Biting Ellipses to Generate Anisotropic Mesh [*]

Xiang-Yang Li   Shang-Hua Teng   Alper Üngör

Abstract. *In numerical simulation where the underlying function is strongly directional, it is desirable to use a mesh that is adaptive both in size and in shape. In such simulation, a metric tensor is used to quantify the ideal size and direction locally at each point in the domain, which in turn defines the local stretching and size of the triangles or quadrilaterals of the mesh. Given a metric tensor, the anisotropic meshing problem is to construct a good quality mesh satisfying the metric tensor. We present a new anisotropic meshing method which is called the* ellipse biting method. *Our algorithm uses the framework of advancing front to generate a close to optimal packing of ellipses. We then use the ρ-Delaunay triangulation of the vertex set to generate the final mesh. Because it generates an ellipse packing that respects the underlying control spacing, this new method produce a high quality mesh whose element size and directionality conform well locally to the given input. As part of this work, we introduces a set of operations including scaling, intersection, and union on tensor metrics. Then operations are used to formally define distance among metrics and to extend Lipschitz condition and the notion of well-shaped meshes from isotropic metrics to anisotropic metrics.*

Keywords. anisotropic mesh generation, metrics, biting, advancing front, sphere packing.

1 Introduction

An essential step in numerical simulation of physical and engineering problems is to find a proper discretization of a continuous domain. This is the problem of *mesh generation.* For problems with complex geometry boundaries and with solutions that change rapidly in both magnitude and direction, we need to use an *unstructured anisotropic mesh* with a varying local topology, spacing, and directionality in order to reduce the problem size and improve the accuracy. A good unstructured anisotropic meshing algorithm uses elements of properly chosen size, shape, and direction that is adaptive to the complex geometry and solution accuracy. In doing so, it generates meshes that are numerically sound and that are also as small as possible.

Several anisotropic meshing methods and heuristics have been developed, implemented, and applied to various applications such as the fluid flow and shock simulation. These methods can be classified into two main families.

The first [1, 3, 4, 9, 13, 17, 18] generates the mesh directly from the control spacing function defined in the domain. Most of these algorithms assume that the control spacing is already "good" enough so that any "well-shaped" mesh conforming to the control spacing are numerically sound.

There are three different approaches to generate the anisotropic mesh directly from the control spacing. The first approach uses advancing front. It selects the new points and connects the new elements layer by layer so that the newly selected elements are well-shaped and respect the control spacing. The second approach uses the particle simulation: initially, it samples a set of particles: ellipse or cells, then moves these particles and adds or deletes some particles until they are stabilized on some predefined energy function. The last approach divides the input domain into some subdomains such that points in each subdomain have similar directionality

[*] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801. Supported in part by a NSF CAREER award (CCR-9502540), an Alfred P. Sloan Research Fellowship, and the U.S. Department of Energy through the University of California under Subcontract number B341494 (DOE ASCI). The emails of authors: xli2@cs.uiuc.edu, steng@cs.uiuc.edu, ungor@uiuc.edu.

and/or size control. By rescaling, each subdomain can be mapped a new subdomain so that the directionality control of the new subdomain is very close to be isotropic. The standard isotropic meshing methods are used for each new subdomain and resulting mesh is mapped back to form a mesh for the original subdomain.

The second family [2, 11] of the anisotropic meshing methods allows modification to mesh connectivity. These methods comprise various discrete operators such as refinement, coarsening, and vertex movement to convert an initial mesh into a final well-shaped mesh. These methods defines a quality measure of the triangle or quadrilateral element based on the control spacing. If a element of the mesh is not good based on the quality measure, then they either add new points, or remove original points, or relocate the points, or change the connectivity of the current mesh to improve the quality locally.

Over the years, several anisotropic meshing methods such as those based on advancing front and Delaunay like triangulations have become popular due to their effectiveness in practical applications. However, these methods do not come with equal strengths. For example, advancing front [14] uses simple data structures and is efficient and relatively easy to implement. It offers a high quality of point placement strategy for the early fronts and the integrity of the boundary. Unfortunately, when fronts meet each other or itself, it is difficulty and time consuming to decide the size and the directionality of the elements in that region. Our objective is to develop a new meshing algorithm that utilizes the strengths of advancing front while prevents the difficulty when two fronts meet.

We show that the advancing front method can be used to efficiently construct a hight quality ellipse-packing. At a high level, this new advancing front based packing algorithm first finds an ellipse packing of the boundary of the domain and then grows the packing towards the interior of the domain. Each time when a new ellipse is added to the interior, a larger protection ellipse is removed (bitten away) from the domain. By doing this, it builds the ellipse packing by adding ellipses one at a time, or a layer at a time, in the same spirit as the standard advancing method; our new method uses advancing front to construct an ellipse packing instead of the mesh elements themselves. We show that this advancing front based method does generate a high quality anisotropic mesh. We will refer this new method as the *ellipse biting method* and show that it can be made as practical as the standard advancing front meshing methods.

The rest of paper is organized as follows. Section 2 reviews some existing anisotropic mesh generation methods. Section 3 discusses the control spacing which specifies the element sizes and directionality, and point densities of a mesh. Section 4 defines some basic notations and operations on tensor metrics. These operations include summation, intersection, and union, and are used to define distance among metrics, the interpolation of the metrics, and the ρ-Delaunay criterion for the construction of the final mesh. Our ellipse biting method is presented in Section 5. We conclude our paper in section 6 with the discussion of future work.

2 Previous Results

In this section, we review some related anisotropic mesh generation methods. One motivation of our new method is to retain the advantages of these previous meshing algorithms.

2.1 Advancing Front Method

Advancing front methods construct a mesh of a domain by moving a front from its boundary towards its interior. It first generates an initial front typically by constructing a surface mesh for the boundary of the domain. It then creates new elements one at a time or a layer at a time and updates the front with these created faces [7, 8, 14]: In the one element-at-a-time model, it chooses a face of the current front and introduces a new mesh element with it as the base face. It can use another vertex on the front or insert a new Steiner point in the interior as the additional vertex of the new element. The base face and potentially some other faces on the front (if the additional vertex is an existing one) are removed from the front, and some faces of the new element are added to the front. This process is repeated until the front is empty, i.e., all fronts have merged upon each other and the domain is fully meshed.

The method involves the simultaneous generation of field points and their connectivity. The selection of the base face and the placement of the new mesh vertex are the two key steps of any advancing front method. These two steps must ensure that the new mesh element is valid, "well-shaped", and respects to the anisotropic spacing control. In addition, they should keep the front in good condition to allow the creation of graceful future elements. Hence, once the base face is chosen, we need to decide where to place the new vertex. Recall that for each base face, we can only place the Steiner point in a particular region near the base face so that the new element conforms to the anisotropic spacing control. Call this region the *feasible region.* For most advancing front based anisotropic meshing algorithms, the size of the elements in the mesh is determined by the spacing of the points on the advancing boundary as it propagates.

In Section 5, we show how to use advancing front methods to help Delaunay like anisotropic mesh generation. In particular, we present a method to construct a high quality ellipse-packing using the advancing front methods.

2.2 Particle Simulation

A particle system [18] is a collection of particles that moves over time according to either a deterministic or a stochastic set of rules or equation of motion. Bossen and Heckbert [4], Shimada *et. al.* [18] applied interacting particle to $2D$ anisotropic and parametric surface meshing respectively. They use the metric tensor to specify an anisotropy in the domain, and generate the anisotropic point distribution using a proximity-based force.

