Noname manuscript No.
(will be inserted by the editor)

Optimal parametrizations for surface remeshing

Emilie Marchandise · Jean-François Remacle · Christophe Geuzaine

Received: date / Accepted: date

Abstract We present different linear parametrizations techniques for the purpose of surface remeshing: the energy minimizing harmonic map, the convex map and the least square conformal map. The implementation of those mappings as well as the associated boundary conditions are presented in a unified manner and the issues of triangle flipping and folding that may arise with discrete linear mappings are discussed. We explore the optimality of these parametrizations for surface remeshing by applying several classical 2D meshing algorithms in the parametric space and by comparing the quality of the generated elements. We present various examples that permit to draw guidelines that a user can follow in choosing the best parametrization scheme for a specific topology, geometry, and characteristics of the target output mesh.

Keywords Parametrization · Convex combination map · Harmonic map · Conformal map · Surface remeshing · Anisotropic mesh · Quadrangular mesh

1 Introduction

Thera are many applications for which it might be desirable to remesh a $3D$ surface. In this paper, we focus on two kinds of applications (see Fig. 1).

The first application concerns medical geometries that are often described only by triangulations (in stereolitography STL format). These triangulations are usually the result of a segmentation procedure from the CT scan or MRI dicom-images. Such triangulations can be oversampled and have triangles of

Emilie Marchandise and Jean-François Remacle
Université catholique de Louvain, Institute of Mechanics, Materials and Civil Engineering (iMMC), Place du Levant 1, 1348 Louvain-la-Neuve, Belgium
emilie.marchandise@uclouvain.be, jean-francois.remacle@uclouvain.be

Christophe Geuzaine
Université de Liège, Department of Electrical Engineering and Computer Science, Montefiore Institute B28, Grande Traverse 10, 4000 Liège, Belgium cgeuzaine@ulg.ac.be

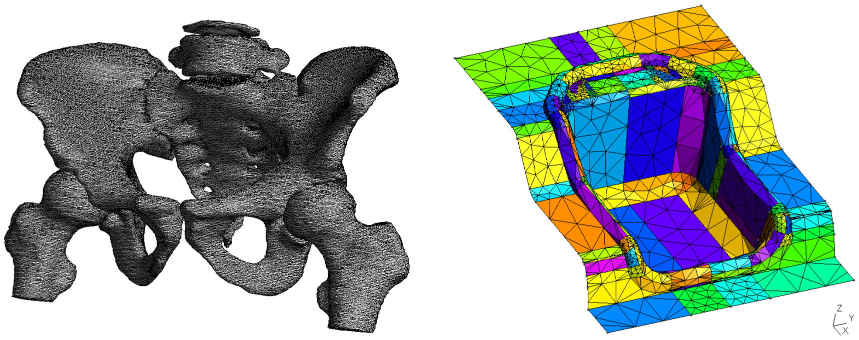


Fig. 1 Examples of geometries for which a remeshing procedure is desirable. Left figure shows an example of an oversampled STL triangulation resulting from a mesh segmentation of a human pelvis and right figure shows the straightforward meshing of a CAD geometry of a maxi-cosi.

poor quality with small elementary angles. The low quality meshes are not suitable for finite element simulations since the quality of the mesh will impact both on the accuracy and efficiency of the numerical method [41,2]. In this case, it is desirable to build a high quality mesh from those low quality meshes before performing any numerical simulation.

The other application is about CAD models. CAD models are often made of a huge amount of patches that have no physical significance and a straightforward meshing of the patches often leads to meshes that are not suitable for finite element simulations. Indeed, as most surface mesh algorithms mesh model faces individually, mesh points are generated on the bounding edges of those patches and if thin CAD patches exist in the model they will result in the creation of small distorted triangles with very small angles (Fig. 2). The low quality elements present in the surface mesh will often hinder the convergence of the FE simulations on those surface meshes. They also prevent the generation of quality volumetric meshes for three-dimensional finite element computations (CFD, structure mechanics, etc.). An efficient manner to build a high quality mesh for such CAD models is then to build from the initial CAD mesh a cross-patch parametrization that enables the remeshing of merged patches.

There are mainly two approaches for surface remeshing: mesh adaptation strategies [20,3,44] and parametrization techniques [6,46,29,42,21,24]. 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

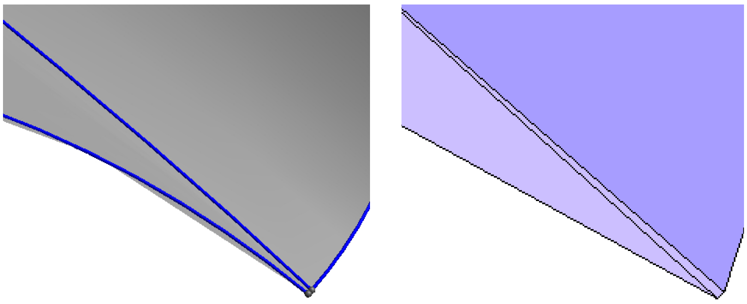


Fig. 2 Example of 2 patches of a CAD geometry (left) for which the mesh (right) contains a very small triangle of poor quality.

been used extensively for applying textures onto surfaces [5,25] and have become a very useful and efficient tool for many mesh processing applications [9, 16,23,39,13]. 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 planar mesh generation procedure and the new triangulation is then mapped back to the original surface [8,30].

The existing methods for discrete parametrization can be classified as follows: linear, non-linear and hybrid methods. Non-linear methods based on discrete or differential-geometric non-linear distortion measure [19,40,47] offer strong guarantees on the absence of triangle folding and flipping at the cost of a generally higher computational effort. Some authors have also suggested hybrid techniques that linearize those non-linear measures at the cost of only a few linear solves [4,45]. Linear methods require only the resolution of a single linear system. Most methods require mapping the vertex of the patch boundary to a given polygon (usually convex) in the parametric plane. This is for example the case of the discrete harmonic map introduced by Eck [8] or the more robust convex combination map [9] that is guaranteed to be one-to-one if the boundary is mapped onto a convex polygon. Among the convex combinations maps we have the barycentric map by Tutte [43] and the more recent convex map that mimics the behavior of the harmonic map [11]. Some authors also suggested boundary-free parameterization by pinning down only two vertices. This is the case for example in the least square conformal maps (LSCM) introduced by Levy et al. [23] and the discrete conformal parametrizations (DCP) of Desbrun et al. [1]. These mappings can achieve lower angle distortion than previous results. However, as the quality of the parametrization can depend significantly on the choice of the constrained vertices, Mullen et al. [31] suggested spreading the constraints throughout the mesh by constraining the barycenter of the mapping to be at $(0,0)$ and the moment of inertia of the boundary to be unity. In [31], those spread constraints are taken into account through recourse to spectral theory.

In this paper, we present and compare (in Sec. 2 to Sec. 5) different types of linear mappings for the discrete parametrization of triangulated surfaces. The implementation of those mappings as well as the boundary conditions are presented in a very comprehensive manner, with the aim to make the exposition accessible to a wider community than the one of computer graphics—in particular to the finite element community. The discrete parametrization aims at computing the discrete mapping $\mathbf{u}(\mathbf{x})$ that maps every triangle of the three dimensional surface $\mathcal{S}$ to another triangle of $\mathcal{S}'$ that has a well known parametrization:

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

We then restrict ourselves to the parametrization of non-closed triangulated surfaces[1] since we have already presented in previous papers [28, 26, 27] efficient techniques to split a closed object into a series of different mesh patches.

Once the discrete mapping has been computed, the metric tensor of the mapping can be computed as follows (see [34] for more details):

$$\mathbf{M_u} = \left(\frac{\partial \mathbf{x}}{\partial \mathbf{u}}\right)^T \cdot \left(\frac{\partial \mathbf{x}}{\partial \mathbf{u}}\right). \tag{2}$$

This metric tensor is used by the 2D mesh generators to compute the lengths and the angles in the parametric space. Most of the planar mesh generators require that the metric tensor is sufficiently smooth.

We then discuss (in Sec. 6) optimal parametrizations for surface remeshing. The aim of the surface mesh generation process is to build a mesh that has controlled element sizes $\delta(x)$ and shapes. We are interested in generating uniform and non-uniform (isotropic and anisotropic) triangular and quadrangular meshes. Our surface remeshing technique relies on a parametrization and the remeshing is performed in the 2D parametric space using planar mesh generators (e.g. Delaunay or Frontal meshers). We define an optimal parametrization as a parametrization that used by the 2D planar mesh generators will give the best quality meshes. Different mesh quality criteria are then defined for several target output meshes. The shape quality of isotropic triangular meshes can be evaluated by computing the aspect ratio of every triangle as follows [15]:

$$\kappa = \alpha \frac{\text{inscribed radius}}{\text{circumscribed radius}} = 4 \frac{\sin \hat{a} \sin \hat{b} \sin \hat{c}}{\sin \hat{a} + \sin \hat{b} + \sin \hat{c}}, \tag{3}$$

$\hat{a}, \hat{b}, \hat{c}$ being the three inner angles of the triangle. With this definition, the optimal equilateral triangles have $\kappa = 1$ and the flat degenerated triangles have $\kappa = 0$. The shape quality quadrangular meshes is evaluated by computing the quality η of every quadrangle as follows:

$$\eta = \max\left(1 - \frac{2}{\pi} \max_k \left(\left|\frac{\pi}{2} - \alpha_k\right|\right), 0\right), \tag{4}$$

[1] Non-closed surfaces are surfaces of zero genius with at least one boundary. The genus of a surface is defined as the number of handles in the surface.

where $\alpha_k, k = 1, .., 4$ are the four angles of the quadrilateral. This quality measure is $\eta = 1$ if the element is a perfect quadrilateral and is $\eta = 0$ if one of those angles is either ≤ 0 or $\geq \pi$. A good computational mesh should have a high mean value of shape quality $\bar{\kappa}$ and a high value of minimum shape quality $\kappa_{\min}$. Having the right element shapes for the triangular and quadrangular mesh is not enough: mesh generators should also provide meshes with controlled edge length. Here, we define the *efficiency index* [48] τ of a mesh as the exponential of the mean value of the difference between each adimensional edge length l_i^h and one:

$$\tau[\%] = 100 \exp\left(\frac{1}{n_e}\sum_{i=1}^{n_e} d_i\right), \tag{5}$$

with $d_i = l_i^h - 1$ if $l_i^h < 1$, $d_i = \dfrac{1}{l_i^h} - 1$ if $l_i^h > 1$ and n_e the number of edges in the mesh. In (5), the adimensional length l_i^h of the mesh edge $\mathbf{y}_i = \mathbf{b} - \mathbf{a}$ of R^3 with respect to the non uniform size field $\delta(\mathbf{x})$ is defined as

$$l = \|\mathbf{y}\| \int_0^1 \frac{1}{\delta(\mathbf{a} + t\mathbf{y})}\, dt. \tag{6}$$

An optimal mesh in term of the size is a mesh for which every edge i is of adimensional size l_i^h equal to one so that the efficiency index should be $\tau = 100\%$. Surface mesh algorithms usually produce meshes with typical values of efficiency index τ around $\tau = 85\%$, with non-dimensional sizes around $1/\sqrt{2} \leq l_i^h \leq \sqrt{2}$.

Finally, after drawing the conclusions (in Sec. 7), we present (in Sec. 8) guidelines that a user can follow in choosing the best parametrization scheme for a specific topology, geometry, and characteristics of the target output mesh.

2 Harmonic mappings

In this section, we introduce three different harmonic mappings: the harmonic map that minimizes the Dirichlet energy, the discrete convex map that mimics the harmonic map and the conformal map which is a harmonic map that moreover conserves the angles. In the next section, we will show how to compute the discrete mappings on a given input triangular mesh.

2.1 Energy minimizing harmonic map

A harmonic map minimizes distortion in the sense that it minimizes the Dirichlet energy of the mapping $\mathbf{u}(\mathbf{x})$:

$$E_D(\mathbf{u}) = \int_{\mathcal{S}} \frac{1}{2} |\nabla \mathbf{u}|^2\, ds, \tag{7}$$

subject to Dirichlet boundary conditions $\mathbf{u} = \mathbf{u}_D$ on one of the N-closed boundaries of the surface[2]. This Dirichlet boundary is denoted $\partial\mathcal{S}^0$. Harmonic maps are not in general conformal and do not preserve angles but they are popular since they are very easy to compute and are guaranteed to be one-to-one for convex regions [32, 7].

