A Construction of Anisotropic Meshes Based on Quasi-conformal Mapping

Yuxue Ren, Na Lei, Hang Si and Xianfeng David Gu

Abstract We propose a novel method which is able to generate planar anisotropic meshes according to a given metric tensor. It is different from the classical metric-based or high dimensional embedding mesh adaptation methods. Our method resolves the anisotropy of a metric tensor field by finding a corresponding Euclidean metric in the plane. This is achieved via quasi-conformal mapping between two Riemannian surfaces. Given a planar source domain together with a metric tensor defined on it, and a target domain with a Euclidean metric, there exists a quasi-conformal mapping between them, such that the mapping is conformal with respect to the metric tensor on the source and the Euclidean metric on the target. A discrete quasi-conformal mapping can be constructed by solving the Beltrami equation on a Riemannian manifold. Our method first computes the Beltrami coefficient which is a complex-valued function from the given metric tensor. It then uses discrete Yamabe flow to construct this quasi-conformal mapping. We then construct an isotropic triangulation on the target domain. The constructed mesh is mapped back to the source domain by the inverse of the quasi-conformal mapping to obtain an anisotropic mesh of the original domain. This method has solid theoretical foundation. It guarantees

Yuxue Ren
School of Mathematics, Jilin University, Changchun, 130012China,
Beijing Advanced Innovation Center for Imaging Technology, Capital Normal University, Beijing, 100048China, e-mail: snow_ren@foxmail.com

Na Lei
DUT-RU ISE, Dalian University of Technology, Dalian, 116620 China,
Beijing Advanced Innovation Center for Imaging Technology, Capital Normal University, Beijing, 100048China, e-mail: nalei@dlut.edu.cn

Hang Si
Weierstrass Institute for Applied Analysis and Stochastics, Mohrenstr. 39, 10117 Berlin, Germany
e-mail: hang.si@wias-berlin.de

Xianfeng David Gu
Department of Computer Science, Stony Brook University, Stony Brook, NY USA e-mail: gu@cs.stonybrook.edu

the correctness for all symmetric positive definite metric tensors. We show experimental results on function interpolation problems to illustrate both of the features and limitations of this method.

1 Introduction

Many physical problems exhibit anisotropic features, i.e., their solutions change more significantly in one direction than others. Examples include in particular convection-dominated problems whose solutions have, e.g., layers, shocks, or corner and edge singularities. Anisotropic meshes have great importance in numerical methods to solve partial differential equations. They improve the accuracy of the solution and decrease the computational cost.

Anisotropy denotes the way distances and angles are distorted. It is naturally related to approximation theory and is important in function interpolation [21, 22, 4, 11]. For example, it has been shown that for a smooth function the anisotropy is best characterized by the Hessian of that function. In practice, a central question is how to efficiently distinguish the anisotropy of a given problem. Another important question is how to characterize the anisotropy in a such a way that an optimal mesh for a given problem can be defined. These are all difficult questions and are active research subjects.

It is well-understood that anisotropic features can be represented by a metric tenser $\mathscr{M}$ defined on the target space $\Omega \subset \mathbb{R}^d$, where the metric tenser of each vertex is a $d \times d$ symmetric positive definite matrix. $\mathscr{M}$ defines a Riemannian metric on Ω, both lengths and angles can be re-defined according to this metric. This allows the use of classical isotropic mesh adaptation techniques to produce anisotropic meshes. It is one of the major approaches for producing anisotropic meshes, see [1, 3, 23, 16, 12, 9, 17, 18, 20]. Although these methods are very successful in practice, there is no theoretical proof that the generated anisotropic meshes are appropriate or good according to the input metric.

Variational mesh adaptation is another useful technique to generate adapted meshes. It is based on the optimisation of a mesh related functional to achieve the best adapted meshes. Such methods are centroidal Voronoi tessellations (CVTs) [10, 15, 24], optimal Delaunay triangulations (ODTs) [5], and monitor functions [13]. Many of these methods are generalised to produce anisotropic meshes by incorporating a metric tensor into the functional. Again, there is no theoretical guarantees on the success of these methods with arbitrary anisotropic metric tensors.

A recent anisotropic meshing technique is through higher dimensional embedding, [2, 14, 26, 6]. Instead of using metric tensors, it increases the dimensions to resolves the anisotropy such that it can be treated isotropic in this high dimensional space. The co-dimensions can be flexibly chosen to emphasis the interested quantities. By using the normal component of the surface, this approach can produce curvature-adapted anisotropic surface meshes [14, 7]. By using the gradient of a function, this approach produces well adapted meshes to interpolate anisotropic functions [8]. However, it is not clear how to choose the co-dimensional coordinates for a metric tensor.

In this paper, we propose a novel method to construct anisotropic meshes in the plane. Assume an anisotropic metric tensor M is given on a planar domain Ω, our goal is to construct an anisotropic mesh with respect to M on Ω. Our method is based on the theory of quasi-conformal mapping. The construction of an anisotropic mesh is achieved by building a quasi-conformal mapping φ from Ω to D, where D is a target domain with Euclidean metric. The Beltrami coefficient of the mapping μ_φ is determined by the metric tensor M. Then an isotropic mesh is calculated on D, and pulled back to Ω by φ. This results an anisotropic mesh on Ω. The quasi-conformal mapping φ is achieved by solving the Beltrami equation using the discrete Yamabe flow method. If the metric tensor is symmetric and positive definite, this method guarantees the success of finding a quasi-conformal mapping.

We tested our method using an application of interpolation of anisotropic functions. Our experiments on some published examples showed that this method is able to effectively reduce the interpolation error (measured in L^2 norm) compared with the uniform meshes. Furthermore, the error is consistently reduced with respect to the increase of number of points.

We conducted preliminary comparisons of our results with the results produced by two public codes. The first one is BAMG [1], which is a metric-based anisotropic mesh generator. Another is Detri2, developed by the third author [2]. It implements the high dimensional embedding mesh adaptation method [8]. First of all, it is noted that both codes produced high quality anisotropic meshes. Their L^2 interpolation errors are about two orders of magnitude smaller than ours. This shows a strong limitation of our method. Our method has a limitation on the input point set. It does not as flexible as the classical mesh adaptation methods. On the other hand, our method could be seen as an effective anisotropic mesh smoothing step compared with the smoothing algorithms used in BAMG and Detri2.

The structure of this paper is as follows. Section 2 explains that given a field of metric tensor, how to compute an quasi conformal mapping. Section 3 introduces the theorem of discrete Yamabe flow. All the algorithms can be found in Section 4. Section 5 shows the results of experiments.

