

# A New Anisotropic Local Meshing Method and Its Application in Parametric Surface Triangulation

W.W. Zhang<sup>1</sup>, Y.F. Nie<sup>1</sup>, and Y.Q. Li<sup>1</sup>

**Abstract:** A new algorithm for anisotropic triangular meshes generation in two dimension is presented. The inputs to the algorithm are the boundary geometry information and a metric tensor that specifies the desired element size and shape. The initial nodes are firstly distributed according to the above mentioned geometrical information, after bubble simulation, the optimized nodes set that meets the requirements of the metric tensor is obtained quickly. Then taking full advantage of the nodes set and the adjacency lists information provided by the process of node placement, a handful of non-satellite nodes are removed from the adjacency lists of the nodes with the Anisotropic Bubble-type Local Mesh Generation (ABLMG) method, the anisotropic meshes are generated rapidly. The developed meshes reflect the metric tensor requirement well, and they don't require remeshing and smoothing. Since the adjacency list is built for each node, it avoids searching neighbor nodes when calculating the resultant force of each node and generating the local mesh around each node, the simulation time is greatly saved. The new algorithm is also used for 3D surface triangulation based on mapping method. The parametric space corresponding to the target surface is triangulated with our algorithm, then mapping the mesh topology in the parametric space back to the original surface, the surface mesh with high quality are obtained. Some numerical examples are given to test the feasibility of the algorithm.

**Keywords:** anisotropic local mesh generation, nodes placement by bubble simulation, parametric surface, metric tensor.

## 1 Introduction

It has been stated by many computation practice that anisotropic mesh is highly efficient for reducing the degrees of freedom of a numerical computation, as well as more accurately capturing physical phenomena [Peraire and Peiro (1992); Frey and

---

<sup>1</sup> Department of Applied Mathematics, School of science, Northwestern Polytechnical University, Xi'an 710129, PR China. E-mail: yfnie@nwpu.edu.cn

Alauzet (2005); Leicht and Hartmann (2010); Legrand, Deleersnijder, Delhez, and Legat (2007)]. For the problems with complex geometry boundaries and with solutions that change rapidly in both magnitude and direction, unstructured anisotropic mesh is often advantageous in terms of computational cost and solution accuracy.

A number of meshing strategies, such as the advancing front technique(AFT) [Lo and Wang (2005)], the Delaunay triangulation method(DTM) [Borouchaki, George, Hecht, Laug, and Saltel (1997); Borouchaki, George, and Mohammadi (1997); Yokosuka and Imai (2009); Bossen and Heckbert (1996)], and the method combining local modification with smoothing or node movement [Habashi, Dompierre, Bourgault, Yahia, Fortin, and Vallet (2000); Yamakawa and Shimada (2003)], have been developed in the last decade for generating anisotropic meshes according to a given metric tensor. The AFT can be used to control the shape and size of the elements through adjusting the location of the new insert node. It offers a high quality node placement strategy for the early fronts. However, when fronts meet each other or itself, it is difficult and time consuming to decide the size and the directionality of the elements in that region. In the DTM, a coarse mesh is usually generated to cover the problem domain at first. By repeatedly insert nodes, this coarse mesh is then refined until the required gradation effect is achieved. In the method combining local modification, it constructs the constrained Delaunay triangulation of the domain firstly, then iteratively smooths, refines, and retriangulates. On each iteration, a node is selected randomly, it is repositioned according to attraction/repulsion with its neighbors, and the neighborhood is retriangulated.

From the above discussion, we find that the location where new nodes are to be inserted has a great impact on the quality of the final anisotropic mesh. And in the DTM and the method combining local modification, the time-consuming remeshing process is needed. In order to avoid ill-shaped elements caused by poorly distributed node locations and the remeshing process, Shimada et al. [Shimada and Gossard (1998)] describe a scheme to pack ellipses by defining proximity-based interacting forces among ellipses and finding a force-balancing configuration using dynamic simulation, the centres of the ellipses are then connected with Delaunay triangulation for a complete mesh topology. Yamakawa et al. [Yamakawa and Shimada (2003)] propose a anisotropic tetrahedral meshing method with bubble packing and AFT. However, for the above two methods, the dynamic simulation efficiency still needs to be improved. Then Qi et al. [Qi, Nie, and Zhang (2012)] have done several modifications to reduce the cost of simulation, such as the selection of viscosity coefficient, and a low order numerical algorithm is also chosen when solving the ordinary differential equations which control the movement of bubbles, the computing cost has been decreased by approximately 40%. In this paper, based on the modified strategy of Qi et al., a new anisotropic meshing method, called

Anisotropic Bubble-type Local Mesh Generation (ABLMG) method is described according to a given metric tensor. The new method consists of two major steps: anisotropic node placement (detailed in Section 3) and node-based local mesh generation (detailed in Section 4). Compared with Shimada's method, the ABLMG method builds the adjacency lists for each node in the process of node placement, which avoids searching neighbor nodes when calculating the resultant force of each node, meanwhile, by removing a handful of non-satellite nodes from the adjacency lists, local mesh around each node is built quickly, it makes the ABLMG method more easy and efficient.

The most direct application of anisotropic mesh generation is surface triangulation. Surface triangulation is one of the most important and yet difficult prerequisites for shell analysis [Bechet, Millet, and Sanchez-Palencia (2008)], biomechanics modeling [Chen and Lu (2011)] as well as mesh generation in three dimensions [Wu and Wang (2005)]. With the result of the rapid development in computer technology, quite a number of new automatic surface mesh generation schemes based on different approaches have been suggested [Miranda, Martha, Wawrzynek, and Ingraffea (2009); Shimada and Gossard (1998); Lee (2001); Lee (2003); Lee (2000); Borouchaki, Laug, and George (2000)]. The mapping method [Lee (2001); Lee (2003); Lee (2000); Borouchaki, Laug, and George (2000)] is one of the most popular method. In the mapping method, the target surface is represented by a bi-variate mapping, such that any node on the 3D surface is mapped to a parametric space. 3D surface mesh generation is turned into mesh generation in the parametric space, and the final 3D surface mesh is obtained by mapping the mesh topology in the parametric space back to the original surface. In the area of controlling the grading and the elements size distribution of the surface mesh, Riemannian metric [Lee (2001); Lee (2003); Lee (2000)] approach is proved to be simple to implement and easy to use.

The objective of this paper is to introduce a new algorithm for anisotropic local mesh generation. The local mesh generation method employed in the previous study [Nie, Chang, and Fan (2007); Chen, Nie, Zhang, and Wang (2012)] is not sufficiently simply and efficiently to handle complex shapes and cannot be extended to the anisotropic problems. Therefore, we developed a new anisotropic local mesh generation algorithm, unlike many other anisotropic mesh generation schemes suggested previously, the new algorithm use the anisotropic nodes placement method by bubble simulation to optimize initial nodes distribution, then with the ABLMG method, the local mesh is generated in a local area around each node without searching neighboring nodes, the union of these local meshes is the coordinating global mesh. The new algorithm is applied on 3D parametric surface mesh generation based on the mapping method, and the metric tensor is co-generated by

the 3D surface metric tensor and the gradient of the parametric surface equations. Numerical examples show that the high quality of the anisotropic meshes and the surface meshes generated in this paper.