The particle system used to generate the mesh is similar to that used in computer graphics in the sense that discrete bodies interact in $2D$ or $3D$ space as a result of the application of pairwise, repulsive/attractive forces [18]. There are several characteristics that make this method particularly suitable for the mesh generation. Particles are packed in the order of the dimension, i.e., vertices first and then edges, faces, and volumes. A quick initial guess at the final particle configuration is obtained by using hierarchical spatial subdivision [18]. The particle size and direction are adjusted individually by the control spacing function. A population control mechanism is used during the relaxation to remove any superfluous particle that is largely overlapped by its neighbors, and to subdivide any lone particle missing some neighbors, so that a given domain is filled with an appropriate number of particles. Shimada *et. al.* [18] define the force for two particles centered at $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ based on the current distance $l = ||\boldsymbol{x}_i - \boldsymbol{x}_j||$, the target stable distance l_0, and the corresponding linear spring constant k_0 at the target distance. The target distance l_0 is calculated as the sum of the two lengths l_{ij} and l_{ji}, measured along the line segment that connects the centers $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, from the center to the boundary of each particles.

Then the force for particles [18] is defined as

$$f(w) = \begin{cases} \frac{k_0}{l_0}(1.25w^3 - 2.375w^2 + 1.125) & 0 \leq w \leq 1.5 \\ 0 & 1.5 \leq w \end{cases}$$

Note that here the particles can be the spheres or cells, if an isotropic mesh is preferred, and it can be ellipses or rectangles if an anisotropic mesh is preferred.

Although these rules of forces and interactions are heuristically defined, the particle simulation methods produce quite good quality meshes in practice.

2.3 Refine or Smooth the Mesh

Another approach to generate a conforming anisotropic mesh is to refine or smooth a previously generated mesh according to some quality measures of its elements. There are various quality measures for a triangle element.

Let V_K, P_K be the area and perimeter of triangle K. Let h^* be the desired mesh size at the location of triangle K, and $h_K = P_K/3$. Buscagli and Dari [5] define the quality Q_k of K as

$$20.78 V_K / P_K^2 F(\frac{h_K}{h^*}),$$

where $F(x) = m(x)^\beta(2 - m(x)^\beta)$, and $m(x) = min(x, 1/x)$, β is a parametric constant. Note that here $20.78 \simeq 12\sqrt{3}$ makes sure that the equilateral triangle will have quality measure 1. For anisotropic mesh, the distance and the area are computed according to a metric stored at the vertices of a fixed background mesh [5]. If the quality of an element is not good, the following operations may be applied in the local neighborhood to improve it: adding new Steiner points at the center of edges, swapping diagonal edges; collapsing internal edges, or moving vertices around.

Castro Diaz *et al* [6] use another approach by just considering the length of the edges in mesh. Let d_i be the length of edge a_i computed with the metric tensor M. Let l_{max} and l_{min} be the acceptable maximal and minimal edge length, respectively. They choose [6] $l_{max} \simeq 1.4$ and $l_{min} \simeq 0.6$. If $d_i > l_{max}$, then edge a_i is split into two edges; if $d_i < l_{min}$, then a_i is contracted by merging two end points of a_i. The above process is repeated until no action taken for any edges in the mesh. The Laplacian like smoothing is then applied to improve the mesh.

Zhang and Trépanier [21] apply different approach by just smoothing the mesh vertices to improve the quality of the mesh. A triangle K is first transformed to K' by using the local objective ellipse. The quality of K is that of K' computed under the isotropic case. The objective is to move the mesh vertex to minimize the total potential energy defined by a spring system.

3 Control Spacing

Each domain Ω and a differential equation u defines a desired local spacing within a domain to specify, for example, the expected element size in a given neighborhood (or point densities near a point) and the element directionality near a point. In this section, we discuss how to determine the local spacing from the geometry of Ω and the numerical condition of u for generating the anisotropic mesh.

3.1 Geometry Condition

The geometry of the boundary of Ω contributes to the local spacing of a high quality anisotropic mesh. In two dimensions, we assume that Ω is given as a *planar-straight-line graph* (PSLG), which is a collection of line segments and points in the plane, closed under intersection. Suppose Ω is described by a PSLG S. Ruppert [16] introduced the concept called *local feature size* $\boldsymbol{lfs}(x)$ to capture the geometry condition. Ruppert has observed that $\boldsymbol{lfs}$ changes slowly within the domain. Formally, a function $f()$ is *α-Lipschitz* if for any two points $\boldsymbol{x}, \boldsymbol{y}$ in the domain, $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq \alpha||\boldsymbol{x} - \boldsymbol{y}||$. The Lipschitz coefficiency of $\boldsymbol{lfs}$ is bounded from above by 1 [16].

There are several ways to describe the spacing function from previous mesh $\mathcal{M}$ over a domain Ω: **Edge-length function,** el_M; **Nearest-neighbor function,** nn_M. See Li *et. al.* [12].

3.2 Numerical Condition

The numerical condition is usually obtained from an *a priori* error analysis, or an *a posteriori* error analysis based on an initial numerical simulation. It defines *numerical spacing functions* for each point $\boldsymbol{x}$ in the domain Ω by a metric tensor $M(\boldsymbol{x})$. The tensor quantifies the desired element size and the stretching of the triangle or quadrilateral in the mesh near a point $\boldsymbol{x}$. The matrix $M(\boldsymbol{x})$ is symmetric, and all of its eigenvalues are positive. If the numerical spacing is considered locally as a constant, then the unit ball is the ellipse $(\boldsymbol{y} - \boldsymbol{x})^T M(\boldsymbol{y} - \boldsymbol{x}) = 1$. In other words, the numerical spacing function specified an ellipse for every point $\boldsymbol{x}$ in the domain. It specifies the edge length requirement along every direction.

Locally at point $\boldsymbol{x}$, function u can be approximated by a quadratic function

$$u(\boldsymbol{x} + d\boldsymbol{x}) = u(\boldsymbol{x}) + d\boldsymbol{x} \bigtriangleup u(\boldsymbol{x}) + \frac{1}{2}(d\boldsymbol{x}Hd\boldsymbol{x}^T),$$

where H is the *Hessian matrix* of u, the matrix of the second partial derivatives. Suppose that continuous piecewise linear approximation is applied to u, then the maximal interpolation error depends on H. The matrix $M(\boldsymbol{x})$, from the interpolation viewpoint, is determined by the eigenvalues and the corresponding eigenvectors of the Hessian matrix of u [20]. In other words, in two dimensions, let $c = \cos(\phi)$, $s = \sin(\phi)$, the metric M has formula

$$\begin{pmatrix} c & -s \\ s & c \end{pmatrix} \begin{pmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{pmatrix} \begin{pmatrix} c & s \\ -s & c \end{pmatrix}$$

where γ_1 and γ_2 are the magnitude values of the eigenvalue of the Hessian matrix; ϕ is the rotation angle of the ellipse. The major radius and the minor radius of the corresponding ellipse defined by M is $r_1 = 1/\sqrt{\gamma_1}$, and $r_2 = 1/\sqrt{\gamma_2}$. See Figure 1 (a).

Note that if all of the eigenvalues are equal, then the ellipse becomes a circle. Hence the anisotropic numerical spacing becomes isotropic. Then we can use the circle biting method recently developed by Li *et. al.* [12] to generate a well shaped and well conformed mesh efficiently.

