easy to prove that (11) is equivalent to the following quadratic minimization problem:

$$\min_{u \in U(\mathcal{S})} J(u) = \frac{1}{2} \int_{\mathcal{S}} \|\nabla^2 u\| ds \tag{13}$$

with

$$U(\mathcal{S}) = \{u \in H^1(\mathcal{S}),\ u = f(\mathbf{x}) \ \text{ on } \ \partial\mathcal{S}\}. \tag{14}$$

Assume the following finite expansions for u

$$u_h(\mathbf{x}) = \sum_{i \in I} u_i \phi_i(\mathbf{x}) + \sum_{i \in J} f(\mathbf{x}_i)\phi_i(\mathbf{x}), \tag{15}$$

where I denotes the set of nodes of $\mathcal{S}_\mathcal{T}$ that do not belong to the Dirichlet boundary, J denotes the set of nodes of $\mathcal{S}_\mathcal{T}$ that belong to the Dirichlet boundary and where ϕ_i are the nodal shape functions associated to the nodes of the mesh. We assume here that the nodal shape function ϕ_i is equal to 1 on vertex $\mathbf{x}_i$ and 0 on any other vertex: $\phi_i(\mathbf{x}_j) = \delta_{ij}$.

Using the expansion (15), the functional J from (13) can be written as

$$\begin{aligned} J(u_1, \ldots, u_N) \quad = \quad & \frac{1}{2} \sum_{i \in I} \sum_{j \in I} u_i u_j \int_{\mathcal{S}_\mathcal{T}} \nabla\phi_i(\mathbf{x}) \cdot \nabla\phi_j(\mathbf{x}) ds + \\ & \sum_{i \in I} \sum_{j \in J} u_i f(\mathbf{x}_j) \int_{\mathcal{S}_\mathcal{T}} \nabla\phi_i(\mathbf{x}) \cdot \nabla\phi_j(\mathbf{x}) ds + \\ & \frac{1}{2} \sum_{i \in J} \sum_{j \in J} f(\mathbf{x}_i) f(\mathbf{x}_j) \int_{\mathcal{S}_\mathcal{T}} \nabla\phi_i(\mathbf{x}) \cdot \nabla\phi_j(\mathbf{x}) ds. \end{aligned} \tag{16}$$

In order to minimize J, we can simply cancel the derivative of J with respect to u_k:

$$\begin{aligned}\frac{\partial J}{\partial u_k} &= \sum_{j\in I} u_j \int_{\mathcal{S}_\mathcal{T}} \nabla\phi_j(\mathbf{x})\cdot\nabla\phi_k(\mathbf{x})ds + \sum_{j\in J} f(\mathbf{x}_j)\int_{\mathcal{S}_\mathcal{T}} \nabla\phi_k(\mathbf{x})\cdot\nabla\phi_j(\mathbf{x})ds \\ &= 0\ ,\quad \forall k\in I. \end{aligned} \tag{17}$$

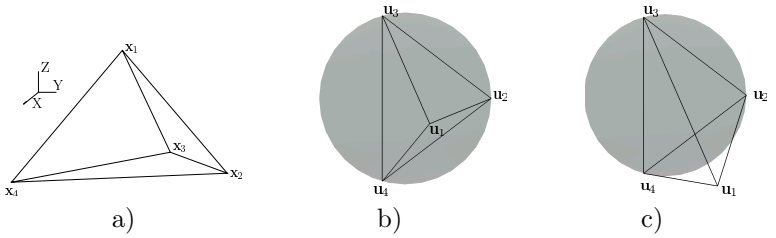
There are as many equations (17) as there are nodes in I. This system of equations can be proven to be symmetric positive definite so that it can be solved easily, e.g. using preconditioned conjugate gradients. If we want to solve (17) with linear finite elements, we can compute the elementary matrix $A_{ij}^{T_k}$ of triangle T_k as:

$$A_{ij}^{T_k} = \int_{T_k} \nabla\phi_i\cdot\nabla\phi_j ds = \int_0^1\int_0^{1-\xi} \nabla^{\xi,\eta}\phi_i\ \mathbf{M}_\xi^{-1}\ \nabla^{\xi,\eta}\phi_j \sqrt{\det \mathbf{M}_\xi}\ d\xi d\eta, \tag{18}$$

where $\mathbf{M}_\xi$ is the metric tensor of the mapping $\mathbf{x}(\xi)$.

3.2. One-to-one discrete harmonic map

In contrast to the continuous harmonic map, it was shown in [25, 28] that the discrete harmonic map first introduced in [22] is not always one-to-one. Indeed, we can see in the next example introduced by Floater [25] that the discrete harmonic map as presented in the previous section is not guaranteed to be one-to-one. Consider a coarse triangulation (Figure 5) made of three triangles $\mathcal{S}_\mathcal{T} = \{(1,2,3),(1,3,4),(1,4,2)\}$ and let $\mathbf{x}_1 = (r,0,1)$ for some real value $r > 0$ and $\mathbf{x}_2 = (1,1,0), \mathbf{x}_3 = (0,0,0), \mathbf{x}_4 = (1,-1,0)$. The three boundary vertices $\mathbf{x}_1, \mathbf{x}_2$ and $\mathbf{x}_3$ are mapped onto the boundary of the unit disk (Figures 5b and c) and the vertex $\mathbf{x}_1$ should be mapped inside the triangle $(\mathbf{u}_2, \mathbf{u}_3, \mathbf{u}_4)$ to ensure a one-to-one mapping. However, numerical methods for solving Laplace equation may not provide solutions that obey to a discrete maximum principle, especially when meshes are distorded [29]—which can lead to discrete harmonic maps which are not one-to-one (Figure 5c).



a) b) c)

Figure 5. a) Triangulation for which the discrete harmonic mapping is not guaranteed to be one-to-one: $\mathbf{x}_1 = (r,0,1)$ for $r > 0$ and $\mathbf{x}_2 = (1,1,0), \mathbf{x}_3 = (0,0,0), \mathbf{x}_4 = (1,-1,0)$. b) Case $r = 1.5$: the mapping is one-to-one , c) Case $r = 3.5$: the mapping is not one-to-one. The point $\mathbf{u}_1$ does not even lie within the unit disk.

