

# A simple and flexible mesh parameterization method

Colin Cartade<sup>1</sup>, Rémy Malgouyres<sup>1</sup>, Christian Mercat<sup>3</sup>, and Chafik Samir<sup>2</sup>

<sup>1</sup> LIMOS, Université d’Auvergne, Complexe des Cézeaux,  
63173 Aubière, France

`{colin.cartade,remy.malgouyres}@u-clermont1.fr`

<sup>2</sup> IUFM, Université Claude Bernard Lyon 1

43 boulevard du 11 novembre 1918, 69622 Villeurbanne

`christian.mercat@gmail.com`

<sup>3</sup> ISIT, Université d’Auvergne, Complexe des Cézeaux,  
63173 Aubière, France

`chafik.samir@u-clermont1.fr`

**Abstract.** We describe a method to compute conformal parameterizations with a natural boundary based on a simple differentiable expression measuring angles between edges by using complex numbers. The method can be adapted to preserve metric properties or map textures with constrained positions. Some illustrations are shown to assess the efficiency of the algorithms.

**Keywords:** parameterization, conformal, constrained texture mapping, natural boundary, minimization.

## Introduction

Parameterizations of discrete surfaces are one to one maps from a triangulated surface in 3D to the plane. They are widely used in computer graphics because they allow one to simplify difficult 3D problems in easy 2D tasks. For instance, texture mapping, a very classical such application, boils down to the trivial task of mapping an image on a rectangular domain. They also allow to consider a mesh as the graph of a function from the plane to the 3D space. Such a representation is useful for applications such as morphing, surface fitting, etc.

Many methods have been introduced to compute conformal parameterizations. These parameterizations, by definition, should preserve angles and thus the local aspect of the mesh. Among them, barycentric methods [2, 7] fix the boundary of the parameterization on a convex shape and then solve a linear system to find the parameters for the others vertices. These algorithms are very fast but introduce a big length distortion near the boundary. Other methods have then been introduced to find parameterizations with a more natural boundary. However, attempts such as the so called *intrinsic parameterization* [1] or *LSCM* method [4] use boundary conditions that are not very natural, leading sometimes to strange unsymmetrical results. The *ABF* method [8–10] gives better results in this respect. But this method is not easy to adapt when using other constraints such as lengths, areas, positions of specific vertices, etc.

In this paper, we introduce a differentiable expression to compute both the angle and the length ratio between two edges. On the one hand, we use it to define a conformal energy measuring angle distortion of a parameterization in a similar way to ABF method. And as this energy is expressed in terms of the position of the vertices of the parameterization and not the angles, it can be adapted to also preserve metric properties or positional constraints. On the other hand, we use it to define a boundary energy to find parameterizations with natural boundary.

The rest of the paper is organized as follows. In Section 1, we define the angular formula and the resulting conformal energy. In Section 2, we detail the minimization algorithm. Section 3 introduces boundary conditions to guarantee convergence of the algorithm and to determine natural boundary. Examples of area preservation and parameterization with constraints are given in the last section.

## 1 Minimizing angle distortion

### 1.1 Main problem

Our goal is to find a parameterization of a triangular or quadrangular mesh, that is to say a flattened version of the mesh. A good parameterization should be as close as possible to the initial mesh, and in particular its faces should look like those of the mesh.

By definition, conformal maps are the applications preserving angles and in consequence they also preserve lengths ratios locally. Therefore, for faces which are small with respect to the whole mesh, faces in a conformal parameterization have almost the same shape as those of the initial mesh. Therefore we look at a *good* flattened mesh as a discretization of a conformal map.

Although continuous conformal maps exist, in general, we cannot find a parameterization whose (linear) faces have exactly the same angles as those of the three dimensional mesh. So a natural approach is to look for the parameterization minimizing the angles distortion. It is used by ABF method which computes the parameterization minimizing the energy

$$\sum (\alpha_i - \beta_i)^2,$$

where the sum is over all the angles  $\alpha_i$  of the mesh and the  $\beta_i$  are the corresponding angles in the parameterization. The unknowns of ABF method are not the vertices of the parameterization but the angles  $\beta_i$ . Consequently the method has three main drawbacks.

- They have to reconstruct the parameterization from the angles which is not straightforward.
- Not all the angles configurations can be obtained. For example, for triangles faces the sum of the angles must be  $\pi$ . Thus, it is a constrained minimization problem which is solved with Lagrange multipliers technique.
- Since the variables are the angles, we cannot add other energy terms, for example an energy measuring the area distortion.