For later convenience, we use the following notations: $M(r_1, r_2, \theta)$ denotes the metric M at a point; $R(\theta) = \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$ denotes the rotation matrix with angle θ; $\Lambda = \begin{pmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{pmatrix}$ denotes the diagonal matrix formed by the eigenvalues of M. We also use its elements (m_{ij}) to denote M. Let $\tilde{M} = M^{-1/2}$. If $M = R(\theta)^T \Lambda R(\theta)$, then $\tilde{M} = R(\theta)^T \Lambda^{-1/2} R(\theta)$. Note that the diagonal of $\Lambda^{-1/2}$ is the major and minor radius of the ellipse defined by M. Hereafter, let $\tilde{M}$ denote the tensor ellipse defined by a matrix M.

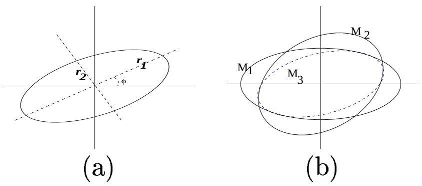


Figure 1: (a):The ellipse defined by matrix M. (b): The maximal ellipse intersected by two ellipses.

4 Operations on Metrics

In this section, we discuss some basic operations on the metrics. Some definitions had been proposed and discussed by previous researchers [4, 10, 19], but for completeness and their importance in our methods and notations, we still include them here. These operations are important in formally defining the size and shape of well-shaped anisotropic meshes and make it possible to extend many standard concepts such as Lipschitz condition of spacing function from isotropic case to anisotropic cases. Here a metric at a point is symmetric positive definite matrix M. We call an ellipse defined by M the tensor ellipse of M, or just ellipse for simplicity.

4.1 Basic Notations and Operations

Let $\boldsymbol{x}$, $\boldsymbol{y}$ be two vectors incident on point $\boldsymbol{c}$. The *dot* product becomes $\boldsymbol{x}^T M(\boldsymbol{c})\boldsymbol{y}$, and the cross product becomes $\sqrt{det(M(\boldsymbol{c}))}(\boldsymbol{x} \times \boldsymbol{y})$, where $\boldsymbol{x} \times \boldsymbol{y}$ is the Euclidean product; $det(M(\boldsymbol{c}))$ is the determinant of matrix $M(\boldsymbol{c})$. The area of the ellipse $M(r_1, r_2, \phi)$ is $\pi r_1 r_2$, i.e., $\pi|det(M)|^{-1/2}$. The perimeter of the ellipse M is $4aE(0, e)$, where e is the eccentricity, $E(\phi, k)$ is the elliptic integral. See [19] for more formulas.

For an edge $\boldsymbol{x}_0\boldsymbol{x}_1$ in the domain, let $\Gamma(t) = (1-t)\boldsymbol{x}_0 + t\boldsymbol{x}_1$ be a parametric description for the segment $\boldsymbol{x}_0\boldsymbol{x}_1$. Then its length [19] is defined as $\int_o^1 \sqrt{(x_1 - x_0)^T M(\Gamma(t))(x_1 - x_0)}dt$. By linearly interpolating the metric $M(\Gamma(t)) = (1-t)M(\boldsymbol{x}_0) + tM(\boldsymbol{x}_1)$, we approximate the segment length as the following.

Definition 4.1 [Edge Length] *[6] The length of segment $\boldsymbol{x}_0\boldsymbol{x}_1$, denoted by $||\boldsymbol{x}_0 - \boldsymbol{x}_1||_M$, is approximated by*

$$\int_0^1 \sqrt{l_0^2 + t(l_1^2 - l_0^2)}dt \quad = \quad \frac{2}{3}\frac{l_0^2 + l_0 l_1 + l_1^2}{l_0 + l_1} \quad (1)$$

where $l_i = \sqrt{(x_1 - x_0)^T M(x_i)(x_1 - x_0)}$,$i = 0, 1$.

Let $\boldsymbol{rp}$, $\boldsymbol{rq}$ be two segments incident on point $\boldsymbol{r}$. Assume that $M(\boldsymbol{r}) = R(\phi)^T \Lambda R(\phi)$. Then by transforming the ellipse $M(\boldsymbol{r})$ to unit circle through $\Lambda^{1/2}R(\phi)$, the angle between $\boldsymbol{rp}$ and $\boldsymbol{rq}$ is computed as the following.

Definition 4.2 [Angle] *Let θ be the angle between segment $\boldsymbol{rp}$ and $\boldsymbol{rq}$. Then*

$$\cos(\theta) = \frac{(\boldsymbol{p} - \boldsymbol{r})^T M(\boldsymbol{r})(\boldsymbol{q} - \boldsymbol{r})}{||\Lambda^{1/2}R(\phi)(\boldsymbol{p} - \boldsymbol{r})|| \cdot ||\Lambda^{1/2}R(\phi)(\boldsymbol{q} - \boldsymbol{r})||}. \quad (2)$$

The area of a domain Ω is defined as $A(\Omega) = \int\int_\Omega \sqrt{det(M(\boldsymbol{x}))}dx_1 dx_2$, where $\boldsymbol{x} = (x_1, x_2)^T$, see [19]. Given the metric at each vertex of a triangle element $\triangle\boldsymbol{pqr}$, we use the linearly interpolation of metric for every point in it. Then the area of $\triangle\boldsymbol{pqr}$ is approximated by the following.

Definition 4.3 [Triangle Area] *The area of $\triangle\boldsymbol{pqr}$ is approximated by*

$$A(\triangle\boldsymbol{pqr}) = \frac{1}{2}\sqrt{det(M_{avg})}(\boldsymbol{p} - \boldsymbol{r}) \times (\boldsymbol{q} - \boldsymbol{r}), \quad (3)$$

where $M_{avg} = (M(\boldsymbol{p}) + M(\boldsymbol{q}) + M(\boldsymbol{r}))/3$.

We then introduce the following definitions.

Definition 4.4 [Scaling] *Let $\beta \otimes M(\boldsymbol{x})$ denote the metric by scaling the tensor ellipse centered at point $\boldsymbol{x}$ by a factor β.*

Hence $\beta \otimes M(r_1, r_2, \phi) = M(\beta r_1, \beta r_2, \phi)$, if $\beta > 0$. Notice $\beta \otimes M(\boldsymbol{x}) = (m_{ij}/\sqrt{\beta})$; if $\beta_1 > 0$, $\beta_2 > 0$, $(\beta_1\beta_2) \otimes M(\boldsymbol{x}) = \beta_1 \otimes (\beta_2 \otimes M(\boldsymbol{x}))$.

Definition 4.5 [Expanding] *Let $M(\boldsymbol{x}) \oplus c$ denote the metric of the ellipse centered at point $\boldsymbol{x}$ by extending (or shrinking if $c < 0$) the major and minor radii of the tensor ellipse by value c.*

Thus $M(r_1, r_2, \phi) \oplus c = M(r_1 + c, r_2 + c, \phi)$.

Definition 4.6 [Rotation] *Let $M^\theta(\boldsymbol{x})$ denote the metric of the ellipse centered at point $\boldsymbol{x}$ by counterclockwise rotating $M(\boldsymbol{x})$ by θ degree.*

Notice that $M(r_1, r_2, \phi)^\theta = M(r_1, r_2, \phi + \theta)$, i.e., $M^\theta(\boldsymbol{x}) = R(\theta)^T M(\boldsymbol{x})R(\theta)$. All above definitions respect to definitions for circles when ellipses degenerate to circles.

