

# Compute Quasi-Conformal Map on Point Cloud by Point Integral Method

Zhen Li\*      Zuoqiang Shi†      Jian Sun ‡

October 27, 2017

## Abstract

We develop a method searching for quasi-conformal maps from point-cloud surfaces to Euclidean plane. The maps can be got by directly solving Beltrami equation with proper boundary condition or solving optimization problem of the variational formulation. The point integral method (PIM) is used for discretization on point clouds. Numerical experiments suggest that the method converges as the density of points increases.

**Keywords:** quasi-conformal map, Beltrami equation, point cloud, Point Integral Method (PIM).

## 1 Introduction

Conformal and quasi-conformal maps appear in many areas of sciences and engineering, such as physics, mechanics and computer science. It is well known in the theory of complex analysis that for two simply-connected domains on the complex plane there is a conformal map from the one to the other. Conformality means that the map preserves local shapes, i.e. it maps right angle to right angle and keep the orientation unchanged. The necessary and sufficient condition for  $\Phi : (x, y) \mapsto (u, v)$  to be conformal is that  $u$  and  $v$  satisfy the well-known Cauchy-Riemann equation

$$\begin{cases} u_x = v_y, \\ u_y = -v_x, \end{cases}$$

where  $(x, y)$  is the Cartesian coordinates on  $\mathcal{M}$  and the suffices stand for partial derivatives. These two equations can be rewritten in a coordinates-independent form

$$\nabla u + \mathbf{J}\nabla v = 0, \tag{1.1}$$

---

\*Yau Mathematical Sciences Center, Tsinghua University, Beijing, China, 100084. *Email:* zli12@mails.tsinghua.edu.cn.

†Yau Mathematical Sciences Center, Tsinghua University, Beijing, China, 100084. *Email:* zqshi@tsinghua.edu.cn.

‡Yau Mathematical Sciences Center, Tsinghua University, Beijing, China, 100084. *Email:* jsun@math.tsinghua.edu.cn.

where  $\nabla$  is the gradient operator on  $\mathcal{M}$  and  $\mathbf{J}$  is the conjugate operator. In this case,

$$\nabla = \begin{pmatrix} \partial_x \\ \partial_y \end{pmatrix} \quad \text{and} \quad \mathbf{J} = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

From above we can easily see that both  $u$  and  $v$  are harmonic functions. They satisfy Laplace equations

$$\begin{cases} \Delta u = 0, \\ \Delta v = 0. \end{cases} \quad (1.2)$$

For this reason,  $(u, v)$  are sometimes called a pair of conjugate harmonic functions. Conjugacy means that their stationary points are the same and their gradients are orthogonal everywhere.

To solve for the conformal map  $\Phi = (u, v)$  from the Cauchy-Riemann equation, we need a boundary condition which defines the boundary of the image of  $\Phi$ , which is a closed curve in the  $uv$ -plane. This can be expressed by  $(u_b, v_b)$ , the boundary value of  $(u, v)$ , as

$$f(u_b, v_b) = 0. \quad (1.3)$$

Generally, this is a nonlinear boundary condition. But we can take a triangular domain in the  $uv$ -plane as the image of  $\Phi$ . In this case the boundary condition is piecewise linear.

According to the theory of complex analysis, the problem of solving (1.1) with boundary condition (1.3) has a solution. However, it is not easy to solve (1.1) with (1.3) directly. In this paper we propose a method to solve (1.1) (1.2) and (1.3) together. We can imagine the following procedure. If  $u_b$  is given, and we can solve  $v_b$  from (1.3) as a function of  $u_b$ , then we have two Dirichlet problems for the Laplace equations (1.2). By solving these problems we get a pair  $(u, v)$ . We can check how good is this  $(u, v)$  as a solution of (1.1). If it is not good enough, we change  $u_b$  and do the same thing again. By repeating this procedure, we may expect to find a  $u_b$  such that (1.1) is perfectly satisfied. This imaginary procedure can be realized as an optimization problem. The objective function is

$$\|\nabla u + \mathbf{J}\nabla v\|, \quad (1.4)$$

and the argument is  $u_b$ . Here  $\|\cdot\|$  is some norm. The solution of the Cauchy-Riemann equation is found when the objective function attains its global minimum 0.

If  $\mathcal{M}$  is a curved surface, the concept of harmonicity and orthogonality can still be defined. Thus conformality can be generalized to surfaces. Since there is a standard global coordinate system  $(u, v)$  on Euclidean plane  $\mathbb{R}^2$ , any map  $\Phi$  from a surface  $\mathcal{M}$  to  $\mathbb{R}^2$  can be seen as two functions  $u$  and  $v$  on  $\mathcal{M}$ .  $\Phi$  is said to be conformal if its components  $(u, v)$  is a pair of conjugate harmonic functions on  $\mathcal{M}$ . In this case the Cauchy-Riemann equation (1.1) is still a necessary and sufficient condition of the conformality of  $\Phi$ . Here the gradient operator and Laplace operator are associated to the metric of  $\mathcal{M}$ .

The concept of conformality can be generalized in another way. The classical conformal map takes infinitesimal circles to infinitesimal circles. If it is allowed to take infinitesimal circles to infinitesimal ellipses, and the eccentric ratio of the ellipses are bounded, then we

get a quasi-conformal map. If we consider general quasi-conformal maps on a surface,  $u$  and  $v$  also satisfy a system of PDEs, which is the Beltrami equation

$$\mathbf{A}^{-1} \cdot \nabla u + \mathbf{J} \cdot \nabla v = 0.$$

Here  $\mathbf{A}$  is a second order tensor field on  $\mathcal{M}$ . If  $\mathbf{A}$  is the identity, we have a Cauchy-Riemann equation. The concept of quasi-conformal map appears naturally if we consider conformal map with respect to a metric that is different from the default one on  $\mathcal{M}$ . If  $\Phi$  is conformal in the new metric, it is quasi-conformal in the default metric. The converse is also true. Any quasi-conformal map is conformal in some other metric. Therefore, the above  $u$  and  $v$  are a pair of conjugate harmonic functions with respect to another metric on the surface. The derivation and some mathematical details of the Beltrami equation will be presented in the next section.

The standard conformal map is a particular case of the quasi-conformal map from surface to plane. The boundary condition can also be defined in general case. The idea of the method for solving the boundary value problem of quasi-conformal map is the same as that of conformal map as described above, i.e., reducing the boundary value problem to an optimization problem.

To discretize the optimization problem on point cloud, we need a numerical method that is suitable for point cloud data. In [8] the authors developed a numerical method, the Point Integral Method (PIM), for solving Laplace equation. As introduced above, the solution of a Cauchy-Riemann equation is a pair of conjugate harmonic functions, which satisfy Laplace equation. We will see in the next section that the solution of a Beltrami equation for quasi-conformal map is a pair of functions that satisfy an elliptic equation, which can also be solved by PIM. In the method described in this paper, we need to solve the Beltrami equation together with the related elliptic equations and boundary conditions. They are discretized to a linear system of algebraic equations by PIM.

There are many numerical methods for computing conformal or quasi-conformal maps on meshes, such as [7, 2, 4, 5, 12]. Sometimes it is inconvenient or even difficult to generate meshes of enough quality for numerical algorithms [3, 10]. This is a serious issue when the geometry of the surface is complicated. Point cloud is a much simpler discretization method for representing domains and data. In recent years, more and more research focused on solving problems on point clouds [1, 6, 9, 8, 11]. It is natural to develop methods for solving conformal and quasi-conformal maps on point clouds, which is the motivation of this work.

The paper is organized in the following way: Section 2 is a brief introduction to quasi-conformal map, Beltrami equation and its boundary value problem. Here we emphasize the geometric aspects. Section 3 and Section 4 describe the discretizations of the boundary value problem by the finite element method (FEM) and the point integral method (PIM) respectively. In Section 6, several examples are presented to show the convergence and performance of the PIM algorithm for solving conformal and quasi-conformal maps.

## 2 Beltrami Equation for Quasi-Conformal Map

In this section, we give an introduction to the theoretical background of the quasi-conformal map and the Beltrami equation. The readers can also find the relevant contents in textbooks

on complex analysis, Riemann surfaces, and differential geometry.

Quasi-conformal map  $w(z) = u(z) + \sqrt{-1}v(z)$  can be defined as the solution of the complex Beltrami equation

$$\frac{\partial w}{\partial \bar{z}} = \mu \frac{\partial w}{\partial z},$$

where  $\mu$  is a given complex function on  $U$ , satisfying some boundedness condition. In this section we will show that from  $\mu$  we can define a metric  $\mathbf{C}$ , such that the quasi-conformal map is conformal with respect to  $\mathbf{C}$ . Furthermore, we can transform the above complex equation into a real form

$$\mathbf{A}^{-1} \cdot \nabla u + \mathbf{J} \cdot \nabla v = 0,$$

in which  $\mathbf{A}$  is explicitly determined by  $\mu$ . Several other interesting and useful consequences will also be introduced.

In the following, let  $\mathcal{M}$  be a surface with a metric  $\mathbf{G}$ , admitting an orthonormal moving frame. Here the surface  $\mathcal{M}$  is not assumed to be embedded in  $\mathbb{R}^3$ . It could be a part of any two dimensional manifold. In later sections, we will require that  $\mathcal{M}$  is embedded in  $\mathbb{R}^3$  and  $\mathbf{G}$  is induced from the Euclidean inner product  $\langle \cdot, \cdot \rangle$  of  $\mathbb{R}^3$ , so that the numerical algorithms can be applied to solve the Beltrami equation. Suppose  $\mathbf{C}$  is another metric which is generally different from the default metric  $\mathbf{G}$ . Let  $\mathbf{g}$  be the Euclidean metric on a plane which is identified with  $\mathbb{R}^2$ . Re-denote  $(u, v)$  by  $(u^a)$ , then the components of  $\mathbf{g}$  constructs an identity matrix  $(\delta_{ab})$ . We will use  $a, b, \dots$  as indices on the plane and  $A, B, \dots$  on  $\mathcal{M}$ . Einstein's summation convention is used and all indices run from 1 to 2 without specifying. All quantities are assume to be smooth.