## 1.2 A differentiable measure of angle distortion

In the sequel we will denote by  $z_i$  the vertices of the mesh and by  $z'_i$  the corresponding two dimensional parameters. We will also consider the  $z'_i$  as complex numbers.

Given a triangular face  $(z_i, z_j, z_k)$ , we want a formula giving the angle in  $z_i$  that we can easily differentiate. Obviously we cannot use a formula such as

$$\arccos\left(\frac{(z_k - z_i) \cdot (z_j - z_i)}{\|z_j - z_i\| \|z_k - z_i\|}\right)$$

for this reason.

However, if we imagine the three vertices as complex numbers in the plane they define, the coefficient

$$\rho = \frac{z_k - z_i}{z_j - z_i} \in \mathbb{C}$$

measures both the angle and the length ratio between the edges  $[z_i, z_j]$  and  $[z_i, z_k]$ . Thus, the nonnegative number  $|z'_k - z'_i - \rho(z'_j - z'_i)|^2$  is a measure of the conformal distortion for the face  $(z_i, z_j, z_k)$  from the vertex  $z_i$ . Moreover, still using complex numbers, we have nice formulas for the derivatives. Indeed, if  $z_i = x_i + i y_i$ ,

$$\begin{aligned} \frac{\partial}{\partial x'_i} &= 2 \operatorname{Re}\left((-1 + \bar{\rho})(z'_k - z'_i - \rho(z'_j - z'_i))\right), \\ \frac{\partial}{\partial y'_i} &= 2 \operatorname{Im}\left((-1 + \bar{\rho})(z'_k - z'_i - \rho(z'_j - z'_i))\right). \end{aligned}$$

For every face, we only need to define a coefficient  $\rho$  on one vertex, to preserve all the angles of the face. Finally we want to minimize the energy

$$H = \sum |z'_k - z'_i - \rho_f(z'_j - z'_i)|^2$$

where the sum is over all the faces  $f = (z_i, z_j, z_k)$  of the mesh.

For quadrangular meshes, we could convert each face to two triangles and thus construct a triangular mesh but it increases the number of faces. We rather define the  $\rho$  coefficients as the ratios of the diagonals. More precisely, for a face  $(z_i, z_j, z_k, z_l)$ , we define

$$\rho = \frac{z_l - z_j}{z_k - z_i},$$

and we minimize the function

$$H = \sum |z'_l - z'_j - \rho_f(z'_k - z'_i)|^2.$$

This definition has the advantage of being more symmetric than the previous one, we do not have to choose the vertex in which we compute  $\rho$ . Besides, it leads to discrete conformal maps in the sense of [5], where many theorems and in particular convergence theorems are given.

## 2 Minimization algorithm

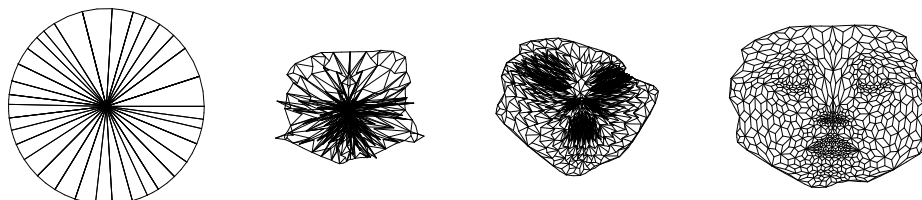
We cannot obtain satisfying results by minimizing the energy term  $H$  as such. Nevertheless, we believe it is important to develop the minimization algorithm first. In particular it allows to understand why we introduce new cost functions in Section 3 and why we choose these terms.

### 2.1 The algorithm

We achieve the minimization using a BFGS algorithm, a very efficient quasi-Newton method described in [6]. In particular, it computes an approximation of the second derivatives of the function from the exact gradient. So we only need to compute the first derivatives.

*Remark 1.* The BFGS method assumes a quadratic behavior around the critical point of the function and thus cannot be used to find the critical point of the ABF Lagrangian.

In practice we start from an initial parameterization whose boundary points are on the unit circle and whose interior points are in  $(0, 0)$ . And we would like the algorithm to unfold the interior points. An example is shown in Figure 1.



