INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 16, Number 5, Pages 745–766

ANISOTROPIC MESH ADAPTATION METHOD BASED ON ANISOTROPIC BUBBLE-TYPE LOCAL MESH GENERATION

WEI GUO, YUFENG NIE AND WEIWEI ZHANG*

Abstract. A new anisotropic adaptive mesh refinement method based on anisotropic bubble-type local mesh generation (ABLMG) for elliptic partial differential equations is proposed. The anisotropic meshes are generated as quasi-uniform meshes in metric spaces with the metric determined on each vertex by anisotropic a posteriori error estimator. Under the new metric, the error is equidistributed in the directions of maximum and minimum stretching on an element, and the mesh size is reduced/coarsend in regions with large/small errors. With the full use of the adjacent lists provided by the node placement method, the local mesh for each vertex is generated through ABLMG method. Compared with other methods, the mesh refining and coarsening can be obtained in the same framework and the mesh suits the metric well at each refinement level. Numerical results in two-dimensions are presented to verify the ability of our metric tensor to generate anisotropic mesh with correct concentration and stretching direction.

Key words. Metric tensor, anisotropic mesh, adaptive finite element, node placement and local mesh generation.

1. Introduction

The advantage of anisotropic adaptive mesh refinement method has been amply demonstrated for improving computational efficiency and enhancing the solution accuracy, especially for the problems with anisotropic features. Through adapting the mesh size, shape and orientation, the mesh can be refined both in regions and directions with large errors. The use of anisotropic mesh refinement method involves several key factors: error estimates, determination of metric tensor and anisotropic mesh generation. Beginning with the pioneering work of D'Azevedo[7] and Simpson[27], these techniques have been developed by many researchers[24, 3, 5, 9, 12, 18].

Deriving an efficient and reliable a posteriori error estimator is a difficult task on highly anisotropic mesh. Two requirements must be satisfied for anisotropic error estimators. The error estimator must perform well on anisotropic meshes and should provide the directional information to refine the mesh with large errors. Unfortunately, the classical isotropic a posteriori error estimators can't suit the requirements. For isotropic error estimator, the effectivity index of estimator depends on the mesh aspect ratio which is unbounded for anisotropic mesh. Since the early nineties of last century, many anisotropic a posteriori error estimators have been proposed, for example, the hierarchical a posteriori error estimator[15], the dual weighted residual estimator[11], local problem estimates[2] and so on. In order to specify the refinement direction most of the present error estimators make use of the gradient or Hessian matrix of the solution which are unavailable in numerical computation. To avoid the difficulty, the information of the solution is approximated by the recovery technique such as the Zienkiewicz-Zhu post-processing[34, 35]. It is worth pointing out that although no convergence can be certified in anisotropic

Received by the editors July 16, 2018 and, in revised form, March 3, 2019.
2000 *Mathematics Subject Classification.* 65N30, 65N50.
*Corresponding author.

mesh, numerical results show that the information obtained from recovery technique can be used to guide refinement and coarsening for anisotropic adaptive mesh refinement.

Obtaining the metric from a posteriori error estimator to guide the mesh generation is important for anisotropic mesh generation. Different from isotropic mesh refinement where only the mesh size need to be optimized, for anisotropic mesh refinement method the shape and orientation also need to be optimized. All of them are described by the metric tensor, determined by error estimates. A number of strategies have been developed to deduce metric tensors. The metric tensors are commonly defined as the Hessian matrix of the solution proposed by D'Azevedo[7]. Based on error estimates for polynomial preserving interpolation estimation, Huang[14] developed a general formula for the metric tensor. For the anisotropic elliptic problems Huang et al.[16] verified that high-accuracy finite element solution and superconvergence on the mesh vertices can be obtained by utilizing the inverse of the anisotropic diffusion matrix as the metric tensor for anisotropic mesh generation.

For high-quality anisotropic adaptive mesh generation, three basic approaches exist to achieve mesh refinement: mesh smoothing, anisotropic re-meshing and mesh splitting[24]. For mesh smoothing method, the nodes are relocated at each refinement level to minimize the error estimates. For example, Schneider and Jimack[25] introduced a new anisotropic mesh adaptation strategy in order to modify the node positions of a given (isotropic) mesh such that the a posteriori error estimate is minimized. However how to choose the initial mesh vertex number for the mesh smoothing method is still an open problem. Mesh splitting is a canonical way to refine the mesh for isotropic adaptive mesh refinement method. Whereas the strong anisotropic mesh can't be obtained no matter which kind of splitting strategies are used (the longest edge bisection method or the newest vertex bisection method), since splitting methods limit the aspect ratio for anisotropic mesh. Many researchers[28] show that the anisotropy of the mesh can be increased by pre-defined refinement patterns. Anisotropic re-meshing method requires generating new anisotropic mesh at each refinement level. The mesh with high quality and strong anisotropy can be arrived in fewer steps. For instance the anisotropic centroidal Voronoi tessellation (ACVT) have been developed by Du and Wang[10] for two dimensional anisotropic mesh generation and optimization. There are also a number of computer codes including BL2D[19], BAMG[13], and MMG3D[8] for generating anisotropic meshes which lead to a large number of publications.