## 2.1 Conformal map with respect to a different metric

A map  $\Phi$  from  $(\mathcal{M}, \mathbf{C})$  to  $(\mathbb{R}^2, \mathbf{g})$  is said to be conformal with respect to  $\mathbf{C}$  if the pullback of  $\mathbf{g}$  by  $\Phi$  is pointwisely proportional to  $\mathbf{C}$ , i.e.,

$$\Phi^* \mathbf{g} = h \mathbf{C},$$

where  $h$  is a positive function on  $\mathcal{M}$ . By definition of  $\mathbf{g}$  we have

$$\begin{aligned} \Phi^* \mathbf{g} &= \Phi^*(du^a du^b \delta_{ab}) \\ &= du^a du^b \delta_{ab}. \end{aligned}$$

The d's in the first line are differentials on  $\mathbb{R}^2$ , and those in the second line are differentials on  $\mathcal{M}$ . Take a moving frame  $\{e_A\}$  which is orthonormal with respect to  $\mathbf{G}$  on  $\mathcal{M}$ , we have

$$\begin{aligned} h \mathbf{C}_{AB} &= h \mathbf{C}(e_A, e_B) \\ &= du^a(e_A) du^b(e_B) \delta_{ab} \\ &= \mathbf{G}(e_A, \nabla u^a) \mathbf{G}(e_B, \nabla u^b) \delta_{ab} \\ &= \nabla_A u^a \nabla_B u^b \delta_{ab} \\ &= F_A^a F_B^b \delta_{ab}, \end{aligned}$$

where  $\nabla$  is the gradient operator associated with  $\mathbf{G}$  and  $F_A^a = \nabla_A u^a$ . Use matrix notation  $C = (C_{AB})$  and  $F = (F_A^a)$ , the above equation becomes

$$hC = FF^T.$$

By taking determinants on both sides, we get

$$h = \frac{\det F}{\sqrt{\det C}}.$$

Plug this back, we get

$$\det FF^{-1} = \sqrt{\det C} F^T C^{-1}.$$

By definition of cofactor of a matrix and symmetry of  $C$ , we have

$$\text{cof}F = \sqrt{\det C} C^{-1} F.$$

The components of the inverse metric  $C^{-1}$  are usually denoted as  $(C^{AB})$ . There are four equations in the above matrix equation, which are

$$\begin{cases} F_2^2 = \sqrt{\det C} C^{1A} F_A^1, & \begin{cases} -F_2^1 = \sqrt{\det C} C^{1A} F_A^2, \\ F_1^1 = \sqrt{\det C} C^{2A} F_A^2. \end{cases} \end{cases} \quad (2.1)$$

The conjugate operator  $\mathbf{J}$  is defined by a rotation

$$\mathbf{J} \cdot e_1 = e_2, \quad \mathbf{J} \cdot e_2 = -e_1.$$

It is easy to verify that the above two pairs of equations are respectively equivalent to two vector-valued equations

$$\begin{cases} -\mathbf{J} \cdot \nabla u^2 = \mathbf{A}^{-1} \cdot \nabla u^1, \\ \mathbf{J} \cdot \nabla u^1 = \mathbf{A}^{-1} \cdot \nabla u^2, \end{cases} \quad (2.2)$$

where  $\mathbf{A}$  is defined by its matrix  $A$ , and  $A^{-1} = \sqrt{\det C} C^{-1}$ . Next we will show that these two equations are equivalent to each other. As defined above,  $A$  is a symmetric matrix with determinant 1. Write out the components of  $A^{-1}$  and  $A$  explicitly,

$$A^{-1} = \begin{pmatrix} A_1^1 & A_1^2 \\ A_2^1 & A_2^2 \end{pmatrix}, \quad \text{and} \quad A = \begin{pmatrix} A_2^2 & -A_1^2 \\ -A_2^1 & A_1^1 \end{pmatrix}$$

Also the matrix of the conjugate operator  $\mathbf{J}$  in the moving frame  $\{e_A\}$  is

$$J = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}.$$

It is easy to check that  $AJ = JA^{-1}$ . This relation is actually independent of choice of orthonormal moving frame, therefore

$$\mathbf{A} \cdot \mathbf{J} = \mathbf{J} \cdot \mathbf{A}^{-1}.$$

By this relation and the property that  $\mathbf{J}^2 = -\text{id}$ , we have

$$\mathbf{A}^{-1} \cdot \nabla u^2 = -\mathbf{J}^2 \cdot \mathbf{A}^{-1} \cdot \nabla u^2 = -\mathbf{J} \cdot \mathbf{A} \cdot \mathbf{J} \cdot \nabla u^2.$$

Use the second equation in (2.2), we have

$$\mathbf{A}^{-1} \cdot \nabla u^2 = \mathbf{J} \cdot \nabla u^1,$$

which is exactly the first equation in (2.2). By similar arguments we can derive the second equation in (2.2) by the first one too. Thus the two vector-valued equations in (2.2) are equivalent. Without loss of generality, we choose the first one

$$\mathbf{A}^{-1} \cdot \nabla u + \mathbf{J} \cdot \nabla v = 0. \quad (2.3)$$

This is a system of PDEs for  $(u^a) = (u, v)$ , so called Beltrami equation, which characterizes conformal map with respect to  $\mathbf{C}$  from surface to Euclidean plane. If we take  $\mathbf{C} = \mathbf{G}$ , then  $\mathbf{A} = \text{id}$  and (2.3) becomes a system of Riemann-Cauchy equations.

## 2.2 Conjugate harmonic functions

We are to show that the solution of (2.3) is a pair of harmonic functions on  $(\mathcal{M}, \mathbf{C})$ . We still take the orthonormal moving frame  $\{e_A\}$  and denote its coframe with respect to  $\mathbf{G}$  as  $\{\omega^A\}$ . Let  $D$  be the associated covariant differential operator and its connection forms be  $\omega_A^B$ . That is

$$d\omega^B = \omega^A \wedge \omega_A^B.$$

Then we have

$$\begin{aligned} D\nabla u^a &= D(F_1^a e_1 + F_2^a e_2) \\ &= dF_1^a e_1 + F_1^a D e_1 + dF_2^a e_2 + F_2^a D e_2 \\ &= (dF_1^a + F_2^a \omega_2^1) e_1 + (dF_2^a + F_1^a \omega_1^2) e_2 \\ &=: DF_1^a e_1 + DF_2^a e_2. \end{aligned}$$

Here  $DF_1^a$  and  $DF_2^a$  are also 1-forms, thus they can be spanned by  $\omega^A$ . Suppose

$$DF_A^a = D_B F_A^a \omega^B.$$

By Poincaré's lemma, there is

$$\begin{aligned} 0 &= d^2 u^a \\ &= d(F_1^a \omega^1 + F_2^a \omega^2) \\ &= dF_1^a \omega^1 + F_1^a d\omega^1 + dF_2^a \omega^2 + F_2^a d\omega^2 \\ &= (dF_1^a + F_2^a \omega_2^1) \omega^1 + (dF_2^a + F_1^a \omega_1^2) \omega^2 \\ &= DF_1^a \omega^1 + DF_2^a \omega^2 \\ &= (D_1 F_2^a - D_2 F_1^a) \omega^1 \wedge \omega^2. \end{aligned}$$

Thus we have

$$D_1 F_2^a = D_2 F_1^a.$$

Apply this relation to the two pairs of equations in (2.2), we get

$$D_A(\sqrt{\det C} C^{AB} F_B^a) = 0.$$

Divide this equation by  $\sqrt{\det C}$ , we find that  $u^a$  satisfies a Laplace equation

$$\Delta_{\mathbf{C}} u^a = 0,$$

where  $\Delta_{\mathbf{C}}$  is the Laplace-Beltrami operator with respect to metric  $\mathbf{C}$ . Therefore the solutions  $(u, v)$  of Beltrami equations (2.3) is a pair of conjugate harmonic functions with respect to  $\mathbf{C}$ . The above Laplace equation can be considered as an elliptic equation in the default metric  $\mathbf{G}$ ,

$$\nabla \cdot (\mathbf{A}^{-1} \cdot \nabla u^a) = 0. \quad (2.4)$$

Here  $\nabla \cdot = \mathbf{div}$  and  $\nabla = \mathbf{grad}$  are divergence and gradient operators associated with  $\mathbf{G}$ .

**Remark 2.1.** *We should emphasize that any solution of (2.3) is a solution of (2.4), but the converse is generally false. The solutions of (2.4) for  $a = 1, 2$  is surely two functions which are harmonic with respect to  $\mathbf{C}$ , but in most cases they are not conjugate to each other, hence is not a solution to (2.3).*

### 2.3 Quasi-conformal map and Beltrami coefficient

The most common form of Beltrami equation is defined in the context of complex analysis. Let  $\Phi$  be a map from some open set  $U \subset \mathbb{C}$  to  $\mathbb{C}$ , and  $w = \Phi(z)$ . Then the classical Beltrami equation is

$$\frac{\partial w}{\partial \bar{z}} = \mu \frac{\partial w}{\partial z},$$

where  $\mu$  is a given complex function on  $U$ . The solution of this equation is called a quasi-conformal map with Beltrami coefficient  $\mu$ .

In the settings of previous part of this section, we can complexify the surface  $\mathcal{M}$ . Let  $\omega = \omega^1 + \sqrt{-1}\omega^2$ , and  $\mu = \alpha + \sqrt{-1}\beta$ . We assign our metric  $\mathbf{C}$  to be

$$\mathbf{C} = f|\omega + \mu\bar{\omega}|^2,$$

where the bar denotes complex conjugacy and  $f$  is any positive function on  $\mathcal{M}$ . Plug in the definitions, we have

$$\mathbf{C} = f(((1 + \alpha)^2 + \beta^2)(\omega^1)^2 + 4\beta\omega^1\omega^2 + ((1 - \alpha)^2 + \beta^2)(\omega^2)^2),$$

or in its matrix form in the moving frame  $\{e_A\}$

$$C = f \begin{pmatrix} (1 + \alpha)^2 + \beta^2 & 2\beta \\ 2\beta & (1 - \alpha)^2 + \beta^2 \end{pmatrix}. \quad (2.5)$$

The Euclidean plane can also be complexified. Let  $w = u + \sqrt{-1}v$ . Suppose

$$\Phi^*dw = w_1\omega + w_2\bar{\omega}.$$

Then for the above choice of the metric  $\mathbf{C}$  with  $\mu = w_2/w_1$ , there is

$$\begin{aligned}\Phi^*\mathbf{g} &= |\Phi^*dw|^2 \\ &= |w_1|^2 \left| \omega + \frac{w_2}{w_1}\bar{\omega} \right|^2 \\ &= \frac{|w_1|^2}{f} \mathbf{C}.\end{aligned}$$

This means that the quasi-conformal map with the Beltrami coefficient  $\mu$  is a conformal map with respect to the metric  $\mathbf{C}$ , and  $\mathbf{C}$  is uniquely determined by  $\mu$  up to a dilation (multiplying a positive function). For this reason, we can compute a quasi-conformal map by means of a conformal map.