One possibility to ensure a discrete maximum principle consists in placing each point of the parameter plane at the center of gravity of its neighbors. This method was introduced by Floater in [25, 28] and called *convex combination map*. It is implemented simply by choosing

$$A_{ij}^{T_k} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{pmatrix} \tag{19}$$

for every element T_k. However, the parametrization resulting from a convex combination map is much more distorted than the standard harmonic one. The equivalent PDE resulting from the convex combination map is an anisotropic diffusion problem, with a piecewise constant diffusion tensor that is equal to $\mathbf{M}_\xi / \sqrt{\det \mathbf{M}_\xi}$. Surfaces with highly distorded metrics are known to make the work of surface meshers more difficult: Figure 6 compares meshes of a human pelvis generated using either standard harmonic mappings or convex combination maps. The quality of elements clearly deteriorates when using convex combination maps. This can

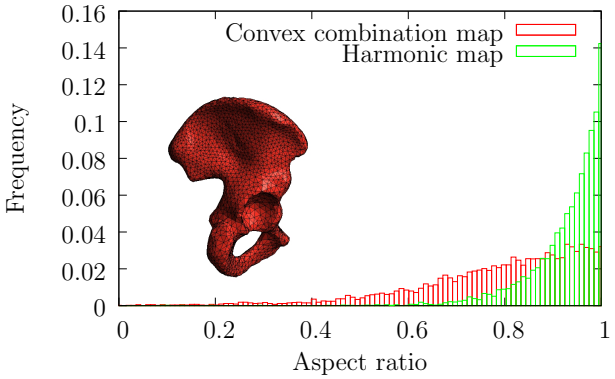


Figure 6. Quality histogram for the remeshing of a human pelvis. Comparison for the harmonic mapping and the convex combination mapping.

be explained by the simpler example of Figure 7. Here, we start from a very bad triangulation (Figure 7 a). We parametrize it using both harmonic and convex combination maps. Iso-values of the x coordinate are drawn for both maps on the unit disk. Even though the mesh issued from the convex combination map is much smoother in the parametric plane that the standard harmonic one, isovalues are much closer to straight lines for the standard map. This means that a straight line in the parameter plane is close to a straight line in the real plane for the harmonic map and hence that the metric tensor $\mathbf{M}$ (8) is much smoother for the harmonic map than it is for the convex combination map.

To solve the problems associated with convex combination maps, we propose a more local way to enforce discrete one-to-one map. The algorithm is called *cavity check* (see Fig. 8) and goes as follows:

1. Compute harmonic map using finite elements;
2. For every interior vertex V_i of the parameter plane, check if each of its neighboring triangles (defining a cavity) is oriented properly[†];
3. If elements are reversed, move the vertex at the center of gravity of the kernel of the polygon P surrounding the point (see Figure 8).

In order to find the kernel of a star-shaped polygon, different algorithms have been proposed in the literature[30, 31]. In this work, as the polygons have a small number of vertices we have implemented a simple quadratic algorithm. It should be noted that the situation presented in

[†]This is simply implemented by comparing normal orientations of the triangles in the parametric space

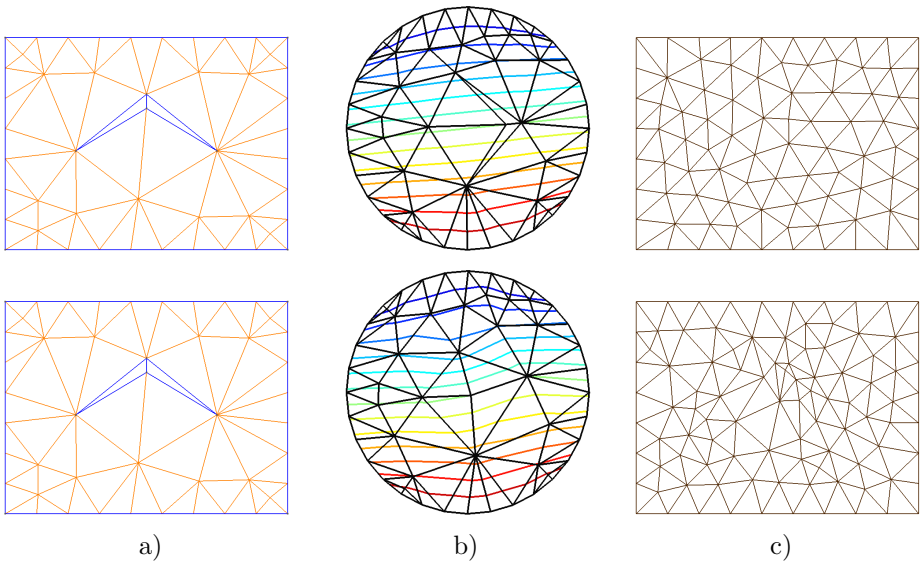


Figure 7. Poor quality initial triangulation (a) that has been remeshed using a harmonic map (top figures) and a convex combination map (bottom figures): b) mapping of the initial mesh onto the unit disk with iso-x values c) the final mesh. For this example a direct mesher based on local mesh modifications (Gmsh/meshadapt) is used to remesh the parametrized surface.

Fig. 8a) does occur very rarely and only occurs for very poor quality initial stl files (one or two cases at most for the examples presented in section 5).

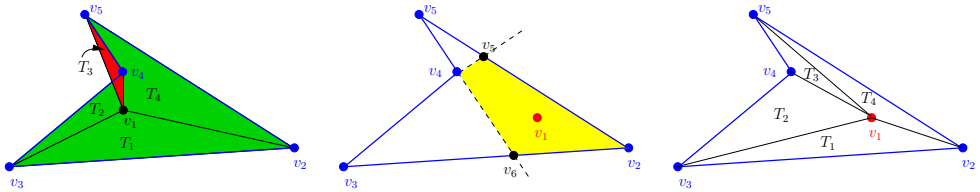


Figure 8. Vertex v_1 with four neighboring triangles $T_1 = (v_1, v_2, v_3)$, $T_2 = (v_1, v_3, v_4)$, $T_3 = (v_1, v_4, v_5)$, $T_4 = (v_1, v_5, v_2)$ defining the polygon $P = (v_2, v_3, v_4, v_5)$. a) Triangle T_3 (in red) is not well oriented and overlaps the triangles T_2 and T_4 . b) The vertex v_1 is moved and placed inside the kernel of the polygon P (yellow area), c) Now, all four triangles become well oriented without overlapping and hence the discrete mapping is guaranteed to be one-to-one for this cavity.

3.3. Harmonic maps for geometries with large aspect ratio

In some cases for which the ratio between the equivalent diameter of the closed loop $\partial\mathcal{S}_1$ and the length in the direction normal to the surface with line loop $\partial\mathcal{S}_1$ is to high, we fail to compute the harmonic map.

In order to explain this, we take a simple example of a surface $\mathcal{S}$ that is a cylinder of height

H and radius R. We can easily compute analytically the harmonic map on this cylinder. Indeed, finding the harmonic map is equivalent to solving the two Laplace equations (11) on a rectangle of height H and length $2L$, with $L = \pi R$ with the boundary conditions shown in Figure 9. The analytical solution of the problem is

$$u(x,y) = \cos\left(\frac{\pi x}{L}\right) Y(y), \quad v(x,y) = \sin\left(\frac{\pi x}{L}\right) Y(y) \tag{20}$$

where

$$Y(y) = \cosh\left(\frac{\pi y}{L}\right) - \left(\tanh\left(\frac{\pi H}{L}\right) \sinh\left(\frac{\pi y}{L}\right)\right). \tag{21}$$

The function $Y(y)$ rapidly tends to zero. This means that, for high geometrical ratios