**Fig. 1.** Four steps of the minimization algorithm, example of good convergence.

### 2.2 Necessity of additional constraints

The global minimum of  $H$  is 0 and is reached when the  $z'_i$  are solutions of the complex linear system :

for all faces  $f = (z_i, z_j, z_k)$

$$z'_k - z'_i - \rho_f(z'_j - z'_i) = 0.$$

The number of equations of this system is the number of faces of the mesh whereas its number of unknowns is the number of vertices of the mesh. As a mesh has always more vertices than faces, the system has an infinity of solutions. But many of them are not valid parameterizations.

In fact, if we minimize  $H$  as such, the boundary points, initially on the unit circle, pull the interior points to the outside whereas the interior points attract the boundary points to origin. As there are much more interior points than boundary ones, if we do not add any energy, the minimization of  $H$  leads to the null parameterization.

### 3 Stabilizing the boundary

#### 3.1 Main idea

The main idea of this section is to add some energy terms to prevent the boundary from being too close to the origin. Constraints should not introduce new distortions. A drastic solution is to fix the boundary points uniformly on a circle. It leads to a very fast and stable minimization process. As all parameterizations methods fixing the boundary, the resulting texture mapping is, in general, unnatural around the boundary. It is particularly true when the boundary of the mesh is very different from fixed boundary of the parameterization.

#### 3.2 Preserving metric boundary

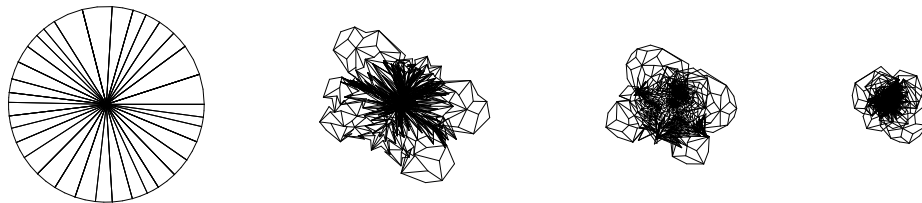
Our first motivation is to add an energy to preserve edges lengths around the boundary. Therefore we introduce the following metric energy

$$L = \sum (|z'_i - z'_j|^2 - \|z_i - z_j\|^2)$$

and we propose to minimize an average  $E$  of the metric energy and the conformal one

$$E = \alpha H + \beta L,$$

where the coefficients  $\alpha$  and  $\beta$  are positive real numbers.



**Fig. 2.** Four steps of the minimization algorithm, example of bad convergence.

It stabilizes the process for meshes with few vertices and it allows convergence toward a good parameterization. When we increase the number of points, the algorithm can still reach an undesirable local minimum. Thus, the boundary points can be attracted by the origin and wind around it in order to respect boundary edges lengths. An exemple is shown on Figure 2. In this case, using a high value of  $\beta$  compared to  $\alpha$  is no more better.

### 3.3 Preserving boundary angles

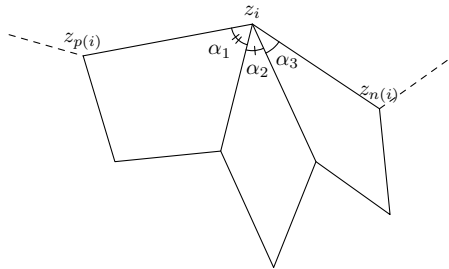
To prevent the winding around behavior of the algorithm, we propose to add an energy term to preserve angles between border edges. And similarly to conformal energy, in order to obtain a differentiable energy, we introduce complex coefficients measuring both angles and lengths ratios.

More precisely, for each boundary vertex  $z_i$ , we denote by  $p(i)$  and  $n(i)$  the indices of the previous and next vertex along the boundary. We want to associate to the vertex  $z_i$  a complex number  $\rho$  representing the angle between the edges  $[z_{p(i)}, z_i]$  and  $[z_i, z_{n(i)}]$  and their ratio of lengths. We define  $\rho$  as  $re^{i\theta}$  where the modulus is

$$r = \frac{\|z_{n(i)} - z_i\|}{\|z_{p(i)} - z_i\|}$$

and the argument  $\theta$  is the sum of the angles in  $z_i$ . In case of Figure 3,