Conversely, if we are given a metric  $\mathbf{C}$  and a conformal map  $\Phi$  with respect to it, it is natural to ask that which quasi-conformal map it stands for, and what is  $\mu$ . The answer is straightforward. We only need to solve equations (2.5) for  $\alpha, \beta$  (real and imaginary parts of  $\mu$ ) from components of  $\mathbf{C}$  in an orthonormal moving frame. The explicit formula for  $\mu$  is

$$\mu = \frac{C_{11} - C_{22} + 2\sqrt{-1}C_{12}}{C_{11} + C_{22} + 2\sqrt{\det C}}.$$

Any quasi-conformal map sends infinitesimal ellipses on  $\mathcal{M}$  to infinitesimal circles on the plane. In fact, a quasi-conformal map  $\Phi$  can be characterized by the shapes and orientations of the ellipses, up to dilation. These informations are all encoded in the Beltrami coefficient  $\mu$ , or equivalently, the metric  $\mathbf{C}$ . For  $\mathbf{C}$  given by (2.5) in the orthonormal moving frame, it is not difficult to find the eigenvalues and eigenvectors of  $C$  to be

$$c_{1,2} = \left(1 \pm \sqrt{\alpha^2 + \beta^2}\right)^2, \quad \theta^B = \omega^A Q_A^B,$$

where the transformation matrix is

$$Q = \frac{1}{\sqrt{2(\alpha^2 + \beta^2 + \alpha\sqrt{\alpha^2 + \beta^2})}} \begin{pmatrix} \alpha + \sqrt{\alpha^2 + \beta^2} & -\beta \\ \beta & \alpha + \sqrt{\alpha^2 + \beta^2} \end{pmatrix}.$$

We have omitted the dilation factor  $f$  here.  $\{\theta^A\}$  is another orthonormal coframe with respect to the default metric  $\mathbf{G}$ . It is also orthogonal with respect to  $\mathbf{C}$  which can be spanned in this coframe as

$$\mathbf{C} = c_1(\theta^1)^2 + c_2(\theta^2)^2.$$

Let  $\{g_A\}$  be the orthonormal moving frame dual to  $\{\theta^A\}$  with respect to  $\mathbf{G}$ . Then the transformation matrix from  $\{e_A\}$  to  $\{g_A\}$  is also  $Q$ . That is

$$g_B = e_A Q^A_B.$$

The quasi-conformal map  $\Phi$  sends an ellipse in the tangent space at each point  $p \in \mathcal{M}$ , whose semi-major axis is  $\sqrt{c_1}g_1$  and semi-minor axis  $\sqrt{c_2}g_2$  to a circle centered at  $\Phi(p)$  on the plane. The function

$$D_\Phi = \sqrt{\frac{c_1}{c_2}} = \frac{1 + \sqrt{\alpha^2 + \beta^2}}{1 - \sqrt{\alpha^2 + \beta^2}} \quad (2.6)$$

is called local dilatation<sup>1</sup> quotient. If  $\sup_{\mathcal{M}} D_\Phi = K$ , then  $\Phi$  is called a  $K$ -quasi-conformal map in some literatures.

In summary, a quasi-conformal map  $\Phi = u + \sqrt{-1}v$  with Beltrami coefficient  $\mu$  from a surface  $\mathcal{M}$  to an Euclidean plane is a solution to the Beltrami equation (2.3). And  $(u, v)$  is a pair of conjugate harmonic functions with respect to the metric  $\mathbf{C}$  defined by  $\mu$ , which satisfy Laplace equations (2.4). Therefore we have reduced the problem of finding a quasi-conformal map to solving a system of PDEs, in which the operator  $\mathbf{A}$  can be expressed in the moving frame  $\{e_A\}$  as

$$A = \frac{C}{\sqrt{\det C}} = \frac{1}{1 - \alpha^2 - \beta^2} \begin{pmatrix} (1 + \alpha)^2 + \beta^2 & 2\beta \\ 2\beta & (1 - \alpha)^2 + \beta^2 \end{pmatrix}. \quad (2.7)$$

Since  $\det A = 1$ , we have  $A^{-1} = \text{cof} A^T$ . Obviously,  $\mathbf{A}$  is just a real equivalence of  $\mu = \alpha + \sqrt{-1}\beta$ . In fact it characterizes a conformal structure on  $\mathcal{M}$ . We can think of it as a metric up to dilation.

**Remark 2.2.** *Assigning a Beltrami coefficient on a surface is more difficult than on a plane. On a plane there is a standard Cartesian coordinate system, thus a Beltrami coefficient is just a complex function. On a surface, there is no canonical choice of orthonormal moving frame. To assign a Beltrami coefficient on a surface, we must assign both the orthonormal moving frame and the complex function. This difference is not hard to understand, since assigning a Beltrami coefficient is equivalent to assigning a metric tensor up to dilation.*

## 2.4 Boundary condition

The commonly used Dirichlet and Neumann conditions for second order PDEs are inadequate for Beltrami equation. It is well-known that a conformal map between two simple domains on the plane is uniquely determined if the image of the boundary is given and three boundary points are fixed. The image of unfixed boundary points are allowed to move freely along the boundary of target domain. This statement holds for quasiconformal map too, because any quasi-conformal map is a conformal map in certain metric.

The image of the boundary is a closed curve in the  $uv$ -plane, which can be expressed by an implicit equation

$$f(u_b, v_b) = 0, \quad (2.8)$$

where  $(u_b, v_b)$  is the boundary value of  $(u, v)$ . Notice that  $(u_b, v_b)$  is the image of all the boundary points, including the fixed images  $\{(u_i, v_i)\}_{i=1}^3$  of three boundary points  $\{\mathbf{p}_i\}_{i=1}^3$ .

---

<sup>1</sup>We use the word *dilatation* to describe the isotropic deformation, and use *dilatation* to describe the anisotropic aspect of a map.

So the boundary condition can be written as

$$\begin{cases} f(u(\mathbf{s}), v(\mathbf{s})) = 0, & \mathbf{s} \in \partial\mathcal{M}, \\ (u(\mathbf{p}_i), v(\mathbf{p}_i)) = (u_i, v_i), & \mathbf{p}_i \in \partial\mathcal{M}, i = 1, 2, 3. \end{cases} \quad (2.9)$$

In order to solve the Beltrami equation for a quasiconformal map, we can assign the value of only three boundary points. At the same time, we need to restrict the image of the rest of  $\partial\mathcal{M}$  to the boundary of the target domain. This boundary condition is related to the usual Dirichlet condition in the sense that if one of  $(u, v)$  is determined, then the other is also determined. This relationship is crucial for solving boundary value problems of Beltrami equations. It will be made more clear in the next part.

The boundary condition (2.9) is generally nonlinear. Hence we need to solve a system of linear PDEs with nonlinear boundary condition. It should be solved by iterative methods. When the target domain is a triangle, however, the boundary condition becomes (piecewise) linear. In this special case the system can be solved in one step. In the following of this paper, we mainly consider linear problems, i.e., the target domain are triangles. The nonlinear problems occupies a secondary part.

## 2.5 Solve linear problems

From above we see that the solution of Beltrami equation is a pair of conjugate functions who satisfy same elliptic equations. The two functions are harmonic with respect to certain metric, and the elliptic equations satisfied by them are Laplace equations in that metric. On the one hand, the Beltrami equation with adequate boundary condition is not easy to solve while the usual Dirichlet condition is inadequate for it. On the other hand, Laplace equations with Dirichlet conditions are easy to solve. Since the boundary condition for Beltrami equation can be related to the Dirichlet condition, the Laplace equations can be used as a bridge between the Beltrami equation and its boundary condition. In other words, we can solve Beltrami equation together with the two related Laplace equations.

Let  $\mathcal{M}$  be a simply-connected surface with boundary. Let  $\mathbb{R}^2$  be an Euclidean plane with standard Cartesian coordinates  $(u, v)$ . Suppose we are given an orthonormal moving frame and a complex function  $\mu$  on  $\mathcal{M}$ . We want to find a quasi-conformal map  $\Phi$  from  $\mathcal{M}$  to  $\mathbb{C} \cong \mathbb{R}^2$ , taking  $\mu$  as its Beltrami coefficient. We can consider the redundant system of PDEs

$$\nabla \cdot (\mathbf{A}^{-1} \cdot \nabla u) = 0, \quad (2.10)$$

$$\nabla \cdot (\mathbf{A}^{-1} \cdot \nabla v) = 0, \quad (2.11)$$

$$\mathbf{A}^{-1} \cdot \nabla u + \mathbf{J} \cdot \nabla v = 0, \quad (2.12)$$

where  $\mathbf{A}$  is a positive self-adjoint operator with determinant 1. It acts on tangent vector fields on  $\mathcal{M}$ . The components of  $\mathbf{A}$  in the orthonormal moving frame can be given by (2.7). Although the Beltrami equation (2.12) implies the Laplace equations (2.10) and (2.11), we still need them as a bridge between the Beltrami equation (2.12) and its boundary conditions.

By solving Dirichlet problems of (2.10) and (2.11), we can express  $(u, v)$  as

$$u = \mathcal{T}u_b, \quad v = \mathcal{T}v_b, \quad (2.13)$$

where  $(u_b, v_b)$  is the boundary value of  $(u, v)$ .  $\mathcal{T}$  is a linear operator defined by any solver of the Dirichlet problem of equation (2.10) or (2.11). It is well-defined since given any boundary value we can solve the Dirichlet problem and get the solution.