The rest of the paper is organized as follows: Section 2 introduces the Riemannian metric tensor. In Section 3, the anisotropic node placement by bubble simulation is discussed in detail. Section 4 presented the anisotropic local mesh generation method ABLMG. The application of the ABLMG method on surface triangulation and a numerical result is discussed in Section 5. Finally, conclusions are described in Section 6.

## 2 Riemannian metric tensor

In many cases, it is desirable to create graded anisotropic meshes, where node spacing is a function of position and direction. For anisotropic mesh generation, the element size function is generalized to describe shape as well as size, with a  $2 \times 2$  Riemannian metric tensor  $M_2$  [Lo and Wang (2005)]

$$M_2(P) = \begin{bmatrix} a(P) & b(P) \\ b(P) & c(P) \end{bmatrix} \tag{1}$$

where  $a(P) > 0$ ,  $c(P) > 0$ , and  $a(P)c(P) - b^2(P) > 0$ .  $P$  is an any node in 2D domain  $\Omega$ . Let  $\lambda_1$  and  $\lambda_2$  are the eigenvalues,  $e_1$  and  $e_2$  are the unit corresponding orthogonal eigenvector, then  $M_2(P)$  can be decomposed as

$$\begin{aligned} M_2(P) &= [e_1, e_2] \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} [e_1, e_2]^T \\ &= \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \frac{1}{h_1^2} & 0 \\ 0 & \frac{1}{h_2^2} \end{bmatrix} \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \end{aligned} \tag{2}$$

where  $h_1$  and  $h_2$  are the mesh sizes along the characteristic direction  $e_1, e_2$  in the  $\Omega$  respectively. Using this tensor, the distance between two nearby nodes  $x$  and  $y$  is approximatively computed as

$$d(x, y) = \sqrt{(x - y)^T M_{avg} (x - y)} \tag{3}$$

with  $M_{avg} = (M(x) + M(y))/2$ . When the nodes placement method by bubble simulation is applied to optimize nodes distribution,  $h_1, h_2$  and  $\theta$  ( $0 \leq \theta < \pi$ ) describe the lengths of the two radii and the rotation angle of the elliptical bubble shown in Fig. 1. If  $h_1 = h_2$ , the metric is isotropic in which case the elliptical bubbles are circles.

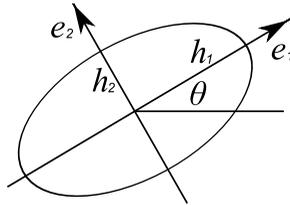


Figure 1: Elliptical bubble controlled by Riemannian metric tensor

### 3 Node placement by elliptical bubble simulation

In the light of the ideas and treatment technologies about molecular dynamics simulation and bubble meshing method presented by Shimada etc. [Shimada and Gosard (1998)], node placement method by elliptical bubble simulation is described in this section. The domain is packed by bubbles with interaction forces, and the nodes are considered as the centers of bubbles, according to the Newton's second law of motion, a force balancing configuration of bubbles is found by performing dynamic simulation, then putting nodes on the center of the bubbles, the obtained nodes is our need. The flow chart of node placement method by elliptical bubble simulation is shown in Fig.2.

Initial bubble placement is very essential to the time requirement of the process of node placement. If the initial bubble configuration is very poor, then a large number of iterative steps will be required before achieving a stable node configuration. How to reasonably place the initial bubbles is one of the problems to be solved. During dynamic simulation, the search of neighboring bubbles is an important aspect of the resultant force calculation. Meanwhile, in order to meet the requirements of the metric tensor, how to control the number of elliptical bubbles is also a problem to be solved. These issues are discussed in the following four subsections.

#### 3.1 Initial elliptical bubble placement

The geometric boundary of the domain is described by the *Planar Straight Line Graph* (PSLG), which is formed by a set of line segments, intersecting only at their end points. The elliptical bubbles are in turn placed on geometric entities, namely, vertices, edge, faces and volumes. Initial boundaries nodes are placed with the sub-binary technology. When bubbles are placed on the surfaces, rhombic cells with inside angles of  $60^\circ$  and  $120^\circ$  are used instead of square cells in order to realize hexagonal arrangement of the bubbles. The initial placement of bubbles in the isotropic case refers to the literature [Liu, Nie, Zhang, and Wang (2010)]. The difference in the anisotropic case is the oblique direction of elliptical bubble

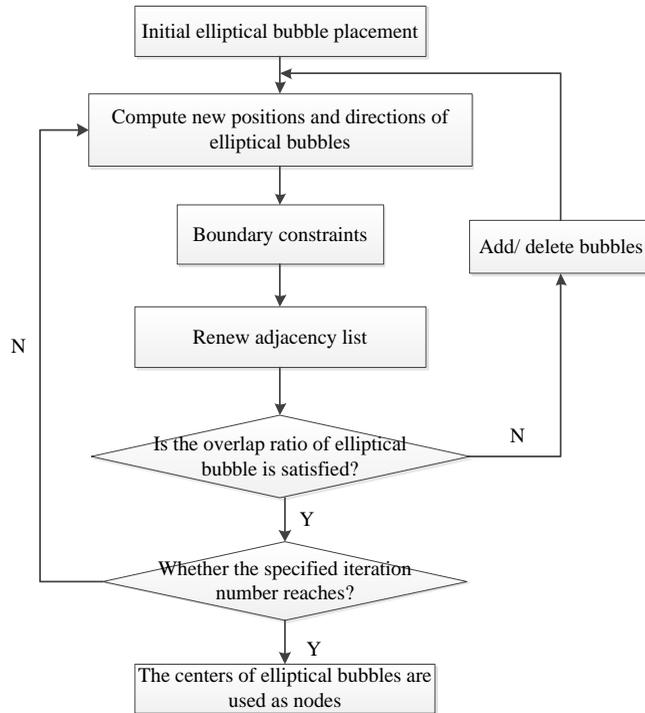


Figure 2: Flow chart of node placement method by elliptical bubble simulation

which is controlled by  $\theta$  in the metric tensor (see Section 2). This scheme has the advantage of being extremely fast and works well according to the desired spacing of nodes. In a square area with a hole, for example, the positions and orientations of the bubbles are controlled by the Riemannian metric

$$M_2 = \begin{bmatrix} \frac{1+4x^2}{h^2} & \frac{4xy}{h^2} \\ \frac{4xy}{h^2} & \frac{1+4y^2}{h^2} \end{bmatrix} \quad (4)$$

and the initial distribution of elliptical bubbles is shown in Fig. 3, Fig. 3a, Fig. 3b and Fig. 3c present initial vertice bubbles, edge bubbles and surface bubbles respectively.

### 3.2 Motion control of elliptical bubbles

According to Newton’s second law of motion, the motion equation of the elliptical bubble is a second order ordinary differential equation [Shimada and Gossard

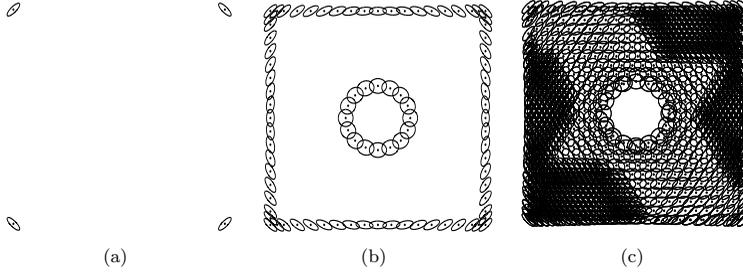


Figure 3: Initial elliptical bubble placement in 2D. (a) Vertice bubbles. (b) Edge bubbles. (c) Surface bubbles.