4.2 Summation, Subtraction, Intersection and Union

The metric summation is often necessary for interpolating the metric based on some discretized values. One rigorous approach to define the summation of M_1, M_2 is to use the idea of simultaneous reduction of M_1 and M_2 [10]. Let (e_1, e_2) be the eigenvectors of the matrix $M_1^{-1}M_2$. Note that the eigenvectors are same as matrix $M_2^{-1}M_1$.

Lemma 4.1 *[10] Let (e_1, e_2) be the eigenvectors of the matrix $M_1^{-1}M_2$. Then $e_1^T M_1 e_2 = e_1^T M_2 e_2 = 0$.*

Proof: Assume δ_1, δ_2 are two eigenvalues of $M_1^{-1}M_2$, i.e., $M_1^{-1}M_2 e_i = \delta_i e_i$, $i = 1, 2$. Then we have $M_2 e_1 = \delta_1 M_1 e_1$, and $M_2 e_2 = \delta_2 M_1 e_2$, which implies that $e_2^T M_2 e_1 = \delta_1 e_2^T M_1 e_1$, and $e_1^T M_2 e_2 = \delta_2 e_1^T M_1 e_2$.
From $e_1^T M_2 e_2 = \delta_2 e_1^T M_1 e_2$, we have $e_2^T M_2 e_1 = (e_1^T M_2 e_2)^T = (\delta_2 e_1^T M_1 e_2)^T = \delta_2 e_2^T M_1 e_1$. Noting that $e_2^T M_2 e_1 = \delta_1 e_2 M_1 e_1$, we have $\delta_1 e_2^T M_1 e_1 = \delta_2 e_2^T M_1 e_1$. Hence we have $(\delta_1 - \delta_2)(e_2^T M_1 e_1) = 0$. If $(e_2^T M_1 e_1) \neq 0$, then $\delta_1 = \delta_2$, which implies that we can select e_1, e_2 such that $e_2^T M_1 e_1 = 0$. And $e_1^T M_1 e_2 = 0$ comes from the fact that $e_2^T M_2 e_1 = \delta_1 e_2^T M_1 e_1$. □
Let $\lambda_i = e_i^T M_1 e_i$, $\mu_i = e_i^T M_2 e_i$, $i = 1, 2$. Then we write the two matrices at the following format [10]:

$$\begin{aligned} M_1 &= (\mathcal{P}^{-1})^T \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix} \mathcal{P}^{-1}, \\ M_2 &= (\mathcal{P}^{-1})^T \begin{pmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{pmatrix} \mathcal{P}^{-1}, \end{aligned}$$

where $\mathcal{P}$ is the matrix formed by the column vectors (e_1, e_2).
Notice that the above approach can be extended to any dimension. Also note that if let $X = x_1 e_1 + x_2 e_2 = \mathcal{P}(x_1, x_2)^T$ be an arbitrary vector in $\mathbb{R}^2$ represented under the *oblique coordinates* (e_1, e_2). Then

$$X^T M_1 X = \lambda_1 x_1^2 + \lambda_2 x_2^2;$$
$$X^T M_2 X = \mu_1 x_1^2 + \mu_2 x_2^2.$$

Definition 4.7 [**Summation**] *The metric of the summation of metrics is*

$$M_1 \oplus M_2 = (\mathcal{P}^{-1})^T \begin{pmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{pmatrix} \mathcal{P}^{-1}, \qquad (4)$$

where $\gamma_i = 1/(\frac{1}{\sqrt{\lambda_i}} + \frac{1}{\sqrt{\mu_i}})^2$, $i = 1, 2$.

Note that expanding $M \oplus c$ is actually $M \oplus (c \otimes I)$. Also note that $\widetilde{M_1 \oplus M_2} = \widetilde{M_1} + \widetilde{M_2}$.

Definition 4.8 [**Subtraction**] *The metric of the subtraction of metrics is*

$$M_1 \ominus M_2 = (\mathcal{P}^{-1})^T \begin{pmatrix} \gamma_1 & 0 \\ 0 & \gamma_2 \end{pmatrix} \mathcal{P}^{-1}, \qquad (5)$$

where $\gamma_i = 1/(\frac{1}{\sqrt{\lambda_i}} - \frac{1}{\sqrt{\mu_i}})^2$, $i = 1, 2$.

Note that the summation and subtraction of two ellipses always define a new ellipse. In [10], they give a way to approximate the intersection of two metrics as following.

Definition 4.9 [**Intersection**]*[10] The intersection $M_1 \cap M_2$ of metrics is defined as*

$$(\mathcal{P}^{-1})^T \begin{pmatrix} \max(\lambda_1, \mu_1) & 0 \\ 0 & \max(\lambda_2, \mu_2) \end{pmatrix} \mathcal{P}^{-1}. \qquad (6)$$

Definition 4.10 [**Union**] *We define the union $M_1 \cup M_2$ of metrics as*

$$(\mathcal{P}^{-1})^T \begin{pmatrix} \min(\lambda_1, \mu_1) & 0 \\ 0 & \min(\lambda_2, \mu_2) \end{pmatrix} \mathcal{P}^{-1}. \qquad (7)$$

Lemma 4.2 *The ellipse defined by $M_1 \cap M_2$ is contained by both $\tilde{M}_1$ and $\tilde{M}_2$; The ellipse defined by $M_1 \cup M_2$ contains both $\tilde{M}_1$ and $\tilde{M}_2$.*

Proof: Note that a point X is inside an ellipse $\tilde{M}$, if and only if $X^T M X \leq 1$. Let $X = x_1 e_1 + x_2 e_2$ be an arbitrary point inside ellipse defined by $M_1 \cap M_2$. Then

$$\max(\lambda_1, \mu_1) x_1^2 + \max(\lambda_2, \mu_2) x_2^2 \leq 1.$$

It implies that $\lambda_1 x_1^2 + \lambda_2 x_2^2 \leq 1$ and $\mu_1 x_1^2 + \mu_2 x_2^2 \leq 1$. Hence X is inside ellipse $\tilde{M}_1$ and $\tilde{M}_2$, which implies the first part of the lemma. The second part follows from the similar proof. □
Note that the above definitions for the intersection and the union do not satisfy the associate rule, i.e., generally, $M_1 \cap (M_2 \cap M_3) \neq (M_1 \cap M_2) \cap M_3$; $M_1 \cup (M_2 \cup M_3) \neq (M_1 \cup M_2) \cup M_3$.
Suppose now, there are several variables are given, then we first have to find a metric such that the maximum interpolation error is minimized for all given variables. In other words, the problem is to find the "biggest" ellipse contained in all ellipses corresponding to the metrics defined by all given variables. See Figure 1 (b). One approach [10] is to use the simultaneous reduction of M_1 and M_2, $M_1 \cap M_2$, i.e.,

$$(\mathcal{P}^{-1})^T \begin{pmatrix} \max(\lambda_1, \mu_1) & 0 \\ 0 & \max(\lambda_2, \mu_2) \end{pmatrix} \mathcal{P}^{-1}.$$

4.3 Transformation, Interpolation

Given a metric $M_1 = R(\phi)^T \Lambda R(\phi)$, the corresponding tensor ellipse becomes a unit circle under the transformation $\Lambda^{1/2}R(\phi)$ of coordinates. We define the following transformation $\Phi_{M_1}()$ that transforms the $\tilde{M_1}$ at a specified point to unit circle and transforms $\tilde{M}_k$ at any other point to another ellipse. As before, we compute the simultaneous reduction eigenvectors e_1, e_2 for matrix $M_1^{-1}M_k$, and let $\mathcal{P}_k$ be the matrix formed by the column vector (e_1, e_2). Let $\lambda_i = e_i^T M_1 e_i$, $\mu_i = e_i^T M_k e_i$ be their eigenvalue, $i = 1, 2$.