2 Quasi conformal mapping

A quasi conformal mapping is a homeomorphism between plane domains which to first order takes small ellipses of bounded eccentricity to small circles, see Figure 1 for an example.

Suppose (S_1, g_1) and (S_2, g_2) are Riemannian surfaces, $\{(U_\alpha, z_\alpha)\}, \{(V_\beta, w_\beta)\}$ are their atlas which are compatible with their Riemannian metric g_1, g_2 respectively,

[1] available in FreeFEM++ (http://www.freefem.org)

[2] http://www.wias-berlin.de/people/si/detri2.html

then on each local chart (U_α, z_α), $g_1 = e^{2\lambda_1} dz_\alpha d\bar{z}_\alpha$, on each local chart (V_β, w_β), $g_2 = e^{2\lambda_2} dw_\beta d\bar{w}_\beta$. Let $f : (S_1, g_1) \to (S_2, g_2)$ be a diffeomorphism, for each vertex $v \in (S_1, g_1)$, locally f maps a chart $(U, z(v))$ to a chart $(f(U), w(f(v)))$, where U is a neighbourhood of v, then the pull back metric of f near v can be defined as

$$f^* g_2 = e^{2\lambda_2} |w_z dz + w_{\bar{z}} d\bar{z}|^2 = e^{2\lambda_2} |w_z|^2 |dz + \mu d\bar{z}|^2 \tag{1}$$

where $w_z = \frac{\partial w}{\partial z}, w_{\bar{z}} = \frac{\partial w}{\partial \bar{z}}$, $\mu = \frac{w_{\bar{z}}}{w_z}$. f is in fact an isometric mapping when it defined on the manifold $(S_1, f^* g_2)$. So for any metric $g_1' = e^{2\lambda_3} |dz + \mu d\bar{z}|^2$ shares the same conformal structure with $f^* g_2$, f is a conformal mapping from (S_1, g_1') to (S_2, g_2), where $\lambda_3 : S_1 \to \mathbb{R}$ is a scalar function defined on S_1.

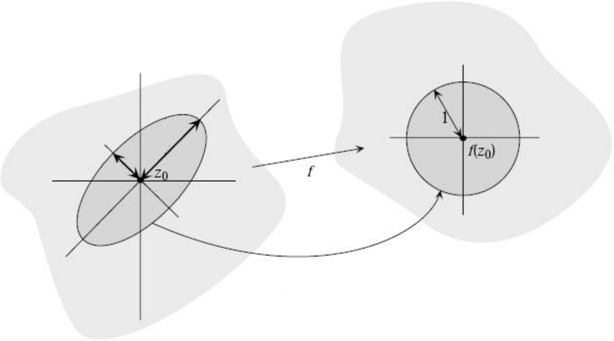


Fig. 1 A quasi-conformal mapping transforms an ellipse into a circle with a bounded eccentricity.

When μ is bounded, f is called a quasi conformal mapping. Quasi conformal mapping is a generalization of conformal mapping, it is a solution to the Beltrami equation

$$\frac{\partial f}{\partial \bar{z}} = \mu(z) \frac{\partial f}{\partial z}, |\mu(z)| \leq k. \tag{2}$$

where $\mu(z)$ is called the complex dilatation Beltrami coefficients, describes the distortion of f .

Suppose S_1 be a simply connected domain in $\mathbb{C}$, ∂S_1 has more than one point, $g_1 = dzd\bar{z}$. If $f_0 : (S_1, g_1) \to (S_1, g_1')$ is a quasi conformal mapping, then $f \circ f_0^{-1} : (S_1, g_1') \to (S_2, g_2)$ is a conformal mapping. This provides us a novel way to convert the problem of producing anisotropic mesh to that of producing isotropic mesh via changing the metric. Let $\mathscr{T}$ be a isotropic triangulation of S_1 under metric g_1', then $\mathscr{T}$ will become an anisotropic mesh if we change the metric to be g_1, and it is obvious that the Beltrami coefficients of f_0 convey the anisotropic feature of $\mathscr{T}$.

Most of the classical methods focus on the construction of isotropic mesh on (S_1, g_1'), while we provide a new idea, that is, the computation of the quasi conformal mapping f_0, the theorem of discrete Yamabe flow guarantees the existence and uniqueness of f_0. Since $f \circ f_0^{-1}$ is a conformal mapping, f, f_0 shares the same

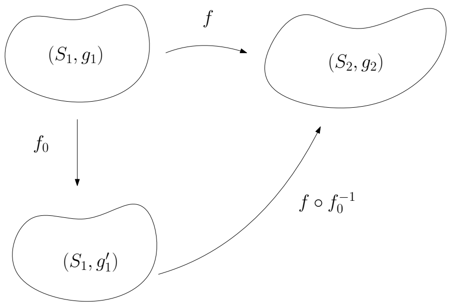


Fig. 2 Convert Quasi-conformal mapping problem to conformal mapping problem.

Beltrami coefficients, so we can compute f_0 by solving the Beltrami equation with f's Beltrami coefficients.

2.1 Computation of Jacobian matrix based on Metric Tensors

In our problem, S_1 is a simply connected domain in $\mathbb{C}$, the anisotropic feature is represented by a field of metric tensors $\mathscr{M} := \{M(v); v \in S_1\}$, in which $M(v)$s are 2×2 symmetric positive definite matrixes for all $v \in S_1$. Let g_3 be the Riemannian metric defined by $\mathscr{M}$, then for an open curve $c \subset (S_1, g_3)$, the length of c is $\|c\|_{g_3} = \int_{t=0}^{1} \sqrt{v(t)^T M(c(t)) v(t)}$, in which $v(t) = \frac{dc(t)}{dt}$. Let $z = x + iy$ be a parameterization of (S_1, g_1), such that $g_1 = dz d\bar{z}$, define $f_0(v) = v, \forall v \in S_1$, then the pull back metric of f_0 can be described as

$$f_0^* g_3 = (dx, dy) M(z) \begin{pmatrix} dx \\ dy \end{pmatrix}.$$

Let $\phi(z) = w$ is the homeomorphism onto (S_1, g_1) such that $f_0 \circ \phi^{-1}$ is conformal, denote $w = u + iv$, then we have

$$\begin{pmatrix} du \\ dv \end{pmatrix} = J(\phi(z)) \begin{pmatrix} dx \\ dy \end{pmatrix},$$

in which $J(\phi)$ is the Jacobi matrix of ϕ. Then the pull back metric of $f_0 \circ \phi^{-1}$ with respect to w is

$$(f_0 \circ \phi^{-1})^* g_3 = (du, dv)(J(\phi(z))^{-1})^T M(z) J(\phi(z))^{-1} \begin{pmatrix} du \\ dv \end{pmatrix}.$$

When $f_0 \circ \phi^{-1}$ is conformal, the pull back metric satisfies $(f_0 \circ \phi^{-1})^* g_3 = c dw d\bar{w}$, so $\mathscr{M}$, ϕ satisfies

$$M(z) = cJ(\phi(z))^T J(\phi(z)),$$

in which c is a positive constant.