(1998)]

$$m\ddot{x}_i + c\dot{x}_i = f_i, \quad i = 1 \dots N \tag{5}$$

where  $m$  is the mass of bubble,  $c$  is the damping coefficient,  $N$  is the number of bubbles,  $x_i$  is the center of elliptical bubble  $i$ ,  $-c\dot{x}_i$  is the viscous damping force from the system, which makes the bubble system converge to a stable configuration.  $f_i$  is the resultant force exerted on the bubble  $i$  by its surrounding bubbles

$$f_i = \sum_{j=1, j \neq i}^N f_{ij} \tag{6}$$

$f_{ij}$  is the interaction force that exert on bubble  $i$  from bubble  $j$ . It is time-consuming to compute the interaction forces from other bubbles one by one. Since the interaction force between bubbles is short-range, only neighboring bubbles have the force effect, we just calculate the force from adjacent bubbles, this will greatly reduce the computation time of the resultant force. The searching of neighboring bubble is to be discussed in detail in Section 3.3.

In order to avoid the interaction force from growing infinitely large when the distance between two bubbles is zero, the interaction force is approximated by the 3rd order polynomials [Shimada and Gossard (1998)] (see Fig. 4)

$$f(w) = \begin{cases} k_0 (1.25w^3 - 2.375w^2 + 1.125) & 0 \leq w \leq 1.5 \\ 0 & 1.5 < w \end{cases} \tag{7}$$

instead of the van der Waals force, where  $k_0$  is a const,  $w$  is the ratio of the real distance  $l$  and the desired distance  $l_0$  between two elliptical bubbles, i.e.  $w = l/l_0$ , and the desired distance  $l_0$  is defined as

$$l_0 = l_{ij} + l_{ji} \tag{8}$$

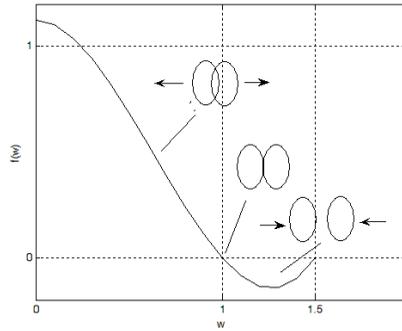


Figure 4: The interaction force between elliptical bubbles

Here  $l_{ij}$  is the distance from bubble  $i$  to the intersection between: (1) the line segment connecting bubble  $i$  and bubble  $j$ , and (2) the boundary of bubble  $i$ . Similarly  $l_{ji}$  is the distance from bubble  $j$  to the intersection between the same line segment and the boundary of bubble  $j$ . From Fig. 4, a repulsive force is applied (shown in Fig. 5(a)) when two elliptical bubbles are located closer than the stable distance  $l_0$  ( $l/l_0 < 1$ ), when  $l/l_0 = 1$ , the interaction force is zero (Fig. 5(b)), or an attractive force is applied (shown in Fig. 5(c)) when the bubbles lie farther apart than  $l_0$  ( $1 < l/l_0 < 1.5$ ), until the attractive force is zero if  $l/l_0 > 1.5$ .

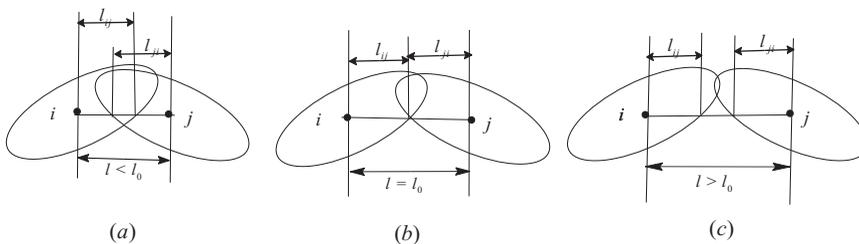


Figure 5: The relationship between force and positions of elliptical bubbles. (a) Repulsive force. (b) Stable. (c) Attractive force.

In the previous bubble meshing method, the high-precision numerical method (the fourth order Runge-Kutta method [Shimada and Gossard (1998)]) is used to solve the equation of motion. In view of the continuous motion of the bubbles during dynamic simulation, Qi et al. [Qi, Nie, and Zhang (2012)] have pointed out that the high-precision numerical solution of every iterative step will inevitably lead to a large number of time consumption, however, the low-precision numerical

method (Euler's method) can save the computing time effectively, although it has little effect on the quality of the nodes. So Euler's method is applied to simulate the motion of the bubbles.

### 3.3 Bubble adjacency list

Since the interaction force between bubbles is short-range force, during dynamic simulation, it is unnecessary to calculate the forces exerted on each bubble by the others one by one, only the forces exerted by adjacent bubbles need to be considered. In order to search the adjacent bubbles around each bubble quickly, an adjacency list for each bubble is built to store the information of adjacent bubbles.

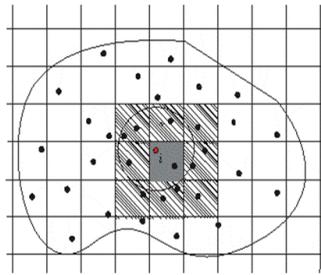


Figure 6: Adjacent bubbles set of a bubble

About the initially establishment of the adjacency list, firstly, the calculation region is divided into a series of small regular areas, these small areas will be referred to as buckets. Each bucket has a bubble list that contains all the bubbles in the bucket, as shown in Fig. 6. Take bubble  $i$  for example, when searching the neighboring bubbles of the bubble  $i$ , the bubbles only covered by the shadow are judged whether lie in the circle centered at the bubble  $i$  with a radius of  $1.7\sigma$ , where  $\sigma$  is the ideal distance between two bubbles, i.e.  $\sigma = l_{ij} + l_{ji}$  (Fig. 5). Meanwhile, the bubble adjacency list should be renewed simultaneously during simulation, from Eq.(7), the interaction force is zero if  $w > 1.5$ , and we set  $w = 1.7$ , so the adjacency list of the bubble  $i$  contains enough adjacent bubbles, and it only need be renewed every  $k$  steps. When updating the adjacency list of bubble  $i$ , for the bubbles of its adjacency list, and the bubbles of the adjacency lists of the neighboring bubbles, checking their new positions whether lie in the circle with the radius of  $1.7\sigma$ , if they are in, as members of the new adjacency list.

### 3.4 Nodes population control

In order to find an appropriate number of bubbles, the population of elliptical bubbles is adaptively adjusted during dynamic simulation, so that bubbles are closely

packed with minimum gaps or overlaps in the final configuration.