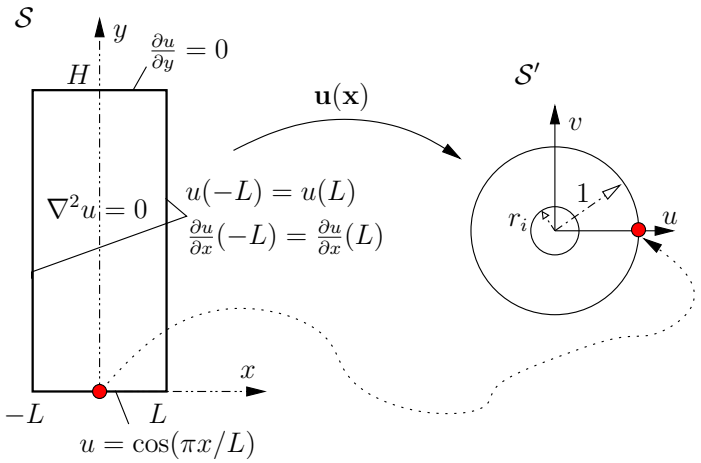


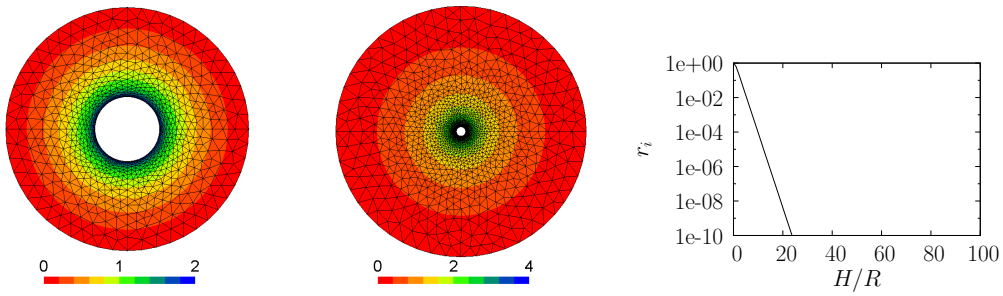
Figure 9. Harmonic mapping of the cylinder onto the unit disk. The left Figure shows the rectangular domain of size $[2L \times H]$ and the boundary conditions used to compute the analytical solution of the Laplace equation on the cylinder of height H and radius $R = L/\pi$. The right Figure shows the harmonic map on the unit disk.

($H/R \approx 6\pi$), computed coordinates become non distinguishable for high y's because of the computer finite precision. Figure 10 shows the map of the cylinder into an annulus. The inner radius of the annulus goes to zero exponentially. In practice, using double precision arithmetic we have experienced that Gmsh's meshers will fail to deliver decent meshes when $H/R > 6\pi$. The algorithm can be improved by scaling the problem by the geometrical aspect ratio of the cylinder H/R and by solving the following Laplace equation with anisotropic coefficients k_x and k_y:

$$k_x u_{,xx} + k_y u_{,yy} = 0, \quad k_x v_{,xx} + k_y v_{,yy} = 0, \quad \text{with} \quad k_x = 1, k_y = H/R. \tag{22}$$

The solution in the $y-$direction is then given by:

$$Y(y) = \frac{e^{\frac{-y}{\sqrt{H}\sqrt{R}}}\left(e^{\frac{2y}{\sqrt{H}\sqrt{R}}} + e^{\frac{2\sqrt{H}}{\sqrt{R}}}\right)}{e^{\frac{2\sqrt{H}}{\sqrt{R}}} + 1}. \tag{23}$$



(a) (b) (c)

Figure 10. Harmonic map of the cylinder of height H and radius R onto the unit disk. Mesh in the unit circle and values of $y(u,v)$ for a) $H/R = 2$ ($r_i = 0.26$), b) $H/R = 4$ ($r_i = 0.036$). c) $r_i = \sqrt{u(L,H)^2 + v(L,H)^2}$ as a function of H/R.

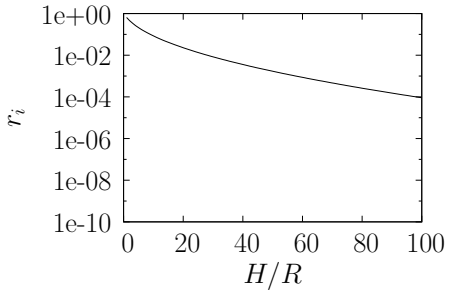


Figure 11. $r_i = \sqrt{u(L,H)^2 + v(L,H)^2}$ as a function of H/R for the scaled Laplacian problem (22)

As can be seen in Fig.11 this function is less stiff and the inner radius r_i does not tend to zero for large values of the geometrical aspect ratio.

This technique could be generalized in different ways. The first idea is to simply compute global anisotropic coefficients from the size of the oriented bounding boxes [32]. A second idea is to compute local anisotropic coefficients from the gradient of the distance function to the boundary $\partial\mathcal{S}_1$. The distance function could be for example computed as the solution of an elliptic PDE as described in [33]. Another possibility, usually used in computer graphics, is to use a partition scheme based on the concept of Voronoi diagrams [22] or inspired by Morse theory [19, 34], which would lead to a partitioning of the mesh into a number of charts that by construction have a uniform geometrical aspect ratio.

3.4. Higher order representation of the geometry

In case of faceted triangulations, it is often desirable to smooth the normals when remeshing. This may be the case for example when the CAD is described by very few triangles or for triangulations obtained from crude segmentation techniques.

This can be achieved by changing only one of the three maps $\mathbf{x}(\xi)$ defined on Fig. 2. Instead

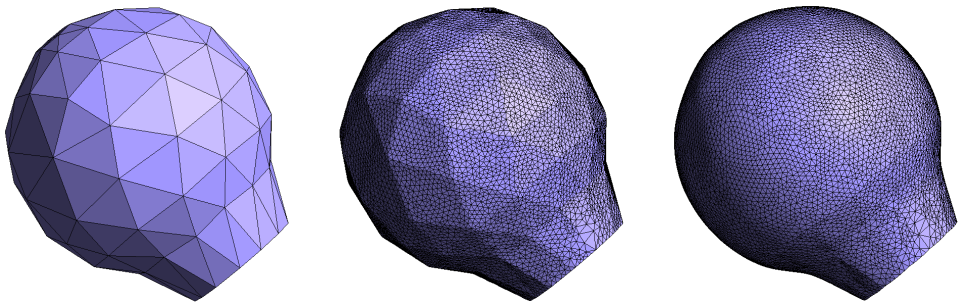
of choosing linear finite elements for this mapping as is done in (3), we have chosen in this work a cubic interpolation that is often used in the community of computer graphics [35, 36]:

$$\begin{aligned}\mathbf{x}(\xi) =& a_{300}\zeta + a_{030}\xi + a_{300}\eta + a_{210}3\zeta^2\xi + a_{120}3\zeta\xi^2 + a_{201}3\zeta^2\eta \\ &+ a_{021}3\xi^2\eta + a_{102}3\zeta\eta^2 + a_{201}3\xi\eta^2 + a_{210}6\xi\eta\zeta, \qquad \text{with} \quad \zeta = 1-\xi-\eta.\end{aligned} \tag{24}$$