Since $M(z)$ is a symmetric positive definite matrix for all z, there always exists a matrix N satisfying $M(z) = N(z)^T N(z)$, $N(z)$ can be computed through $M(z)$'s orthogonal decomposition $M(z) = P(z)^T \Lambda(z) P(z)$. $\Lambda(z)$ is a diagonal matrix, its diagonal elements $\lambda_1 > \lambda_2 > 0$ are characteristic values of $M(z)$, P is a 2×2 orthogonal matrix, it can be denoted as

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \tag{3}$$

which is a rotation matrix with degree clockwise θ. Its row vectors of $P(z)$ are characteristic vectors of $M(z)$. let $N(z) = \Lambda^{1/2}(z)P(z)$, then $M(z) = N(z)^T N(z)$. So we have $J(\phi(z)) = cN(z)$. $\phi(z)$'s Beltrami coefficients can be computed based on $N(z)$.

2.2 Compute Beltrami Coefficients Based on Jacobian matrix

Actually, ϕ maps an infinitesimal circle to an infinitesimal ellipse, whose long and short axis' lengthes are characteristic values of $J(\phi(z))$, and the direction of them are $J(\phi(z))$'s characteristic vectors. We know

$$\begin{aligned} J(\phi(z)) &= c\Lambda^{1/2}P \\ &= c\begin{pmatrix} \sqrt{\lambda_1} & 0 \\ 0 & \sqrt{\lambda_2} \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \\ &= c\begin{pmatrix} \sqrt{\lambda_1}\cos\theta & -\sqrt{\lambda_1}\sin\theta \\ \sqrt{\lambda_2}\sin\theta & \sqrt{\lambda_2}\cos\theta \end{pmatrix} \end{aligned} \tag{4}$$

so we have $\frac{\partial u}{\partial x} = c\sqrt{\lambda_1}\cos\theta$, $\frac{\partial u}{\partial y} = -c\sqrt{\lambda_1}\sin\theta$, $\frac{\partial v}{\partial x} = c\sqrt{\lambda_2}\sin\theta$, $\frac{\partial v}{\partial y} = c\sqrt{\lambda_2}\cos\theta$.

Denote $\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}$, $\frac{\partial v}{\partial x}, \frac{\partial v}{\partial y}$ as u_x, u_y, v_x, v_y. Without loss of generality, assume $\lambda_1 > \lambda_2$ Then

$$dw = w_z dz + w_{\bar{z}} d\bar{z} = (\frac{u_x + v_y}{2} + i\frac{v_x - u_y}{2})dz + (\frac{u_x - v_y}{2} + i\frac{v_x + u_y}{2})d\bar{z}. \tag{5}$$

Therefor,

$$w_z = \frac{u_x + v_y}{2} + i\frac{v_x - u_y}{2}, w_{\bar{z}} = \frac{u_x - v_y}{2} + i\frac{v_x + u_y}{2}. \tag{6}$$

Then the Beltrami coefficient is

$$\mu = \frac{w_{\bar{z}}}{w_z} = \frac{\sqrt{\lambda_1} - \sqrt{\lambda_2}}{\sqrt{\lambda_1} + \sqrt{\lambda_2}}(\cos 2\theta - i\sin 2\theta). \tag{7}$$

We can see that the modulus of μ is decided by the characteristic values of $J(\phi)$ everywhere, and the angle of μ is two times of the intersection angle of longer axis with $x-$axis, while μ have no connection with c. On the other hand, consider ϕ's Jacobian matrix, we have

$$det(J(\phi)) = c^2\sqrt{\lambda_1\lambda_2}cos^2\theta + c^2\sqrt{\lambda_1\lambda_2}sin^2\theta = c^2\sqrt{\lambda_1\lambda_2} > 0,$$

so ϕ is local homeomorphism, furthermore, according to Theorem 3 of [**?**], ϕ is a global homeomorphism when it maps the boundary ∂S_1 of S_1 homeomorphically onto itself, in Section4, we will discuss that the output of our algorithm satisfies this condition. On the other hand, μ describes the feature of the infinitesimal ellipse, while it cannot describe long and short axis' lengthes. In fact, μ decides the torsion of the mapping, we can control the extent of torsion by controlling the modula of μ, which decides the extent of anisotropic feature. In the following, meshes with different extent of anisotropic are shown.

After computing the Beltrami coefficients, discrete surface Yamabe flow can be used to solve the Beltrami equation to compute the quasi conformal mapping corresponding to Beltrami coefficients.

3 Discrete Surface Yamabe Flow

Yamabe flow is a powerful tool to design Riemannian metric according to prescribed curvature. In this process the conformal structure of the manifold is preserved. In our problem, we can use it to compute an Euclidean metric with the given conformal structure. The Euclidean metric is in fact our target metric, then a new embedding of a mesh can be computed and it forms the image of the quasi conformal mapping we compute.

Let $\mathscr{T}$ be a triangular mesh embedded in $\mathbb{C}$, let e_{ij} be an edge of $\mathscr{T}$, $l_{ij}^{(0)}$ be the initial length. The discrete conformal factor is a function defined on the set of vertices $u : V \to \mathbb{R}$. During the process of Yamabe flow, the length of e_{ij} is defined as

$$l_{i,j} = e^{u(v_i)+u(v_j)} l_{ij}^{(0)}. \tag{8}$$

Let $K_i = K(v_i)$ be the discrete Gaussian curvature of v_i, then $K(v_i) = 2\pi - \sum_{[v_i,v_j,v_k]\in F} \theta_i^{jk}$ if v_i is an interior vertex of $\mathscr{T}$, $K(v_i) = \pi - \sum_{[v_i,v_j,v_k]\in F} \theta_i^{jk}$ if v_i is a boundary vertex, where θ_i^{jk} is a corner angle in the face $[v_i, v_j, v_k]$ at the vertex v_i.