$$\theta = \alpha_1 + \alpha_2 + \alpha_3.$$



**Fig. 3.** An example of definition of  $\rho = re^{i\theta}$  where  $\theta = \alpha_1 + \alpha_2 + \alpha_3$ .

Then we introduce the energy

$$B = \sum |z'_{n(i)} - z'_i - \rho_i(z'_i - z'_{p(i)})|^2$$

and minimize

$$E = \alpha H + \beta L + \gamma B$$

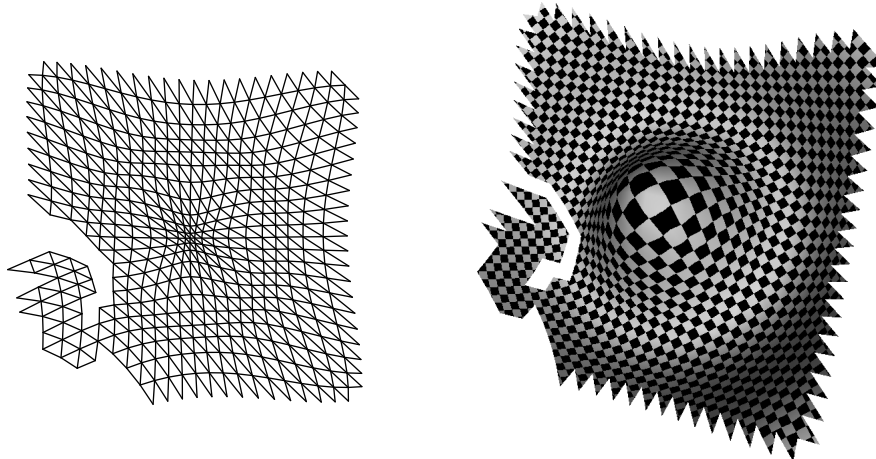
for convenient positive real number  $\alpha$ ,  $\beta$  and  $\gamma$ .

In fact, the energy  $B$  is a little redundant with the conformal one. The conformal energy intends to preserve all the angles and in particular the boundary ones. However it is important to strengthen the condition on boundary angles to keep the boundary points from being quickly attracted by the interior points. Thus, at each step of the minimization the interior points remains inside the area delimited by the boundary points allowing convergence toward the right local minimum. Shown in Figure 1 are four intermediate iterations of the algorithm to illustrate the successful convergence.

*Remark 2.* We could believe it is more natural to define directly  $\theta$  as the angle between the two edges. But it boils down to choosing an orientation of the plane  $(z_{p(i)}, z_i, z_{n(i)})$ . Although the mesh is oriented it seems difficult to make a consistent choice.

The resulting algorithm is quite stable in practice. We made many experiments with different meshes and various boundaries, and for most of them, we verify convergence toward the desired parameterization. Moreover the texture mapping are always good near the boundary even with very distorted boundaries.

An example is shown in Figure 4: on the left we represent the parameterization, and on the right, we map a checkerboard on the mesh.



**Fig. 4.** Parameterization of a triangular mesh with  $E = H + B + L$ .

### 3.4 Finding a natural boundary

The energies  $B$  and  $L$  of the previous section can also be used to find a natural boundary. Indeed we propose the following three steps algorithm.

1. First we minimize the energy

$$L + B$$

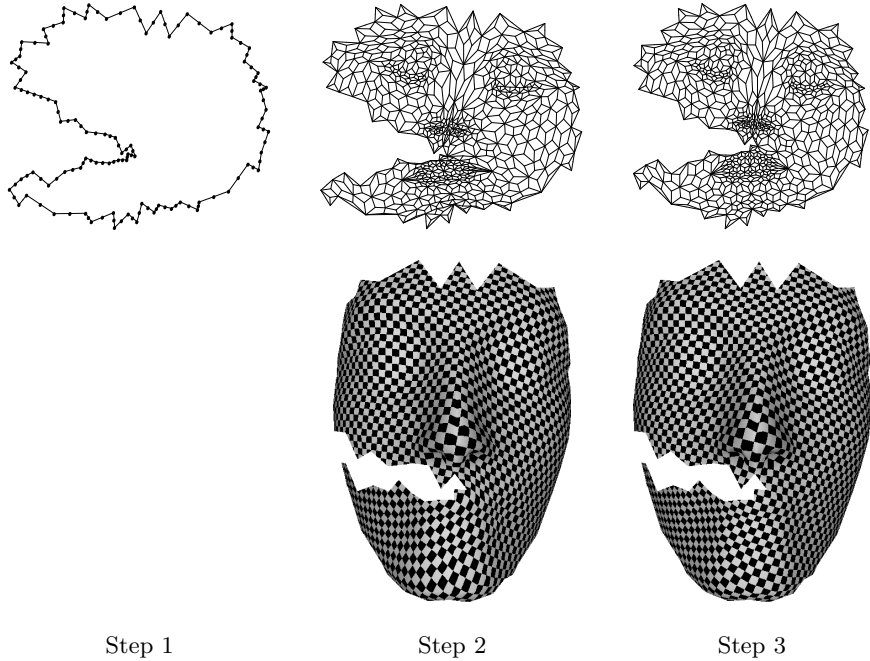
to obtain a boundary with almost the same edges lengths and angles between the edges as the initial boundary of the mesh.