The strong formulation corresponding to this minimization problem is given by:

$$\nabla^2 u = 0, \quad \nabla^2 v = 0 \quad \text{on} \quad \mathcal{S} \tag{8}$$

with appropriate Dirichlet and Neumann boundary conditions on the N boundaries of $\mathcal{S}$, denoted $\partial\mathcal{S}^i, i = 1, ..., N$:

$$\mathbf{u} = \mathbf{u}_D(\mathbf{x}), \quad \text{on} \ \ \partial\mathcal{S}^0, \quad \frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0, \qquad \text{on} \ \ (\partial\mathcal{S} - \partial\mathcal{S}^0). \tag{9}$$

This means that if one can compute numerically the solution to those two partial differential equations (8), the solutions $u(\mathbf{x})$ and $v(\mathbf{x})$ define the discrete harmonic mapping $\mathbf{u}(\mathbf{x})$.

2.2 Convex combination map

In contrast to the continuous harmonic map (7), Floater et al. showed in [10, 12] that the discrete version of (7) is not always one-to-one. To ensure a discrete maximum principle, they introduced a convex combination with coefficients λ_{ik} that are such that every interior vertex $\mathbf{u}_i$ is a convex combination of its d_i neighboring vertices:

$$\mathbf{u}_i = \sum_{k=1}^{d_i} \lambda_{ik} \mathbf{u}_k, \ \text{with} \quad \sum_{k=1}^{d_i} \lambda_{ik} = 1. \tag{10}$$

The coefficients λ_{ik} can also be expressed as:

$$\lambda_{ik} = \frac{w_{ik}}{\sum_k w_{ik}}, \tag{11}$$

where w_{ik} are the weights of the convex mapping. The simplest convex mapping is the *barycentric convex mapping* by Tutte [43] for which the coefficients are uniform for every interior vertex $\mathbf{u}_i$ and are equal to the inverse of the valency (d_i) of the interior vertex $\lambda_{ik} = 1/d_i$. These coefficients can also be build by assembling the contributions for every mesh element e (see Fig. 3, where e and e' are the two mesh elements sharing the edge ik):

$$w_{ik} = w_{ik}^e + w_{ik}^{e'} \ \text{with} \quad w_{ik}^e = w_{ik}^{e'} = \frac{1}{2} \tag{12}$$

[2] For example, a non-closed surface with one hole (see Fig. 9) has $N = 2$ closed boundaries.

We will show however that this *barycentric convex mapping* leads to a non-smooth metric tensor (Eq. (2)) and that some 2D mesh generators are not comfortable with such non-smooth metric tensors.

Another alternative is to design a convex mapping that mimics the behavior of the harmonic map. The idea in [11] is that harmonic functions satisfy the mean value theorem, i.e that at every point of the domain the value of the harmonic function $\mathbf{u}_i(\mathbf{x}_i)$ is equal to the average of its values around any circle at that point:

$$\mathbf{u}_i = \frac{1}{2\pi r_i} \int_{\Gamma_i} \mathbf{u}(\mathbf{x})\, ds. \tag{13}$$

We can then derive that independently of $r_i > 0$, this equation can be written as a convex combination where the weights of this *mean value convex map* are (see Fig. 3):

$$w^e_{ik} = \frac{\tan(\alpha^e_i/2)}{\|\mathbf{x}_i - \mathbf{x}_k\|}, \quad w^{e'}_{ik} = \frac{\tan(\alpha^{e'}_i/2)}{\|\mathbf{x}_i - \mathbf{x}_k\|}. \tag{14}$$

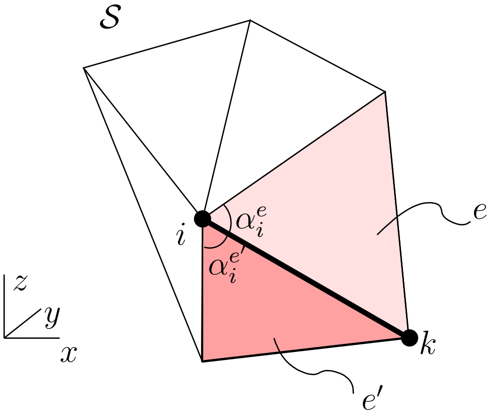


Fig. 3 Illustration of the vertices and the angles which needs to be evaluated for the computation of the weights of the mean value convex map. The two mesh elements sharing the edge ik are noted e and e'.

2.3 Least square conformal map

The least square conformal map as introduced by Levy at al. [23] asks that the gradient of u and the gradient of v be as orthogonal as possible in the parametrization and have the same norm. This can be seen as an approximation of the Cauchy-Riemann equations. For a piecewise linear mapping, the least square conformal map can be obtained by minimizing the energy:

$$E_{\text{LSCM}}(\mathbf{u}) = \int_{\mathcal{S}} \frac{1}{2} \left|\nabla u^{\perp} - \nabla v\right|^2 ds, \tag{15}$$

where $^{\perp}$ denotes a counterclockwise 90° rotation in $\mathcal{S}$. For a 3D surface defined by a normal vector $\mathbf{n}$, the counterclockwise rotation of the gradient can be written as: $\nabla u^{\perp} = \mathbf{n} \times \nabla \mathbf{u}$ (see Fig. 4).

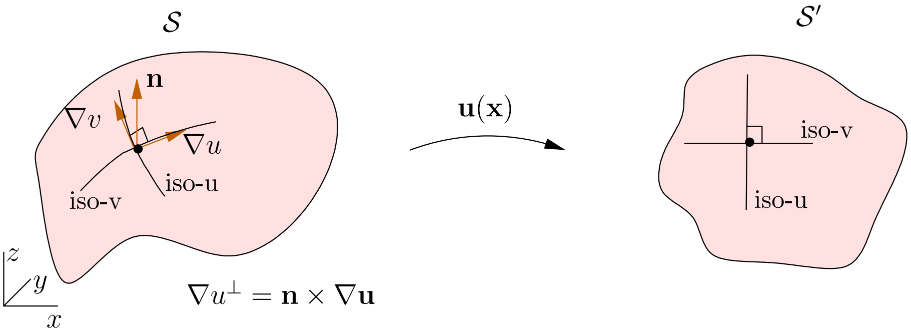


Fig. 4 Definitions for a conformal mapping. $\nabla u^{\perp}$ denotes the counterclockwise 90° rotation of the gradient ∇u for a 3D surface.

Equation (15) can be simplified and rewritten as follows:

$$\begin{aligned} E_{\text{LSCM}}(\mathbf{u}) &= \int_{\mathcal{S}} \frac{1}{2} \left(\nabla u^{\perp} \cdot \nabla u^{\perp} + \nabla v \cdot \nabla v - 2\nabla u^{\perp} \cdot \nabla v\right) ds \\ &= \int_{\mathcal{S}} \frac{1}{2} \left(\nabla u \cdot \nabla u + \nabla v \cdot \nabla v - 2\left(\mathbf{n} \times \nabla u\right) \cdot \nabla v\right) ds. \end{aligned} \tag{16}$$

Recalling the idenity that a “dot” and a “cross” can be interchanged without changing the result, we have

$$\begin{aligned} E_{\text{LSCM}}(\mathbf{u}) &= \int_{\mathcal{S}} \frac{1}{2} \left((\nabla u)^2 + (\nabla v)^2\right) - \mathbf{n} \cdot (\nabla u \times \nabla v)\, ds \\ &= E_D(\mathbf{u}) - \mathcal{A}(\mathbf{u}), \end{aligned} \tag{17}$$

where $\mathcal{A}(\mathbf{u})$ is the area functional. It follows from (17) that the conformal energy is then simply defined as the dirichlet functional $E_D(\mathbf{u})$ minus the area functional $\mathcal{A}(\mathbf{u})$.

Figure 5 shows two examples of mappings $\mathbf{u}(\mathbf{x})$ of a 3D surface $\mathcal{S}$. The top figures shows the harmonic map that is computed by minimizing the Dirichlet energy $E_D(\mathbf{u})$ and the bottom figures show a conformal map that is obtained by minimizing the least square conformal energy $E_{LSCM}(\mathbf{u})$. The iso-u and iso-v values that are visible on the 3D surface show that the conformality is also almost achieved for the harmonic map except near the fixed boundaries.

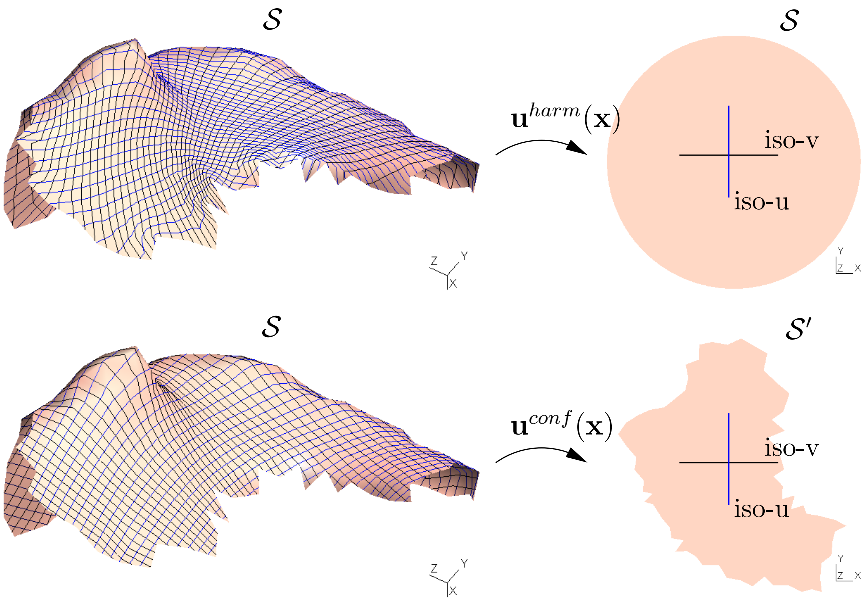


Fig. 5 Example of harmonic (top) and conformal (bottom) maps for the mapping of a surface $\mathcal{S}$, computed respectively by minimizing the Dirichlet energy $E_D(\mathbf{u})$ and the least square conformal energy $E_{LSCM}(\mathbf{u})$. The iso-u and iso-v values are visible on the 3D surfaces; note how the conformality is almost achieved for the harmonic map except near the fixed boundaries.

3 Discrete harmonic mappings

We start this section by deriving first the finite element formulation of the quadratic minimization problems (7) and (15). We denote by the functional J either the Dirichlet energy E_D or the least-square conformal energy E_{LSCM} to be minimized:

$$\mathbf{u}^* = \arg\min_{\mathbf{u}\in\mathbf{U}(\mathcal{S})} J(\mathbf{u}), \quad \text{with} \quad \mathbf{U}(\mathcal{S}) = \{\mathbf{u} \in H^1(\mathcal{S}),\ \mathbf{u} = \mathbf{u}_D(\mathbf{x}) \text{ or } \frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0 \text{ on } \partial\mathcal{S}^i\}, \tag{18}$$

where $\mathbf{u}_D$ denotes the Dirichlet boundary conditions. We assume the following finite expansion for $\mathbf{u}$:

$$\mathbf{u}_h(\mathbf{x}) = \sum_{i\in I} \mathbf{u}_i \phi_i(\mathbf{x}) + \sum_{i\in J} \mathbf{u}_D(\mathbf{x}_i)\phi_i(\mathbf{x}), \tag{19}$$

where I denotes the set of nodes of $\mathcal{S}$ that do not belong to the Dirichlet boundary, where J denotes the set of nodes of $\mathcal{S}$ 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: $\phi_i(\mathbf{x}_j) = \delta_{ij}$.