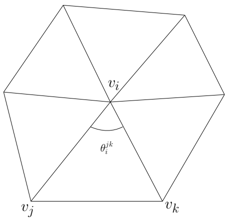


Fig. 3 Geometric interpretation of Gaussian curvature.

It is easy to prove the following discrete Gauss-Bonnet Theorem.

Theorem 1. *Suppose $\mathscr{T}$ is a triangular mesh with a discrete metric, then*

$$\sum_{v\in T} K(v) = 2\pi\chi(\mathscr{T}), \tag{9}$$

where $\chi(\mathscr{T})$ is the Euler characteristic of the mesh, $\chi(\mathscr{T}) = |V| - |E| + |F|$, $|V|$, $|E|$, $|F|$ are the numbers of vertices, edges and faces, respectively.

Yamabe flow is the process to update the conformal factor u according to Gaussian curvature K,

$$\frac{du(t)}{dt} = 2(\bar{K} - K(t)), \tag{10}$$

where t is time parameter, $\bar{K}$ is the prescribed curvature which must satisfy Gauss-Bonnet theorem. For our purpose of generating mesh on planar domain, $\bar{K}(v_i) = 0$ for all interior vertices of $\mathscr{T}$. While when v_i is a boundary vertex, its target curvatures are decided by the shape of S_1. For example, if S_1 is a circle domain, the target curvatures on all the boundary vertices are defined as $\frac{2\pi}{m}$, where m is the number of boundary vertices. While when S_1 is an orthogon, there are four vertices on the boundary whose target curvatures are $\pi/2$ and other boundary vertices' target curvatures are zero. It is easy to check that for both cases the total target curvatures are 2π, satisfying Gauss-Bonnet theorem.

The convergence of Yamabe flow has been proven in [19]. In fact, the solution of Yamabe flow can be seen as an extremal point of a convex energy – Yamabe energy. Let $\mathbf{u} = (u_1, u_2, \cdots, u_n)$ be the conformal factor, define a differential 1-form $\omega = \sum_{i=0}^{n}(\bar{K}_i - K_i)du_i$, the differential of ω is

$$d\omega = \sum_{i,j=0}^{n}\left(\frac{\partial K_i}{\partial u_j} - \frac{\partial K_j}{\partial u_i}\right)du_i \wedge du_j, \tag{11}$$

It is easy to verify that

$$\frac{\partial K_i}{\partial u_j} = \frac{\partial K_j}{\partial u_i}, \tag{12}$$

so $d\omega = 0$, ω is a closed 1-form. The Yamabe energy is defined as

$$E(\mathbf{u}) = \int_{\mathbf{u_0}}^{\mathbf{u}} \sum_{i=1}^{n}(\bar{K}_i - K_i)du_i, \tag{13}$$

$E(\mathbf{u})$ is well defined and convex, so it has an extremal point, we can use Newton's method to solve it. More details of these results can be found in [25].

It is easy to compute the Hessian matrix of $E(\mathbf{u})$. Let e_{ij} be an interior edge of $\mathscr{T}$, $\mathbf{f}_{ijk}$ and $\mathbf{f}_{jil}$ are two faces which are adjacent to the edge, then we can define the weight of e_{ij} as follows

$$\mathbf{w}_{ij} = \cot\theta_k^{ij} + \cot\theta_l^{ij} \tag{14}$$

If e_{ij} is on the boundary of $\mathscr{T}$, there is only one face adjacent to it, the weight is

$$\mathbf{w}_{ij} = \cot\theta_k^{ij}. \tag{15}$$

Then the element on the $i-$th row and $j-$th column of the Hessian matrix of $E(\mathbf{u})$ is

$$\frac{\partial^2 E(\mathbf{u})}{\partial u_i \partial u_j} = -\frac{\partial K_i}{\partial u_j} = \begin{cases} \mathbf{w}_{ij} & i \neq j \\ -\sum_k \mathbf{w}_{ik} & i = j. \end{cases} \tag{16}$$

It has been proven by [19] that the Hessian matrix is positive on the linear subspace $\{\mathbf{u} | \sum_{i=1}^{n} u_i = 0\}$. [25] has proved that the admissible metric space for $\mathscr{T}$ with fixed connectivity is not convex, so during the process of Newton's method, the connectivity of $\mathscr{T}$ should be transformed if necessary to ensure that $\mathbf{u}$ is in the admissible metric space during each step. The method that is usually used is edge swap. For an edge e_{ij}, if $\theta_k^{ij} + \theta_l^{ij} > \pi$, we swap it and denote it as e_{kl}, demonstrated in Fig 4.

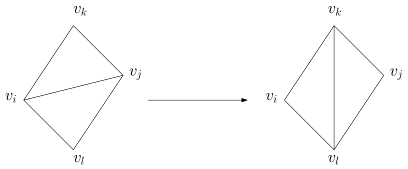


Fig. 4 Edge swap.

4 Algorithm

The problem we consider here is that given a complex domain S_1 and metric tensor $\mathscr{M}$ defined on S_1, to construct an anisotropic mesh with respect to the given metric tensor. The main idea is to convert the construction of an anisotropic mesh to the construction of isotropic mesh through computing a quasi conformal mapping $\phi : (S_1, g_1) \to (S_1, g_3)$. The image of the isotropic mesh for the quasi conformal mapping is an anisotropic mesh with respect to the given metric tensor.

Step one Compute the Beltrami coefficients based on the given metric tensor.(Algorithm 1)

Step two Solve Beltrami equation by Yamabe flow method to compute the quasi conformal mapping based on its Beltrami coefficients.(Algorithm 2)

Step three Construct an isotropic mesh, compute its image for the quasi conformal mapping. (Algorithm 3).

4.1 Compute Beltrami coefficients based on metric tensor

Given a background triangulation $\mathscr{T}$ on the complex domain (S_1, g_1), discrete metric tensor $\mathscr{M} = \{M(v); v \in V\}$ is a matrix-valued function defined on the vertices set of $\mathscr{T}$. According to the description of Section 2, since $M(v)$ is a positive symmetric matrix, there is a matrix $N(v)$ such that $M(v) = N(v)^T N(v)$ which can be computed based on $M(v)$'s orthogonal decomposition. Let ϕ be the quasi conformal mapping which maps infinitesimal ellipse $(x,y)M(v)\begin{pmatrix} x \\ y \end{pmatrix} = \varepsilon$ to infinitesimal circle, as the description of Section 2, $N(v)$ is the Jacobian matrix of the quasi conformal mapping ϕ on v, so the Beltrami coefficients of ϕ on v can be computed according to $N(v)$. More details have been provided in Section 2.