The overlap ratio is first used by Shimada [Shimada and Gossard (1998)] in isotropic case to control the number of nodes, based on the adjacency list, the overlap ratio in the anisotropic case is deduced as

$$\alpha_i = \sum_{j=0}^N \frac{1}{l_{ij}} (2l_{ij} + l_{ji} - D_{ij}) \quad (9)$$

where  $l_{ij}$  and  $l_{ji}$  are shown in the Fig. 5,  $D_{ij}$  is the real distance between the centers of bubble  $i$  and bubble  $j$ ,  $N$  is the number of the adjacent bubbles corresponding to the bubble  $i$ , usually  $N = 8$ , however,  $N$  is chosen to be the number of all the bubbles in Shimada's method, which will spent more time to calculate the overlap ratio than ours.

Actually, the overlap ratio describes the number of neighboring bubbles of each bubble. In ideal case, the standard overlap ratio of nodes on a line, on a surface or in an internal volume are 2, 6 and 12, respectively. By computing the overlap ratios of the bubbles, the bubbles are to be deleted automatically when their overlap ratios are too large, or new bubbles are to be added when their overlap ratios are too small. For example, after the dynamic simulation and the adaptive population adjustment of the elliptical bubbles in Fig. 3, the resulting bubbles and the node distribution are shown in Fig. 7.

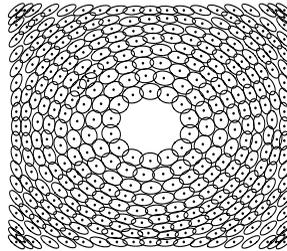


Figure 7: Nodes distribution when bubble system is in the dynamic equilibrium

#### 4 Anisotropic bubble-type local mesh generation

Noting that good node distribution alone cannot guarantee a quality mesh, the way in which the nodes are connected is also an important factor. In references [Nie, Chang, and Fan (2007); Yagawa (2004); Fujisawa, Inaba, and Yagawa (2003)], mesh generation begins by appropriately distributing nodes in the analysis domain. However, the process of node generation has not been mentioned, they need to build

complicated data structure to search neighboring nodes for mesh generation. However, in our algorithm, the node placement by elliptical bubble simulation provides a high-quality nodes set, as well as the adjacency list information, which make the anisotropic mesh generation more simple and efficient, the new local mesh generation method is called ABLMG (Anisotropic Bubble-type Local Mesh Generation).

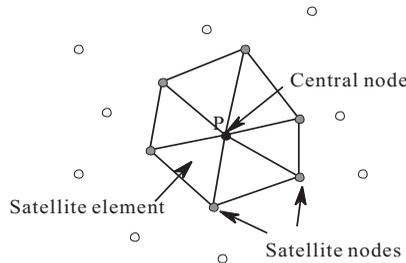


Figure 8: A local mesh associated to a central node  $P$

Following some definitions are to be introduced. A node  $P$  among the distributed nodes is designated as a *central node*, which can be seen in Fig. 8. A local mesh is generated around the central node  $P$  using ABLMG method described later. Elements formed by the local mesh are called *satellite elements*, and a node of a satellite element, associated with the central node  $P$  is considered as a *satellite node*.

The adjacency list of each central node contains its neighboring nodes with  $w < 1.7$ , which includes all satellite nodes and a small number of non-satellite nodes (no more than 2) [Chen, Nie, Zhang, and Wang (2012)]. In Delaunay mesh, the mesh edges are built by connecting the central nodes and their satellite nodes. Therefore, the very small number of non-satellite nodes should be removed from the adjacency list of the central node, and this part will be discussed in the following.

Taking full advantage of the generated nodes set and the adjacency list, the local meshes associated to the central nodes are generated with the ABLMG method by removing the non-satellite nodes, and the combination of local meshes is harmonious global mesh.

#### 4.1 Deleting non-satellite nodes

The deletion of the non-satellite nodes associated to the central nodes will be discussed in this section, the following is the procedure:

1. All the nodes are connected with all their adjacent nodes from their adjacency lists. Since the adjacency list of each central node contains a small

number of non-satellite nodes (no more than 2), and the mesh edges are built by connecting the central nodes and their satellite nodes, the edges which connect the central node and its non-satellite nodes will cross other edges.

2. Sorting the nodes from the adjacency list in counterclockwise order, taking the central node  $P$  for example, to get the sequence ...  $P_{j-1}, P_j, P_{j+1}$ ... which is shown in Fig. 9.
3. Checking node  $P_j$  ( $j=1,2,\dots,n$ ) whether or not to be a satellite node of the central node  $P$ ,  $n$  is the size of the adjacency list. If node  $P_{j-1}$  and  $P_{j+1}$  don't lie in each other's adjacency lists respectively, then node  $P_j$  is a satellite node of the central node  $P$ . Otherwise, according to step 1, node  $P_{j-1}$  and node  $P_{j+1}$  will be connected. Then the intersection test [Li and Hua (2003)] between line segment  $PP_j$  and  $P_{j-1}P_{j+1}$  is implemented.

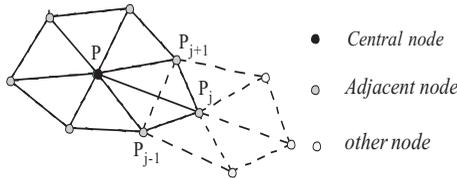


Figure 9: Intersection test for the local mesh of the central node  $P$

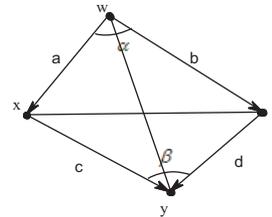


Figure 10: Edge swap

For the step 3, if line segment  $PP_j$  and  $P_{j-1}P_{j+1}$  intersect, Delaunay criteria can be used in the isotropic case to check the position relationship between the node  $P_j$  and the circumscribed circle of  $\Delta PP_{j-1}P_{j+1}$ . However, Delaunay criteria is disabled in the anisotropic case. So the edge swap criterion [Bossen and Heckbert (1996)] is used to remove the non-Delaunay edges which allows us to generate local mesh quickly. In Fig. 10, if  $\angle\alpha + \angle\beta > 180^\circ$ , i.e.  $\sin(\alpha + \beta) < 0$ , the edge  $xz$  should be deleted, the expression can be simplified in the following way in Euclidean geometry

$$\begin{aligned} \sin(\alpha + \beta) &= \sin \alpha \cos \beta + \sin \beta \cos \alpha \\ &= \frac{a \times b}{|a| \cdot |b|} \cdot \frac{c \cdot d}{|c| \cdot |d|} + \frac{c \times d}{|c| \cdot |d|} \cdot \frac{a \cdot b}{|a| \cdot |b|} \\ &\propto (a \times b)(c \cdot d) + (c \times d)(a \cdot b) < 0 \end{aligned} \tag{10}$$

Its Riemannian equivalent is:

$$\sqrt{\det M_{avg}}(a \times b)(c^T M_{avg} d) + \sqrt{\det M_{avg}}(c \times d)(a^T M_{avg} b) < 0 \tag{11}$$

and since  $\sqrt{\det M_{avg}}$  is positive:

$$(a \times b)(c^T M_{avg} d) + (c \times d)(a^T M_{avg} b) < 0 \tag{12}$$

where  $M_{avg} = (M(w) + M(x) + M(y) + M(z))/4$ , and  $u \times v$  is the 2D cross product with scalar value  $u_1 v_2 - u_2 v_1$ . If Eq. (12) holds, the edge  $xz$  should be deleted, the nodes  $x$  and  $z$  will be deleted from each other's adjacency lists respectively. If  $\sin(\alpha + \beta) = 0$ , there are four nodes lying on a same ellipse, and the method will become unstable. To avoid this problem, two end nodes of the line segment with the smallest vertex abscissa value are considered as each other's non-satellite nodes respectively. This ensures the uniqueness of the local meshes, and the union of the local meshes is consistent with the global Delaunay triangulation. If  $\sin(\alpha + \beta) > 0$ , the edge  $yw$  should be deleted, and the nodes  $y$  and  $w$  will be deleted from each other's adjacency lists respectively.