According to the boundary condition (2.9), the boundary values  $(u_b, v_b)$  satisfy an implicit equation

$$f(u_b, v_b) = 0. \quad (2.14)$$

Generally we cannot express  $v_b$  by  $u_b$ , vice versa. In the linear case, however, the target domain  $\Phi(\mathcal{M})$  is a triangle in the  $uv$  plane. For each edge of the triangle, we can express  $v_b$  by  $u_b$ . For example, each point  $(u_b, v_b)$  lying on the segment between  $(u_1, v_1)$  and  $(u_2, v_2)$  satisfies the linear equation

$$v_b = \frac{v_2 - v_1}{u_2 - u_1}(u_b - u_1) + v_1.$$

Similar for the segment between  $(u_2, v_2)$  and  $(u_3, v_3)$  and that between  $(u_3, v_3)$  and  $(u_1, v_1)$ . With more abstract notation, we have

$$v_b = \mathcal{S}u_b, \quad (2.15)$$

where  $\mathcal{S}$  is a piecewise linear operator defined by the triangular boundary of the target domain. Notice that  $(u_b, v_b)$  contain all the boundary points, including the three points  $\{\mathbf{p}_i\}_{i=1}^3$  whose images are fixed as  $\{(u_i, v_i)\}_{i=1}^3$ .

With the help of the operators  $\mathcal{T}$  and  $\mathcal{S}$  we can relate the boundary values to the Beltrami equation. Plug (2.13) and (2.15) into the Beltrami equation (2.12), we have

$$\begin{aligned} 0 &= \mathbf{A}^{-1} \cdot \nabla u + \mathbf{J} \cdot \nabla v \\ &= \mathbf{A}^{-1} \cdot \nabla \mathcal{T}u_b + \mathbf{J} \cdot \nabla \mathcal{T}v_b \\ &= \mathbf{A}^{-1} \cdot \nabla \mathcal{T}u_b + \mathbf{J} \cdot \nabla \mathcal{T}\mathcal{S}u_b. \end{aligned}$$

So we get an equation for  $u_b$ :

$$(\mathbf{A}^{-1} \cdot \nabla \mathcal{T} + \mathbf{J} \cdot \nabla \mathcal{T}\mathcal{S})u_b = 0. \quad (2.16)$$

This is the equation to discretize and solve. After we get  $u_b$ , we can get  $v_b$  by  $v_b = \mathcal{S}u_b$ . Then the quasiconformal map  $(u, v)$  is get by solving the Laplace equations (2.10) and (2.11) with Dirichlet conditions  $u_b$  and  $v_b$  respectively.

### 3 FEM Discretization

Although our main concern in this paper is to solve for quasi-conformal maps from point cloud, the formulation of the problem does not rely on any particular discretization method. It can also be implemented by FEM on meshes, which provides a method to compare with. In this section we describe briefly how to find quasiconformal maps by FEM. The reason that we put the FEM part ahead of the PIM part is that the FEM discretization of equation (2.16) is simpler and easier to understand. It helps to understand the PIM discretization, which is slightly different.

### 3.1 FEM discretization of differential operators

Suppose the finite element basis functions defined on  $\mathcal{M}$  are  $\{\phi_i\}_{i=1}^N$ . In this paper we use linear elements, i.e. hat functions, as basis. Let  $U$  be the column vector representing the discretized  $u$ , that is

$$u = \sum_{j=1}^N U_j \phi_j.$$

Then the discretization of the Laplace equation (2.10) is

$$\mathcal{L}U = 0,$$

where the matrix operator  $\mathcal{L}$  is given by

$$\mathcal{L}_{ij} = \int_{\mathcal{M}} \nabla \phi_i \cdot \mathbf{A}^{-1} \cdot \nabla \phi_j \, dV.$$

The discretization of  $\nabla$  is given by  $\mathcal{D} = \mathcal{G}$ , where

$$\mathcal{G}_{ij} = \int_{\mathcal{M}} \phi_i \nabla \phi_j \, dV.$$

Similarly,  $\mathbf{A}^{-1} \cdot \nabla$  and  $\mathbf{J} \cdot \nabla$  are discretized as

$$\mathcal{D}_{ij}^A = \int_{\mathcal{M}} \phi_i \mathbf{A}^{-1} \cdot \nabla \phi_j \, dV, \quad \mathcal{D}_{ij}^J = \int_{\mathcal{M}} \phi_i \mathbf{J} \cdot \nabla \phi_j \, dV.$$

These discretizations of differential operators are got by standard procedures of FEM. Other differential operators can be discretized in similar way.

### 3.2 Solving linear problems with FEM

In the discrete setting, instead of attacking (2.16) directly, we first discretize the PDE system (2.10) (2.11) and (2.12). Then we repeat the same procedure of deriving (2.16) in Section 2 to derive its discretization. Without specifying, discretizations of functions and operators are represented by column vectors and matrices.

Suppose the system (2.10) (2.11) and (2.12) can be discretized as

$$\mathcal{L}U = 0, \tag{3.1}$$

$$\mathcal{L}V = 0, \tag{3.2}$$

$$\mathcal{A}^{-1}\mathcal{D}U + \mathcal{J}\mathcal{D}V = 0, \tag{3.3}$$

where  $\mathcal{L}, \mathcal{D}, \mathcal{A}, \mathcal{J}$  are matrices and  $U, V$  are column vectors representing the values of discretized  $u, v$ . There are two type of boundary conditions. The one is fixed. We pick out three points on  $\partial\mathcal{M}$  and map them to the vertices of a triangular domain on  $\mathbb{R}^2$ . We indicate these points on  $\partial\mathcal{M}$  by suffix 'F', thus the values of  $u, v$  on these points are denoted by column vectors  $U_F, V_F$ . The other is partially constrained. The other boundary points are mapped to the segments of the triangular domain on  $\mathbb{R}^2$ . We indicate these points on

$\partial\mathcal{M}$  by suffix ‘ $S$ ’, thus the values of  $u, v$  on these points are denoted by two column vectors  $U_S, V_S$ . The values of  $u, v$  on interior points are denoted by  $U_I, V_I$ . Since the segments in  $\mathbb{R}^2$  are expressed by linear equations in  $u, v$ , the second type of boundary condition can be written as

$$V_S = \mathcal{B}U_S + \mathcal{C}, \quad (3.4)$$

where  $\mathcal{B}$  is a matrix and  $\mathcal{C}$  is a column vector. Of course, we assume that none of edges of the triangular domain is parallel with the  $v$ -axis. By denoting  $\mathcal{D}^A = \mathcal{A}^{-1}\mathcal{D}$  and  $\mathcal{D}^J = \mathcal{J}\mathcal{D}$ , and separate the matrices and column vectors into blocks with different indices, the discretized system can be rewritten as

$$\mathcal{L}_I U_I + \mathcal{L}_S U_S + \mathcal{L}_F U_F = 0, \quad (3.5)$$

$$\mathcal{L}_I V_I + \mathcal{L}_S V_S + \mathcal{L}_F V_F = 0, \quad (3.6)$$

$$\mathcal{D}_I^A U_I + \mathcal{D}_S^A U_S + \mathcal{D}_F^A U_F + \mathcal{D}_I^J V_I + \mathcal{D}_S^J V_S + \mathcal{D}_F^J V_F = 0. \quad (3.7)$$

By the boundary condition (3.4) and the first two equations, we can eliminate  $V_S, U_I, V_I$  in the third one. This gives us an linear system of equations for  $U_S$ ,

$$\begin{aligned} & (\mathcal{D}_I^A \mathcal{L}_I^{-1} \mathcal{L}_S - \mathcal{D}_S^A + (\mathcal{D}_I^J \mathcal{L}_I^{-1} \mathcal{L}_S - \mathcal{D}_S^J) \mathcal{B}) U_S \\ & = (-\mathcal{D}_I^A \mathcal{L}_I^{-1} \mathcal{L}_F + \mathcal{D}_F^A) U_F + (-\mathcal{D}_I^J \mathcal{L}_I^{-1} \mathcal{L}_F + \mathcal{D}_F^J) V_F \\ & + (-\mathcal{D}_I^J \mathcal{L}_I^{-1} \mathcal{L}_S + \mathcal{D}_S^J) \mathcal{C}. \end{aligned} \quad (3.8)$$

Since the interior points are much more than boundary points, the equations are much more than the unknowns. We can solve  $U_S$  by method of least square. Then we can compute  $V_S$  by (3.4) and solve (3.5) and (3.6) for  $U_I$  and  $V_I$ .

## 4 PIM Discretization

The above description of the method and the equation (2.16) is conceptual and abstract. In order to compute quasi-conformal maps on point cloud, we need to implement the method by proper numerical solvers of PDEs. In this paper we use the point integral method (PIM) to discretize the equation (2.16). PIM is a method for discretizing differential operators. Here we only introduce two aspects that are most relevant to the current problem. The first one is the solution of the Dirichlet problem of Poisson-type elliptic equation. The second one is the computation of partial derivatives. In this part, we assume that the domain  $\mathcal{M}$  is embedded in  $\mathbb{R}^d$ . If  $\mathcal{M}$  is a planar domain,  $d = 2$ . If  $\mathcal{M}$  is on a surface,  $d = 3$ . Without specifying, the indices of summation symbol  $\sum$  goes from 1 to  $d$ .