Algorithm 1 Computation of Beltrami Coefficients

```
Require: A triangular mesh $\mathscr{T}$, metric tensor $\mathscr{M}$.
Ensure: Beltrami coefficients of a quasi conformal mapping with respect to $\mathscr{M}$.
function BELTRAMI COEFFICIENTS($\mathscr{T}$, $\mathscr{M}$)
    for Each vertex $v \in V$ do
        compute the orthogonal decomposition of $M(v)$.
        Compute the Jacobian matrix of the quasi conformal mapping $J(\phi(v))$.
        Compute Beltrami coefficients $\mu(v)$ of $\phi$.
    end for
return $\mu$.
end function
```

4.2 The Algorithm of Computing Quasi Conformal Mapping According to Its Beltrami Coefficients

Suppose $\phi_0 : \mathscr{T} \to \mathbb{C}$ be an embedding of $\mathscr{T}$. Let $e_{ij} = [v_i, v_j]$ be an edge of $\mathscr{T}$, denote $z_i = \phi_0(v_i), z_j = \phi_0(v_j)$, then define the initial length of e_{ij} in the very beginning of Yamabe Flow as follows

$$l_{ij}^{(0)} := |e_{ij}|_{g_3} = |(z_j - z_i) + s \cdot \frac{\mu(v_i) + \mu(v_j)}{2} \cdot (\bar{z}_j - \bar{z}_i)|. \tag{17}$$

In the formula, $|z|$ is the modulus of a complex number z. This kind of edge length might not correspond to an embedding to the complex space. It changes the conformal structure in the discretion view, we call them the anisotropic lengthes. We add a parameter s to describe the extent of the anisotropic property. Then we use Yamabe Flow to computer a proper value of e^{λ_3} to calculate a new metric, denoted as $|e_{ij}|_{g_4} = e^{\lambda(v_i)+\lambda(v_j)}|e_{ij}|_{g_3}, \forall i, j$, which satisfies the condition that for all interior vertices v of $\mathscr{T}$, $K(v) = 0$, curvatures of vertices on the boundary conform the shape of S_1.

We know that if the lengths of $\mathscr{T}$'s edges are given, all the vertices' discrete gaussian curvature can be calculated according to the cosine formula and discrete Gaussian curvature formula. We denote the Gaussian curvature calculated by $|e|_{g_3}$ as K_0, put K_0 as the initial Gaussian curvature. Put $\bar{K}$ as the target Gaussian curvature, which corresponds to an embedding of $\mathscr{T}$, then $\bar{K}(v) = 0$ for all interior vertices v of $\mathscr{T}$. The shape of S_1 decides $\bar{K}(v)$ for boundary vertices of $\mathscr{T}$.

There is an important essential that should be paid attention to. During the process of Yamabe flow, the connectivity of $\mathscr{T}$ should be transformed if necessary at each step. During the judgement at edge swap, define the length of e_{ij} as $e^{\lambda_t(v_i)+\lambda_t(v_j)}|e|_{g_3}$, in which λ_t is the conformal factor at time t. While when we transform the connectivity of $\mathscr{T}$, all new edges' length in g_3 should be real-time computed. Suppose that there is an new edge e_{kl} connecting v_k, v_l, then

$$|e_{kl}|_{g_3} = |(z_l - z_k) + s \cdot \frac{\mu(v_k) + \mu(v_l)}{2} \cdot (\bar{z}_l - \bar{z}_k)|. \tag{18}$$

so its length at time t is $l_{kl} = e^{\lambda_t(v_k)+\lambda_t(v_l)}|e_{kl}|_{g_3}$.

Our last step is to compute an embedding $\phi_1 : \mathscr{T} \to \mathbb{C}$ of $\mathscr{T}$ according to the metric $|e_{ij}|_{g_4} = e^{\lambda(v_i)+\lambda(v_j)}|e_{ij}|_{g_3}$. In the following we simplify $|e_{ij}|_{g_4}$ to d_{ij}. There are a lot of methods to compute $\mathscr{T}$'s embedding . For example, the problem can be translated to the process of minimizing an energy, or, it can be solved by a branch of computation of intersection points of two circles, see Fig 5. In our case, we choose the second algorithm. Since our algorithm guarantees that all the triangles of $\mathscr{T}$ are non-degenerate, the lengths of edges satisfy $d_{ij} < d_{ik} + d_{jk}$, so at each step of computation, those two circles must have two intersections.

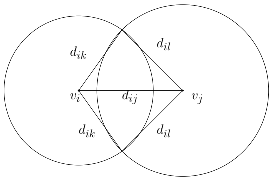


Fig. 5 The intersection points of two circles.

The choice of solutions decides the direction of f_{ijk}'s normal vector. If we regard f_{ijk} to be a face in $\mathbb{R}^3$, its normal vector is either $(0,0,1)$ or $(0,0,-1)$. During the process of this algorithm, the directions of all the faces' normal vector must be the same.

Algorithm 2 Yamabe Flow Algorithm

```
Require: A mesh $\mathscr{T}$ with embedding $\phi_0$, Beltrami Coefficients $\mu$, parameter $s$.
Ensure: $\mathscr{T}$'s new embedding $\phi_1$.
function YAMABE FLOW($\mathscr{T}$, $\mu$, $s$)
    for $e_{ij} \in E$ do
        Compute the initial length of $e_{ij}$
            $$l_{ij}^{(0)} := |(z_j - z_i) + s \cdot \frac{\mu(v_i) + \mu(v_j)}{2} \cdot (\bar{z}_j - \bar{z}_i)|.$$
    end for
    for $v_i \in V$ do
        Initialize conformal factor $\gamma_i := 1.0$. Compute the target curvature $\bar{K}_i$, initial curvature$K_i$.
    end for
    while $max|K_i - \bar{K}_i| > \varepsilon$ do
        for Edge $e_{ij} = [v_i, v_j] \in E$ do
            Compute edge length $l_{ij} := l_{ij}^{(0)} \cdot \gamma_i \cdot \gamma_j$.
        end for
        Edge swap.
        for $v_i \in V$ do
            Compute the Gaussian curvature of $v_i$.
        end for
        for Edge $e_{i,j} \in E$ do
            Compute the edge weight $w_{ij}$ to form the Hessian matrix $\Delta$.
        end for
        $d\gamma = e^{\Delta^{-1}(\bar{K} - K)}$. Normalize $d\gamma$ such that $\Pi d\gamma_i = 1.0$. $\gamma := \gamma \cdot d\gamma$.
    end while
    Compute a new embedding $\phi_1 : V \to \mathbb{C}$ according to $\{l_{ij}; i, j <= |V|\}$.
return $\phi_1$.
end function
```

4.3 Construct Anisotropic Mesh Based On Quasi Conformal Mapping

The discretion of the quasi conformal mapping ϕ we compute is a piecewise linear mapping, it can be represented as two coordinates of $\mathscr{T}$, locally on each cell **f** of $\mathscr{T}$, ϕ is a linear mapping which can be decided by the image of **f**'s vertices. So given a triangulation T, the image of T can be decided by the image of all the vertices of T. Then the problem is converted to the problem of computing the image of a given vertex. Firstly, one should find out which cell $\mathbf{f} \in \mathscr{T}$ includes the given vertex, then compute the image according to the images of vertices of **f**.