After the above process, for each central node, by eliminating non-satellite nodes in its adjacency list, its corresponding local Delaunay mesh can be obtained. Compared with the local mesh generation method presented in the literature [Yagawa (2004);Fujisawa, Inaba, and Yagawa (2003)], the ABLMG method avoids the establishment of the bucket data structures and the local search process of the adjacent nodes. It generates local meshes more quickly, and all the local meshes are merged to form a harmonious global mesh.

#### **4.2 Anisotropic mesh quality**

In order to test whether the anisotropic mesh meeting the requirements of Riemannian metric tensor  $M_2$ , a relative error is calculated between the desired distance  $l_0$  and the real distance  $l$  of each pair connected nodes, and it is described by  $\varepsilon$

$$\varepsilon = \frac{|l - l_0|}{l_0} \tag{13}$$

$\varepsilon$  more tends to 0, the anisotropic meshes generated by the ABLMG method more reflect the features of their respective metric tensors. To measure the anisotropic triangular element quality, the formulae discussed in [Borouchaki, George, Hecht, Laug, and Saltel (1997);Du and Wang (2005)] are used. All programs in this paper are run in the PC (CPU basic frequency is 2.31 GHZ, 1.00 GB memory), using VC++ 6.0 compiler to compile.

As numerical examples, a union circle region is calculated with the same Riemannian metric tensor as Eq. (4), where  $h = 0.05$ . The final number of nodes and elements are 665 and 1264. The packing elliptical bubbles are shown in Fig. 11(a), and they are tightly packed with minimum gaps or overlaps. Then putting nodes

on the centers of the bubbles, and the obtained nodes set is just we need. Finally using the ABLMG method, the anisotropic mesh is obtained with good configuration and gradualness which is shown in Fig. 11(b). The average values of element quality is 0.952, the maximum value and the minimum values of the element quality are 0.999 and 0.686 respectively. According to Eq. (13) of the edge deviation, let  $e = 5\%$ , the distribution of the edge deviation is shown in Table 1, we can see that the edge deviation factor of more than 90% of all the edges is less than  $e$ , the anisotropic mesh generated with the ABLMG method meets the requirements of Riemannian metric tensor  $M_2$ .

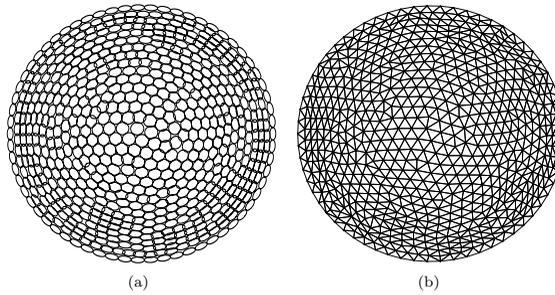


Figure 11: Anisotropic mesh quality examination. (a) The packing elliptical bubbles. (b) Anisotropic bubble-type local mesh generation.

Table 1: Edge deviation factor distribution

	$\leq e$	$e < \varepsilon \leq 2e$	$2e < \varepsilon \leq 3e$	$3e < \varepsilon \leq 4e$	$4e < \varepsilon \leq 5e$
Percentage of edges	0.948466	0.00291	0.004497	0.003175	0.040952

### 4.3 Numerical examples of local mesh generation

In this section, several numerical examples are provided for showing the performance of the ABLMG method. The first example is for a square domain  $[-3, 3]^2$ . Nodes are first generated with elliptical bubble simulation, then each node is connected with its adjacent nodes from its adjacency list, and the initial local meshes are shown in Fig. 12(a). Finally, after 157317 times intersection checking(of which there are 1286 times line segment intersection, intersecting segments occupy 0.82% in intersection test), deleting the non-satellite nodes of each central node, the result mesh can be obtained which is shown in Fig. 12(c), and close-up views at (-1.5,

-1.5) are shown in Fig.12(b) and Fig.12(d). The number of nodes and elements are 2301 and 4525 respectively. The average values of element quality is 0.969094, the maximum value and the minimum values of the element quality are 0.999 and 0.71625, respectively.

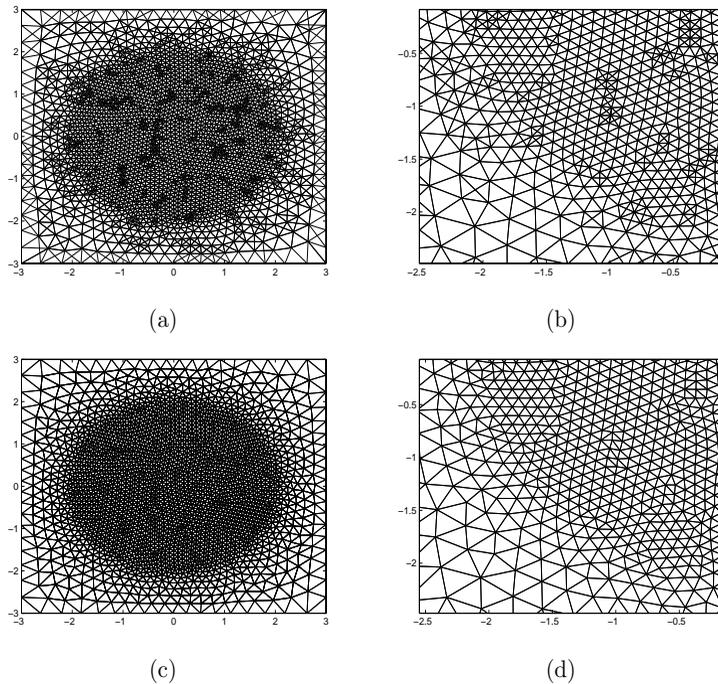


Figure 12: Local meshes generation example. (a) Connecting nodes with all their adjacent nodes. (b) A close-up view of (a) at (-1.5, -1.5). (c) Final mesh after deleting non-satellite nodes of each node. (d) A close-up view of (c) at (-1.5, -1.5).

Another example (shown in Fig. 13) is for a circle domain with radius of 10, and three given metric tensors are as follows: The first is a constant metric tensor that corresponds to  $h_1 = 0.5$ ,  $h_2 = 0.2$ , and  $\theta = \pi/4$ . The second is a nonuniform metric tensor with  $h_1 = 0.15|\sqrt{x^2 + y^2} - 6| + 0.2$ ,  $h_2 = 0.5h_1$ , and  $\theta = 0$ , there is a line refinement at  $\sqrt{x^2 + y^2} = 6$  to be introduced. The last one is a nonuniform metric tensor that corresponds to  $h_1 = 0.1|y| + 0.2$ ,  $h_2 = 0.5h_1$ , and  $\theta = 0$ , this means that a line refinement at  $y = 0$  is to be introduced. After the node placement by elliptical bubble simulation, the packing bubbles are shown in Fig. 13, and their corresponding meshes generated with the ABLMG method are shown in Fig. 14. The average values of element quality for these three meshes are 0.9253, 0.9119,

0.9727, respectively. Obviously, the meshes reflect the features of their respective metric tensors.