### 4.1 PIM discretization of Poisson-type equations

PIM uses integrals to approximate derivatives. Consider the Dirichlet problem of the Poisson-type equation like (2.10)

$$\begin{cases} \nabla \cdot (\mathbf{A}^{-1} \cdot \nabla u) = 0, & \mathbf{y} \in \mathcal{M} \\ u = u_b, & \mathbf{y} \in \partial\mathcal{M} \end{cases} \quad (4.1)$$

According to PIM, it can be approximated by an integral equation

$$L_t u(\mathbf{x}) + \frac{2}{\beta} \int_{\partial\mathcal{M}} u(\mathbf{y}) R_t(\mathbf{x}, \mathbf{y}) \, dA_{\mathbf{y}} = \frac{2}{\beta} \int_{\partial\mathcal{M}} u_b(\mathbf{y}) R_t(\mathbf{x}, \mathbf{y}) \, dA_{\mathbf{y}}, \quad (4.2)$$

where  $\beta > 0$  is a small constant, such as  $10^{-4}$ . The integral operator  $L_t$  is defined by

$$L_t u(\mathbf{x}) := \frac{1}{2t} \int_{\mathcal{M}} (u(\mathbf{x}) - u(\mathbf{y})) (2 + M(\mathbf{x}, \mathbf{y})) R_t(\mathbf{x}, \mathbf{y}) \, dV_{\mathbf{y}}$$

where the two functions  $R_t(\mathbf{x}, \mathbf{y})$  and  $M(\mathbf{x}, \mathbf{y})$  are defined for any two points  $\mathbf{x}$  and  $\mathbf{y}$  on  $\mathcal{M}$ . If  $\mathcal{M}$  is a planar domain,

$$R_t(\mathbf{x}, \mathbf{y}) = C_t \exp \left( -\frac{1}{4t} \sum_{i,j} a_{ij}(\mathbf{x}) (y^i - x^i)(y^j - x^j) \right),$$

$$M(\mathbf{x}, \mathbf{y}) = \sum_{j,k,l} D_l a^{lj}(\mathbf{x}) a_{jk}(\mathbf{x}) (y^k - x^k),$$

where the parameter  $t$  is a positive number and  $C_t = (4\pi t)^{-d/2}$ .  $A = (a_{ij})$  is the matrix form of the tensor  $\mathbf{A}$ , and  $(a^{ij})$  its inverse. Since  $\mathbf{A}$  is known (given by the metric  $\mathbf{C}$ ), these two functions can be computed explicitly. If  $\mathcal{M}$  is on a surface, let it be equipped with an orthonormal moving frame  $\{e_1, e_2\}$ . Since the terms  $R_t(\mathbf{x}, \mathbf{y})$  and  $M(\mathbf{x}, \mathbf{y})$  contain  $\mathbf{x} - \mathbf{y}$ , which is computed in  $\mathbb{R}^3$ , we need to extend  $A = (a_{ij})$  to  $\mathbb{R}^3$ . We first extend  $\{e_1, e_2\}$  to  $\{e_1, e_2, e_3\}$ , where  $e_3$  is a smooth unit normal vector field of  $\mathcal{M}$  in  $\mathbb{R}^3$ . The extended coefficient matrix in the extended frame  $\{e_1, e_2, e_3\}$  is

$$\hat{A} = \begin{pmatrix} A & \\ & 1 \end{pmatrix}$$

Transform to the Cartesian coordinates,

$$\bar{A} = (e_1, e_2, e_3) \hat{A} (e_1, e_2, e_3)^T. \quad (4.3)$$

By replacing  $A$  by  $\bar{A} = (\bar{a}_{ij})$  in  $R_t(\mathbf{x}, \mathbf{y})$  and  $M(\mathbf{x}, \mathbf{y})$  defined above, we get the two functions that are needed in the case  $\mathcal{M}$  is on a surface.

To discretize (4.2), let  $\{\mathbf{x}_i\}$  be the points in the point cloud representing  $\mathcal{M}$ . Let  $\{V_i\}$  be a set of area elements associated to the point cloud, and  $\{A_i\}$  a set of length elements of boundary points  $\{\mathbf{s}_i\}$ . Then we define two matrices  $\mathcal{L}_t$  and  $\mathcal{I}^\beta$  by

$$\mathcal{L}_{t,ij} = \begin{cases} -\frac{1}{2t} R_t(\mathbf{x}_i, \mathbf{x}_j) (2 + M(\mathbf{x}_i, \mathbf{x}_j)) V_j, & \text{if } i \neq j \\ -\sum_{i \neq j} \mathcal{L}_{t,ij}, & \text{if } i = j. \end{cases} \quad (4.4)$$

$$\mathcal{I}_{ij}^\beta = \frac{2}{\beta} R_t(\mathbf{x}_i, \mathbf{s}_j) A_j. \quad (4.5)$$

Define  $\bar{\mathcal{L}} = \mathcal{L}_t + \mathcal{I}^\beta$ . Then the Dirichlet problem (4.1) is discretized as

$$\bar{\mathcal{L}} \bar{U} = \mathcal{I}^\beta U_B,$$

where the column vector  $\bar{U} = (u(\mathbf{x}_i))$ , and  $U_B = (u_b(\mathbf{s}_i))$  contains its the boundary components. The inner components of  $\bar{U}$  can be solved from this equation.

## 4.2 PIM discretization of partial derivatives

Let  $f$  be any smooth function on  $\mathcal{M}$ . We want to compute the partial derivative of  $f$  along coordinate  $x^k$  of  $\mathbb{R}^d$ . Let  $\hat{R}_t(\mathbf{x}, \mathbf{y})$  be the normalization of  $R_t(\mathbf{x}, \mathbf{y})$  defined by

$$\hat{R}_t(\mathbf{x}, \mathbf{y}) = R_t(\mathbf{x}, \mathbf{y}) \left/ \int_{\mathcal{M}} R_t(\mathbf{x}, \mathbf{y}) \, dV_{\mathbf{y}} \right.$$

It can be considered as an approximation of the Dirac  $\delta(|\mathbf{y} - \mathbf{x}|)$ . The PIM approximation of  $\partial_k f$  is given by the integral

$$\partial_k f(\mathbf{x}) = \frac{1}{2t} \int_{\mathcal{M}} (f(\mathbf{x}) - f(\mathbf{y})) \sum_l a_{kl}(\mathbf{x})(x^l - y^l) \hat{R}_t(\mathbf{x}, \mathbf{y}) \, dV_{\mathbf{y}}.$$

Note that when  $\mathcal{M}$  is a surface, in this expression of the partial derivative operator  $\partial_k$ , we need to extend the matrix  $A = (a_{ij})$  to  $\bar{A} = (\bar{a}_{ij})$  by (4.3), just as what we do for  $R_t(\mathbf{x}, \mathbf{y})$  and  $M(\mathbf{x}, \mathbf{y})$ .

The partial derivative operator  $\partial_k$  is discretized as a matrix. Suppose  $\mathcal{M}$  is represented by a point cloud as described above.  $F$  is a column vector whose  $i$ -th entry is  $f(\mathbf{x}_i)$ . The partial derivative of  $f$  along coordinate  $x^k$  is discretized as  $\partial_k F$ , where the partial derivative operator  $\partial_k$  is a matrix whose entries are defined by

$$\partial_{k,ij} = \begin{cases} -\frac{1}{2t} \sum_l a_{kl}(\mathbf{x}_i)(x_i^l - x_j^l) \hat{R}_t(\mathbf{x}_i, \mathbf{x}_j) V_j, & \text{if } i \neq j \\ -\sum_{i \neq j} \partial_{k,ij}, & \text{if } i = j. \end{cases} \quad (4.6)$$

## 4.3 Solving linear problems with PIM

In this part, we derive the PIM discretization of the equation (2.16). Two types of operators need to be discretized. The one is the  $\mathcal{T}$  in (2.13), which send boundary value of a Dirichlet problem to its solution. The other is the gradient operator  $\nabla$  of a function, which is used in the Beltrami equation.

Consider  $\mathcal{T}$  first. According to previous introduction to PIM, the PIM discretization of the Dirichlet problem of equation (2.10) takes the form

$$\bar{\mathcal{L}}\bar{U} = \mathcal{I}^\beta U_B, \quad (4.7)$$

in which the column vector  $\bar{U}$  is the discrete solution of the equation. It consists of  $U_I$ ,  $\bar{U}_S$  and  $\bar{U}_F$ . The boundary value  $U_B$  is a column vector consisting of  $U_S$  and  $U_F$ . The meanings of the suffix  $I, B, S, F$  are the same as in previous part.  $I$  stands for inner components,  $B$  stands for boundary components,  $F$  and  $S$  are fixed and unfixed boundary components respectively. We add a bar ‘-’ to some variables because the  $\bar{U}_S$  and  $\bar{U}_F$  solved from the linear system (4.7) are generally different from the given  $U_S$  and  $U_F$ . There are small errors. It doesn’t matter because we only need  $U_I$ . The matrices  $\bar{\mathcal{L}}$  and  $\mathcal{I}^\beta$  are defined by (4.4) and (4.5).

With (4.7) we can express  $\bar{U}$  as

$$\bar{U} = \bar{\mathcal{L}}^{-1} \mathcal{I}^\beta U_B = \bar{\mathcal{L}}^{-1} \mathcal{I}_S^\beta U_S + \bar{\mathcal{L}}^{-1} \mathcal{I}_F^\beta U_F. \quad (4.8)$$

But  $\bar{U}_S$  and  $\bar{U}_F$  will not be used. What we need is only  $U_I$ . Therefore we take

$$U_I = (\bar{\mathcal{L}}^{-1}\mathcal{I}_S^\beta)_I U_S + (\bar{\mathcal{L}}^{-1}\mathcal{I}_F^\beta)_I U_F. \quad (4.9)$$

If we denote  $U_B = (U_S, U_F)^T$ , then (4.9) can be put into a block matrix form

$$U_I = \left( (\bar{\mathcal{L}}^{-1}\mathcal{I}_S^\beta)_I, (\bar{\mathcal{L}}^{-1}\mathcal{I}_F^\beta)_I \right) U_B.$$

This is the PIM discretization of the operator  $\mathcal{T}$  in (2.13) for solving Dirichlet problem of elliptic equation. Similarly, we have

$$V_I = (\bar{\mathcal{L}}^{-1}\mathcal{I}_S^\beta)_I V_S + (\bar{\mathcal{L}}^{-1}\mathcal{I}_F^\beta)_I V_F. \quad (4.10)$$

Now consider the gradient operator  $\nabla$ . It is defined by the discretized partial derivative operators  $\{\partial_k\}_{k=1}^d$  in Euclidean space  $\mathbb{R}^d$ , which are given by (4.6). If  $\mathcal{M}$  is a planar domain,  $\nabla$  is just the Euclidean gradient. It is discretized as  $\mathcal{D} = (\partial_1, \partial_2)$ . If  $\mathcal{M}$  is on a surface,  $\nabla$  is the covariant gradient operator. We need to project the Euclidean gradient  $(\partial_1, \partial_2, \partial_3)$  to the tangent space of  $\mathcal{M}$  at each point to get  $\mathcal{D}$ . Again, let  $\{e_1, e_2\}$  be an orthonormal moving frame on  $\mathcal{M}$ . The two components of  $\mathcal{D}$  are given by matrices

$$\mathcal{D}_i = \sum_{k=1}^d e_a^k \partial_k, \quad a = 1, 2.$$

The gradient operator is just  $\mathcal{D} = (\mathcal{D}_1, \mathcal{D}_2)$  now. With this discretization of the gradient operator  $\mathcal{D}$ , we can express the operator  $\mathcal{D}^A = (\mathcal{D}_1^A, \mathcal{D}_2^A)$  with

$$\mathcal{D}_a^A = a^{ab} \mathcal{D}_b, \quad a = 1, 2,$$

and  $\mathcal{D}^J = (-\mathcal{D}_2, \mathcal{D}_1)$ . Therefore the gradient operator  $\nabla$  and the related operators has been discretized by PIM.