Algorithm 3 Quasi conformal mapping

```
Require: A background mesh $\mathscr{T}$ with two embeddings $\phi_0, \phi_1$, a triangular mesh $T$ with an embedding $\psi_0$.
Ensure: The image of the quasi conformal mapping $\psi_1$.
function QUASI CONFORMAL MAPPING($\mathscr{T}$, $\phi_0$, $\phi_1$, $T$)
    for Each vertex $v \in V$ do
        Find out which domain $\phi_0(\mathbf{f})$, $\mathbf{f} \in \mathscr{T}$ includes $\psi_0(v)$.
        Compute $\psi_1(v)$ in the domain $\phi_0(\mathbf{f})$.
    end for
return $\psi_1$.
end function
```

5 Experiments and Comparisons

In this section, we will show experimental results of our method on interpolation of anisotropic functions as well as comparison of our method with other methods implemented in publicly available mesh adaptation codes.

5.1 A Metric Tensor from the Gradient

Most of the methods consider to choose the Hessian matrix or recovery Hessian matrix of the input function to be metric tensor. [8] provides a novel anisotropic mesh adaptation technique based on higher dimensional embedding which contains the information of the function f itself, its gradient and Hessian information. In this section, we describe an anisotropic mesh creation method based on the information with respect to the gradient of the interpolated function.

Let S_1 be a domain in $\mathbb{C}$, define Riemannian metric $g_1 = dzd\bar{z}$ on S_1. Let $f : S_1 \to \mathbb{R}$ be a real-valued C^1 continuous function, then the set $\{(x,y,f(x,y))|(x,y) \in S_1\}$ forms a surface embedded in $\mathbb{R}^3$, denoted as S_2, actually S_2 is a Riemannian surface with Euclidean metric of $\mathbb{R}^3$, denoted as g_2. Define a mapping $\phi : (S_1,g_1) \to (S_2,g_2)$ as

$$\phi(x,y) = (x,y,f(x,y)),$$

obviously ϕ is a diffeomorphism. Let $p := (x_0,y_0,f(x_0,y_0)) \in (S_2,g_2)$, we put an infinitesimal $\varepsilon-$circle on (x_0,y_0), denoted as $\gamma = \{(x_0+\varepsilon\cos\theta, y_0+\varepsilon\sin\theta)|0 \le \theta < 2\pi\}$, then its image is

$$\phi(\gamma) := \{(x_0+\varepsilon\cos\theta, y_0+\varepsilon\sin\theta, f(x_0+\varepsilon\cos\theta, y_0+\varepsilon\sin\theta))|0 \le \theta < 2\pi\}. \quad (19)$$

Since f is C^1, we have

$$f(x_0+\varepsilon\cos\theta, y_0+\varepsilon\sin\theta) = f(x_0,y_0) + f_x\varepsilon\cos\theta + f_y\varepsilon\sin\theta + O(\varepsilon^2) \quad (20)$$

Where $f_x := \frac{\partial f}{\partial x}(x_0,y_0), f_y := \frac{\partial f}{\partial y}(x_0,y_0)$.

Assume that q is a point in $\phi(\gamma)$, $q = (x_0+\varepsilon\cos\theta_0, y_0+\varepsilon\sin\theta_0, f(x_0+\varepsilon\cos\theta_0, y_0+\varepsilon\sin\theta_0))$, then

$$\begin{aligned}&\|q-p\|^2 = \varepsilon^2\cos^2\theta_0 + \varepsilon^2\sin^2\theta_0 + (f_x\varepsilon\cos\theta + f_y\varepsilon\sin\theta)^2 + O(\varepsilon^2)\\ &= (f_x^2+1)\varepsilon^2\cos^2\theta_0 + 2f_xf_y\varepsilon^2 cos\theta_0\sin\theta_0 + (f_y^2+1)\varepsilon^2\sin^2\theta_0 + O(\varepsilon^2)\\ &= (\varepsilon\cos\theta_0, \varepsilon\sin\theta_0)\begin{pmatrix} f_x^2+1 & f_xf_y \\ f_xf_y & f_y^2+1\end{pmatrix}\begin{pmatrix}\varepsilon\cos\theta_0 \\ \varepsilon\sin\theta_0\end{pmatrix} + O(\varepsilon^2),\end{aligned} \quad (21)$$

thus $\phi(\gamma)$ is approximately contained by a tiny ellipsoid, its projection to the tangent plane of (S_2,g_2) at p is an infinitesimal ellipse with centre p. It is obvious that, the image of this tiny ellipse

$$\{(x,y)|(x-x_0,y-y_0)\begin{pmatrix} f_x^2+1 & f_xf_y \\ f_xf_y & f_y^2+1 \end{pmatrix}^{-1}\begin{pmatrix} x-x_0 \\ y-y_0 \end{pmatrix}=\varepsilon^2\}$$

is a circle. So as it described above, if there is a quasi conformal mapping ϕ_0 defined on S_1 which maps the tiny ellipse above to a tiny circle, $\phi\circ\phi_0^{-1}$ is a conformal mapping. So we choose the metric tensor $\mathscr{M}=\{M(z)|z\in S_1\}$ in S_1 to construct the quasi conformal mapping, in which

$$M(z)=\begin{pmatrix} f_x^2+1 & f_xf_y \\ f_xf_y & f_y^2+1 \end{pmatrix}^{-1}. \tag{22}$$

$\mathscr{M}(z)$ induces an new metric on S_1, denoted as g_3. In fact, the direction of long axis of γ on the surface is the direction in which the function changes fastest and the direction of short axis is the direction in which f changes slowest, it means that they are the principal direction of (S_2,g_2).