2. Then we fix this boundary and minimize  $H$  (the boundary does not move during the algorithm).
3. Finally we use this minimum as initial condition to minimize

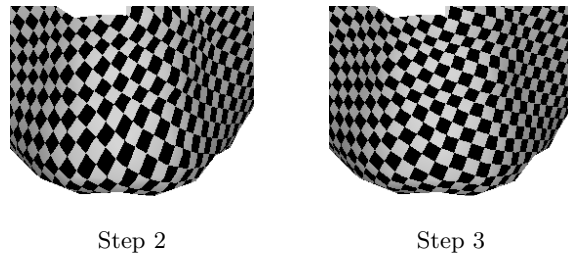
$$\alpha H + \beta L + \gamma B.$$

Although this algorithm needs three minimization steps, it is in general faster than the one of the previous section. Step 1 is very fast because the function to minimize has few variables and step 2 too because the boundary is fixed. Finally step 3 is fast because the algorithm start close to the minimum.

An example is shown on Figure 5. Although, the boundary is fixed during step 2, we can see that the texture mapping is quite good. Besides, if we need a fast algorithm to parameterize a mesh with a large number of points and a simple boundary it can be sufficient. The result is better after step 3. The boundary is more natural and the parameterization more conformal. The difference is visible near the chin of the mask. To make it clear, Figure 6 shows a zoom in on interesting regions of step 2 and step 3 of Figure 5.



**Fig. 5.** Parameterizations and texture mappings after each step of the method.



**Fig. 6.** Zooms in on texture mappings at steps 2 and 3 respectively in Figure 5.



## 4 Introduction of new constraints

We now further study the advantage of minimizing a conformal energy that expresses in term of points instead of angles. It allows one to add quite easily additional constraints according to the applications.

### 4.1 Adding energies : areas, lengths preservation

Even if the main feature of parameterization techniques is to preserve angles, we could also want to preserve other geometrical properties. In particular, on Figure 4, we would like to reduce the strong metric distortion in regions with high curvature.

In general it is not possible to preserve both angles and lengths. An alternate solution is to relax conformal constraints by adding a new metric energy. Thus, we only obtain a quasi-conformal parameterization but it leads to a better texture mapping.

In case of triangular meshes, to preserve areas we could introduce the energy

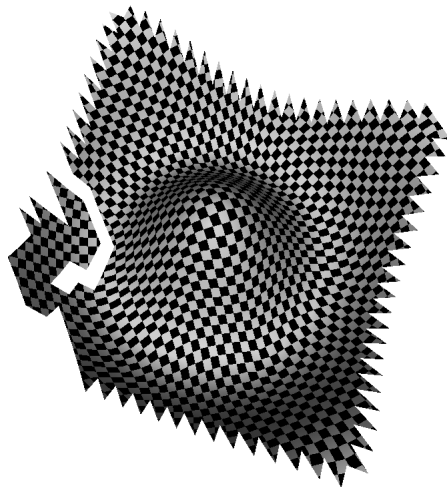
$$A = \sum \left( \operatorname{Im}(z'_k - z'_i) \overline{(z'_j - z'_i)} - \|(z_k - z_i) \wedge (z_j - z_i)\|^2 \right)^2$$

where the sum is over all the faces  $(z_i, z_j, z_k)$ .

Then, we minimize an average of the form

$$E = \alpha H + \beta A + \gamma L + \delta B$$

The texture mapping of Figure 7 is obtained using such an energy.