Definition 4.11 [**Transformation**] *Then we define the transformation as*

$$\Phi_{M_1}(M_k) = (\mathcal{P}_k^{-1})^T \left(\begin{array}{cc} \mu_1/\lambda_1 & 0 \\ 0 & \mu_2/\lambda_2 \end{array} \right) \mathcal{P}_k^{-1}. \quad (8)$$

Note that $\Phi_{M_1}(M_k)$ is symmetric and positive definite. Also note that if we apply $\Phi_{M_1}()$ to M_1 itself, then $M_1^{-1}M_1$ becomes the unit matrix. Then the eigenvectors e_1, e_2 of M_1 are chosen as the eigenvectors of $M_1^{-1}M_1 = I$.

For mesh generation, we do not need to compute the spacing functions exactly. A common approach to approximate them is to store discrete values on the vertices of a background mesh such as a quadtree/octree decomposition of the domain, or the previous mesh. When we need to evaluate the spacing function at an arbitrary point $\boldsymbol{x}$ in the domain, we simply interpolate these discrete values. Assume $\boldsymbol{x}$ is contained in a simplex element K of the mesh.

Definition 4.12 [**Interpolation**] *Let $\boldsymbol{p}_1$, $\boldsymbol{p}_2$, $\boldsymbol{p}_3$ be the vertices of K. Assume that $\boldsymbol{x} = \sum_{i=1}^{3} \alpha_i \boldsymbol{p}_i$, where $\sum_{i=1}^{3} \alpha_i = 1$ and $\alpha_i \geq 0$, then the linearly interpolated spacing for $\boldsymbol{x}$ is*

$$M(\boldsymbol{x}) = M(\sum_{i=1}^{3} \alpha_i \boldsymbol{p}_i) = \bigoplus_{i=1}^{3} (\alpha_i \otimes M(\boldsymbol{p}_i)). \quad (9)$$

Note that the linear interpolation of the metric does not keep the *α-Lipschitz*. In other words, if the metric on all background mesh vertices satisfies the *α-Lipschitz* condition, then the interpolated metric on all points does not always have the *α-Lipschitz* condition.

4.4 Norm, Distance and α-Lipschitz

The local feature size $\boldsymbol{lfs}$ and the numerical condition $M(\boldsymbol{x})$ together define the global control spacing function. Recall that for generating well shaped isotropic mesh that conforms to the control spacing $f()$, we need the smooth condition of the spacing function $f()$, i.e., $f()$ satisfies the α-Lipschitz condition. We expect that for generating a mesh that conforms well to the control tensor metric spacing $M()$, we also need the "smoothness" condition of $M()$. Intuitively, for any two points $\boldsymbol{x}$ and $\boldsymbol{y}$, if the Euclidean distance $||\boldsymbol{x} - \boldsymbol{y}||$ is small compared to the local objective ellipse, then the ellipses $M(\boldsymbol{x})$ and $M(\boldsymbol{y})$ can not have dramatical change either in the size or in the direction. Otherwise, it is impossible to generate mesh elements in that region which both satisfy the size requirement and the shape requirement well. For continuous metrics M, we can define the derivate of M along any direction. Notice however, we did not have any analog definition about the smoothness of the tensor metrics $M(\boldsymbol{x})$ compared with the Lipschitz condition for isotropic case.

First of all, we need to define the distance of two ellipses, then we can define the Lipschitz condition of the tensor metric $M(\boldsymbol{x})$. Noting that, if the metric is isotropic, i.e., the ellipse becomes circles, the distance of two circles is the radius difference. The definition of the distance of two ellipses should capture both the shape difference and the direction difference.

Let a, b be the major and minor radii of the ellipse $M(\boldsymbol{x})$, and let ϕ be the angle between the major radius and the x axes. Let $f_M(\theta)$ be the *polar equation* for the ellipse in the usual polar coordinate system with origin at point $\boldsymbol{x}$. Then we have

$$f_M(\theta) = ab/\sqrt{a^2 \sin^2(\theta - \phi) + b^2 \cos^2(\theta - \phi)}. \quad (10)$$

Then we define the norm of a metric M as the average segment size.

Definition 4.13 [**Norm**] *Given an ellipse M_1, we define its norm $||M_1||$ as*

$$||M_1|| = \frac{1}{2\pi} \int_0^{2\pi} |f_{M_1}(\theta)| d\theta \quad (11)$$

Notice that if the ellipse degenerated to a circle, its norm is the radius of the circle.

Definition 4.14 **[Ellipses Distance]** *Given any two ellipses M_1, and M_2, we define their distance $||M_1 - M_2||_d$ as the following.* [1]

$$||M_1 - M_2||_d = \frac{1}{2\pi}\int_0^{2\pi} |f_{M_1}(\theta) - f_{M_2}(\theta)|d\theta \quad (12)$$

Note that the above definition satisfies the three conditions of the distance definition: (1) $||M_1 - M_2|| \geq 0$, for all ellipses M_1, and M_2; (2) $||M_1 - M_1|| = 0$; (3) $||M_1 - M_2|| \leq ||M_1 - M_3|| + ||M_3 - M_2||$, for all ellipses M_1, M_2 and M_3. The third property comes from the following observation: $|f_{M_1}(\theta) - f_{M_2}(\theta)| \leq |f_{M_1}(\theta) - f_{M_3}(\theta)| + |f_{M_3}(\theta) - f_{M_2}(\theta)|$, for any θ. Also note that the above definition respects to the distance definition for circles, when the ellipse degenerates to a circle. However, it is difficult to compute the norm and distance according to above definitions in practice. We can use the Frobenius norm of the matrix to define the norm and the distance of metrics. Let $\widetilde{m_{i,j}}$, $\widetilde{n_{i,j}}$ be the elements of matrix $\widetilde{M}$, $\widetilde{N}$ respectively.

Definition 4.15 **[F-Norm]** *Given an ellipse M, we define its F-norm $||M||$ as*

$$||M|| = (\sum_{i,j} \widetilde{m_{i,j}}^2/2)^{1/2}.$$

Definition 4.16 **[F-Distance]** *Given any two ellipses M, and N, we define their F-distance $||M - N||$ as following.*

$$||M - N|| = (\sum_{i,j}(\widetilde{m_{i,j}} - \widetilde{n_{i,j}})^2/2)^{1/2}.$$

Notice that the definition of distance using the Frobenius norm satisfies the triangle-inequality property.

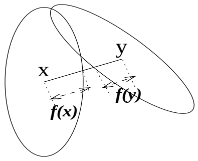


Figure 2: The segment along the line by two centers.

We also can use the idea of Shimada [18] defining the force between particles to define the distance of ellipses. See section 2.2. Let $f(\boldsymbol{x})$, $f(\boldsymbol{y})$ be the lengths measured along the line segment that connects the centers $\boldsymbol{x}$ and $\boldsymbol{y}$, from the center to the boundary of each ellipses $M(\boldsymbol{x})$, $M(\boldsymbol{y})$. See Figure 2. We approximate the distance of $M(\boldsymbol{x})$, $M(\boldsymbol{y})$ as following.