Figure 6 shows the ellipses in the domain $[-1,1]\times[-1,1]$ with respect to $\mathscr{M}(z)$ of the Gaussian function $f(x,y)=4e^{-(x^2+y^2)/2}$.

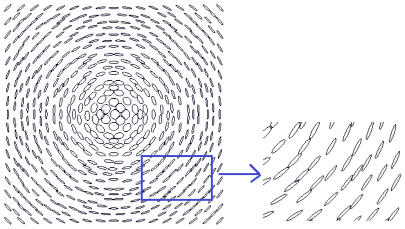


Fig. 6 Ellipse of function $f(x,y)=4e^{-(x^2+y^2)/2}$.

5.2 Experiments on function interpolation

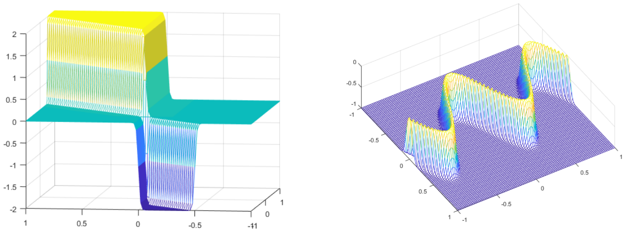
In this section, we show experiments on function interpolation using our method. We experimented two functions, the first function is found in the paper [8], which are:

$$\begin{aligned} f_1(x,y) &= \tanh(60x)-\tanh(60(x-y)-30), \\ f_2(x,y) &= \tanh(-100(y-0.4\sin(2\pi x))^2). \end{aligned}$$

Fig7 shows the functions we are going to interpolate.

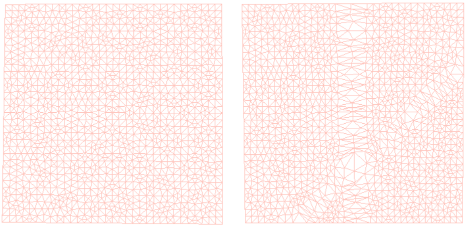
Figure8 shows two embeddings of a triangulation with 1404 vertices, which is the background mesh to describe the quasi conformal mappings with respect to f_1.

Figure9 shows the anisotropic mesh of f_1 with different number of vertices and their pre-image of the quasi conformal mapping.



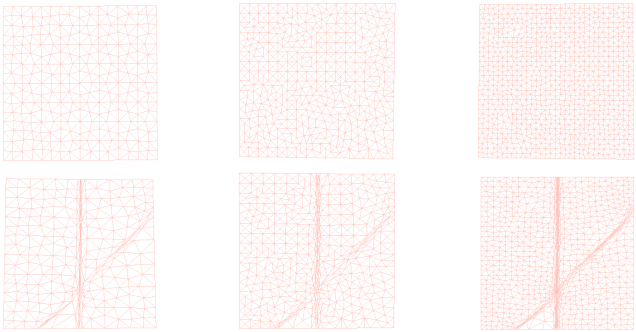
f_1 f_2

Fig. 7 Two functions



isotropic embedding anisotropic embedding

Fig. 8 Two embeddings of background mesh of f_1



meshes with 300 vertices meshes with 500 vertices meshes with 1000 vertices

Fig. 9 Anisotropic mesh of f_1

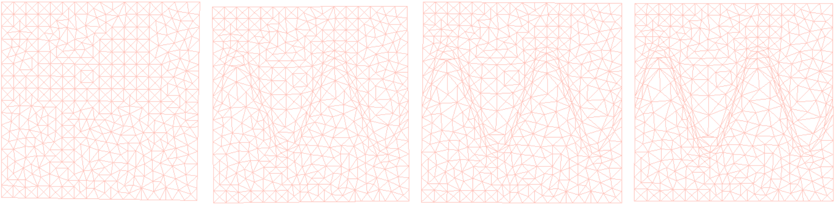
Table1 reports the L^2-errors of the interpolation functions corresponding to these anisotropic meshes.

We know that during the process of construction of quasi conformal mapping, we add a parameter s to control the extent of ”anisotropic”, the next figure shows the background mesh of quasi conformal mappings with different parameters with respect to f_2. Figure10 shows the anisotropic mesh of f_2 with different parameters.

Table2 reports the L^2-errors of the interpolation functions corresponding to these anisotropic meshes.

$\lvert V\rvert$	Isotropy	Anisotropy
300	0.434039	0.169213
500	0.3414	0.152131
1000	0.26307	0.0925087

Table 1 The L^2-errors of $f_1(x,y) = \tanh(60x) - \tanh(60(x-y)-30)$ on different meshes.



$s = 0.0$ $s = 0.6$ $s = 0.8$ $s = 1.0$

Fig. 10 Anisotropic mesh of f_2 with different parameters.

s	0.0	0.2	0.4	0.6	0.8	1.0
Error	0.250927	0.205359	0.179804	0.160007	0.130494	0.142274

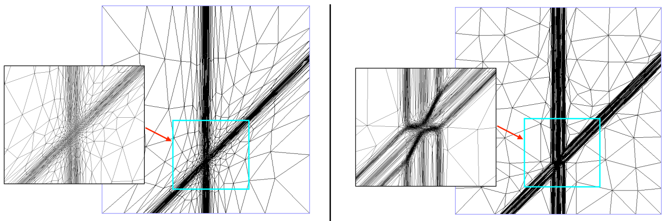
Table 2 The L^2-errors of $f_2(x,y) = \tanh(-100(y-0.4\sin(2\pi x))^2)$ on different meshes.

From these experiments, we observe that our method is able to capture the anisotropic features of the interpolated functions. The resulting meshes improved the accuracy of the interpolation compared with uniform meshes (those meshes with $s = 0$). Moreover, the interpolation error consistently decreases according to the increase of the number of points.

5.3 *Comparisons with other methods*

In this section, we conducted preliminary comparisons of our results with the results produced by two public codes, one is `BAMG` which implements the classical metric-based methods. Another is `Detri2` which implements the high dimensional embedding method [8]. First of all, it is noted that both codes produced high quality anisotropic meshes, with a much smaller interpolation error which is about two orders of magnitude smaller than those errors of our results, see Figure 11. This shows the limitation of our method compared to mesh adaptation methods. It is necessary to add/remove vertices, while our method does not change the number of vertices.