Thanks to expansion Eq. (19), the functional J defining the energy of the least square conformal map Eq. (17) can be rewritten as

$$\begin{aligned} J(\mathbf{u}_1, \ldots, \mathbf{u}_N) = {} & \frac{1}{2} \sum_{i \in I} \sum_{j \in I} u_i u_j \int_{\mathcal{S}} \nabla\phi_i \cdot \nabla\phi_j \, ds + \sum_{i \in I} \sum_{j \in J} u_i u_D(\mathbf{x}_j) \int_{\mathcal{S}} \nabla\phi_i \cdot \nabla\phi_j \, ds + \\ & \frac{1}{2} \sum_{i \in I} \sum_{j \in I} v_i v_j \int_{\mathcal{S}} \nabla\phi_i \cdot \nabla\phi_j \, ds + \sum_{i \in I} \sum_{j \in J} v_i v_D(\mathbf{x}_j) \int_{\mathcal{S}} \nabla\phi_i \cdot \nabla\phi_j \, ds + \\ & \sum_{i \in I} \sum_{j \in J} u_D(\mathbf{x}_i) u_D(\mathbf{x}_j) \int_{\mathcal{S}} \nabla\phi_i \cdot \nabla\phi_j \, ds + \\ & \sum_{i \in I} \sum_{j \in J} v_D(\mathbf{x}_i) v_D(\mathbf{x}_j) \int_{\mathcal{S}} \nabla\phi_i \cdot \nabla\phi_j \, ds - \\ & \sum_{i \in I} \sum_{j \in J} u_i v_j \int_{\mathcal{S}} \mathbf{n} \cdot (\nabla\phi_i \times \nabla\phi_j) \, ds - \\ & \sum_{i \in I} \sum_{j \in J} u_D(\mathbf{x}_i) v_j \int_{\mathcal{S}} \mathbf{n} \cdot (\nabla\phi_i \times \nabla\phi_j) \, ds - \\ & \sum_{i \in I} \sum_{j \in J} u_i v_D(\mathbf{x}_i) \int_{\mathcal{S}} \mathbf{n} \cdot (\nabla\phi_i \times \nabla\phi_j) \, ds - \\ & \sum_{i \in I} \sum_{j \in J} u_D(\mathbf{x}_i) v_D(\mathbf{x}_i) \int_{\mathcal{S}} \mathbf{n} \cdot (\nabla\phi_i \times \nabla\phi_j) \, ds. \end{aligned} \tag{20}$$

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

$$\begin{aligned} \frac{\partial J}{\partial u_k} = {} & \sum_{j \in I} u_j \underbrace{\int_{\mathcal{S}} \nabla\phi_k \cdot \nabla\phi_j \, ds}_{A_{kj}} + \sum_{j \in J} u_D(\mathbf{x}_j) \underbrace{\int_{\mathcal{S}} \nabla\phi_k \cdot \nabla\phi_j \, ds}_{A_{kj}} - \\ & \sum_{j \in I} v_j \underbrace{\int_{\mathcal{S}} \mathbf{n} \cdot (\nabla\phi_k \times \nabla\phi_j) \, ds}_{C_{kj}} - \sum_{j \in I} v_D(\mathbf{x}_j) \underbrace{\int_{\mathcal{S}} \mathbf{n} \cdot (\nabla\phi_k \times \nabla\phi_j) \, ds}_{C_{kj}} \\ = {} & 0 \ , \quad \forall k \in I. \end{aligned} \tag{21}$$

The same can be done for the derivative with respect to v_k.

There are as many equations (21) as there are nodes in I. This system of equations can be written as:

$$\begin{pmatrix} \bar{\bar{A}} & \bar{\bar{C}} \\ \bar{\bar{C}}^T & \bar{\bar{A}} \end{pmatrix} \begin{pmatrix} \bar{U} \\ \bar{V} \end{pmatrix} = \begin{pmatrix} \bar{0} \\ \bar{0} \end{pmatrix}, \tag{22}$$

where $\bar{\bar{A}}$ is a symmetric positive definite matrix and $\bar{\bar{C}}$ is an antisymmetric matrix, both built by assembling the elementary matrices for every mesh triangle e: $A^e = A_{ij}$ and $C^e = C_{ij}$. Hence the resulting matrix in Eq. 22 is symmetric definite positive and efficient direct sparse symmetric-positive-definite solvers can be used. The vectors $\bar{U}$ and $\bar{V}$ denote respectively the vector of unknowns u_k and v_k.

In the case of the energy minimizing harmonic map, the matrix C vanishes, which makes the system of equations (22) uncoupled:

$$\bar{\bar{A}}\bar{U} = \bar{0}, \quad \bar{\bar{A}}\bar{V} = \bar{0}, \tag{23}$$

where the matrix $\bar{\bar{A}}$ is the assembly of the elementary finite element stiffness matrices defined in (21).

Finally, in the case of a convex combination map the system of equations is also uncoupled as in (23), the matrix $\bar{\bar{A}}$ being now the assembly of the following elementary matrices:

$$A^e = A_{ij} = \begin{pmatrix} (w_{12}^e + w_{13}^e) & -w_{12}^e & -w_{13}^e \\ -w_{21}^e & (w_{22}^e + w_{23}^e) & -w_{23}^e \\ -w_{31}^e & -w_{32}^e & (w_{31}^e + w_{32}^e) \end{pmatrix}, \tag{24}$$

where the weights w_{ij}^e are the convex weights given by Eq. (12) for the barycentric convex map and by Eq. (14) for the mean value map. Notice that for the mean value convex map, the resulting matrix $\bar{\bar{A}}$ is not symmetric anymore since $w_{ij} \neq w_{ji}$, so that a symmetric-positive-definite solver can not be used in this case.

4 Boundary conditions

It is necessary to impose appropriate boundary conditions to guarantee that the discrete minimization problem has a unique solution and that this unique solution defines a one-to-one mapping (and hence avoids the degenerate solution $\mathbf{u}$ =constant).

We first consider the **Dirichlet boundary conditions** that are often used for the discrete harmonic and convex mappings:

$$\mathbf{u} = \mathbf{u}_D(\mathbf{x}) \quad \text{on} \quad \partial\mathcal{S}^0. \tag{25}$$

This boundary condition allows mapping of one boundary of $\mathcal{S}$ ($\partial\mathcal{S}^0$) to a fixed boundary in the parametric space (a circle, a square, an ellipse, the projection of $\partial\mathcal{S}$ on its mean plane etc.). The fixed boundaries that are considered are convex polygons since the continuous harmonic map is proven to be one-to-one if the mapped surface $\mathcal{S}'$ is convex. This result is called the Radò-Kneser-Choquet (RKC) theorem [32,7] and it strongly depends on the fact that the harmonic mappings obeys a strong maximum principle: $\mathbf{u}(\mathbf{x})$ 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 6a with one single boundary ($N_B = 1$) that is mapped to a unit circle with appropriate Dirichlet boundary conditions. Surface $\mathcal{S}'$ is convex so that 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 also 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 $v = v_0$.

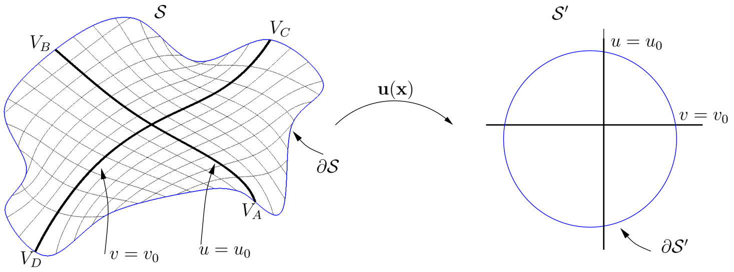


Fig. 6 Iso-values of coordinates u and v on a surface $\mathcal{S}$ for which the mapping $\mathbf{u}(\mathbf{x})$ is computed as the solutions of the Laplace equation with Dirichlet boundary conditions that map $\partial\mathcal{S}$ on the unit circle $\partial\mathcal{S}'$.

Figure 7 illustrates the Dirichlet boundary conditions that map the boundary of the triangulation of the Toutankhamon mask $\partial\mathcal{S}^0$ to a unit circle. This boundary condition can be implemented by choosing arbitrarily a starting vertex on the boundary $\partial\mathcal{S}^0$. Then, the curvilinear abscissa l_i of point $\mathbf{x}_i$ along the curve can computed for all vertices along the curve of total length L. The mapping of the boundary nodes (the position (u_i, v_i)) of the vertex $\mathbf{x}_i$ can then be written as:

$$u_D(\mathbf{x}_i) = \cos\left(\frac{2\pi l_i(\mathbf{x}_i)}{L}\right) \quad , \quad v_D(\mathbf{x}_i) = \sin\left(\frac{2\pi l_i(\mathbf{x}_i)}{L}\right). \tag{26}$$

Instead of fixing all the boundary nodes $\partial\mathcal{S}^0$ to a convex polygon, one might fix two (u, v) coordinates, thus pinning down two vertices in the parameter plane with Dirichlet boundary conditions. Indeed, for least square conformal maps, the mapping (22) has full rank only when the number of pinned vertices is greater or equal to 2 [23]. Pinning down two vertices will set the translation, rotation and scale of the solution when solving the linear system (22) and will lead to what is called a free-boundary parametrization. It was independently found by the authors of the LSCM [23] and the DCP [1] that picking two boundary vertices the farthest from each other seems to give good results in general. However, the quality of the conformal parametrization depends drastically on the choice of these constraint vertices. Indeed, global distortion can ensue and a degradation of conformality can be observed around the pinned vertices. Fig. 8 compares a LSCM with two pinned vertices with

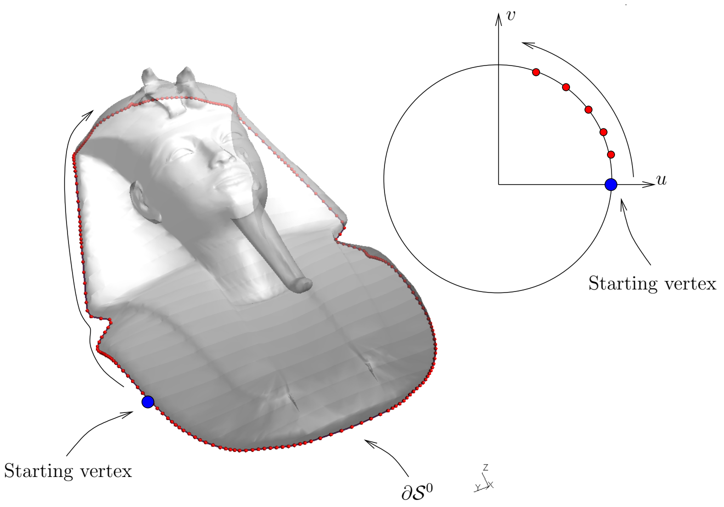


Fig. 7 Dirichlet boundary conditions: mapping the boundary of the Toutankhamon mask $\partial\mathcal{S}^0$ to a unit circle.

a less distorted LSCM that spreads the constraints throughout the mesh (we call this approach the constrained LSCM or CLSCM)

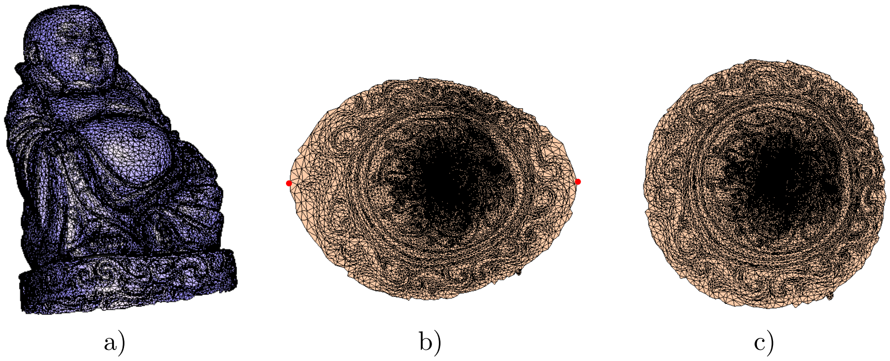


Fig. 8 Initial triangulation $\mathcal{S}$ of a bouddha statue (a) that has been parametrized by computing the LSCM with two constrained vertices (shown in red) (b) and the constrained LSCM solved with a spectral method (c).