The coefficients a_{ijk} are the 6 control points of the curved PN triangle [36] that can be computed from the triangle $T_j \in \mathcal{R}^3$ that is defined by its three coordinates $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$ and its three vertex normals $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$:

$$\begin{aligned} a_{300} &= \mathbf{x}_1, \quad a_{030} = \mathbf{x}_2, \quad a_{003} = \mathbf{x}_3, \\ w_{ij} &= (\mathbf{x}_j - \mathbf{x}_i)\cdot\mathbf{n}_i, \\ a_{210} &= (2\mathbf{x}_1 + \mathbf{x}_2 - w_{12}\mathbf{n}_1)/3, \quad a_{120} = (2\mathbf{x}_2 + \mathbf{x}_1 - w_{21}\mathbf{n}_2)/3, \\ a_{021} &= (2\mathbf{x}_2 + \mathbf{x}_3 - w_{23}\mathbf{n}_2)/3, \quad a_{012} = (2\mathbf{x}_3 + \mathbf{x}_2 - w_{32}\mathbf{n}_3)/3, \\ a_{102} &= (2\mathbf{x}_3 + \mathbf{x}_1 - w_{31}\mathbf{n}_3)/3, \quad a_{201} = (2\mathbf{x}_1 + \mathbf{x}_3 - w_{13}\mathbf{n}_3)/3, \\ E &= (a_{210} + a_{120} + a_{021} + a_{012} + a_{102} + a_{201})/6, \quad V = (\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3)/3, \\ a_{111} &= E + (E - V)/2. \end{aligned} \tag{25}$$

The implemented method is extremely fast and enables to smooth nicely a faceted model. Figure 12 shows an initial triangulation and the new meshes computed with harmonic maps for both a linear and cubic mapping $\mathbf{x}(\xi)$. The advantage of this method compared to smoothing-based approaches combined with direct methods is that our technique does not result in any feature loss.



(a) (b) (c)

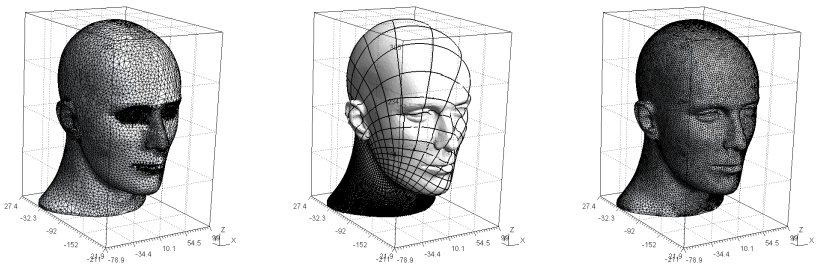
Figure 12. (a) Initial STL triangulation, b) remeshing with a linear map $\mathbf{x}(\xi)$ and c) remeshing with a cubic map $\mathbf{x}(\xi)$.

4. Computational algorithm

The presented algorithm for remeshing consists of different steps as illustrated in Fig.13:

1. Start from an initial triangulation

2. Compute the mapping:
   (a) Divide the surface into surfaces of genus $G = 0$
   (b) Solve two Laplace equations for computing u and v using finite elements
   (c) Verify that the discrete harmonic map is locally one-to-one. If not, proceed as explained in section 3.2
3. Use standard surface meshers to remesh in the parametric space and map the triangulation back to the original surface
4. From the surface mesh, use standard volume meshers to build a 3D finite element mesh



(1) (2) (3)

Figure 13. Remeshing algorithm. 1) Initial triangulation, 2) Harmonic map $u(\mathbf{x})$ and $v(\mathbf{x})$ and 3) new mesh based on the harmonic map.

We will now detail some of the steps involved in the remeshing algorithm cited above.

For step $2(a)$, as explained in section 2, the surface should be of zero genus and have at least one boundary. In case the conditions to compute the harmonic map are not satisfied, the mesh is split into different parts that each satisfy the conditions. We are currently working on an optimal automatic splitting algorithm (numerical homology) that will be presented in an upcoming paper. For step $2(b)$, we use the high-performance direct solver TAUCS to solve the linear system that arises from the finite element discretization of the Laplace equation (16). For step (3), we mesh in the parametric space such that all edges $\mathbf{e}$ have a non-dimensional length of $l_e = 1$, where the non-dimensional length is defined as:

$$l_e = \int_e \frac{1}{\delta(\mathbf{x})} dl, \tag{26}$$

with δ denoting the mesh size field [1] and where dl is given by (9).

5. Examples

The parametrization and remeshing procedure described above has been implemented in the open source finite element mesh generator Gmsh [1]. We present several examples for which this approach may be of interest, namely STL triangulations and triangulations that come from a CAD representation. We then show results of direct mesh generation based on surfaces that are parametrized with harmonic maps.

5.1. Remeshing (low-input) STL triangulations

In this section, we show examples of surfaces that do not have a standard CAD representation such as STL triangulations. Those triangulations can be found in many domains such as 3D scanned images, computer game characters, terrain data, and medical data issued form a segmentation. Figure 14 shows some examples of parametrizations of triangulated surfaces. The examples were found on the INRIA web site [‡] of Eric Saltel.

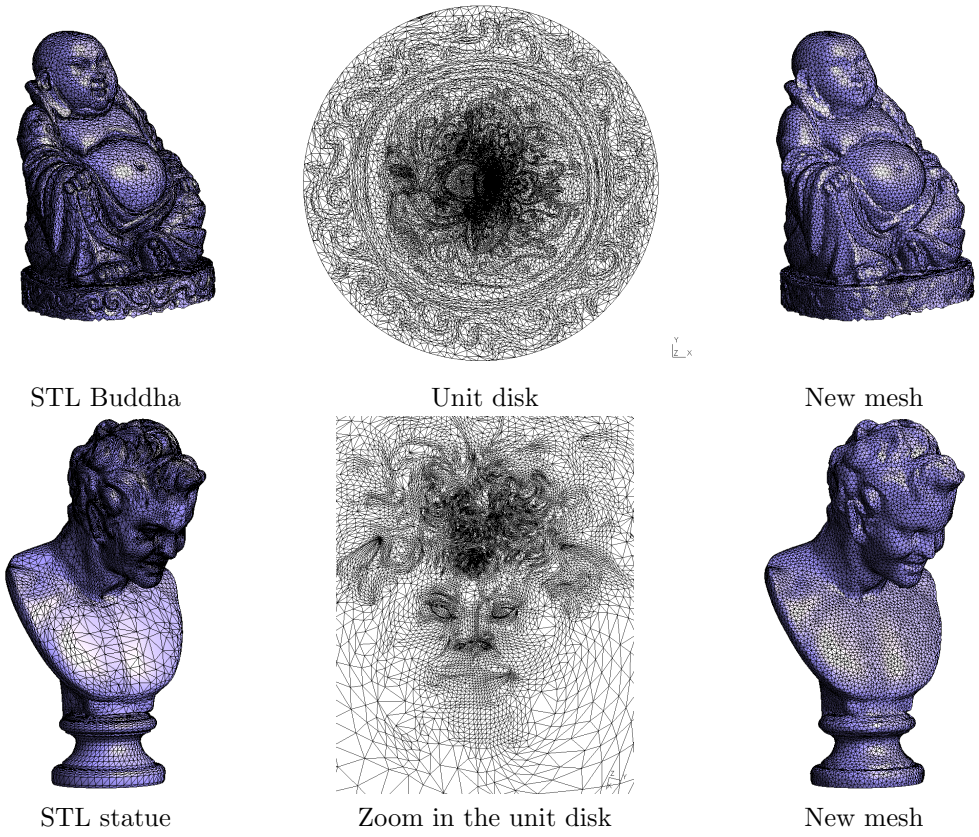


Figure 14. Two examples of parametrizations of STL triangulations.