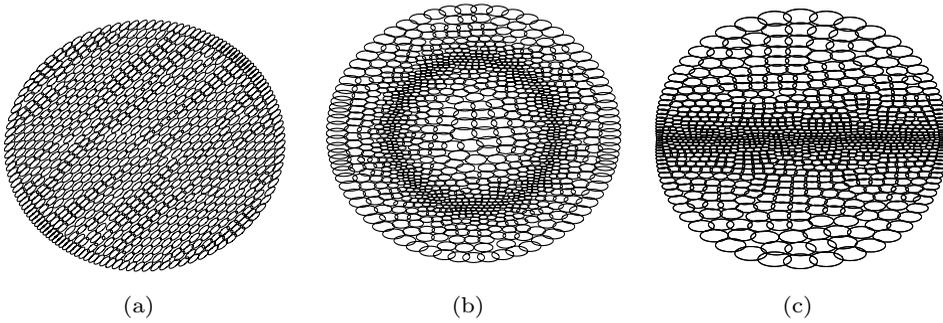


Figure 13: Nodes placement by elliptical bubble simulation with three given metric tensors in a circle. (a)  $h_1 = 0.5$ ,  $h_2 = 0.2$ , and  $\theta = \pi/4$ . (b)  $h_1 = 0.15|\sqrt{x^2 + y^2} - 6| + 0.2$ ,  $h_2 = 0.5h_1$ , and  $\theta = 0$ . (c)  $h_1 = 0.1|y| + 0.2$ ,  $h_2 = 0.5h_1$ , and  $\theta = 0$ .

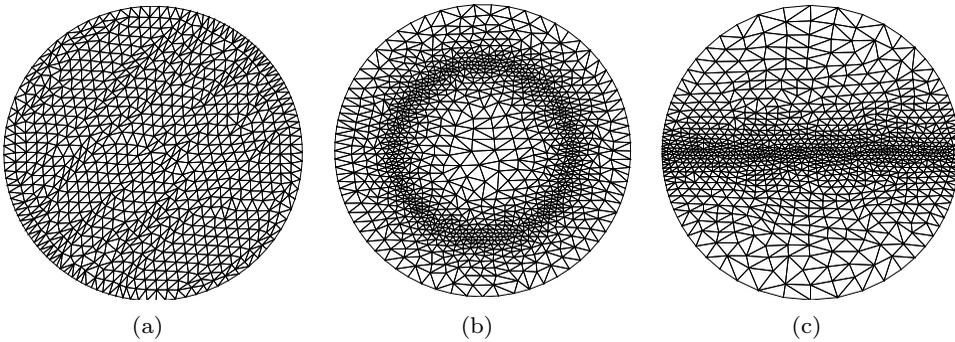


Figure 14: Anisotropic meshes generated with the ABLMG method.

A number of meshing strategies have been developed in the last decade for generating anisotropic meshes according to a given metric tensor, e.g. Ani2D package [Lipnikov and Vasilevski (2005)]. Ani2D generates conformal anisotropic triangular meshes with the prescribed number of triangles in a given metric tensor. The input data for their generator is an initial conformal triangulation, and Ani2D changes the initial mesh through a sequence of local modifications. Compared to the Ani2D mesh generator, our method avoids complicated remeshing, and appropriate number of nodes and elements are generated adaptively according to the given metric

tensor. For example, in a square domain, Ani2D and our method are used respectively to generate anisotropic mesh with a same metric tensor, and both meshes are shown in Fig.15. In the Ani2D packages, the number of nodes and elements are 137 and 232, the average values of element quality is 0.893102, the maximum value and the minimum values of the element quality are 0.997852 and 0.704087, respectively. However, the number of nodes and elements are 132 and 212 in our method, and the average values of element quality is 0.928566, the maximum value and the minimum values of the element quality are 0.999927 and 0.724101, respectively.

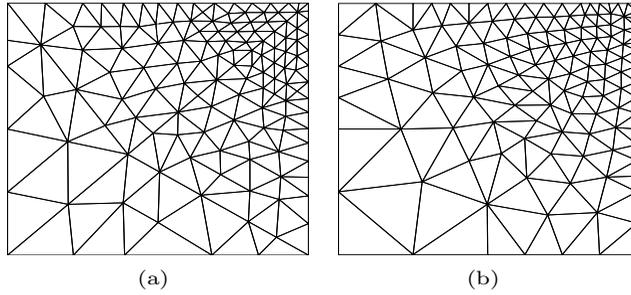


Figure 15: Anisotropic mesh generated by Ani2D and our method. (a) By Ani2D. (b) By our method.

### 5 Application in parametric surface triangulation

In this section, the ABLMG method is used for 3D surface triangulation based on the mapping method [Lee (2001); Lee (2003); Lee (2000)]. The surface to be meshed is represented by a bi-variate mapping such that any node on the 3D surface is mapped to a parametric space. The parametric space is meshed with the ABLMG method, the generated mesh does not need a smoothing process, and it can be directly projected back to the 3D target surface according to the mapping function, the target surface mesh is finally obtained. A Riemannian metric tensor [Lee (2001); Lee (2003); Lee (2000)] is defined so that the desired mesh is obtained in 3D surface. The  $3 \times 3$  metric tensor  $M_3$  is positive-definite, of the form

$$\begin{aligned}
 M_3(P) &= \begin{bmatrix} a(P) & b(P) & c(P) \\ b(P) & d(P) & e(P) \\ c(P) & e(P) & f(P) \end{bmatrix} \\
 &= [e_1 \ e_2 \ e_3] \begin{bmatrix} \lambda_1 & & 0 \\ & \lambda_2 & \\ 0 & & \lambda_3 \end{bmatrix} [e_1 \ e_2 \ e_3]^T
 \end{aligned} \tag{14}$$

which defines element size and direction characteristics, where  $P$  is any node on the 3D surface,  $\lambda_i$  and  $e_i (i = 1, 2, 3)$  is the eigenvalues and the corresponding eigenvectors of  $M_3(P)$  respectively, such that  $e_i \cdot e_j = \delta_{ij}$  and  $\lambda_i > 0 (i = 1, 2, 3)$ ,  $h_i = 1/\sqrt{\lambda_i} (i = 1, 2, 3)$  is the principal element size in the  $e_i$  direction.  $M_3(P)$  is often given based on the surface geometric feature or the numerical requirements, and it also can be got according to the posteriori error estimate in the previous iteration in an adaptive solver.