How can we define a less distorted least square conformal map (CLSCM) without pinning down two vertices? The idea is to add the two following constraints to the minimization problem that that set the translation, rotation and scale of the solution: (i) the barycenter of the solution must be at zero and (ii) the moment of inertia of the boundary $\partial\mathcal{S}^0$ must be unit. Those constraints can be taken into account through recourse to spectral theory. This idea was derived by Mullen et al. [31] and named **spectral conformal parametrization**. In what follows, we try to present the spectral conformal map in a more comprehensive manner than the way it is presented in [31]. The constrained least square conformal map corresponds to the following discrete constrained minimization problem. Find $\mathbf{U}^*$ such that

$$\mathbf{U}^* = \arg\min_{\mathbf{U}} \frac{1}{2}\mathbf{U^T L_C U}, \quad \text{subject to} \quad \mathbf{U^T E} = 0, \mathbf{U^T B U} = 1. \tag{27}$$

The first constraint in (27) $\mathbf{U^T E} = 0$ states that the barycenter of the solution must be at zero. Indeed, as $\mathbf{E}$ denotes the $2I \times 2$ matrix that is such that $E_{i1} = 1$, $i = 1, ..., I$ and $E_{i2} = 1$, $i = I+1, ..., 2I$ (the other entries of r$\mathbf{E}$ being zero). The second constraint $\mathbf{U^T B U} = 1$ indicates that the moment of inertia of the boundary must be unit, the $\mathbf{B}$ matrix being a $2I \times 2I$ diagonal matrix with 1 at each diagonal element corresponding to boundary vertices and 0 everywhere else. There are two different ways to solve this constrained minimization problem. The first method tries to find the optimum of the following Lagrangian function $\mathcal{L}\left(\mathbf{U}, \boldsymbol{\mu}\right)$ with Lagrange multipliers $\mu_i \geq 0$:

$$\mathcal{L}\left(\mathbf{U}, \boldsymbol{\lambda}, \boldsymbol{\mu}\right) = \frac{1}{2}\mathbf{U^T L_C U} - \boldsymbol{\mu}(\mathbf{U^T E}) - \boldsymbol{\lambda}(\mathbf{U^T B U} - 1). \tag{28}$$

The second method is based on spectral theory that shows that the solution of the constrained minimization problem Eq. (27) is the generalized eigenvector $\mathbf{U}^*$ associated to the smallest non-zero eigenvalue of the matrix $\mathbf{L_C}$, i.e the vector satisfying

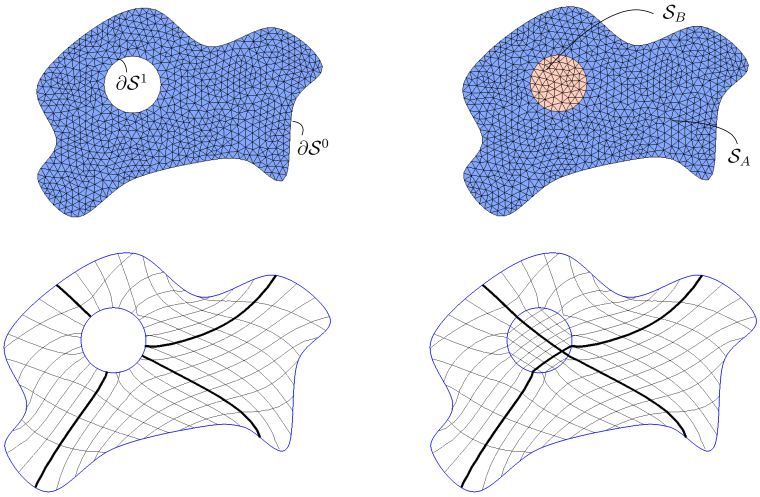
$$\mathbf{L_C U} = \boldsymbol{\lambda} \mathbf{B U}, \tag{29}$$

where λ is the smallest non-zero eigenvalue of $\mathbf{L_C}$. This generalized eigenvector $\mathbf{U}^*$ is called the Fiedler vector of $\mathbf{L_C}$. It can be shown that optimizing (28) is equivalent to finding the Fiedler vector $\mathbf{U}^*$ (29). From a numerical point of view, there exists very efficient eigensolvers that find the Fiedler vector $\mathbf{U}^*$ of the sparse generalized eigenvalue problem we need to solve (29). Those methods usually proceed through Choleski decomposition (to turn the problem into a conventional eigenvalue problem) and Lanczos iterations, particularly fast in our case since we deal with sparse matrices. Software libraries such as Slepc [18] or Arpack [22] provide all those methods for solving the generalized eigenproblem efficiently.

So far, we have discussed Dirichlet boundary conditions and have introduced constrained LSCM in order to avoid pinning down two vertices for the LSCM. Another type of boundary conditions to be considered are natural boundaries, also called **zero Neumann boundary conditions**:

$$\frac{\partial \mathbf{u}}{\partial \mathbf{n}} = 0, \quad \text{on} \quad \partial \mathcal{S}^i. \tag{30}$$

From an implementation point of view, those boundaries conditions correspond to a “do nothing BC” since the Neumann boundary condition is already taken



a) b)

Fig. 9 Triangulation $\mathcal{S}_A$ with one hole on which a discrete harmonic map $\mathbf{u}(\mathbf{x})$ is computed as solution of the Laplace equations $\alpha\nabla^2 u = 0, \alpha\nabla^2 v = 0$ with boundary conditions that map $\partial\mathcal{S}^0$ on the unit circle. The bottom (a) figure shows the iso-u and iso-v values for the mapping ($\alpha = 1$) of the surface $\mathcal{S}$ that excludes the interior disk and for which zero Neumann boundary conditions are applied on $\partial\mathcal{S}^1$. The bottom (b) figure shows the iso-u and iso-v values for the mapping of $\mathcal{S}$ that also includes the interior disk, and for which $\alpha = 1$ inside $\mathcal{S}_A$ and $\alpha = 1.e^{-8}$ (small value) inside $\mathcal{S}_B$.

into account in the minimization problem (18). Consider for example a surface $\mathcal{S}$ that has two boundaries denoted $\partial\mathcal{S}^0$ and $\partial\mathcal{S}^1$ (see Fig. 9a).

Suppose now, we compute the discrete harmonic map by solving (23) with Dirichlet boundary conditions (26) on $\partial\mathcal{S}^0$ and zero Neumann boundary conditions (30) on $\partial\mathcal{S}^1$. The iso-u and iso-v values of this mapping are shown in Fig. 9b). Figures 9a and b) show that imposing Neumann boundary conditions on $\partial\mathcal{S}^0$ is equivalent to the resolution of a Laplace problem on the whole domain $\mathcal{S} = \mathcal{S}_A + \mathcal{S}_B$ while defining a small diffusivity constant α (solving $\alpha\nabla^2 u = 0, \alpha\nabla^2 v = 0$) inside the hole $\mathcal{S}_B$ (Fig. 9b). This means that this problem also 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. However, as can be seen clearly in Fig. 9 the conformality is deteriorated near the Neumann boundary (compare the iso-u and iso-v values in Fig. 9b and Fig. 6). That is the reason for which we have taken the approach for surfaces $\mathcal{S}$ that have several interior boundaries $\partial\mathcal{S}^i$ to mesh the holes, i.e the surfaces such as $\mathcal{S}_B$ that are bounded by $\partial\mathcal{S}^i, i > 0$. The mapping is then computed for an extended surface $\tilde{\mathcal{S}}$ that includes the mesh of the holes (see also [38]) $\tilde{\mathcal{S}} = \mathcal{S} \cup \sum_i \mathcal{S}_{Bi}$.

5 Issues with discrete linear maps

A very important aspect when computing the parametrization is to obtain a bijective (one-to-one) map, where each point of the parametric domain corresponds to exactly one point of the mesh. Let us look at the five different discrete linear maps we have presented so far:

1. the **finite element harmonic** map: solution of the linear system (23) with the elementary matrices defined in (21);
2. the **barycentric convex** map: solution of the linear system (23) where the entries of the elementary matrices (24) are given by (12);
3. the **mean value convex** map: solution of the linear system (23) where the entries of the elementary matrices (24) are given by (14);
4. the **finite element conformal** map: solution of the linear system (22) with the elementary matrices defined in (21);
5. the **spectral conformal** map: solution of the generalized eigenvalue problem (29).

The harmonic and convex mappings map the surfaces onto convex boundaries (e.g. unit circle), while the conformal mappings are open boundary parametrizations. While convex combination maps are always guaranteed to be bijective, non-bijective parametrizations might occur when computing discrete harmonic or conformal map as the solution of linear systems. Figure 10 shows two examples of non-bijective parametrizations. The top figure shows a discrete harmonic map for which the local bijectivity condition is violated. Indeed, the mapping of adjacent mesh triangles intersects and the parametrization contains *triangle flips.* The triangle that has been flipped in the parametric space is the very thin blue triangle on the bottom left of surface mesh. The bottom figure 10 shows a parametrization with a global *domain overlap* (global bijectivity violated). Besides the issues of bijectively, there is an additional important numerical issue for triangulations with large geometrical aspect ratio: the mapping of those surfaces often contains clustered triangles with numerically indistinguishable node coordinates. As this issue has already been discussed in [34,26], we will only discuss the two first issues and present different "check and repair" algorithms to guarantee a one-to-one map.

The first issue concerns **triangle flipping** and is illustrated on the top of Fig. 10. In this figure, looking at the triangle normals in the parametric space, one can see that one triangle has been reversed (the normal of the face is flipped). This issue might occur for harmonic or conformal mappings. One way to enforce the discrete mapping to be one-to-one is to implement a *local cavity check algorithm* that locally modifies interior mesh cavities in which flipping occurs (see [34] for more details). In the case of conformal mappings, triangle flipping might occur also near the open boundaries. We have therefore implemented a *boundary check algorithm* that checks all the parametric triangles adjacent to the boundaries and that reverts the orientation of the triangle by moving the parametric point $\mathbf{u}_i$ to point $\mathbf{u}'_i$ (see Fig. 11). This new point is located along the projection line of the point $\mathbf{u}_i$ on the segment $(\mathbf{u}_{i-1}, \mathbf{u}_{i+1})$.

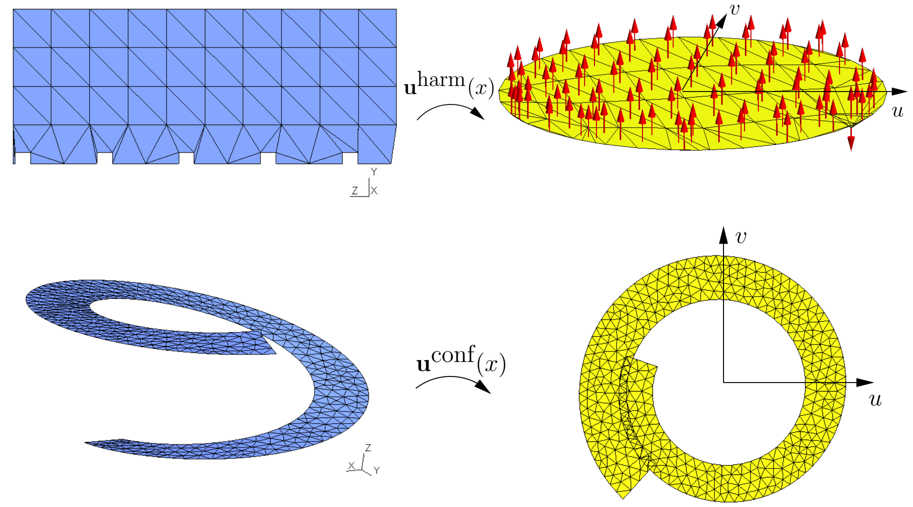


Fig. 10 Non-bijective parametrizations might occur when computing the discrete harmonic or conformal map as the solution of linear systems: a parametrization which contains triangle flips (top) and a parametrization with a global overlap of triangles (bottom).

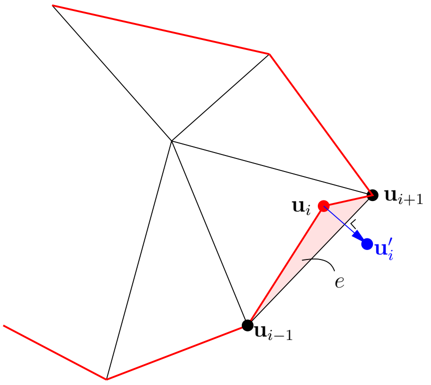


Fig. 11 Boundary check algorithms for repairing triangle flips (such as triangle e) near boundaries. The parametric point $\mathbf{u}_i$ is moved to the the point $\mathbf{u}'_i$ that is moved along the projection line of the point $\mathbf{u}_i$ on the segment $(\mathbf{u}_{i-1}, \mathbf{u}_{i+1})$.