Figure 15 shows the quality histogram for the initial STL triangulation of a pelvis and arterial bypass presented in Fig.3 and the remeshed geometry. The quality histogram shows the aspect ratio of the surface mesh that is defined as:

$$\eta = K \frac{\text{inscribed radius}}{\text{circumscribed radius}} \tag{27}$$

[‡] `http://www-c.inria.fr/Eric.Saltel/saltel.php`

where K has been chosen so that the equilateral triangle has $\eta = 1$.

As the surface of the pelvis is of genus $G = 1$, one cut is made[§] to generate two surfaces of zero genus and a parametrization based on harmonic maps is performed for each of those two new genus zero surfaces. We can see that with the remeshing procedure, we greatly enhance the quality of the mesh.

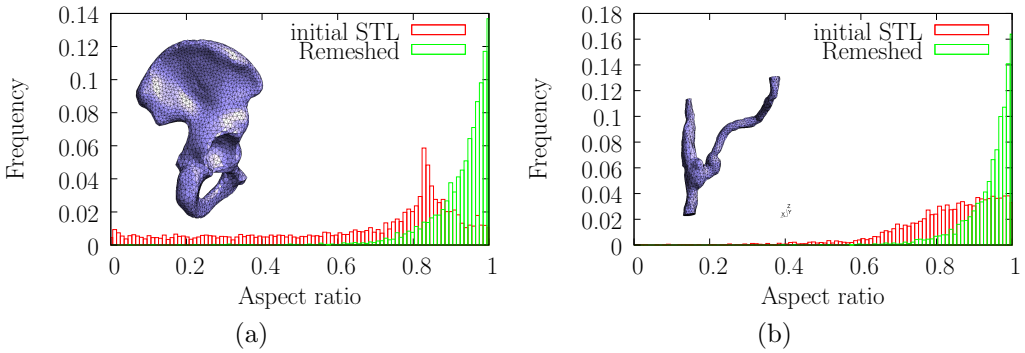


Figure 15. Plot of the quality histogram of both the STL triangulation and the remeshed part of a) a pelvis and b) a bypass of a femoral artery.

Figure 16 compares the efficiency and quality of the proposed method with three alternative remeshing algorithms:

- Local mesh modifications in MeshLab [37] combines a planar flipping optimization algorithm with a subdivision surfaces method [38] that aims at smoothing the surface by successive refinements of the mesh. In order to establish a comparison with the other methods, we apply a zero threshold angle, which leads to a uniform refinement over the mesh.
- The Robust Implicit Moving Least Squares (RIMLS) method described in [39] and implemented in MeshLab [37], whose goal is to obtain smooth representations of surfaces while preserving fine details. We notice that this method is designed for graphical purposes rather than computational meshes.
- The local mesh modification strategy described in [40] and implemented in the MAdLib package [41, 42], which modifies the initial mesh to make it comply with criteria on edge lengths and element shapes by applying a set of standard mesh modifications (edge splits, edge collapses and edge swaps, ...) in an optimal order.

For the purpose of comparison only uniform size fields are prescribed in this test. The computations were performed on a MacBook Pro 2.33GHz Intel Core 2 Duo. We can see that the additional computational effort is quite reasonable for our method and is worth it compared to direct remeshing methods since the mean quality of the mesh is about $\bar{\eta} = 0.94$ for the harmonic map (Gmsh/del2d) method and respectively $\bar{\eta} = 0.78$ for the direct methods that

[§]The cut can be easily realized with Gmsh by using the Mesh > Reclassify tool.

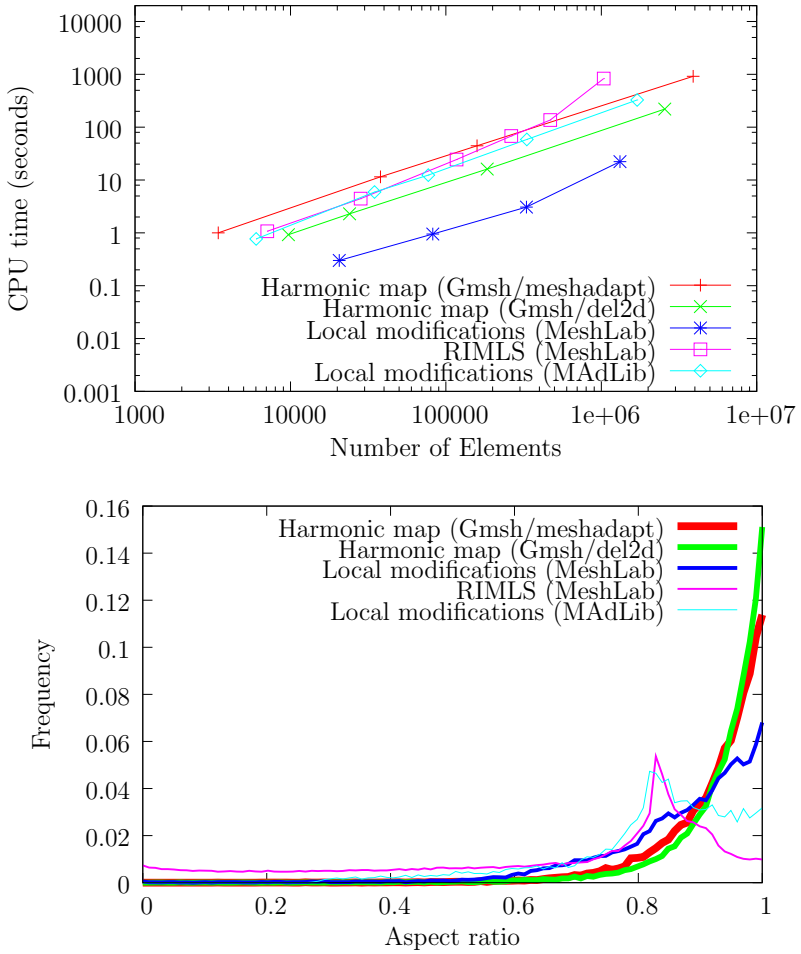
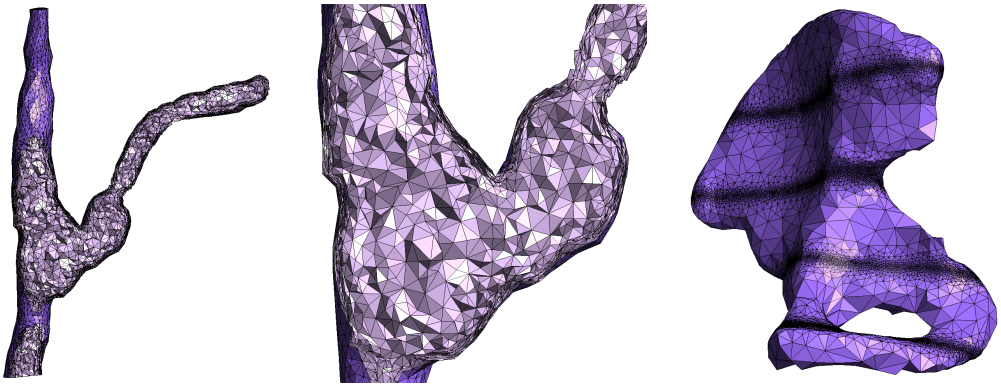


Figure 16. Comparison of the proposed method based on harmonic maps (using two different surface meshing algorithms: Gmsh/meshadapt and Gmsh/del2d [1]) with two direct remeshing methods based on local mesh modifications (MeshLab [37] and MAdLib [41]) and with the RIMLS reparametrization method [39]. Top: comparison of the CPU time requirement. Bottom: comparison of the quality of the surface meshes. (The quality is found to be independent of the mesh size.)

use local mesh modifications with MAdLib and $\bar{\eta} = 0.85$ with MeshLab. Only the subdivision method is faster than the other methods, but it is not designed to handle 3D meshes or prescribed element sizes since the refinement level depends only on the angle between the triangles. Another observation is that our method behaves well regarding the low-quality elements. Finally, the RIMLS method gives meshes with both low quality elements and a poor mean quality ($\bar{\eta} = 0.65$) since it is not intended to produce computational meshes.