Definition 4.17 **[S-Distance]** *Given any two metrics $M(\boldsymbol{x})$, and $M(\boldsymbol{y})$, we approximate their distance $||M(\boldsymbol{x}) - M(\boldsymbol{y})||$ as $|f(\boldsymbol{x}) - f(\boldsymbol{y})|$.*

Notice that the definition respects to the definition for circles when ellipses degenerate to circles, but it does not satisfy the triangle property.
We define the α-Lipschitz condition for the metric $M()$ as the following.

Definition 4.18 **[α-Lipschitz Metric]** *A metric $M()$ is Lipschitz with a coefficiency α, if for any two points $\boldsymbol{x}$ and $\boldsymbol{y}$ in the domain, $||M(\boldsymbol{x}) - M(\boldsymbol{y})|| \leq \alpha||\boldsymbol{x} - \boldsymbol{y}||$, where $||\boldsymbol{x} - \boldsymbol{y}||$ is the Euclidean distance of points $\boldsymbol{x}, \boldsymbol{y}$.*

4.5 Smoothing the Metric

Recall that, for possibly generating a mesh that conforms the control spacing, we expect the control spacing to have some smoothness condition. Let us consider discrete control spacing $(T(\Omega), M_{T(\Omega)}())$, where $T(\Omega)$ is a background mesh defined on domain Ω; $M_{T(\Omega)}()$ is a discrete field of metrics associated with the vertices of $T(\Omega)$. We are interested in a *reasonable* correction on $M_{T(\Omega)}()$ to reduce the size and direction variation, i.e., to smooth the metric such that it satisfies the α-Lipschitz condition. Note that, for isotropic case, we apply the following operation to get the α-Lipschitz smooth spacing function $f_s()$ from old spacing function $f()$:

$$f_s(\boldsymbol{x}) = min(f(\boldsymbol{x}), min_{\boldsymbol{y}\in T(\Omega)}(f(\boldsymbol{y}) + \alpha||\boldsymbol{x} - \boldsymbol{y}||)).$$

For anisotropic spacing $M()$, the above approach can not be applied directly. One approach is to define the *min* operation using the *intersection* operation. See definition 4.9. in Section 4.2. Let $D(M, c)$ be a set of ellipses that have distance c to M. Then the analog definition $M_s(\boldsymbol{x})$ for anisotropic smoothing will be as the following:

Definition 4.19 **[Smoothing]** *The new spacing metric of point $\boldsymbol{x}$ is defined as*

$$M_s(\boldsymbol{x}) = \bigcap_{\boldsymbol{y}\in T(\Omega)} (M(\boldsymbol{x}) \cap (M(\boldsymbol{y}) \oplus \alpha||\boldsymbol{x} - \boldsymbol{y}||)). \quad (13)$$

[1] From now on, we use $||M_1 - M_2||$ to denote the distance of two ellipses defined by two matrices M_1 and M_2. It does not mean the determinant of the matrix $M_1 - M_2$.

For saving the time of smoothing, we do not have to compute the intersection for all vertices $\boldsymbol{y} \in T(\Omega)$. We can just compute it for vertices $\boldsymbol{y}$ connected to $\boldsymbol{x}$ in the background mesh. Then we can iteratively apply above step to get much more smoothing.

4.6 Delaunay Criterion

To build a Delaunay like anisotropic triangulation for a set of points, we need perform the circumcircle test. The current most often used anisotropic Delaunay criteria [3, 18] use the following approach. Assume $\boldsymbol{x}$,$\boldsymbol{y}$,$\boldsymbol{z}$,$\boldsymbol{w}$ are the four vertices of the quadrilateral being checked, and they are on the counterclockwise relation.

Definition 4.20 [**Delaunay Criterion**]*[3] The current diagonal edge $\boldsymbol{xz}$ is swapped if*

$$[(\boldsymbol{z}-\boldsymbol{y}) \times (\boldsymbol{x}-\boldsymbol{y})](\boldsymbol{x}-\boldsymbol{w})^T M_{avg}(\boldsymbol{z}-\boldsymbol{w}) + (\boldsymbol{z}-\boldsymbol{y})^T M_{avg}(\boldsymbol{x}-\boldsymbol{y})[(\boldsymbol{x}-\boldsymbol{w}) \times (\boldsymbol{z}-\boldsymbol{w})] < 0,$$

where $M_{avg} = (M(\boldsymbol{x}) + M(\boldsymbol{y}) + M(\boldsymbol{z}) + M(\boldsymbol{w}))/4$ or $M_{avg} = M((\boldsymbol{x}+\boldsymbol{y}+\boldsymbol{z}+\boldsymbol{w})/4)$.

First note that the above two computations for M_{avg} are not same usually. Also notice that the above definition for the Delaunay criterion is not consistent in the following sense. There exist four points, none of the diagonal edge is locally Delaunay. To address this problem. we propose the following concept to construct the Delaunay like triangulation.
First for isotropic case, let $\boldsymbol{pq}$ be the current checked edge, and $\triangle\boldsymbol{pqr}$, $\triangle\boldsymbol{pqs}$ be two incident triangles. Let $B(\boldsymbol{c_r}, r_{\boldsymbol{r}})$, and $B(\boldsymbol{c_s}, r_{\boldsymbol{s}})$ be the circumcircles of elements $\triangle\boldsymbol{pqr}$, $\triangle\boldsymbol{pqs}$ respectively. Let ρ be a positive constant less than 1. For convenience, let $\alpha_K(\boldsymbol{x}) = ||\boldsymbol{x}-\boldsymbol{c}||/r$, [10], where $\boldsymbol{c}$, r are the center and radius of the circumcircle of K. Then the following definition is introduced.

Definition 4.21 [**ρ-Locally-Delaunay**] *Edge $\boldsymbol{pq}$ is called ρ-locally-Delaunay if $\boldsymbol{r}$ is not in the interior of circle $B(\boldsymbol{c_s}, \rho r_{\boldsymbol{s}})$, and $\boldsymbol{s}$ is not in the interior of circle $B(\boldsymbol{c_r}, \rho r_{\boldsymbol{r}})$, i.e., $\alpha_{\triangle\boldsymbol{pqs}}(\boldsymbol{r}) \geq \rho$, and, $\alpha_{\triangle\boldsymbol{pqr}}(\boldsymbol{s}) \geq \rho$.*

Definition 4.22 [**ρ-Delaunay**] *A mesh is called ρ-Delaunay, if all edges are ρ-Locally-Delaunay.*

Hence the traditional Delaunay triangulation is 1-Delaunay under our definition.

For anisotropic mesh generation, the distance is measured under the metric tensor. The circumcenter $\boldsymbol{c}$ of the triangle element $\triangle\boldsymbol{pqr}$ is the point satisfies the following equations: $||\boldsymbol{c}-\boldsymbol{p}||_M = ||\boldsymbol{c}-\boldsymbol{q}||_M = ||\boldsymbol{c}-\boldsymbol{r}||_M$. The segment length is computed according to equation 4.1.

5 Ellipse Biting

Miller *et al.* [15] have designed a sphere-packing based meshing method which generates quality mesh. Li *et al.* [12] recently designed a scheme called biting to generate high quality isotropic mesh combining the spirit of advancing front and sphere-packing method. In this section, we present a new anisotropic mesh generation method which uses biting as the framework.

5.1 β-Ellipse Packing

Suppose $M()$ is the desired edge-length or nearest-neighbor function with direction control of a high quality anisotropic mesh for a domain Ω. In other words, $M(\boldsymbol{x})$ defines an ellipse at point $\boldsymbol{x}$. We now introduce some definitions to capture the quality of ellipse packing.