The second issue concerns **overlapping of triangles** that might occur when computing linear conformal maps with open boundaries. In order to detect this issue, we use an idea similar to the one suggested by Sheffer et al. [38] that checks the presence of edge folding. Edge foldings are detected by looping over the edges of the boundary $\partial\mathcal{S}^0$ and by computing if there exist intersections between the edges (see Fig. 12). The intersection of two edge segments (p_1, p_2) and (p_3, p_4) of parametric equations:

$$\mathbf{p_a} = \mathbf{p_1} + t_a(\mathbf{p_2} - \mathbf{p_1}), \quad \mathbf{p_b} = \mathbf{p_1} + t_b(\mathbf{p_4} - \mathbf{p_3}),$$

can be computed by solving for the point where $\mathbf{p_a} = \mathbf{p_b}$. This gives the following linear system of equations:

$$\begin{pmatrix} x_2 - x_1 & x_3 - x_4 \\ y_2 - y_1 & y_3 - y_4 \end{pmatrix} \begin{pmatrix} t_a \\ t_b \end{pmatrix} = \begin{pmatrix} x_3 - x_1 \\ y_3 - y_1 \end{pmatrix}. \qquad (31)$$

If $t_a \in [0,1]$ and $t_b \in [0,1]$, we are in the case of segment intersections and hence triangle folding. In this case, we switch to another type of conformal parametrization (spectral or finite element) or recompute the conformal parametrization by choosing two different vertices to pin.

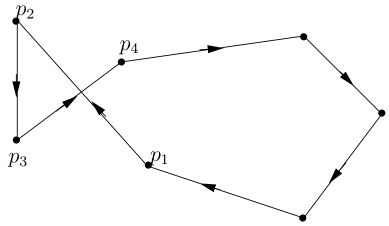


Fig. 12 Intersection boundary $\partial\mathcal{S}^0$ that is present in the case of triangle folding.

6 Surface remeshing examples

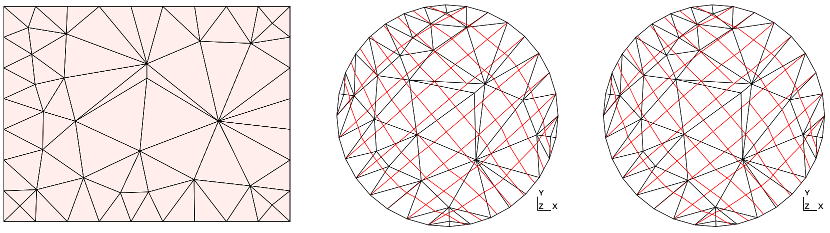
In this section, we discuss the optimal parametrization for different mesh generators and hence for several types of surface meshes: isotropic or anisotropic, triangular or quadrangular. Five different discrete parametrizations are considered (finite element harmonic, barycentric convex, mean value convex, finite element conformal and spectral conformal) as well as five different planar mesh generators available in the Gmsh software: a planar Delaunay algorithm [14] (herafter called "Delaunay"), a Frontal-Delaunay algorithm [33] (hereafter simply called "Frontal"), an algorithm based on local mesh adaptation (called "MeshAdapt", see [15] for more details), the Bidimensional Anisotropic Mesh Generator ("BAMG") developed by F. Hecht [17], and the new Delquad algorithm [35] coupled with the Blossom recombination algorithm [36] in order to produce quadrangular meshes (hereafter called "Delquad-Blossom").

As far as the conformal maps are concerned, we take as default the spectral conformal map which has less distortion as explained in Sec. 4 and Fig. 8 and switch to the finite element conformal map in the case we detect some overlap (see previous section). This will be discussed also later in the guidelines section 8.

We compare timings as well as mesh qualities for the new triangular or quadrilateral meshes. The computations are performed on a MacBook Pro 2.66GHz 4GB RAM Intel Core i7.

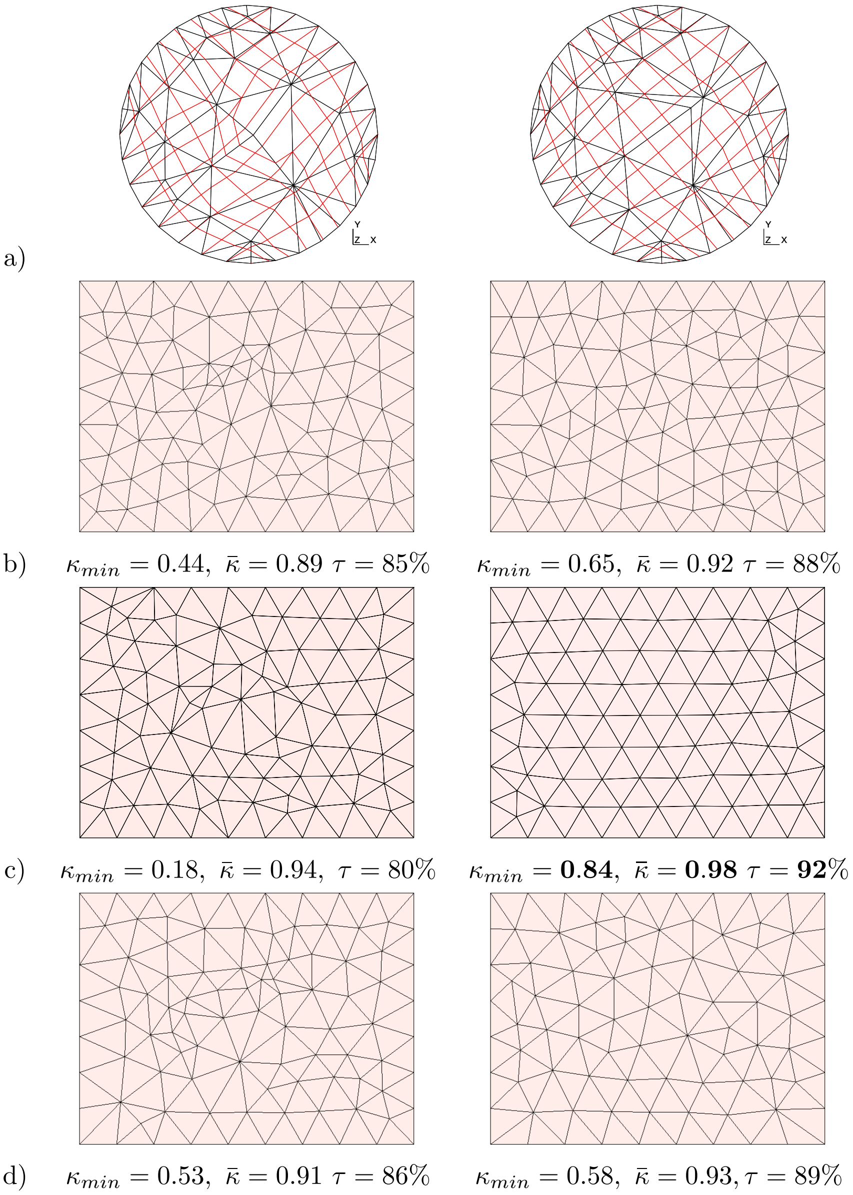
6.1 Isotropic surface meshes

In the first examples, we look at isotropic triangular meshes with a given mesh size field. We first compare the isotropic remeshing of a surface $\mathcal{S}$ of low quality (see Fig. 13) using three different mappings with boundaries mapped onto a unit disk: the finite element harmonic map and the two different convex combination maps (the barycentric and the mean value). The total time for the parametrization and the remeshing is less than $0.005s$ for all mappings. The final mesh contains about 180 triangles. First of all, we notice in Fig. 13 that there are no significant differences between the harmonic map and the mean value map. The mean value map is however better as it is mathematically guaranteed to be one-to-one (which is not the case of the discrete harmonic map in the presence of triangles of poor quality). We therefore advise in the context of surface remeshing to compute always a mean value convex map instead of a finite element harmonic map. Fig. 14 compares the mapping and the quality (aspect ratio κ and efficiency index τ) of the created meshes for the two convex mappings and for three mesh generators: Delaunay, Frontal and MeshAdapt. We can see from the iso-x and iso-y values represented in red in the parametric uv-plane (top Figs. 14) that the metric tensor $\mathbf{M}_\mathbf{u}$ (Eq. (2)) associated with the barycentric map $\mathbf{u}(\mathbf{x})$ is much more distorted than the one obtained with the mean value convex map. Looking at the remeshed surface, we see that the 2D planar meshers are not able to produce high quality meshes using the non-smooth metric of the barycentric map. In this example, we can see that the optimal remeshing is obtained by combining the mean value convex mapping with a frontal mesh generator. Even if we have presented a single small experiment with less than 200 mesh elements, this example is adequate as it is representative of the remeshing of a part of a bigger mesh using convex mappings combined with different surface meshers.



a) b) c)

Fig. 13 Initial triangulation $\mathcal{S}$ of poor quality (a) and mapping of this mesh onto a unit circle with the energy minimizing harmonic map (b) and the mean value convex map (c). The iso-x and iso-y values are shown in red in the parametric plane.



b) $\kappa_{min} = 0.44,\ \bar{\kappa} = 0.89\ \tau = 85\%$ $\kappa_{min} = 0.65,\ \bar{\kappa} = 0.92\ \tau = 88\%$

c) $\kappa_{min} = 0.18,\ \bar{\kappa} = 0.94,\ \tau = 80\%$ $\kappa_{min} = \mathbf{0.84},\ \bar{\kappa} = \mathbf{0.98}\ \tau = \mathbf{92}\%$

d) $\kappa_{min} = 0.53,\ \bar{\kappa} = 0.91\ \tau = 86\%$ $\kappa_{min} = 0.58,\ \bar{\kappa} = 0.93, \tau = 89\%$

Fig. 14 Poor quality initial triangulation $\mathcal{S}$ that has been remeshed using a barycentric convex map (left figures) and a mean value convex map (right figures) using different 2D planar meshers. Mapping of the initial mesh onto the unit disk with iso-x and iso-v values in red (a), final mesh obtained using a planar Delaunay mesher (b), Frontal mesher (c) and MeshAdapt mesher (d).

In the next examples, we compute the mean value and the conformal map for the isotropic remeshing of different STL triangulations of very low quality

($\bar{\kappa} < 0.7$ and $\kappa_{\min} < 0.05$): a tooth, an aorta and a pelvis[3]. When the topology is non-closed or when the geometrical aspect ratio of the triangulation is high, the initial mesh is automatically split by our algorithm into different mesh patches. The splitting is performed by combining a Metis mesh partitioning algorithm and our max-cut mesh partitioner based on a multiscale Laplacian map [26]. As can be seen from Fig. 15, the mapped meshes computed with the mean value map present much more distortion close to the boundaries. Moreover, the magnified views of the iso-u and iso-v values on one of the mesh patches (see Fig. 16) of the aorta shows that the mesh metric (2) is much smoother near the boundaries with the conformal map than with the mean value map. Indeed the mean value map introduces some distortion due to the fixed mapping onto a unit circle, while the conformal mapping is an open boundary mapping. We compare the quality of the new surface meshes using different meshing algorithms. The prescribed mesh size field is uniform for the tooth and function of the vessel radius, computed as the distance to the centerlines.

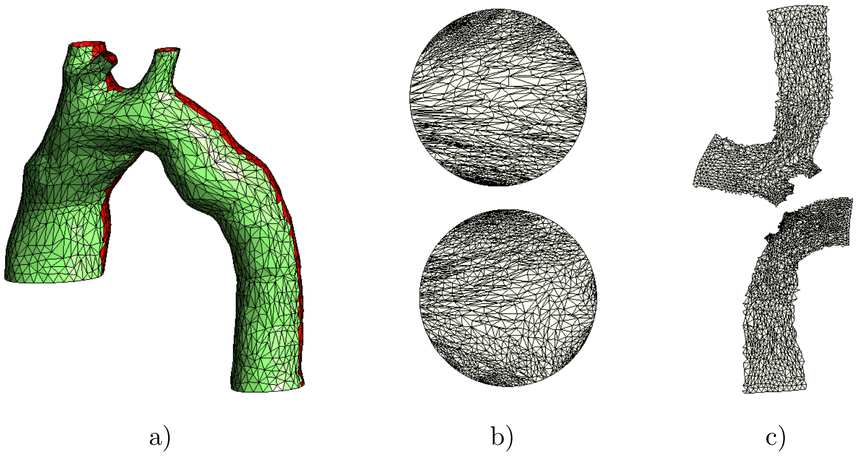


Fig. 15 Remeshing of an STL triangulation of a human aorta that has been split into two mesh patches (a). Mean value mapping (b) and conformal mapping (c) for those two patches.