Figure 17 shows surface and volume meshes created from low-quality triangulations for actual biomedical applications: an arterial bifurcation and a pelvis. Figure 17b shows how

the remeshed surface can be used to construct high-quality boundary layer meshes for cardiovascular blood flow simulations using MAdLib [41, 42].



(a) (b) (c)

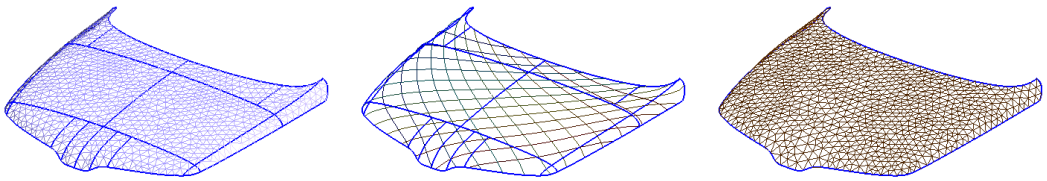
Figure 17. Meshes created from STL triangulations obtained from CT-scans: a) Arterial bifurcation with a uniform edge length on the boundary and a boundary layer mesh, b) zoom of the boundary layer, c) Pelvis with a sinusoidal edge length.

5.2. Remeshing CAD patches

The next example shows a CAD model of a car hood made of 9 different patches that are smoothly connected together (Figure 18). Standard surface meshers mesh each of those patches separately as shown in Fig. 18a. One common issue in engineering analysis is the presence of small sub-patches for the description of one smooth surface which induces the presence of small elements, leading to difficulties in the finite element analysis. It is therefore highly useful to reparametrize those patches into one single surface. This has been done in 2 steps: a mesh has been generated on the multiple patches (Fig. 18a) and the reparametrization (Fig. 18b) has been computed on this first mesh. Then we can remesh the whole compound using any of the surface mesh generators available (Fig. 18c). Note that, in the case of multiple CAD patches reparametrization, points on the reparametrized surface are subsequently projected on the exact CAD model. We use an initial triangulation that is conforming to the patches so that every triangle of this initial triangulation lies on only one CAD patch. It is then easy to compute local CAD coordinates of every point in the new mesh and to compute thereafter their exact location on the CAD model.

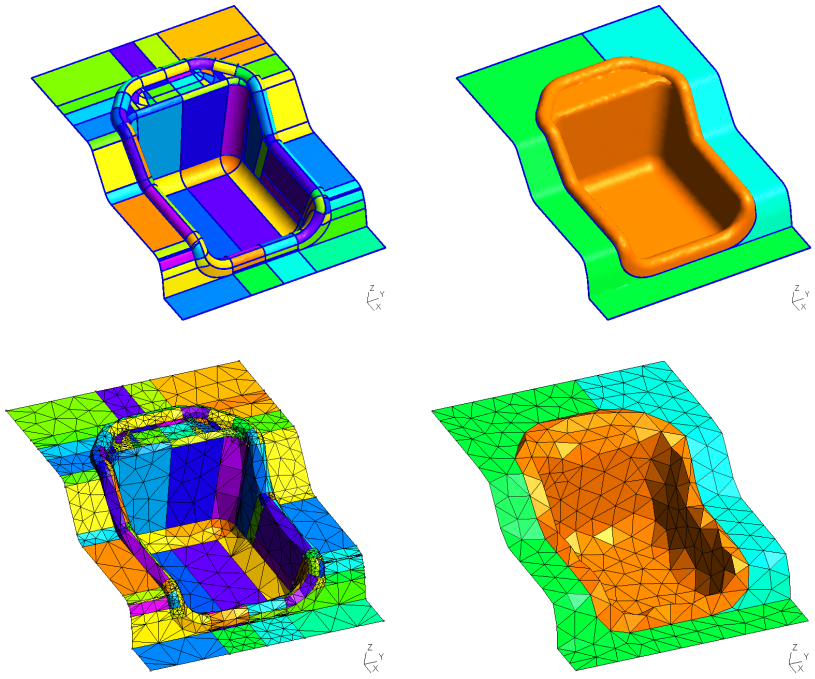
The example of Figure 19 shows another CAD model composed of multiple patches that has been reparametrized into three patches. This example shows clearly the advantage of the approach when coarse meshes have to be generated. Here, reparametrizing a geometry allows to generate smooth uniform meshes. Small details in the initial CAD model induce the generation of small ill-shaped elements. Note the number of reparametrized surfaces was done arbitrarily.

As another an example of a moderately complicated CAD model, we consider the Airbus A319. The aircraft is initially composed of 89 surface patches. After reparametrization, this number has been reduced to 25 (Figure 20). Moreover, lots of curves were reparametrized,



(a) (b) (c)

Figure 18. Surface mesh improvement with a single grouped patch and a parametrization with a harmonic mapping. (a) Meshed CAD model with multiple patches, (b) inverse parametrization $\mathbf{u}(\mathbf{x})$ of all the patches with a single harmonic map and (c) new mesh of the compound.

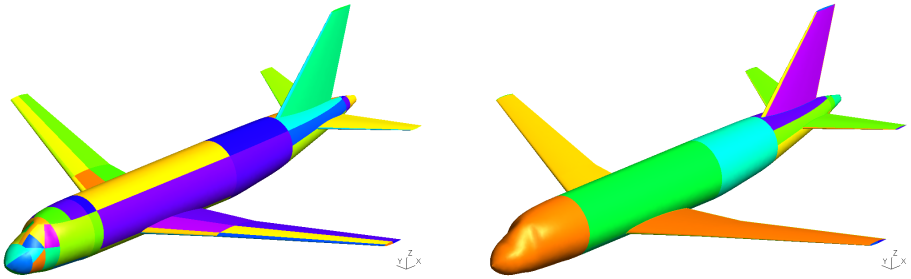


a) b)

Figure 19. Meshing CAD surfaces composed of multitudes of Bezier patches (courtesy of SAMTECH). a) Meshing all the patches separately, b) Using harmonic maps to reparametrize and remesh the CAD into only 3 patches.

especially at the junction between the wings and the fuselage. Figure 20 shows both initial and reparametrized CAD models. Figure 21 shows a global view of the surface mesh as well as a zoom on the mesh of the left wing. The mesh size was adapted to the curvature of the model. Figure 22 shows a large part of the fuselage that has been reparametrized. The reparametrized patch has three holes: two for the wings and one for the right end part. Those three holes, that

are clearly visible in the parametric plane, are highly distorted. Even though, Gmsh's surface meshers were able to produce high quality meshes (Figure 21) with such highly distorded input.