Definition 5.1 [**β-Ellipse-Packing**] *Let β be a positive real constant. A set S of ellipses is a β-ellipse-packing with centers P of Ω with respect to a spacing function $M()$ if*

- *For each point p of P, $0.5 \otimes M(p) \in S$;*
- *The interiors of any two ellipses s_1 and s_2 in S do not overlap; and*
- *For each point $q \in \Omega$, there is an ellipse in S that overlaps with $\beta \otimes M(q)$.*

We also can use the idea of Shimada [18] defining the force between particles to define a weak-β-Ellipse-Packing. See section 2.2. Let $L_{\boldsymbol{x}}(\boldsymbol{y})$, $L_{\boldsymbol{y}}(\boldsymbol{x})$ be the segments, on the line that connects the centers $\boldsymbol{x}$ and $\boldsymbol{y}$, from the center to the boundary of each ellipses $M(\boldsymbol{x})$, $M(\boldsymbol{y})$. See Figure 2.

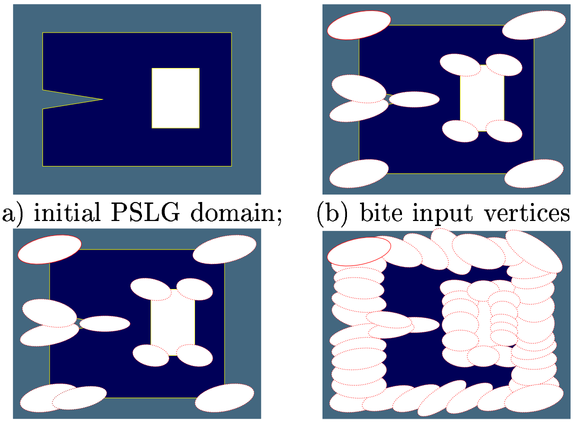
Definition 5.2 [**Weak-β-Ellipse-Packing**] *Let β be a positive real constant. A set S of ellipse is a β-ellipse-packing with centers P of Ω with respect to a spacing function $M()$ if*

- *For each point p of P, $0.5 \otimes M(p) \in S$;*

- *For any two ellipses s_1 and s_2 centered at $\boldsymbol{x}$, $\boldsymbol{y}$, the segment $L_{\boldsymbol{x}}(\boldsymbol{y})$ and $L_{\boldsymbol{y}}(\boldsymbol{x})$ do not overlap.*
- *For each point $q \in \Omega$, there is an ellipse in S that overlaps with $\beta \otimes M(q)$.*

Notice that a β-Ellipse-packing is always a weak-β-Ellipse-packing.

5.2 Biting Ellipse Scheme



(a) initial PSLG domain; (b) bite input vertices;

(c)bite non-input vertex; (d) bite a layer.

Figure 3: A snapshot of the *biting* scheme.

The biting method moves a front from the boundary of the domain to the interior and adds new mesh points in the process. See Figure 3. These mesh points are chosen such that in the mesh generated from this points set, the edges connected to each point approximately satisfy the ellipse property. At each step, we place a new point on the current front rather than place it in the interior. Each time we add a point, we remove an ellipse, the *biting ellipse* from the remaining interior domain. The boundary between the union of biting ellipses and the remaining interior domain defines the new front. This process is called *biting.* A biting ellipse at a point $\boldsymbol{x}$ is $c_b \otimes M(\boldsymbol{x})$, denoted by $\overline{M}(\boldsymbol{x})$, where c_b is a constant satisfying $\alpha c_b \leq 0.5$. The following is a formal description of the biting method:

Algorithm `Biting Ellipse`

1. Compute the control spacing function $M()$ of Ω by combining the local feature size and the numerical condition;
2. Let the boundary of the domain be the initial front, see Figure 3 (a);
3. [**Vertex Protection**]: Bite all the input vertices by removing their biting ellipses from the interior of the domain, see Figure 3 (b);
   Modify the front which becomes a set of segments and arcs. Segments are represented by the endpoints and arcs are represented by the center of the biting ellipse.
4. [**Edge Protection**]: Bite ellipses centered on the input boundary: choose a vertex $\boldsymbol{x}$ on the front and remove its biting ellipse. We apply the same boundary protection technique as in [12]. See Figure 3 (c) and (d).
   Modify the front by introducing the arc of the new biting ellipses and removing the intersection of it with the front.
   Repeat until all initial input boundaries are bitten;
5. [**Interior Biting**]: Choose a vertex $\boldsymbol{x}$ on the front and remove its biting ellipse.
   Modify the front by introducing the arc of the new biting ellipse and removing the intersection of it with the front.
   Repeat until the advancing front is empty.
6. Construct the Delaunay like triangulation of the centers of the biting ellipses as the final mesh.

The biting method always choose the next Steiner point on the front itself. In other words, the front itself is a subset of the feasible region for the selection of new mesh vertices, making it easier to choose the next point. The intersection of two arcs or an arc and a boundary segment provides a good candidate for a new Steiner point, whose biting ellipse will reduce the interior.

The biting method constructs an ellipse packing with respect to the spacing function. The removal of its biting ellipse ensures that the future edge connection will respect to the ellipse shape. The final mesh is the triangulation of the resulting point set by applying the ρ-locally-Delaunay rule.

Let V be the mesh vertices set generated by our biting scheme. Let $E(V) = \{\frac{c_b}{2+\alpha c_b} \otimes M(\boldsymbol{x}) | \boldsymbol{x} \in V\}$ be set of ellipses centered at the mesh vertices.

Theorem 5.1 *If the distance of metric is measured by the S-Distance (4.17), our biting scheme generates a weak $\frac{1+\alpha c_b}{1-\alpha c_b}$ ellipse packing $E(V)$ with respect to $\frac{2c_b}{2+\alpha c_b} \otimes M()$.*

Proof: First, for any two vertices $\boldsymbol{x} \in V$, $\boldsymbol{y} \in V$, we have $|f(\boldsymbol{x}) - f(\boldsymbol{y})| \leq \alpha||\boldsymbol{x} - \boldsymbol{y}||$. See definition 4.17 for the definition of $f()$. Assume that vertex $\boldsymbol{x}$ is bitten before $\boldsymbol{y}$, i.e., $||\boldsymbol{x} - \boldsymbol{y}|| \geq c_b f(\boldsymbol{x})$. Then $f(\boldsymbol{x}) + f(\boldsymbol{y}) \leq 2f(\boldsymbol{x}) + \alpha||\boldsymbol{x} - \boldsymbol{y}||$. It follows that $(\frac{2c_b}{2+\alpha c_b} f(\boldsymbol{x}))/2 + (\frac{2c_b}{2+\alpha c_b} f(\boldsymbol{y}))/2 \leq ||\boldsymbol{x} - \boldsymbol{y}||$. In other words, it satisfies the second condition of the definition for weak-β-packing.
Notice that any point $\boldsymbol{z}$ in the domain is covered by at least one biting ellipse. Assume that $\boldsymbol{z}$ is covered by ellipse $c_b \otimes M(\boldsymbol{x})$, i.e., $||\boldsymbol{z} - \boldsymbol{x}|| \leq c_b f(\boldsymbol{x})$. $f(\boldsymbol{z}) \geq f(\boldsymbol{x}) - \alpha||\boldsymbol{x} - \boldsymbol{z}||$ implies that $f(\boldsymbol{z}) \geq (1/c_b - \alpha)||\boldsymbol{x} - \boldsymbol{z}||$. Let $\beta = \frac{1+\alpha c_b}{1-\alpha c_b}$, then $\beta \frac{c_b}{2+\alpha c_b} f(\boldsymbol{z}) + \frac{c_b}{2+\alpha c_b} f(\boldsymbol{x}) \geq ||\boldsymbol{z} - \boldsymbol{x}||$. Then the theorem follows. □
For solving the finite element system, it is crucial to generate a mesh that has high quality element, especially on the boundary of the mesh. Our biting method generates the vertices of the mesh starting from the input boundary. It first makes sure that there are no vertices close to the boundary, compared with its local size requirement. To do so, the basic approach of the biting method is to make sure that after biting all boundary edges, the new fronts are not close to the input boundary.