It is assumed that the target surface  $\Sigma$  to be meshed is represented by a bi-variate mapping

$$r(u, v) = (x(u, v), y(u, v), z(u, v))^T \tag{15}$$

where  $(u, v)$  are the parametric co-ordinates of a node on the surface and  $r$  is the mapping function vector of the parameterization.  $(u, v) \in \Omega$ ,  $\Omega$  is the 2D parametric space. When  $r$  is a  $C^2$  continuous function, the metric tensor corresponding to the node  $P$  in the parametric space is defined as [Lee (2000)]

$$M_2(P) = [r_u, r_v]^T M_3(P) [r_u, r_v] \tag{16}$$

where  $[r_u, r_v]^T = \begin{bmatrix} \partial x/\partial u & \partial y/\partial u & \partial z/\partial u \\ \partial x/\partial v & \partial y/\partial v & \partial z/\partial v \end{bmatrix}$ . The Riemannian structure is induced into the parametric space to determine the distribution of nodes by bubble simulation. Once the metric tensor  $M_2(P)$  is evaluated according to Eq.(16), surface meshing problem in 3D is turned into mesh generation with anisotropy and non-uniform distribution in 2D parametric space, this part is described in Section 4. Meanwhile, once finishing the mesh generation with the ABLMG method in the parametric space, there is no need to smooth the topology and geometry of the mesh, which can be directly projected back to the target surface based on the mapping function, and the surface mesh is obtained finally.

### 6 Numerical example

The following numerical example is given for the quantitative research about the quality of the generated surface mesh. We choose the coefficient  $k_0 = 1.0$ , the mass coefficient  $m = 1.0$ , the viscosity coefficient  $c = 3.8429$ , the time step  $\Delta t = 0.02$ . In this study, calculating and setting the Riemannian metric tensor  $M_3$  in three-dimensional space is a complex matter, so we just generate uniform isotropic surface mesh, and the node spacing for the surface is  $h$ . The Riemannian metric tensor at each node in the following examples are assumed as the following

$$M_3 = \begin{bmatrix} \frac{1}{h^2} & & 0 \\ & \frac{1}{h^2} & \\ 0 & & \frac{1}{h^2} \end{bmatrix} \tag{17}$$

In Fig. 16, the parametric surface is defined by

$$\begin{cases} x = u \\ y = v \\ z = \cos^2(\pi(u^2 + v^2)) \end{cases} \quad (18)$$

the parametric space is a square with  $[-1, 1]^2$ ,  $h = 0.02$ . It can be seen that a tight packing among bubbles without large overlaps or gaps in Fig. 16(a). The nodes set generated by bubble simulation have good configuration and gradualness in the parametric space.

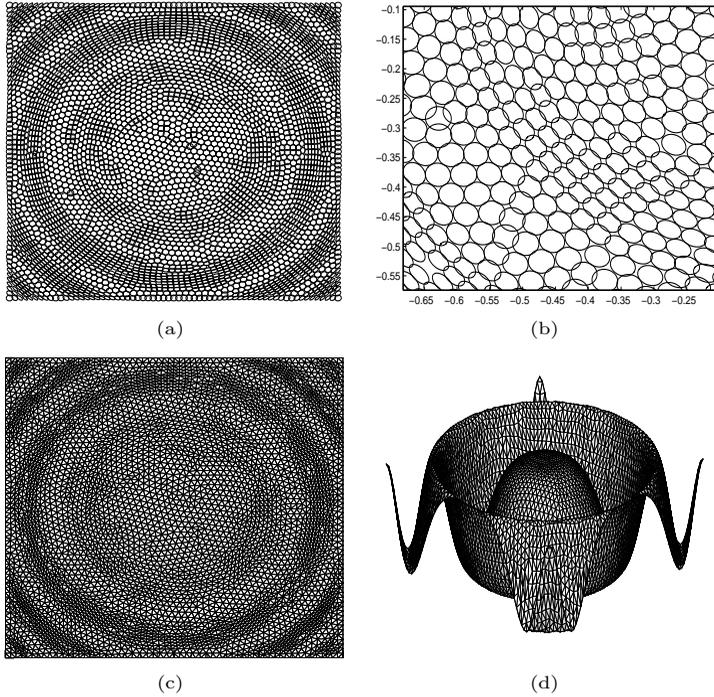


Figure 16: Complex surface mesh generation. (a) Elliptical bubbles in parametric domain. (b) Partial enlarged view of (a). (c) Mesh in parametric domain. (d) Surface mesh.

Mesh quality factor  $\alpha$  for a triangle  $P_1P_2P_3$  proposed in [Lee (2001)] is used to measure the quality of the surface mesh

$$\alpha = \min(\hat{\alpha}(P_1), \hat{\alpha}(P_2), \hat{\alpha}(P_3)) \quad (19)$$

where

$$\hat{\alpha}(P_i) = 2\sqrt{3} \frac{Det(M_i) \cdot Det([P_2 - P_1, P_3 - P_1])}{d(M_i, P_1 P_2)^2 + d(M_i, P_2 P_3)^2 + d(M_i, P_1 P_3)^2} \tag{20}$$

In Eq.(20),  $M_i$  is the surface metric tensor at  $P_i$  while  $Det(M)$  is the determinant of  $M$ ,  $d(M, \cdot \cdot)$  describes the distance under metric tensor  $M$ , it is calculated according to Eq. (3), the parameter  $2\sqrt{3}$  ensures that the quality factor lies in the interval (0, 1),  $\alpha$  is more closer to 1, the quality of the surface mesh element is more better. Table 2 shows the statistic information of the numerical example,  $N$  and  $E$  describe the number of nodes and surface elements respectively, and the quality of elements is measured with Eq.(19) and Eq.(20). The distribution of the elements quality is also presented in Table 2,  $q_{min}$ ,  $q_{max}$  and  $q_{aver}$  are the minimum, maximum and average value respectively. It can be seen that surface mesh in Fig. 16(d) has high overall quality. Although the worst elements quality is less than 0.4, these elements are located at the sharp corners of the original geometry. In order to improve the minimum element quality, appropriate metric tensor must be set.

Table 2: The statistic information of surface mesh quality distribution in numerical examples

Example	N	E	The quality distribution of surface mesh							
			0.0-0.2	0.2-0.4	0.4-0.6	0.6-0.8	0.8-1.0	$q_{min}$	$q_{max}$	$q_{aver}$
Fig. 16	4385	8512	0	0.39%	4.04%	29.9%	65.7%	0.329	0.999	0.899

## 7 Conclusions

A new algorithm for anisotropic local mesh generation has been presented in this paper. Firstly, anisotropic nodes placement method by bubble simulation is applied to optimize nodes distribution, the node spacing is controlled by the Riemannian metric tensor. Since the neighbor node set of every node is stored in its adjacency list, so after the simulation, it is easy to get the neighbor nodes information of each node. This additional information is very valuable for finding the nodes within the influence domain of the integration points in meshless analysis. Secondly, the generated nodes are connected with the ABLMG method, here, we just need remove a handful of non-satellite nodes from the adjacency lists of the nodes, which is very easy to implement, and generates well-shaped elements, meanwhile, the smoothing process isn't required. Finally, the new algorithm for anisotropic mesh generation

is applied on the surface triangulation based on the mapping method, the topological structure of nodes and elements in the parametric space is mapped back to the 3D surface, and high-quality surface mesh is obtained.