(a) (b)

Figure 20. The model of the airbus A319. Figure(a) shows the initial CAD data made of 89 patches and Figure(b) shows the reparametrized CAD made of only 25 patches.

5.3. Remeshing in Gmsh

As previously mentioned, the remeshing algorithm based on harmonic maps is implemented within the open-source software Gmsh. We show a simple example of how to use it. We suppose that we have an initial surface mesh and write the following text file "remesh.geo":

```
// Merge initial mesh (in .stl, .msh, .mesh, .brep, .medit, etc. format)
Merge "bypass.stl";

// If the initial mesh contains different topological entities,
// then re-create the topology
CreateTopology;

// If necessary, create a topological volume
Surface Loop(55) = {15, 16};
Volume(56) = {55};

// Remesh the Edges and Faces (and Volumes) with harmonic maps
Compound Line(10) = {2, 3}; // merge 2 edges
Compound Surface(100) = {1:24}; // auto-detect boundary
Compound Volume(1000) = {56}
```

6. Conclusions and future work

In this paper, we have presented an efficient method for surface and subsequent volume remeshing. The method is based on the parametrization of a genus zero surface with a harmonic map.

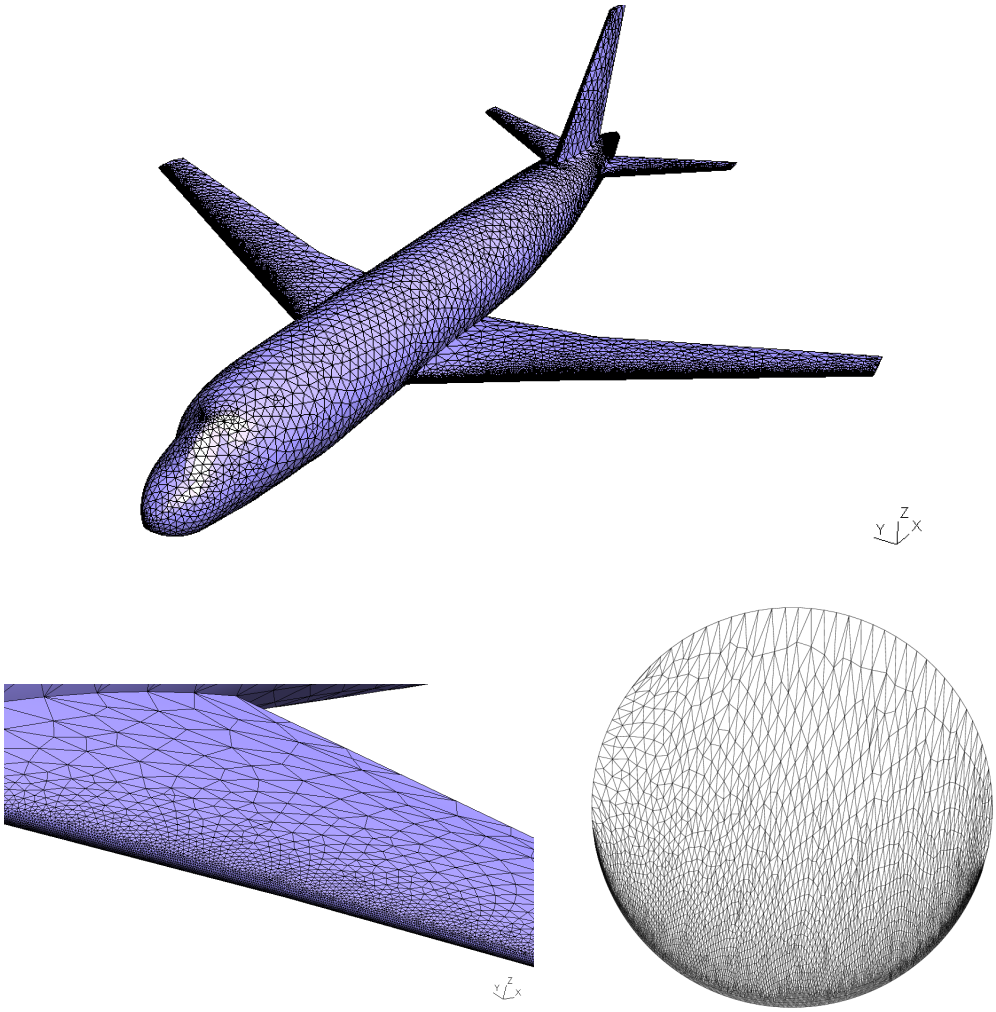


Figure 21. Figure shows the surface mesh of the airbus A319 for which the mesh size is adapted to the principal curvature of the model. Top Figure shows a global view of the mesh while bottom Figures show a zoom of the mesh of a wing, both in the real space and in the unit circle.

We have presented the discrete finite element harmonic map with appropriate boundary conditions. We have introduced a local "cavity check" algorithm to enforce the discrete one-to-one mapping and showed that this approach leads to higher quality meshes than the convex combination map of Floater. A higher order approximation of the geometry based on curved PN triangles was introduced to smooth faceted STL triangulations. Our procedure is easy to implement and very robust against low-quality input triangulations. Compared to smoothing-based approaches our technique does not result in any feature loss and naturally offers refinement options (boundary layer mesh, curvature, etc.). As it enables to remesh

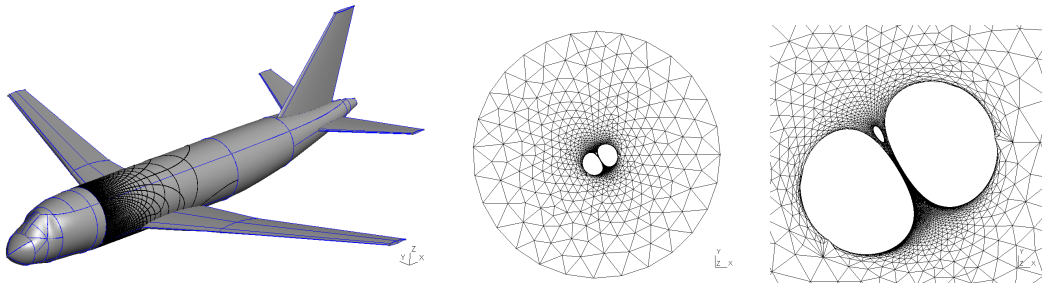


Figure 22. Reparametrization of a large part of the fuselage. Left figure shows iso-contours of u and v. Center and right figures show the mesh in the parametric plane, i.e. inside the unit disk.

multiple CAD patches, the approach can be used to substantially reduce the time required to prepare CAD surface definition for surface mesh generation. The time required to generate the surface mesh is less than $100s$ per 10^6 elements. Furthermore the generated elements have a high mean quality measure which is a clear demonstration of the suitability of the meshes for finite element simulations.