With the above preparation, we can derive the PIM equation for the boundary value problem of Beltrami equation, i.e. the PIM implementation of (3.8). Instead of using (4.7) directly, we use (4.9) in PIM to implement (3.5). Then we plug (4.9) (4.10) and (3.4) into (3.7) to get

$$\begin{aligned} & \left( \mathcal{D}_I^A (\bar{\mathcal{L}}^{-1}\mathcal{I}_S^\beta)_I + \mathcal{D}_S^A + \left( \mathcal{D}_I^J (\bar{\mathcal{L}}^{-1}\mathcal{I}_S^\beta)_I + \mathcal{D}_S^J \right) \mathcal{B} \right) U_S \\ &= - \left( \mathcal{D}_I^A (\bar{\mathcal{L}}^{-1}\mathcal{I}_F^\beta)_I + \mathcal{D}_F^A \right) U_F - \left( \mathcal{D}_I^J (\bar{\mathcal{L}}^{-1}\mathcal{I}_F^\beta)_I + \mathcal{D}_F^J \right) V_F \\ & \quad - \left( \mathcal{D}_I^J (\bar{\mathcal{L}}^{-1}\mathcal{I}_S^\beta)_I + \mathcal{D}_S^J \right) \mathcal{C}. \end{aligned} \quad (4.11)$$

This is the PIM equation for the boundary value problem of Beltrami equation. We solve this equation by method of least square and get  $U_S$ . Finally, use (4.9) (4.10) and (3.4) again to get  $U_I$  and  $V_I$ .

## 5 Nonlinear Boundary

In previous sections we have seen how to find quasiconformal map from surface to a triangle by solving Beltrami equation with piecewise linear boundary condition. In order to find quasiconformal maps to more general target domains, it is necessary to impose nonlinear boundary conditions. Generally, nonlinear problems need to be solved iteratively. There are many possible ways to construct an iteration that can be used to find quasiconformal maps. In this section we describe two of them. The one is based on the variational formulation of the problem. The other is based on the discretized Beltrami equation as described in previous sections.

In this paper, the initialization of the iteration is always chosen as the solution of a linear problem. Given the source domain  $\mathcal{M}$  and the target domain  $\Phi(\mathcal{M})$ , we arbitrarily choose three points  $\{\mathbf{p}_i\}_{i=1}^3$  from  $\partial\mathcal{M}$  and assign their images as three points  $\{(u_i, v_i)\}_{i=1}^3$  on  $\partial\Phi(\mathcal{M})$ , which are also arbitrarily chosen. Using method described in previous sections, the domain  $\mathcal{M}$  is mapped to the triangle determined by  $\{(u_i, v_i)\}_{i=1}^3$  in the target domain. This quasiconformal map is used as the initial solution of the iteration. As iteration goes on,  $\{(u_i, v_i)\}_{i=1}^3$  will remain unchanged, while the images of the other points will be updated.

### 5.1 Variational formulation

There is another approach to characterize the quasiconformal map from one domain to another, which do not use the language of PDE. The Beltrami equation can be easily reformulated as an optimization problem for a functional defined on the domain  $\mathcal{M}$ . The objective functional is a symmetric bilinear form

$$\begin{aligned} F_0(u, v) &= \frac{1}{2} \|\mathbf{A}^{-1} \cdot \nabla u + \mathbf{J} \cdot \nabla v\|_{L^2(\mathcal{M}; \mathbf{A})}^2 \\ &= \frac{1}{2} \int_{\mathcal{M}} (\nabla u \cdot \mathbf{A}^{-1} - \nabla v \cdot \mathbf{J}) \cdot \mathbf{A} \cdot (\mathbf{A}^{-1} \cdot \nabla u + \mathbf{J} \cdot \nabla v) \, dV \\ &= \frac{1}{2} \int_{\mathcal{M}} (\nabla u \cdot \mathbf{A}^{-1} \cdot \nabla u + \nabla v \cdot \mathbf{A}^{-1} \cdot \nabla v) \, dV \\ &\quad - \int_{\mathcal{M}} \nabla u \cdot \mathbf{J} \cdot \nabla v \, dV. \end{aligned}$$

Notice that the critical points of  $F_0$  may not be unique. They correspond to harmonic maps in metric  $\mathbf{A}$ . Among such critical points, only the global minimizer of  $F_0$  with the boundary constraint (2.9) is the desired quasiconformal map. And the global minimum is 0.

The boundary constraint (2.9) can be reformulated as a penalty functional

$$F_1(u, v) = \|f(u_b, v_b)\|_{L^1(\partial\mathcal{M})} = \int_{\partial\mathcal{M}} |f(u, v)| \, dA. \quad (5.1)$$

The extended objective functional is

$$F(u, v) = F_0(u, v) + \lambda F_1(u, v), \quad (5.2)$$

where  $\lambda$  is properly chosen in the computation.

If we can discretize  $F_0$  and the constraint (5.1), it is possible to find the quasiconformal map by directly solving the optimization problem instead of solving nonlinear boundary value problem of Beltrami equation. In the following, we describe the FEM discretization of the functionals and its gradient descent algorithm.

Firstly, let's derive the variational derivatives of  $F_0$ . As in Section 3, suppose the finite element basis functions defined on  $\mathcal{M}$  are  $\{\phi_i\}_{i=1}^N$ . Let  $(U, V)$  be the column vector representing the discretized  $(u, v)$ , that is

$$u = \sum_{j=1}^N U_j \phi_j, \quad v = \sum_{j=1}^N V_j \phi_j.$$

Denote the unit tangent vector of  $\partial\mathcal{M}$  as  $\boldsymbol{\tau}$  and the arc parameter  $\tau$ . It is not difficult to derive that

$$\begin{aligned} \left. \frac{\partial}{\partial \xi} \right|_{\xi=0} F_0(u + \xi \phi_i, v) &= \int_{\mathcal{M}} \nabla u \cdot \mathbf{A}^{-1} \cdot \nabla \phi_i - \nabla v \cdot \mathbf{J} \cdot \nabla \phi_i \, dV \\ &= \int_{\mathcal{M}} \nabla \phi_i \cdot \mathbf{A}^{-1} \cdot \nabla u \, dV - \int_{\partial\mathcal{M}} \phi_i \frac{\partial v}{\partial \tau} \, dA \\ &= \mathcal{L}U - \mathcal{Z}V, \end{aligned}$$

where

$$\mathcal{L}_{ij} = \int_{\mathcal{M}} \nabla \phi_i \cdot \mathbf{A}^{-1} \cdot \nabla \phi_j \, dV, \quad \mathcal{Z}_{ij} = \int_{\partial\mathcal{M}} \phi_i \frac{\partial \phi_j}{\partial \tau} \, dA.$$

Similarly,

$$\left. \frac{\partial}{\partial \xi} \right|_{\xi=0} F_0(u, v + \xi \phi_i) = \mathcal{L}V + \mathcal{Z}U.$$

For  $F_1$ , we can optimize it pointwisely on the boundary. It's straight forward to get

$$\begin{aligned} \frac{\partial}{\partial u} |f(u, v)| &= \text{sign}(f(u, v)) \frac{\partial f(u, v)}{\partial u}, \\ \frac{\partial}{\partial v} |f(u, v)| &= \text{sign}(f(u, v)) \frac{\partial f(u, v)}{\partial v}. \end{aligned}$$

Therefore the gradient descent update rule for the extended objective functional  $F$  is

$$U^{(k+1)} = U^{(k)} - \epsilon (\mathcal{L}U^{(k)} - \mathcal{Z}V^{(k)}), \quad (5.3)$$

$$V^{(k+1)} = V^{(k)} - \epsilon (\mathcal{L}V^{(k)} + \mathcal{Z}U^{(k)}), \quad (5.4)$$

$$U_S^{(k+1)} = U_S^{(k+1)} - \epsilon \lambda \text{sign}(f(U_S^{(k)}, V_S^{(k)})) \frac{\partial f}{\partial u}(U_S^{(k)}, V_S^{(k)}), \quad (5.5)$$

$$V_S^{(k+1)} = V_S^{(k+1)} - \epsilon \lambda \text{sign}(f(U_S^{(k)}, V_S^{(k)})) \frac{\partial f}{\partial v}(U_S^{(k)}, V_S^{(k)}). \quad (5.6)$$

Notice that according to our initialization method, on the boundary only free points  $(U_S, V_S)$  will be updated.

## 5.2 Discrete iteration

The Beltrami equation itself is a linear equation whose solution is also the solution of a pair of Laplace equations. If we can find the right boundary value of  $(u, v)$ , then we can get the solution of Beltrami equation in one step by solving the Dirichlet problems of the Laplace equations with the right boundary value. The nonlinearity exists only in the boundary condition. This fact suggests that we can use gradient descent algorithm to update the boundary condition of  $(u, v)$  and solve a pair of boundary value problems of Laplace equation in each iteration step.