Table 1 shows the quality of the meshes for the different parametrizations and planar mesh generators. The values in the table are mean values obtained for $n = 100$ runs. It is found that the mean aspect quality $\bar{\kappa}$ and the efficiency index τ have a very small standard deviation ($\sigma_{\bar{\kappa}} < 10^{-3}, \sigma_{\tau} < 0.5\%$), while the minimum aspect quality κ_{min} is found to have a larger standard deviation $\sigma_{\kappa_{min}} \approx 0.15$. Moreover, the quality measures $\bar{\kappa}$ and τ are found to be

[3] The STL triangulation of the aorta and the tooth can be downloaded from the INRIA database `http://www-roc.inria.fr/gamma/gamma.php` and the STL triangulation of the pelvis is presented in [28].

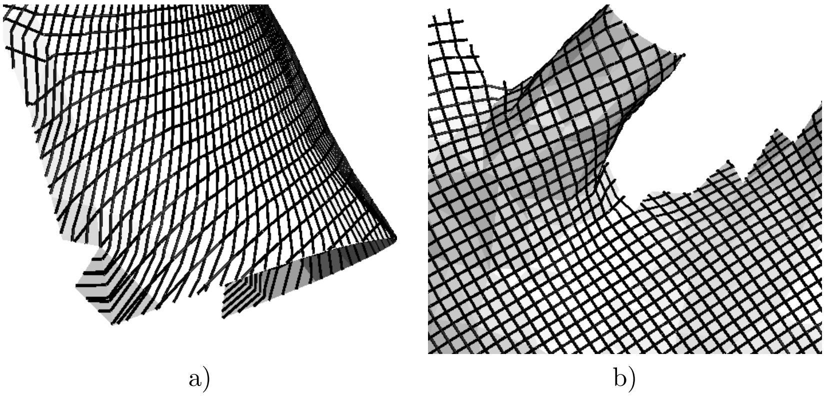


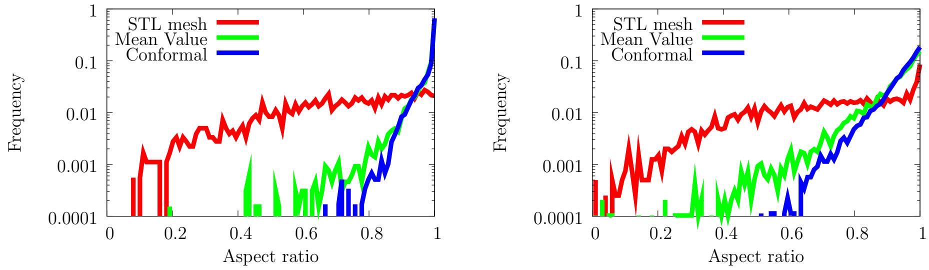
Fig. 16 Magnified views of the iso-u and iso-v values on one of the mesh patches of the aorta (see Fig. 15a) for the mean value map (a) and the conformal map (b).

independent of the prescribed mesh size field as well as from the quality of the initial triangulation. Fig. 17 shows the resulting shape quality histograms for both the tooth and the aorta using a Delaunay mesher.

Geometry/ 2D Mesher	Mean value map					Conformal map				
	κ_{min}	$\bar{\kappa}$	$\tau(\%)$	#	$t(s)$	κ_{min}	$\bar{\kappa}$	$\tau(\%)$	#	$t(s)$
Tooth										
Delaunay	0.08	0.94	86	7036	0.58	0.56	0.94	85	7458	0.67
Frontal	0.19	0.97	91	7022	0.58	**0.62**	**0.98**	**90**	6704	0.59
MeshAdapt	0.18	0.94	88	5748	1.12	0.60	0.94	85	5650	1.03
Aorta										
Delaunay	0.007	0.91	78	10617	1.9	0.61	0.94	86	10439	2.4
Frontal	0.01	0.95	82	9079	2.1	**0.63**	**0.98**	**91**	9573	2.1
MeshAdapt	0.02	0.92	85	9448	5.4	0.51	0.94	89	8523	3.0
Pelvis										
Delaunay	0.09	0.94	78	25960	9.3	0.45	0.94	81	26448	10.9
Frontal	0.01	0.97	82	25440	9.5	**0.48**	**0.98**	**88**	24448	10.42
MeshAdapt	0.03	0.94	88	18394	12.1	0.44	0.95	87	18044	12.3

Table 1 Quality of the surface meshes using different planar mesh generators in the parametric space, where the parametrization has been computed with the mean value and conformal mappings. The qualities we look at are the the minimum aspect ratio κ_{min}, the mean aspect ratio $\bar{\kappa}$ and the efficiency index τ. The number of mesh elements # and the CPU time $t\,(s)$ are also given.

It can be clearly seen from Table 1 that the best planar mesh generator is Gmsh's Frontal algorithm. This is not a surprise: frontal techniques tend to produce meshes that are aligned with principal directions. If the planar domain that has to be meshed is equipped with a smooth metric that conserves angles (i.e. when the mapping is conformal), then the angles between the principal directions are conserved. Frontal algorithms also tend to produce good



a) b)

Fig. 17 Remeshing of the tooth with a Frontal mesher (a) and the aorta with a Delaunay mesher (b). Quality histograms for the initial triangulation (STL), the remeshed surface using a conformal map and a mean value map.

meshes for mean value maps since this mapping mimics a harmonic map and that harmonic maps are almost conformal except close to the boundaries. We observe that the mean value mapping has few triangles (close to the boundaries) of low quality. This can be explained by a loss in conformality at the boundaries that gives rise to a less smoother mesh metric (see the u and v iso-values on the initial surface in Fig. 16).

In the next example, we compare timings for the conformal mapping with other remeshing packages presented in the literature. We consider the well-known Stanford bunny mesh model of $70k$ triangles[4] (see Fig. 18). The original mesh has 5 holes and is of zero genus. For the remeshed bunny of $25k$ triangles presented in Fig. 18, we have a minimum quality of $\kappa_{min} = 0.56$ and a mean quality is $\bar{\kappa} = 0.97$. Prior to computing the parametrization, two different mesh partitioners have been called: a multilevel mesh partitioner (Metis) and a max-cut mesh partitioner based on a multiscale harmonic map [26]. We compare in Table 2 some statistics and timings of our algorithm with the least square conformal map (LSCM) of Levy et al. [23], with the multiresolution remeshing of Eck et al. [8] and with the angle based parametrization (ABF) of Zayer [45]. We can see from Table 2 that our method is quite competitive in terms of computational time with the other methods presented in the literature. As shown in the previous examples, the timings for the other parametrizations is even faster than the timings for the conformal map.

Many other examples of isotropic surface remeshing can be found in references [34], [26] and [27].

6.2 Quadrangular meshes

In the next example, we look at producing high quality quadrilateral meshes. In a recent paper, we have developed an new indirect quadrilateral mesh gen-

[4] The model can be downloaded at the following web site:
`http://www.sonycsl.co.jp/person/nielsen/visualcomputing/programs/bunny-conformal.obj`.

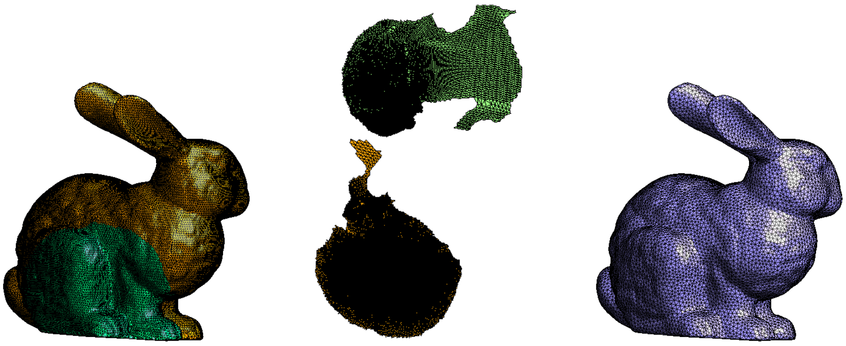


Fig. 18 Remeshing of the bunny mesh model of $70k$ triangles that has been split into 2 mesh partitions. Left figure shows the two partitions, middle figure shows the conformal harmonic parametrization that has been computed for both mesh partitions and right figure shows the remeshed bunny with about $25k$ triangles.

Remeshing	Number of partitions	Partition time (s)	Parametrization time (s)	Total remeshing time (s)
LSCM Levy [23] (1.3Ghz)	23	30	95	–
Eck [8] (1.3Ghz)	88	-	-	33.5
ABF++ Zayer [45]	2	-	13	-
LinABF Zayer [45]	2	-	2	-
Presented method (2.4Ghz)				
* harmonic partitioner	2	16.7	1.4	25
* multilevel partitioner	10	7	1.4	14

Table 2 Remeshing statistics and timings for the remeshing of the bunny mesh model of $70k$ triangles (new mesh of $25k$ triangles) with a conformal map. Comparison (when available) of the presented method with other techniques presented in the computer graphics community.

erator called Delquad-Blossom [35,36]. The approach uses distances in the L^∞ norm as a base for inserting new points and generate right triangles that are then recombined into quadrangles.

The combination of a conformal map with the Delquad-Blossom algorithm allows building meshes of very high quality while starting with a triangulation only. The fact that the conformal map preserves the angles allows us to define directions in the parametric plane that the mesher can follow, enabling generation of quad meshes with both prescribed size and direction. Figure 19 shows the quadrangular remeshing of a Falcon aircraft. Starting with an initial triangulation, triangles of model surfaces have been patched together to create 12 compounds of surfaces that have been parametrized (only half of them are visible in Fig. 19). The colors of the triangles indicate the different compounds of surfaces of the model.

Figure 20 shows the mesh that has been created using the conformal mapping. The final surface mesh is made of 53297 quadrangles and exhibits excellent statistics with $\eta_{min} = 0.26$ and $\bar{\eta} = 0.84$. The efficiency of the mesh is $\tau = 92\%$ which is again very good. The total time for the surface meshing

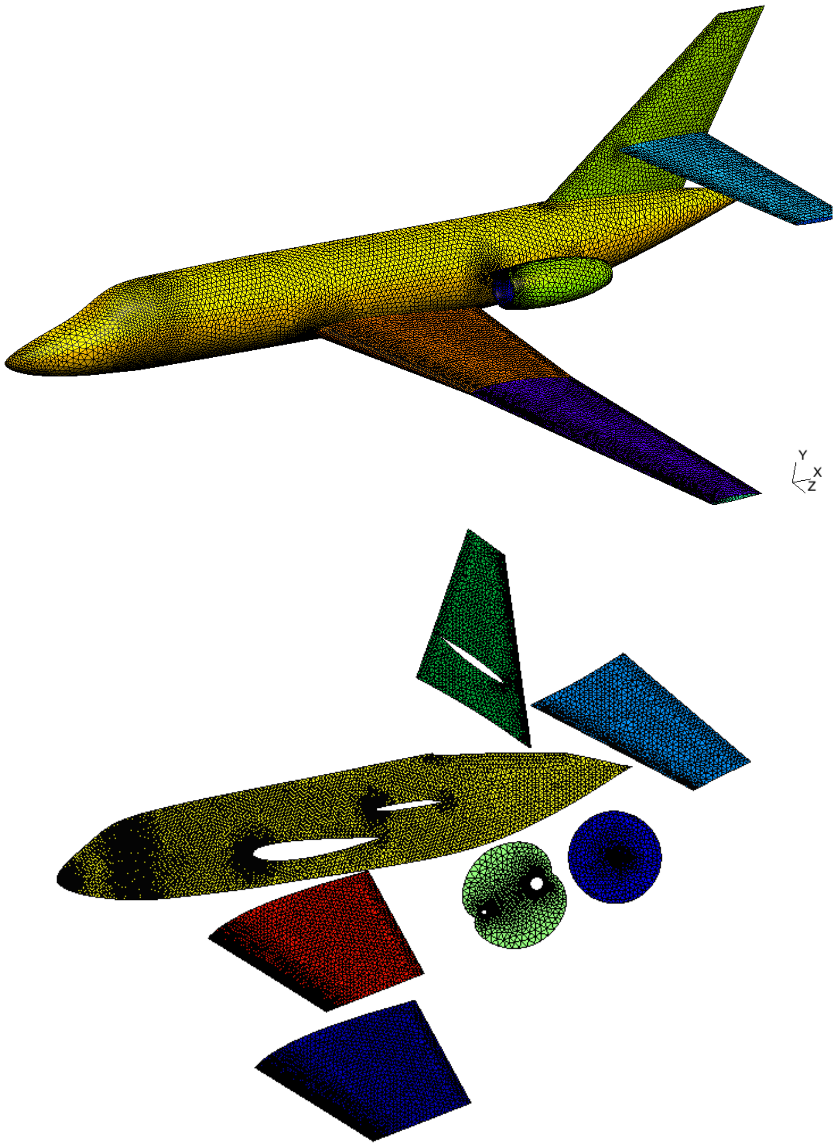


Fig. 19 Initial triangular mesh of half of the Falcon aircraft that has been split into 12 different colored mesh patches (only 7 of those patches are visible) (top) and spectral conformal parametrization of the 7 visible surfaces patches (bottom).