We are currently working on an optimal numerical homology algorithm that will automatically cut a initial surface onto different surfaces of genus zero with uniform geometrical aspect ratio. With this upcoming algorithm we hope to obtain a fully automatic method for high quality remeshing of any topological surface without any geometrical constraint.

Acknowledgements

J.-F. Remacle and C. Geuzaine would like to thank Dr. B. Levy for the fruitful discussions they had on reparametrization techniques at the Trophées du Libre 2009 in Soissons.

REFERENCES

1. Geuzaine C, Remacle JF. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering* 2009; **79**(11):1309–1331.
2. Shewchuk JR. What is a good linear element? interpolation, conditioning, and quality measures. *11th International Meshing Roundtable*, Laboratories SN (ed.), 2002; 115–126.
3. Szczerba D, McGregor R, Szekely G. High quality surface mesh generation for multi-physics bio-medical simulations. *Computational Science – ICCS 2007*, vol. 4487. Springer Berlin, 2007; 906–913.
4. Marcum DL, Gaither A. Unstructured surface grid generation using global mapping and physical space approximation. *Proceedings, 8th International Meshing Roundtable*, 1999; 397–406.
5. Aftosmis M, Delanaye M, Haimes R. Automatic generation of cfd-ready surface triangulation from cad geometry. *AIAA Paper* 1999; **1**(09-0776).
6. Ito Y, Nakahashi K. Direct surface triangulation using stereolithography data. *AIAA Journal* 2002; **40**(3):490–496.
7. Bechet E, Cuilliere JC, Trochu F. Generation of a finite element mesh from stereolithography (stl) files. *Computer-Aided Design* 2002; **34**(1):1–17.
8. Wang D, Hassan O, Morgan K, Weatheril N. Enhanced remeshing from stl files with applications to surface grid generation. *Commun. Numer. Meth. Engng* 2007; **23**:227–239.

9. Borouchaki H, Laug P, George P. Parametric surface meshing using a combined advancing-front generalized delaunay approach. *International Journal for Numerical Methods in Engineering* 2000; **49**:223–259.
10. Zheng Y, Weatherill N, Hassan O. Topology abstraction of surface models for three-dimensional grid generation. *Engrg. Comput.* 2001; **17**(28-38).
11. Marcum DL. Efficient generation of high-quality unstructured surface and volume grids. *Engrg. Comput.* 2001; **17**:211–233.
12. Tristano J, Owen S, Canann S. Advancing front surface mesh generation in parametric space using riemannian surface definition. *Proceedings of 7th International Meshing Roundtable*. Sandia National Laboratory, 1998; 429–455.
13. Laug P, Boruchaki H. Interpolating and meshing 3d surface grids. *International Journal for Numerical Methods in Engineering* 2003; **58**:209–225.
14. M Attene MS B Falcidieno, Wyvill G. A mapping-independent primitive for the triangulation of parametric surfaces. *Graphical Models* 2003; **65**(260-273).
15. Bennis C, Vézien JM, Iglésias G. Piecewise surface flattening for non-distorted texture mapping. *ACM SIGGRAPH Computer Graphics* 1991; :237 – 246.
16. Maillot J, Yahia H, Verroust A. Interactive texture mapping. *Proceedings of ACM SIGGRAPH'93*, 1993; 27–34.
17. Floater MS. Parametrization and smooth approximation of surface triangulations. *Computer aided geometric design* 1997; **14**(231-250).
18. Greiner G, Hormann K. Interpolating and approximating scattered 3d data with hierarchical tensor product splines. *Surface Fitting and Multiresolution Methods*, 1996; 163–172.
19. Levy B, Petitjean S, Ray N, Maillot J. Least squares conformal maps for automatic texture atlas generation. *Computer Graphics (Proceedings of SIGGRAPH 02)*, 2002; 362 – 371.
20. Sheffer A, Praun E, Rose K. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.* 2006; **2**(2):105–171.
21. Floater MS, Hormann K. Surface parameterization: a tutorial and survey. *Advances in Multiresolution for Geometric Modelling* 2005; .
22. Eck M, DeRose T, Duchamp T, Hoppe H, Lounsbery M, Stuetzle W. Multiresolution analysis of arbitrary meshes. *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 1995; 173–182.
23. Schoen R, Yau S. *Lectures on Harmonic Maps*. International Press, Harvard University, 1997.
24. Alliez P, Meyer M, Desbrun M. Interactive geometry remeshing. *Computer graphics (Proceedings of the SIGGRAPH 02)* 2002; :347–354.
25. Floater MS. Parametric tilings and scattered data approximation. *International Journal of Shape Modeling* 1998; **4**:165–182.
26. Rado T. Aufgabe 41. *Math-Verien* 1926; :49.
27. Choquet C. Sur un type de représentation analytique généralisant la représentation conforme et définine au moyen de fonctions harmoniques. *Bull. Sci. Math* 1945; **69**(156-165).
28. Floater MS. One-to-one piecewise linear mappings over trinagulations. *Math. Comp* 2003; **72**(685-696).
29. M P, G EM. Quasi-monotonic continuous darcy-flux approximations in 3-d for any element type. *SPE Reservoir Simulation Symposium*, Houston TX, USA, 2007.
30. Lee DT, Preparata F. An optimal algorithm for finding the kernel of a polygon. *Journal of the ACM, Volume 26* 1979; **26**(3):415–421.
31. Icking C. earching for the kernel of a polygon: A competitive strategy using self-approaching curves. *11th Annu. ACM Sympos. Comput. Geom*, 1995; 258–266.
32. Chang CT, Gorissen B, Melchior S. Fast computation of the minimal oriented bounding box on the rotation group so(3). *ACM Transactions on Graphics* 2009, to appear; .
33. Legrand S, E D, Hanert E, Legat V, Wolanski E. High-resolution, unstructured meshes for hydrodynamic models of the great barrier reef, australia. *Estuarine, coastal and shelf science* 2006; **68**(1-2):26–46.
34. Shinagawa Y, Kunii T, Kergosien YL. Surface coding based on morse theory. *IEEE Computer Graphics and Applications* 1991; **11**(5):66–78.
35. Boubekeur T, Alexa M. Phong tessellation. *ACM Transactions on Graphics* 2008; **27**(5):141:1–141:5.
36. Vlachos A, Peters J, Boyd C, Mitchell JL. Curved pn triangles. *Proceedings on 2001 Symposium on INteractive 3D graphics*, 2001; 159–166.
37. 3D-CoForm. Meshlab 2009. http://meshlab.sourceforge.net/.
38. Loop C. Smooth subdivision surfaces based on triangles. Master's Thesis, Department of Mathematics, University of Utah August 1987.
39. Oztireli A, Guennebaud G, Gross M. Feature preserving point set surfaces based on non-linear kernel regression. *EUROGRAPHICS* 2009; **28**(2).
40. Compère G, Remacle JF, Marchandise E. Transient mesh adaptivity with large rigid-body displacements. *Proceedings of the 17th International Meshing Roundtable*, vol. 3, Garimella R (ed.), Springer, 2008;

213–230.
41. Compère G, Remacle JF. A mesh adaptation framework for large deformations. *International Journal for Numerical Methods in Engineering* 2009; Accepted.
42. Compère G, Remacle JF. Website of madlib: Mesh adaptation library 2008. http://www.madlib.be.