We can not impose the correct boundary condition in one step because we only know where the image boundary curve is, but we don't know where the image points should be on that curve. The gradient descent algorithm can make the image points approach to the boundary curve, but it can not make them move along the curve. PIM is particularly suitable in this situation because it can move image points along the curve automatically. As described by (4.8) in Section 4, when solving a Dirichlet problem, although the boundary value  $U_B$  is given, the PIM still get new values  $\bar{U}$  on all points, including boundary points. In contrast, FEM only update inner points.

The discrete iteration is as follows:

$$U_S^{(k+1)} = U_S^{(k)} - \epsilon\lambda \operatorname{sign}(f(\bar{U}_S^{(k)}, \bar{V}_S^{(k)})) \frac{\partial f}{\partial u}(\bar{U}_S^{(k)}, \bar{V}_S^{(k)}), \quad (5.7)$$

$$V_S^{(k+1)} = V_S^{(k)} - \epsilon\lambda \operatorname{sign}(f(\bar{U}_S^{(k)}, \bar{V}_S^{(k)})) \frac{\partial f}{\partial v}(\bar{U}_S^{(k)}, \bar{V}_S^{(k)}), \quad (5.8)$$

$$\bar{U}^{(k+1)} = \bar{\mathcal{L}}^{-1} \mathcal{I}_S^\beta U_S^{(k+1)} + \bar{\mathcal{L}}^{-1} \mathcal{I}_F^\beta U_F, \quad (5.9)$$

$$\bar{V}^{(k+1)} = \bar{\mathcal{L}}^{-1} \mathcal{I}_S^\beta V_S^{(k+1)} + \bar{\mathcal{L}}^{-1} \mathcal{I}_F^\beta V_F. \quad (5.10)$$

Notice that  $(U_F, V_F)$  remains unchanged since they are the fixed boundary values.

## 6 Numerical Examples

In this section, several examples are presented to show the performance of the PIM algorithm described in previous sections. We first show the convergence of our method for linear problems (the target domains are triangles). The results are compared with the computation by an open package online—SC Toolbox.<sup>2</sup> The two examples indicate that the method converges as the points of discretizations of the domains increase. Then we show some examples of quasiconformal map with nonlinear boundary. Finally we use the quasiconformal map to generate textures on several more complicated surfaces.

To illustrate the domains and maps in the examples, lattice textures are employed. We use relative  $L^2$  error

$$\operatorname{err}(u, v) = \frac{\|(u, v) - (u, v)_{gt}\|_{L^2(\mathcal{M})}}{\|(u, v)_{gt}\|_{L^2(\mathcal{M})}}$$

to compare the results of computations.

<sup>2</sup>See <http://www.math.udel.edu/~driscoll/SC/>

## 6.1 Convergence for linear problems

### 6.1.1 Conformal map on unit disk

Conformal maps on plane are commonly seen in textbooks. The gradients of the two components  $(u, v)$  of such a map are orthogonal to each other. Consider a map  $\Phi$  from the unit disk  $\mathcal{M}$  to a triangle  $\Phi(\mathcal{M})$  whose vertices are  $(1, 0)$ ,  $(0, 1)$  and  $(-1, 0)$ . These vertices are precisely on the unit circle  $\partial\mathcal{M}$ . We assume that  $\Phi$  fixes these points and maps the other boundary points of the disk to the edges of  $\Phi(\mathcal{M})$ . The disk is discretized with 2610, 10191, and 40269 points.

We take the results of SC Toolbox as the ground truth and compute the errors. The textured domain and image of the computed  $\Phi$  are shown in Figure 1. The relative  $L^2$  errors are listed in Table 1. The result of the PIM algorithm converges as the number

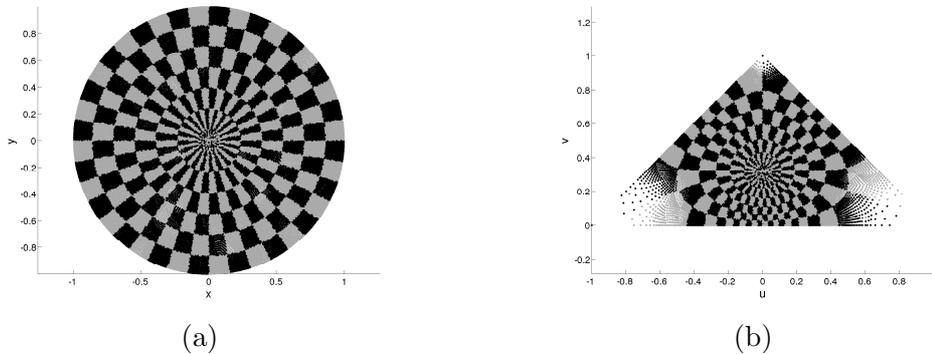


Figure 1: Conformal map from the unit disk to a triangle (40269 points): (a) the unit disk; (b) image of  $\Phi$ .

$ V $	2610	10191	40296
PIM	0.028669	0.018098	0.011850

Table 1: Relative  $L^2$  errors: conformal map from the unit disk to a triangle

of points increases. Even for the discretization with 2610 points, the error is very small. We can see from Figure 1 that  $\Phi$  maps right angles to right angles. This means that the conjugacy of the map is well-preserved by the computation method.

### 6.1.2 Quasi-conformal map from a cap

Our second example is a quasi-conformal map. As argued in Section 2, any quasi-conformal map with a Beltrami coefficient is a conformal map respective to some metric, and vice versa. Thus giving a Beltrami coefficient is equivalent to giving a metric, up to dilation. In this example we will looking for a conformal map with respect to a given metric.

Let  $\mathcal{M}$  be a spherical cap, a part of the unit 2-sphere in  $\mathbb{R}^3$ . The height of the cap is  $1/2$ , thus the cap angle is  $\pi/3$ . Since no coordinates is taken on  $\mathcal{M}$ , we use the coordinates

$(x, y, z)$  of the ambient space  $\mathbb{R}^3$  to represent points on  $\mathcal{M}$ . We choose an orthonormal frame on the cap as

$$e_1 = \frac{(-z, 0, x)}{\sqrt{1-y^2}}, \quad e_2 = \frac{(xy, y^2-1, yz)}{\sqrt{1-y^2}}.$$

See Figure 2(a) for an illustration of the frame and see (b) for the textured domain.

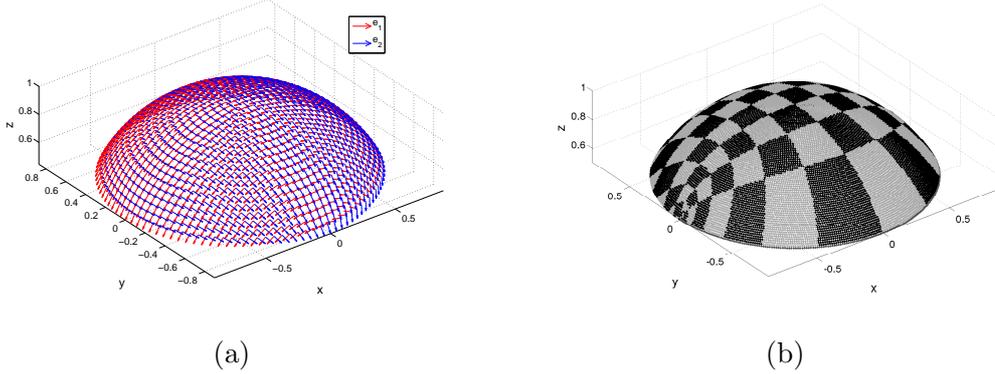


Figure 2: Orthonormal moving frame and texture on the cap (18540 points): (a) Orthonormal frame on the cap; (b) A texture on the cap

We define a metric on  $\mathcal{M}$  by pullback. Let  $\phi = s \circ P$ , in which  $P$  is the normal projection of the cap onto a disk in the  $x-y$  plane and  $s$  is a scaling that maps the image of  $P$  to the unit disk. Pullback by  $\phi$  the standard Euclidean metric of the unit disk to the cap, we get

$$C = \frac{4}{3(1-y^2)} \begin{pmatrix} z^2 & -xyz \\ -xyz & x^2y^2 + (1-y^2)^2 \end{pmatrix}.$$

This is the matrix  $C_{AB}$  of the pullback metric  $\mathbf{C}$  in the orthonormal frame. The matrix that is used directly in solving for quasi-conformal map is

$$A^{-1} = \frac{4}{3(1-y^2)} \begin{pmatrix} x^2y^2 + (1-y^2)^2 & xyz \\ xyz & z^2 \end{pmatrix}.$$

The boundary condition is as follows. Let  $\Phi$  map respectively

$$\left(\frac{\sqrt{3}}{2}, 0, 0\right) \mapsto (1, 0), \quad \left(0, \frac{\sqrt{3}}{2}, 0\right) \mapsto (0, 1), \quad \left(-\frac{\sqrt{3}}{2}, 0, 0\right) \mapsto (-1, 0).$$

We can not use the SC Toolbox to produce a quasi-conformal map directly from a surface to the planar region. It can produce a conformal map  $f$  from the unit disk  $\mathbb{D}$  to the triangle  $\Phi(\mathcal{M})$  which fixed the vertices of the triangle. Therefore  $f^{-1} \circ \Phi$  is a map from  $\mathcal{M}$  to  $\mathbb{D}$ . This is a quasi-conformal map to  $\mathbb{D}$  that has the same Beltrami coefficient as  $\Phi$ , since composing a conformal map does not change Beltrami coefficient. On the other hand, by our definition of the scaled projection  $\phi = s \circ P$  and the pullback metric  $\mathbf{C}$ , it is clear that

$\phi$  is also a quasi-conformal map with the same Beltrami coefficient. By the uniqueness of quasi-conformal map with given Beltrami coefficient and boundary conditions, there must be  $\phi = f^{-1} \circ \Phi$ , or  $\Phi = f \circ \phi$ . Equivalently, the following diagram commutes.

$$\begin{array}{ccc} \mathcal{M} & & \\ \phi \downarrow & \searrow \Phi & \\ \mathbb{D} & \xrightarrow{f} & \Phi(\mathcal{M}) \end{array}$$

Since  $\phi$  is given by exact formula and  $f$  can be computed by SC Toolbox, we can assume  $f \circ \phi$  as the ‘ground truth’.