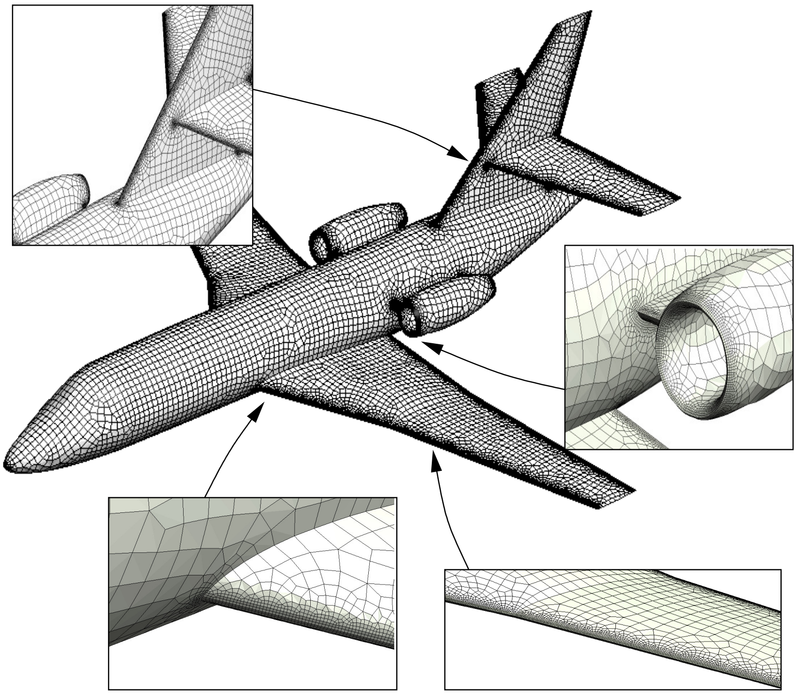


Fig. 20 Quad mesh of a falcon aircraft: $\eta_{min} = 0.17$, $\bar{\eta} = 0.86$, $\tau = 92\%$.

is 22 seconds. This time includes (i) the reparametrization of the 12 surfaces (3 sec.), (ii) the Delquad algorithm applied to the 12 surfaces (10 sec.), the Blossom recombination algorithm applied to the 12 surfaces (9 sec.).

Other examples of quadrangular surface remeshing using the conformal mapping can be found in references [35] and [36].

6.3 Anisotropic meshes

In the last example, we look at producing high quality anisotropic meshes using the Bidimensional Anisotropic Mesh Generator (BAMG) developed by F. Hecht [17]. We remesh the geometry of an arterial bypass with an anisotropic mesh size field based on the curvature. The discrete curvature is computed using the method of Rusinkiewicz [37]. Here again, the initial mesh $\mathcal{S}$ has been automatically split by our algorithm into two different mesh patches $\mathcal{S}_1$ and $\mathcal{S}_2$. Fig. 21 shows the resulting anisotropic mesh with the computed mesh qualities together with the iso-u and iso-v values of the parametrization using a conformal map with open boundaries and a mean value convex map onto a unit disk. We can see that away from the boundary the BAMG mesh generator is able to build an anisotropic mesh in agreement with the curvature-based metric. However, close to the boundaries BAMG fails (see red circles in Fig. 21 with the corresponding magnified views in Fig. 22) to compute a correct anisotropic mesh in the case of the mean value parametrization. This can be again explained by a highly distorted mesh metric $\mathbf{M_u}$ at those locations that is caused by the fixed boundaries (unit circle). As in the case of quadrilateral meshes, we therefore recommend using a conformal mapping for the purpose of anisotropic surface remeshing.

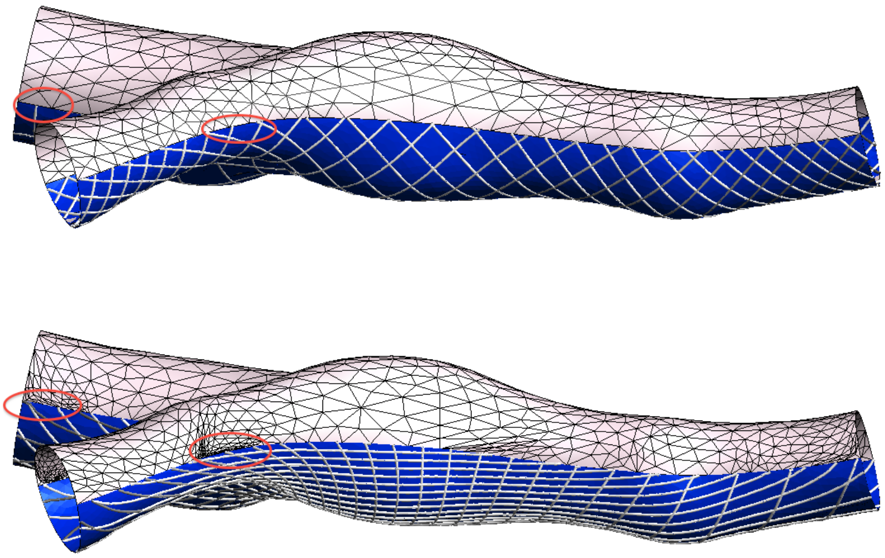


Fig. 21 Anisotropic curvature-based remeshing of an arterial bypass with a conformal map (top) and a mean value map onto a unit disk (bottom). We show the iso-u and iso-v values of the parametrizations on $\mathcal{S}_1$ together with the new anisotropic mesh of $\mathcal{S}_2$.

6.4 Non-bijective parametrizations: flipping and folding

For all the surface remeshing examples, we have checked if the computed parametrizations are bijective by looking at possible triangle flipping and folding in the parametric space.

It was observed that triangle flipping occurs very rarely and is directly related to the quality of the initial mesh. With the presented cavity check and boundary check algorithms, we were able to repair all the flipped triangles in order to recover a bijective parametrization.

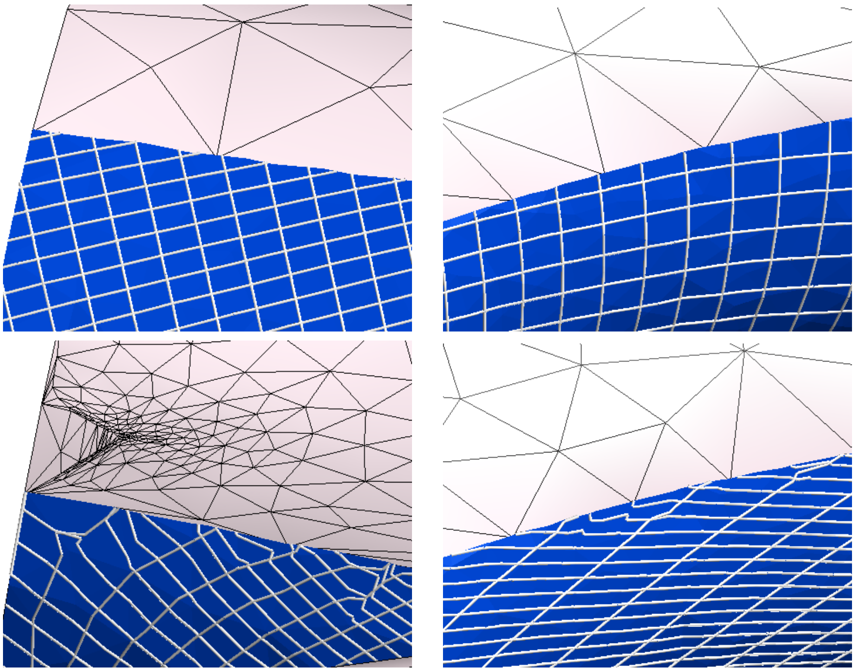


Fig. 22 Magnified views of the two portions of Fig. 21 marked with red circles. The top figures correspond to the conformal map and the bottom to the mean value map onto a unit disk. We show the iso-u and iso-v values of the parametrizations on $\mathcal{S}_1$ together with the new anisotropic mesh of $\mathcal{S}_2$.

Triangle folding was found to occur also very rarely when computing spectral conformal maps. The occurrence of folding is directly related to the shape of the mesh patch to be parametrized. Indeed, the more the ratio of the moment of inertia of the mesh patch about the u-axis and the v-axis, the more likely the parametrization is about to fold with the spectral conformal map. In the cases in which such a folding occurs, we switch to the finite element conformal map by choosing carefully the two mesh vertices to be pinned as a function of the moment of inertia of the mesh patch. For all the presented mesh experiments, we were able to compute a bijective conformal parametrization (either spectral or finite element).

7 Conclusion

In this paper, five different discrete parametrizations have been reviewed: the finite element harmonic map, the barycentric convex map, the mean value convex map, the least square finite element conformal map and the spectral conformal map. The implementation of those mappings as well as the boundary conditions have been presented in a comprehensive and unified manner and the issues of triangle flipping and folding that may arise with discrete linear mappings have been discussed. Different remeshing examples have been presented in order to find the optimal parametrization for the generation of either isotropic or anisotropic triangular or quadrangular meshes.

We have showed that the finite element harmonic mapping should be replaced by the mean value convex mapping, which is a mapping that mimics the harmonic map and is moreover guaranteed to be one-to-one if the boundary is mapped onto a convex polygon. It was also found that the spectral conformal map should be preferred to the finite element conformal map as it introduces less distortion.

We showed that among the different mesh generators, the Frontal algorithm performs best for the generation of high quality triangular meshes. Finally we have shown that in the context of anisotropic meshes (quadrangular or triangular) meshes, the optimal parametrization is the conformal mapping. This mapping has two nice advantages for the planar meshers: (1) it preserves the angles, (2) it is a boundary-free parametrization, so that the resulting mesh metric $\mathbf{M_u}$ is smoother close to the boundaries. We have showed for example that the BAMG mesh generator cannot deal well with locally non-smooth mesh metric close to the boundaries.

The timings show that the presented remeshing techniques based on discrete linear parametrizations are efficient and that there exist an optimal parametrization that renders the highest quality meshes.

8 Guidelines for optimal parametrization for surface remeshing

This section aims at presenting explicit guidelines that a user can follow in choosing the best parametrization scheme for a specific topology, geometry, and characteristics of the target output mesh. Those guidelines are based on the results of this paper and on the conclusions we have just drawn.

- The topology of the mesh should be non-closed, i.e of zero genus with at least one boundary. If this is not the case, the mesh should be partitioned into different non-closed mesh patches prior to computing parametrizations. Care should also be taken for non-closed topologies of high geometrical aspect ratio. This is discussed more in details in [26].
- If the non-closed mesh patches contains several hole (N boundaries > 1), then the holes should be filled with mesh elements in order to have a smooth parametrization across the holes. Indeed, the resulting mesh metric $\mathbf{M_u}$ will be smoother so that the planar mesher will be able to generate higher quality meshes. In order to close the holes, we have taken the approach to compute the mean plane of every hole boundary and to use a planar Delaunay mesher to fill the mean plane of the hole with triangular mesh elements. Those new elements are added for the assembling of the system of equations (22).
- The best quality triangular and quadrangular meshes are obtained when combining the spectral conformal mapping with either a Frontal mesher

for isotropic triangles, the Delquad mesher for quadrangular meshes and an anisotropic mesher such as Bamg for anisotropic triangles. Hence the spectral conformal map should be the method of choice for all target output meshes.