5.3 Conformity of the Triangulation

For isotropic mesh, we say a mesh conforms well to the given control spacing, if the nearest neighbor value, or edge length function is within a constant factor of the control spacing. In this section, we give definitions for the conformity measure of the final triangulation.
It is easy to prove that, if a triangle $K = \triangle \boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3$ is a non-degenerated element, then there exists a unique metric M_K such that K is equilateral with unit edge lengths under M_K, i.e., $||\boldsymbol{x}_i - \boldsymbol{x}_j||_{M_K} = 1$, for $1 \leq i < j \leq 3$. Assume that $\boldsymbol{c}$ is the barry-center of K, and $M_{\boldsymbol{c}}$ is the required control spacing for $\boldsymbol{c}$. Then we use the following definition to measure how bad K conforms to a given control spacing.

Definition 5.3 [**Nonconformity**] *The nonconformity of the triangle K is defined as $||M_K - M_{\boldsymbol{c}}||/||M_{\boldsymbol{c}}||$.*

Notice that the perfect matched element will have the nonconformity 0; the larger the nonconformity of any element, the worse the element conforms to the given control spacing. If an element K has conformity bound above from a constant γ, then the angle at each vertex of K computed according to the given control spacing has a lower bound.

Definition 5.4 [**γ-Well-Conformed Mesh**] *A mesh is γ-well-conformed if every triangle element of the mesh has nonconformity at most γ,*

The connection between a good ellipse packing and the γ-Well-Conformed Mesh is still unknown.

6 Conclusion

In this paper, we have defined a set of operations on tensor metrics. An important application of these operations is that it enable us to define the distance between the ellipses, which is used to measure the Lipschitz property of anisotropic spacing function. The Lipschitz condition is very important in the definitions of well-spaced ellipse-packing and the definition of the locally Delaunay condition. These definitions are the bases for us to extend various provably good meshing methods from isotropic domain to anisotropic domains.
Built upon these definitions, we develop a new anisotropic meshing scheme which combines the practicality characteristics of the advancing front methods and packing methods. Our scheme, ellipse biting method, applies some variations of the advancing front method to generate a good ellipse packing. We proposed a ρ-locally Delaunay triangulation to construct the final mesh. It is as simple and as practical as the advancing front methods.
The standard advancing front methods, however, have some trouble to construct elements in the region where the fronts meet. Our ellipse biting method resolves the difficulty that occurs at the end of the advancing front methods. It is also efficient to generate a good ellipse packing by biting method. We are in the process of conducting the experiments of our ellipse biting program.

7 Acknowledgments

The first author thanks Xiaosong Ma of UIUC for the discussion of definition and proof of the metric distance defined in 4.14.

References

[1] Borouchaki, H., Castro-Diaz, M., George, P., Hecht, F., and Mohammadi, B. Anisotropic adaptive mesh generation in two dimensions for cfd. In *5th International Conference On Numerical Grid Generation in Computational Field Simulations* (Mississppi State University, Apr 1996), vol. 3, pp. 197–206.

[2] Borouchaki, H., and Frey, P. J. Adaptive trangular-quadrilateral mesh generation. *International Journal for Numerical Methods in Engineering 41* (1998), 915–934.

[3] Bossen, F. J. Anisotropic mesh generation with particles. Tech. Rep. CMU-CS-96-134, School of Computer Science, Carnegie Mellon University, May 1996. http://ltswww.epfl.ch/ bossen.

[4] Bossen, F. J., and Heckbert, P. S. A pliant method for anisotropic mesh generation. In *5th International Meshing Roundtable* (Sandia National Laboratories, Oct 1996), pp. 63–76.

[5] Buscaglia, G. C., and Dari, E. A. Anisotropic mesh optimization and its application in adaptivity. *International Journal for Numerical Methods in Engineering 40* (1997), 4119–4136.

[6] Diaz, M. J. C., Hecht, F., and Mohammadi, B. New progress in anisotropic grid adaptation for inviscidand viscous flows simulations. Tech. Rep. 2671, INRIA, Oct 1995.

[7] Frey, P. J., and Borouchaki, H. surface mesh evaluation. In *6th International Meshing Roundtable* (1997), pp. 363–374.

[8] Frey, P. J., Borouchaki, H., and George, P. L. Delaunay tetrahedralization using an advancing-front approach. In *5th International Meshing Roundtable* (Sandia National Laboratories, Oct 1996), pp. 31–46.

[9] Garimella, R. V., and Shephard, M. S. Boundary layer meshing for viscous flows in complex domains. In *7th International Meshing Roundtable* (Sandia National Lab, Oct 1998), pp. 107–118.

[10] George, P. L., and Borouchaki, H. *Delaunay Triangulation and Meshing: Application to Finite Elements.* Hermes, 1998.

[11] Khamayseh, A., and Kuprat, A. Anisotropic smoothing and solution adaption for unstructured grids. *International Journal for Numerical Methods in Engineering 39* (1996), 3163–3174.

[12] Li, X. Y., Teng, S. H., and Üngör, A. Biting: advancing front meets sphere packing. In *2nd Symposium on trends in unstructured mesh generation* (Boulder, Colorado., 1999).

[13] Lohner, R., and Cebral, J. Generation of non-isotropic unstructured grids via directional enrichment. In *2nd Symposium on Trends in Unstructured Mesh Generation* (University of Colorado, Boulder, 1999).

[14] Lohrer, R. Progress in grid generation via the advancing front technique. *Engineering with Computers 12* (1996), 186–210.

[15] Miller, G. L., Talmor, D., Teng, S.-H., and Walkington, N. A delaunay based numerical method for three dimensions: generation, formulation, and partition. In *Proc.* 27^{th} *Annu. ACM Sympos. Theory Comput.* (1995), pp. 683–692.

[16] Ruppert, J. A new and simple algorithm for quality 2-dimensional mesh generation. In *Third Annual ACM-SIAM Symposium on Discrete Algorithms* (1992), pp. 83–92.

[17] Shimada, K., and Liao, J. H. Quadrilateral meshing with directionality control through the packing of square cells. In *7th International Meshing Roundtable* (Sandia National Laboratories, Oct 1998), pp. 61–76.

[18] Shimada, K., Yamada, A., and Itoh, T. Anisotropic triangular meshing of parametric surfaces via close packing of ellipsoidal bubbles. In *6th International Meshing Roundtable* (Sandia National Laboratories, Oct 1997), pp. 375–390.

[19] Springer, C. E. *Tensor and vector analysis; with applications to differential geometry.* New York, Roland Press Co., 1962.

[20] Strang, G., and Fix, G. J. *An Analysis of the Finite Element Method.* Prentice-Hall, 1973.

[21] Zhang, H., and Trépanier, J. An algorithm for the optimization of directionality stretched triangulations. *Computer Methods in Applied Mechanics and Engineering* (1994), 1481–1493.