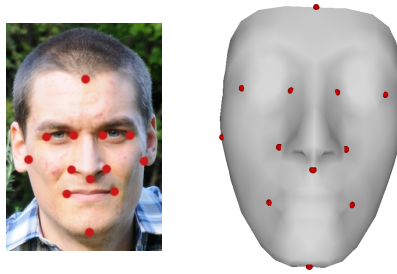


**Fig. 7.** Parameterization of a triangular mesh with  $E = H + A + B$ .

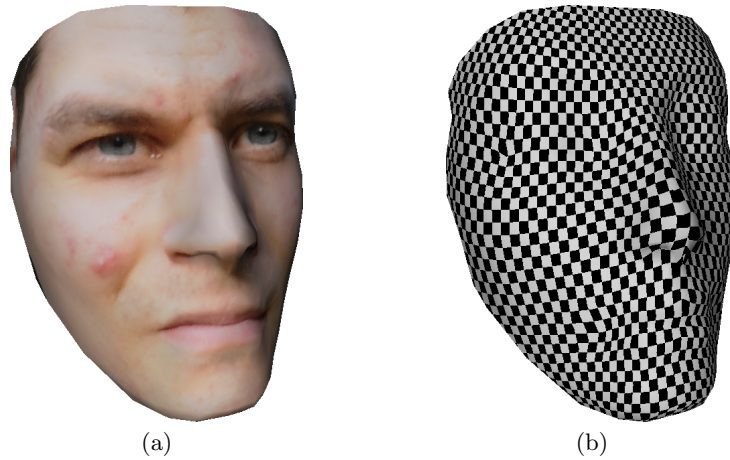
## 4.2 Constrained texture mapping

Another important application of parameterization techniques is texture mapping of a 2D image on a 3D model. The main features of the image and the model must fit. Therefore positions of the corresponding points of the parameterization must be fixed. Our method can be adapted to that case. In fact it only reduces the number of variables of the function to minimize: we consider the points that are not fixed.

In the example of Figure 9 we map an image of a face on a mask of Nefertiti. We select manually 13 corresponding points. They are displayed with thin points on the figure. On Figure 9 (a), we map the image on the mesh to see that the 13 points are mapped to the right position. On Figure 9 (b) we display the map of a checkerboard with the same parameterization, it shows that the parameterization is still conformal.



**Fig. 8.** Corresponding 13 points in the image and the mesh.



**Fig. 9.** Textured 3D model with (a) an image face and (b) a checkerboard. Same parameterization computed with  $E = H + B$ .

## Conclusion

We have described a method of conformal parameterization of triangular and quadrangular meshes. It consists in introducing a complex number to represent angles between edges and minimizing an energy whose variables are the vertices of the parameterization. Energies measuring angular and metric distortions along the boundary are added to compute parameterizations with natural boundaries. The method is general and can be adapted to preserve areas or to do constrained texture mapping. Many examples are shown to assess the efficiency of our method.

## References

1. Desbrun, M., Meyer, M., Alliez, P.: Intrinsic parameterizations of surface meshes. In: *Computer Graphics Forum*. vol. 21, pp. 209–218 (2002)
2. Floater, M.: Mean value coordinates. *Computer Aided Geometric Design* 20(1), 19–27 (2003)
3. Kharevych, L., Springborn, B., Schröder, P.: Discrete conformal mappings via circle patterns. *ACM Transactions on Graphics (TOG)* 25(2), 438 (2006)
4. Lévy, B., Petitjean, S., Ray, N., Maillot, J.: Least squares conformal maps for automatic texture atlas generation. *ACM Transactions on Graphics* 21(3), 362–371 (2002)
5. Mercat, C.: Discrete Riemann surfaces and the Ising model. *Communications in Mathematical Physics* 218(1), 177–216 (2001)
6. Nocedal, J., Wright, S.: *Numerical optimization*. Springer verlag (1999)
7. Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. *Experimental mathematics* 2(1), 15–36 (1993)
8. Sheffer, A., De Sturler, E.: Surface parameterization for meshing by triangulation flattening. In: *Proc. 9th International Meshing Roundtable*. Citeseer (2000)
9. Sheffer, A., De Sturler, E.: Parameterization of faceted surfaces for meshing using angle-based flattening. *Engineering with Computers* 17(3), 326–337 (2001)
10. Sheffer, A., Lévy, B., Mogilnitsky, M., Bogomyakov, A.: ABF++: Fast and robust angle based flattening. *ACM Transactions on Graphics* 24(2), 311–330 (2005)
11. Wegert, E.: Nonlinear Riemann-Hilbert problems—history and perspectives. In: *Computational methods and function theory 1997 (Nicosia)*, Ser. Approx. Decompos., vol. 11, pp. 583–615. World Sci. Publ., River Edge, NJ (1999)