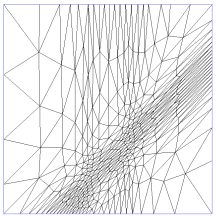
On the other hand, our method could be seen as an effective anisotropic mesh smoothing step compared with the heuristic smoothing algorithms used in `BAMG` and `Detri2`. Figure 12 reports the $L2$-error when only using `BAMG`'s smoothing option. We did this experiment by using the `adaptmesh()` function provided in `FreeFEM++`, and we only call this function once, with different iterations of s-



BAMG (884 points, *L*2-error: 0.0040) | *Detri*2 (2346 points, *L*2-error: 0.0044)

Fig. 11 Anisotropic mesh of f_1 generated by BAMG (left) and Detri2 (right).

moothing, `nbsmooth`=*xxx*, where *xxx* is the given number of smoothing iterations. We used the parameter `nbvx`=1500 to set a limit of number of points.



`BAMG` (341 points)

`nbsmooth`	L^2 error
5	0.510075
10	0.51022
20	0.510206
50	0.509879
100	0.509742

Fig. 12 Left: The adapted mesh generated by `BAMG` with only one iteration. Right: The report of L^2 error on meshes produced by different iterations of smoothing.

From this experiment, we could observe that the mesh smoothing algorithm (which is a heuristic relaxation method) used in *BAMG* has no obvious effect on the resulting meshes. In this case, our method could be used to improve there anisotropic smoothing algorithm in the mesh adaptation process. This could be an interesting future work.

References

1. Frank J Bossen and Paul S Heckbert. A pliant method for anisotropic mesh generation. In *5th Intl. Meshing Roundtable*, pages 63–74. Citeseer, 1996.
2. Guillermo D. Canas and Steven J. Gortler. Surface remeshing in arbitrary codimensions. *Vis. Comput.*, 22(9):885–895, September 2006.
3. MJ Castro-Diaz, F Hecht, B Mohammadi, and O Pironneau. Anisotropic unstructured mesh adaption for flow simulations. *International Journal for Numerical Methods in Fluids*, 25(4):475–491, 1997.
4. L. Chen, P. Sun, and J. Xu. Optimal anisotropic meshes for minimizing interpolation errors in l^p-norm. *Mathematics of Computation*, 76:179–204, 2007.
5. Long Chen and Jin-Chao Xu. Optimal Delaunay triangulations. *Journal of Computational Mathematics*, 22(2):299–308, 2004.

6. Franco Dassi, Andrea Mola, and Hang Si. Curvature-adapted remeshing of CAD surfaces. *Procedia Engineering*, page 13, 2014.
7. Franco Dassi and Hang Si. A curvature-adapted anisotropic surface re-meshing method. In Simona Perotto and Luca Formaggia, editors, *New Challenges in Grid Generation and Adaptivity for Scientific Computing*, volume 5, pages 19–41. Springer International Publishing.
8. Franco Dassi, Hang Si, Simona Perotto, and Timo Streckenbach. Anisotropic finite element mesh adaptation via higher dimensional embedding. *Procedia Engineering*, 124:265–277, 2015.
9. Cécile Dobrzynski and Pascal Frey. Anisotropic delaunay mesh adaptation for unsteady simulations. In *Proceedings of the 17th international Meshing Roundtable*, pages 177–194. Springer, 2008.
10. Qiang Du and Desheng Wang. Anisotropic centroidal voronoi tessellations and their applications. *SIAM Journal on Scientific Computing*, 26(3):737–761, 2005.
11. Luca Formaggia and Simona Perotto. New anisotropic a priori error estimates. *Numerische Mathematik*, 89(4):641–667, 2001.
12. Pascal-Jean Frey and Frédéric Alauzet. Anisotropic mesh adaptation for cfd computations. *Computer methods in applied mechanics and engineering*, 194(48-49):5068–5082, 2005.
13. Weizhang Huang. Mathematical principles of anisotropic mesh adaptation. In *Commun. Comput. Phys*. Citeseer, 2006.
14. B. Lévy and N. Bonneel. Variational anisotropic surface meshing with voronoi parallel linear enumeration. In *Proc. 21st International Meshing Roundtable*, pages 349–366, 2012.
15. Yang Liu, Wenping Wang, Bruno Lvy, Feng Sun, Dong-Ming Yan, Lin Lu, and Chenglei Yang. On centroidal voronoi tessellationłenergy smoothness and fast computation. *ACM Transactions on Graphics*, 28(4):1–17, August 2009.
16. Rainald Löhner and Juan Cebral. Generation of non-isotropic unstructured grids via directional enrichment. *International Journal for Numerical Methods in Engineering*, 49(1-2):219–232, 2000.
17. Adrien Loseille and Frédéric Alauzet. Continuous mesh framework part I: well-posed continuous interpolation error. *SIAM J. Numer. Anal.*, 49(1):38–60, 2011.
18. Adrien Loseille and Frédéric Alauzet. Continuous mesh framework part II: validations and applications. *SIAM J. Numer. Anal.*, 49(1):61–86, 2011.
19. Feng Luo. Combinatorial yamabe flow on surfaces. *Communications in Contemporary Mathematics*, 6(05):765–780, 2004.
20. David Marcum and Frdric Alauzet. Aligned metric-based anisotropic solution adaptive mesh generation. *Procedia Engineering*, 82:428–444, 2014.
21. Shmuel Rippa. Long and thin triangles can be good for linear interpolation. *SIAM Journal on Numerical Analysis*, 29(1):257–270, 1992.
22. Jonathan R. Shewchuk. What is a good linear element? interpolation, conditioning, and quality measures. In *Proceedings of 11th International Meshing Roundtable*, pages 115–126, Ithaca, New York, September 2002. Sandia National Laboratories.
23. Kenji Shimada, Atsushi Yamada, Takayuki Itoh, et al. Anisotropic triangular meshing of parametric surfaces via close packing of ellipsoidal bubbles. In *6th International Meshing Roundtable*, pages 375–390, 1997.
24. Dong-Ming Yan, Kai Wang, Bruno Levy, and Laurent Alonso. Computing 2d Periodic Centroidal Voronoi Tessellation. pages 177–184. IEEE, June 2011.
25. Xiaotian Yin, Miao Jin, Feng Luo, and Xianfeng David Gu. Discrete curvature flows for surfaces and 3-manifolds. In *Emerging Trends in Visual Computing*, pages 38–74. Springer, 2009.
26. Zichun Zhong, Xiaohu Guo, Wenping Wang, Bruno Lvy, Feng Sun, Yang Liu, and Weihua Mao. Particle-based anisotropic surface meshing. *ACM Transactions on Graphics*, 32(4):1, July 2013.