- As the conformal map is not guaranteed to be bijective (flipping and folding might occur), we advise to check the presence of flipping and folding and to try to repair locally the parametrization using the presented cavity check and boundary check algorithms. If, however, the repair does not succeed, we advice to switch to the finite element conformal map.
- When switching to a finite element conformal map, if a bijective parametrization cannot be computed (this occurs only in very few cases in which some elements of the input meshes have a very low mesh quality or if the boundaries of the mesh patch are very wiggly), we advise to switch to the mean value convex map which is guaranteed to be one-to-one.

The flowchart in Fig. 23 represents the remeshing algorithm using parametrizations that is implemented in the open-source mesh generator Gmsh [15] and that is based on the presented guidelines. Examples of how to use the remeshing with parametrizations can be found on the Gmsh wiki: `https://geuz.org/trac/gmsh`[5].

References

1. P. Alliez, M. Meyer, and M. Desbrun. Interactive geometry remeshing. *Computer graphics (Proceedings of the SIGGRAPH 02)*, pages 347–354, 2002.
2. M. Batdorf, L.A. Freitag, and C. Ollivier-Gooch. A computational study of the effect of unstructured mesh quality on solution efficiency. In *Proc. 13th AIAA Computational Fluid Dynamics Conf.*, 1997.
3. E. Bechet, J.-C. Cuilliere, and F. Trochu. Generation of a finite element mesh from stereolithography (stl) files. *Computer-Aided Design*, 34(1):1–17, 2002.
4. M. Ben-Chen, C. Gotsman, and G. Bunin. Conformal flattening by curvature prescription and metric scaling. *Computer Graphics Forum*, 27(2), 2008.
5. C. Bennis, J-M Vézien, and G. Iglésias. Piecewise surface flattening for non-distorted texture mapping. *ACM SIGGRAPH Computer Graphics*, pages 237 – 246, 1991.
6. H Borouchaki, P. Laug, and P.L. George. Parametric surface meshing using a combined advancing-front generalized delaunay approach. *International Journal for Numerical Methods in Engineering*, 49:223–259, 2000.
7. C. Choquet. Sur un type de représentation analytique généralisant la représentation conforme et défininie au moyen de fonctions harmoniques. *Bull. Sci. Math*, 69(156-165), 1945.
8. Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 173–182, 1995.
9. M. S. Floater. Parametrization and smooth approximation of surface triangulations. *Computer aided geometric design*, 14(231-250), 1997.
10. M. S. Floater. Parametric tilings and scattered data approximation. *International Journal of Shape Modeling*, 4:165–182, 1998.
11. M. S. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20(1):19–37, 2003.

[5] Access the wiki with username *gmsh* and password *gmsh*.

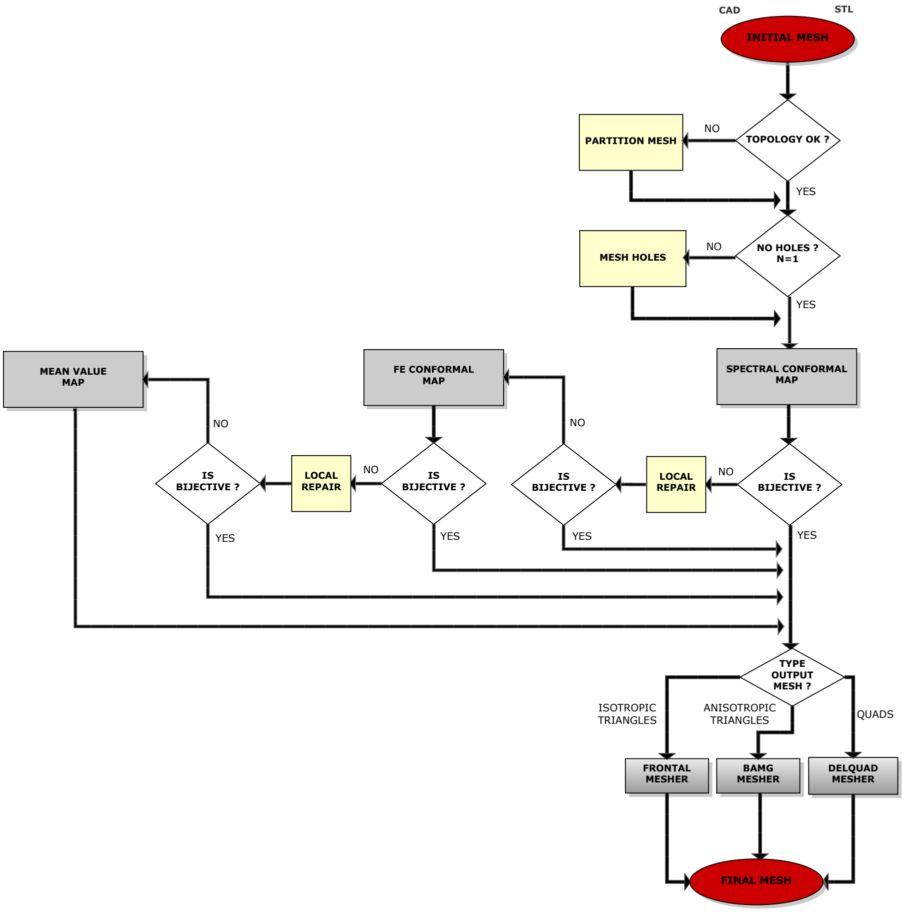


Fig. 23 Flowchart that represents the remeshing algorithm using parametrizations that is implemented in the open-source mesh generator Gmsh [15] and that is based on the presented guidelines.

12. M. S. Floater. One-to-one piecewise linear mappings over triangulations. *Math. Comp*, 72(685-696), 2003.
13. M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. *Advances in Multiresolution for Geometric Modelling*, 2005.
14. P.-L. George and P. Frey. *Mesh Generation.* Hermes, 2000.
15. C. Geuzaine and J.-F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.
16. G. Greiner and K. Hormann. Interpolating and approximating scattered 3d data with hierarchical tensor product splines. In *Surface Fitting and Multiresolution Methods*, pages 163–172, 1996.
17. F. Hecht. Bamg: Bidimensional anisotropic mesh generator, 2006. `http://www.freefem.org/ff++`.
18. V. Hernandez, J.E. Roman, and V. Vidal. Slepc: A scalable and flexible toolkit for the solution of eigenvalue problems. *ACM Trans. Math. Soft.*, 31(3):351–362, 2005.

19. K. HORMANN, , and G. 2000 GREINER. Mips: An efficient global. parametrization method. In *Curve and Surface Design*. Vanderbilt University Press, 2000.
20. Y. Ito and K Nakahashi. Direct surface triangulation using stereolithography data. *AIAA Journal*, 40(3):490–496, 2002.
21. P. Laug and H. Boruchaki. Interpolating and meshing 3d surface grids. *International Journal for Numerical Methods in Engineering*, 58:209–225, 2003.
22. R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK Users Guide: Solution of Large Scale Eigenvalue Problems by Implicitly Restarted Arnoldi Methods.* Society for Industrial and Applied Mathematics, 1997.
23. B. Levy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. In *Computer Graphics (Proceedings of SIGGRAPH 02)*, pages 362 – 371, 2002.
24. M. Spagnuolo M. Attene, B. Falcidieno and G. Wyvill. A mapping-independent primitive for the triangulation of parametric surfaces. *Graphical Models*, 65(260-273), 2003.
25. J. Maillot, H. Yahia, and A. Verroust. Interactive texture mapping. In *Proceedings of ACM SIGGRAPH'93*, pages 27–34, 1993.
26. E. Marchandise, C. Carton de Wiart, W.G. Vos, C. Geuzaine, and J-F. Remacle. High quality surface remeshing using harmonic maps. Part II: Surfaces with high genus and of large aspect ratio. *International Journal for Numerical Methods in Engineering*, 86(11):1303–1321, June 2011.
27. E. Marchandise, P. Crosetto, C. Geuzaine, J.F. Remacle, and E. Sauvage Quality open source mesh generation for cardiovascular flow simulations. *Chapter of of modelling physiological flows. Springer Series on Modeling, Simulation and Applications*,Berlin Heidelberg , 2011.
28. E. Marchandise, G. Compère, M. Willemet, G. Bricteux, C. Geuzaine, and J.-F. Remacle. Quality meshing based on stl triangulations for biomedical simulations. *International Journal for Numerical Methods in Biomedical Engineering*, 83:876–889, 2010.
29. David L. Marcum. Efficient generation of high-quality unstructured surface and volume grids. *Engrg. Comput.*, 17:211–233, 2001.
30. David L. Marcum and Adam Gaither. Unstructured surface grid generation using global mapping and physical space approximation. In *Proceedings, 8th International Meshing Roundtable*, pages 397–406, 1999.
31. P. Mullen, Y. Tong, P. Alliez, and M. Desbrun. Spectral conformal parameterization. In *In ACM/EG Symposium of Geometry Processing*, 2008.
32. T. Rado. Aufgabe 41. *Math-Verien*, page 49, 1926.
33. S. Rebay. Efficient unstructured mesh generation by means of delaunay triangulation and bowyer-watson algorithm. *Journal of Computational Physics*, 106:25–138, 1993.
34. J.-F. Remacle, C. Geuzaine, G. Compère, and E. Marchandise. High quality surface meshing using harmonic maps. *International Journal for Numerical Methods in Engineering*, 83:403–425, 2010.
35. J-F. Remacle, F. Henrotte, T. Carrier-Baudouin, E. Bechet, C. Geuzaine, and T. Mouton. A frontal delaunay quad mesh generator using the l_∞ norm. *International Journal for Numerical Methods in Engineering*, 2011.
36. J-F. Remacle, J. Lambrechts, B. Seny, E. Marchandise, A. Johnen, and C. Geuzaine Blossom-Quad: a non-uniform quadrilateral mesh generator using a minimum cost perfect matching algorithm. *International Journal for Numerical Methods in Engineering*, 89:1102–119, 2011.
37. S. Rusinkiewicz. Estimating curvatures and their derivatives on triangle meshes. In *Symposium on 3D Data Processing, Visualization, and Transmission*, September 2004.
38. A. Sheffer and E. de Sturler. Parameterization of faceted surfaces for meshing using angle-based flattening. *Engineering with Computers*, 17(3):1435–5663, 2001.
39. A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, 2006.
40. Alla Sheffer, Bruno Lévy, Inria Lorraine, Maxim Mogilnitsky, and Er Bogomyakov. Abf++: fast and robust angle based flattening. *ACM Transactions on Graphics*, 24(311-330), 2005.
41. D. Szczerba, R. McGregor, and G. Szekely. High quality surface mesh generation for multi-physics bio-medical simulations. In *Computational Science – ICCS 2007*, volume 4487, pages 906–913. Springer Berlin, 2007.

42. J.R. Tristano, S.J. Owen, and S.A. Canann. Advancing front surface mesh generation in parametric space using riemannian surface definition. In *Proceedings of 7th International Meshing Roundtable*, pages 429–455. Sandia National Laboratory, 1998.
43. W.T Tutte. How to draw a graph. In *Proceedings of the London Mathematical Society*, volume 13, pages 743–768, 1963.
44. D. Wang, O. Hassan, K. Morgan, and N. Weatheril. Enhanced remeshing from stl files with applications to surface grid generation. *Commun. Numer. Meth. Engng*, 23:227–239, 2007.
45. Rhaleb Zayer, Bruno Lévy, and Hans-Peter Seidel. Linear angle based parameterization. In *ACM/EG Symposium on Geometry Processing conference proceedings*, 2007.
46. Y. Zheng, N.P. Weatherill, and O. Hassan. Topology abstraction of surface models for three-dimensional grid generation. *Engrg. Comput.*, 17(28-38), 2001.
47. Gil Zigelman, Ron Kimmel, and Nahum Kiryati. Texture mapping using surface flattening via multi-dimensional scaling. *IEEE Trans. on Visualisation and Computer Graphics*, 8(198-207), 2002.
48. P. Frey and PL. George Mesh Generation - Application To Finite Elements. Wiley, 2008.