We compute  $\Phi$  by the PIM algorithm here and compare it with  $f \circ \phi$ . Figure 3(a) is the image of the projection  $\phi$ . Figure 3(b) is the images of  $\Phi$  which are computed by the PIM algorithm. The map from 3(a) to 3(b) is just the conformal map  $f$ . From the textures we can see that the conjugacy of  $f$  is well-preserved by our algorithm. The relative  $L^2$  errors

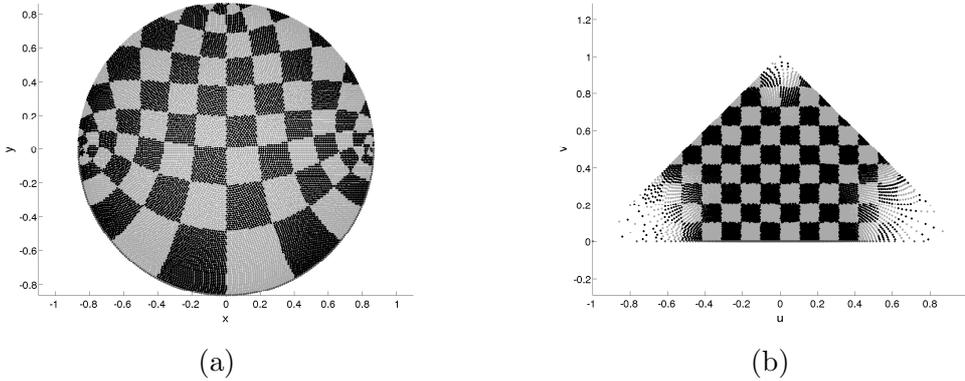


Figure 3: Quasi-conformal map from the cap to a triangle (18540 points): (a) image of the projection  $\phi$ ; (b) image of  $\Phi$ .

are listed in Table 2. It suggests that both implementations of the method converge.

$ V $	1199	4689	18540
PIM	0.079714	0.056181	0.040528

Table 2: Relative  $L^2$  errors: conformal map from cap to a triangle

## 6.2 Nonlinear examples

We first show an trivial example: identity map on the unit disk. This problem is nontrivial for our algorithms because it initially maps the unit disk to a triangle, which is then deformed to the unit disk iteratively. From this example we can see the accuracy of the algorithms. Figure 4 shows the results of two methods. The one is the discrete iteration of PIM on the point cloud, the other is the FEM discretization of variational formulation on

the mesh. From this figure we can see that both results are reasonable, but the distortion near boundary is larger. The relative  $L^2$  error of PIM is 0.077869, while that of FEM is 0.010702.

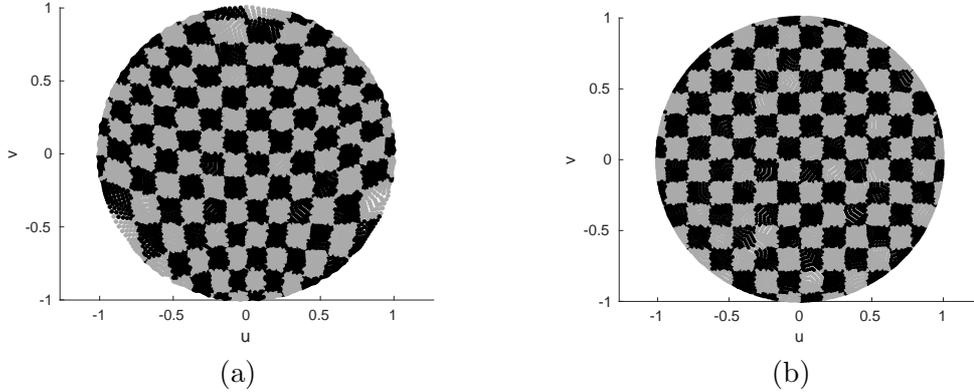


Figure 4: Identity map from the unit disk to itself (10191 points): (a) image by PIM; (b) image by FEM.

Another example is the conformal map from a human face (Alex) to the unit disk on the plain. The reason that we choose conformal map instead of more general quasiconformal map is that the quality of conformal maps are easily seen. For maps whose the exact solutions are unknown, it is not possible to compute  $L^2$  errors of the numerical computations. But from the textures we can check whether right angles are mapped to right angles. If they are, it is reasonable to conclude that the maps are conformal.

See Figure 5 for the results of PIM. The texture on the domain surface is the pull-back of the chessboard texture on the target domain. We can see that the image is indeed an unit disk. The image points are not uniformly distributed. Also, the chessboard lines on the domain surface are nearly orthogonal. But near the boundary the angles are distorted more. Here we only want to show that PIM can be used to find quasiconformal map with nonlinear boundary. But the performance needs further improvements.

### 6.3 More textures

Texture generation is one of applications of quasiconformal maps. Figure 6 show some examples of textures generated by conformal maps from surfaces to planar triangles.

## 7 Conclusion

In this paper we develop and implemente a method of looking for quasi-conformal maps on point clouds. The problem is reduced to solving Beltrami equation and the related elliptic equation with proper boundary condition. Then we apply the point integral method (PIM) to solve the equations numerically. Experiments on some examples on Euclidean plane and surfaces indicate the convergence of the PIM algorithm. Therefore we believe this method to be a reliable tool for solving quasi-conformal maps on point clouds.

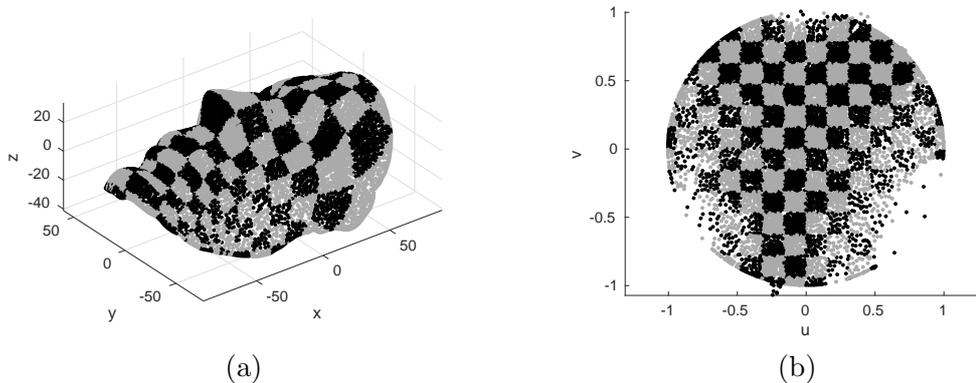


Figure 5: Conformal map from Alex to the unit disk (10597 points): (a) domain with pull-back texture; (b) image.

Here we have only considered quasi-conformal maps defined on simply-connected domains on Euclidean plane and on surfaces. The image of the maps are also confined to triangular regions on plane. In future work, we hope to generalize the method to more general situations.

**Acknowledgments.** This research was supported by NSFC Grants 11371220 and 11671005.

## References

- [1] M. Belkin and P. Niyogi. Towards a theoretical foundation for laplacian-based manifold methods. In *COLT*, pages 486–500, 2005.
- [2] M. Desbrun, M. Meyer, and P. Alliez. Intrinsic parameterizations of surface meshes, 2002.
- [3] G. Dziuk. Finite elements for the beltrami operator on arbitrary surfaces. In S. Hildebrandt and R. Leis, editors, *Partial differential equations and calculus of variations*, volume 1357 of *Lecture Notes in Mathematics*, pages 142–155. Springer, 1988.
- [4] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3d meshes. In *PROCEEDINGS OF THE 2006 SYMPOSIUM ON INTERACTIVE 3D GRAPHICS AND GAMES, MARCH 1417, 2006*, pages 28–29, 2003.
- [5] X. Gu and S.-T. Yau. Global conformal surface parameterization, 2003.
- [6] R. Lai, J. Liang, and H. Zhao. A local mesh method for solving pdes on point clouds. *Inverse Problem and Imaging*, to appear.
- [7] B. Lévy, S. Petitjean, N. Ray, and J. Maillot. Least squares conformal maps for automatic texture atlas generation. *ACM Trans. Graph.*, 21(3):362–371, July 2002.

- [8] Z. Li, Z. Shi, and J. Sun. Point Integral Method for Solving Poisson-type Equations on Manifolds from Point Clouds with Convergence Guarantees. *ArXiv e-prints*, Sept. 2014.
- [9] J. Liang and H. Zhao. Solving partial differential equations on point clouds. *SIAM Journal of Scientific Computing*, 35:1461–1486, 2013.
- [10] J. R. Shewchuk. What is a good linear finite element? - interpolation, conditioning, anisotropy, and quality measures. Technical report, In Proc. of the 11th International Meshing Roundtable, 2002.
- [11] Z. Shi and J. Sun. Convergence of the Laplace-Beltrami Operator from Point Cloud. *ArXiv e-prints*, Mar. 2014.
- [12] W. Zeng, F. Luo, S. T. Yau, and X. D. Gu. Surface quasi-conformal mapping by solving beltrami equations. In *Proceedings of the 13th IMA International Conference on Mathematics of Surfaces XIII*, pages 391–408, Berlin, Heidelberg, 2009. Springer-Verlag.



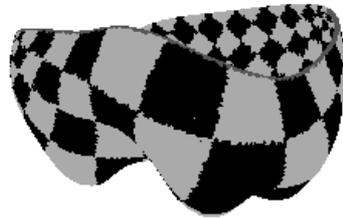
(a)



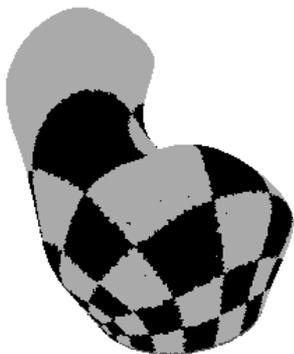
(b)



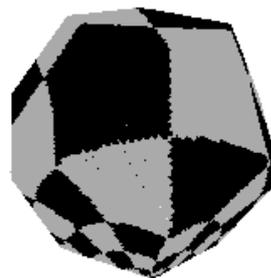
(c)



(d)



(e)



(f)

Figure 6: Conformal map from surfaces: (a) Alex's face; (b) Planck's head; (c) David's head; (d) A tooth; (e) A bone; (f) A dodecahedron;