There are several areas for future work. The algorithm could probably be sped up. For example, when lower mesh quality is acceptable, the algorithm could do less simulation. Meanwhile, the algorithm should be tested empirically with an FEM solver and its results are compared to other mesh generators.

In addition, since the defined inter-bubble force is short-range, for two distant bubbles, their positions and velocities can be updated simultaneously and independently during simulation, this node placement method by elliptical bubble simulation has inherently parallelism.

**Acknowledgement:** This research was supported by National Natural Science Foundation of China (No: 11071196, 90916027), and the Doctorate Foundation of Northwestern Polytechnical University.

## References

**Bechet, F.; Millet, O.; Sanchez-Palencia, E.** (2008): Adaptive and anisotropic mesh strategy for thin shell problems. Case of inhibited parabolic shells. *International Journal of Solids and Structures*, vol. 46, no. 3-4, pp. 534–556.

**Borouchaki, H.; George, P.; Hecht, F.; Laug, P.; Saltel, E.** (1997): Delaunay mesh generation governed by metric specifications .1. Algorithms. *Finite Element Analysis and Design*, vol. 25, no. 1-2, pp. 61–83.

**Borouchaki, H.; George, P.; Mohammadi, B.** (1997): Delaunay mesh generation governed by metric specifications .2. Applications. *Finite Element Analysis and Design*, vol. 25, no. 1-2, pp. 85–109.

**Borouchaki, H.; Laug, P.; George, P. L.** (2000): Parametric surface meshing using a combined advancing-front generalized Delaunay approach . *International Journal for Numerical Methods in Engineering*, vol. 49, no. 1-2, pp. 233–259.

**Bossen, F. J.; Heckbert, P. S.** (1996): A pliant method for anisotropic mesh generation. *Proceedings of the 5th International Meshing Roundtable, Pittsburgh*, pp. 63–74.

**Chen, M. X.; Lu, B. Z.** (2011): TMSmesh: A Robust Method for Molecular Surface Mesh Generation Using a Trace Technique. *Journal of Chemical Theory and Computation*, vol. 7, no. 1, pp. 203–212.

**Chen, W. W.; Nie, Y. F.; Zhang, W. W.; Wang, L.** (2012): A fast local mesh generation method about high-quality node set. *Chinese Journal of Computational Mechanics*, vol. 29, no. 5, pp. 704–709.

**Du, Q.; Wang, D.** (2005): Anisotropic centroidal voronoi tessellations and their applications. *Siam Journal on Scientific Computing*, vol. 26, no. 3, pp. 737–761.

**Frey, P.; Alauzet, F.** (2005): Anisotropic mesh adaptation for CFD computations. *Computer Methods in applied mechanics engineering*, vol. 194, no. 48-49, pp. 5068–5082.

**Fujisawa, T.; Inaba, M.; Yagawa, G.** (2003): Parallel computing of high-speed compressible flows using a node-based finite-element method. *International Journal for Numerical Methods in Engineering*, vol. 58, no. 3, pp. 451–511.

**Habashi, W.; Dompierre, J.; Bourgault, Y.; Yahia, D.; Fortin, M.; Vallet, M.** (2000): Anisotropic mesh adaptation: towards user-independent, mesh-independent and solver-independent CFD. Part I: general principles. *International Journal for Numerical Methods in Fluids*, vol. 32, no. 6, pp. 725–744.

**Lee, C. K.** (2000): Automatic metric advancing front triangulation over curved surfaces. *Engineering Computations*, vol. 17, no. 1, pp. 48–74.

**Lee, C. K.** (2001): On curvature element-size control in metric surface mesh generation. *International Journal for Numerical Methods in Engineering*, vol. 50, no. 4, pp. 787–807.

**Lee, C. K.** (2003): Automatic metric 3D surface mesh generation using subdivision surface geometrical model. Part 1: Construction of underlying geometrical model. *International Journal for Numerical Methods in Engineering*, vol. 56, no. 11, pp. 1593–1614.

**Legrand, S.; Deleersnijder, E.; Delhez, E.; Legat, V.** (2007): Unstructured, anisotropic mesh generation for the Northwestern European continental shelf, the continental slope and the neighbouring ocean. *Continental Shelf Research*, vol. 27, no. 9, pp. 1344–1356.

**Leicht, T.; Hartmann, R.** (2010): Error estimation and anisotropic mesh refinement for 3d laminar aerodynamic flow simulations. *Journal of Computational Physics*, vol. 229, no. 19, pp. 7344–7360.

**Li, Y. H.; Hua, Y. X.** (2003): A method to quickly assessing the intersection of line segments. *Bulletin of Surveying and Mapping*, , no. 7, pp. 30–31.

**Lipnikov, K.; Vasilevski, Y.** (2005): Generators of anisotropic adaptive grids. <http://math.lanl.gov/lipnikov/Research/AniGrids/AniGrids.html>.

**Liu, Y.; Nie, Y. F.; Zhang, W. W.; Wang, L.** (2010): Node placement method by bubble simulation and its application. *CMES-Computer Modeling in Engineering & Sciences*, vol. 55, no. 1, pp. 89–109.

**Lo, S.; Wang, W.** (2005): Generation of anisotropic mesh by ellipse packing over an unbounded domain. *Engineering with Computers*, vol. 20, no. 4, pp. 372–383.

**Miranda, A. C. O.; Martha, L. F.; Wawrzynek, P. A.; Ingraffea, A. R.** (2009): Surface mesh regeneration considering curvatures. *Engineering with Computers*, vol. 25, no. 2, pp. 207–219.

**Nie, Y. F.; Chang, S.; Fan, X. K.** (2007): The Parallel Mechanism of Node-Based Seamless Finite Element Method. *CMES: Computer Modeling in Engineering & Sciences*, vol. 19, no. 2, pp. 135–142.

**Peraire, J.; Peiro, J.** (1992): Adaptive remeshing for three-dimensional compressible flow computations. *Journal of Computational Physics*, vol. 103, no. 2, pp. 269–285.

**Qi, N.; Nie, Y. F.; Zhang, W. W.** (2012): A Fast Node Placement Method by Bubble Simulation. *Chinese Journal of Computational Physics*, vol. 29, no. 3, pp. 333–339.

**Shimada, K.; Gossard, D. C.** (1998): Automatic triangular mesh generation of trimmed parametric surfaces for finite element analysis. *Computer Aided Geometric Design*, vol. 15, no. 3, pp. 199–222.

**Wu, B. H.; Wang, S. J.** (2005): Automatic triangulation over three-dimensional parametric surfaces based on advancing front method. *Finite Element Analysis and Design*, vol. 41, no. 9-10, pp. 892–910.

**Yagawa, G.** (2004): Node-by-node parallel finite elements: a virtually meshless method. *International Journal for Numerical Methods in Engineering*, vol. 60, no. 1, pp. 69–102.

**Yamakawa, S.; Shimada, K.** (2003): Anisotropic tetrahedral meshing via bubble packing and advancing front. *International Journal for Numerical Methods in Engineering*, vol. 57, no. 13, pp. 1923–1942.

**Yokosuka, Y.; Imai, K.** (2009): Guaranteed-quality anisotropic mesh generation for domains with curved boundaries. *Computer-Aided Design*, vol. 41, no. 5, pp. 385–393.