In this paper, the focus is on the anisotropic adaptive mesh refinement method based on anisotropic bubble-type local mesh generation (ABLMG) method. The ABLMG-based adaptive mesh generation method proposed in this paper is an anisotropic re-meshing method. Initially, Shimada et al. [26] proposed the bubble packing method (BPM) based on the fact that the force-balance configuration of bubbles forms a well designed node set. The BPM can be used to generate the anisotropic mesh such as the parametric surface mesh [32] and polygonal surface mesh [29], in which the circle bubbles are replaced by ellipse bubbles. In order to avoid using mesh topology, a pure node placement method by bubble simulation (NPBS) was proposed by Liu et al.[20] in which the adjacent list structure is set up to reduce the time of calculating interaction forces. For the node set with high quality generated by NPBS, a fast bubble-type local mesh generation method (BLMG) is presented in [6] and the anisotropic version (ABLMG) is presented in

[31]. The BLMG method also has great potential in adaptive mesh refinement combining with the a posteriori error estimator[30]. While the use of ABLMG method in adaptive mesh refinement has not been explored, thus in the study, we aim to develop a anisotropic adaptive finite element method with the ABLMG for elliptic problems.

A brief outline of our method is now given. First, a coarse mesh with uniform mesh size is generated, then anisotropic adaptive mesh refinement process is executed iteratively until the solution error requirement is satisfied. The iterative process consists of computing the finite element solution on the current mesh, computing an anisotropic a posteriori error estimator, determining the metric tensor and reconstructing of the new mesh. At each refinement level, based on the anisotropic a posteriori error estimator the new metric tensor is defined on each vertex to determine the mesh size, shape and orientation, meahwhile, the corresponding mesh metric field function is obtained by linear interpolation. The nodes are relocated to near-optimal positions according to the Newton's second law of motion. Finally, The patch of elements is formed through the ABLMG method around each node, and the union of the patches forms the anisotropic Delaunay triangulation of the domain.

The rest of the paper is organised as follows. In Section 2, the anisotropic a posteriori error estimator proposed by Picasso[21, 22, 23] is reviewed. In Section 3, the strategy for determining the metric tensor is firstly given according to the anisotropic a posteriori error estimator. Then the key points of node placement method are described and the ABLMG method is introduced. In Section 4 our anisotropic mesh adaptation method is described in detail, and some numerical examples are presented in Section 5. Finally, conclusions and future work are given in Section 6.

2. Anisotropic a posteriori error estimates

Let Ω be a polygon of R^2 with boundary $\partial\Omega$. The second order elliptic partial differential equation

$$\begin{cases} -\Delta u = f & \text{in}\,\Omega \\ u = 0 & \text{on}\,\partial\Omega \end{cases} \tag{1}$$

where $f \in L^2(\Omega)$ is considered.

Let $\mathscr{T}_h$ be a triangulation of Ω with mesh size h. Let V_h be the space of continuous piecewise linear finite element space associated with triangulation $\mathscr{T}_h$. The finite element approximation corresponding to the model problem (1) is: find $u_h \in V_h$ such that

$$\int_\Omega \nabla u_h \cdot \nabla v_h \,\mathrm{d}\Omega = \int_\Omega f v_h \,\mathrm{d}\Omega \quad \forall v_h \in V_h \tag{2}$$

In order to describe the mesh anisotropy in triangulation $\mathscr{T}_h$, the frame work in [11, 12] is used. Let $T_K : \hat{K} \to K$ be the standard invertible affine map from reference element $\hat{K}$ to the element K in the triangulation $\mathscr{T}_h$. Let $\mathbf{M}_K \in R^{2\times 2}$ and $\mathbf{t}_K \in R^2$ be the matrix and the vector such that

$$\mathbf{x} = T_K(\hat{\mathbf{x}}) = \mathbf{M}_K \hat{\mathbf{x}} + \mathbf{t}_K \tag{3}$$

where $\hat{\mathbf{x}} \in \hat{K}$ and $\mathbf{x} \in K$.

Since $\mathbf{M}_K$ is invertible, it admits a singular value decompose $\mathbf{M}_K = R_K^T \Lambda_K P_K$, where R_K and P_K are orthogonal matrices and Λ_K is diagonal matrix with positive

entries. We set the matrix with the following form

$$\Lambda_K = \begin{pmatrix} \lambda_{1,K} & 0 \\ 0 & \lambda_{2,K} \end{pmatrix} \quad \text{and} \quad R_K = \begin{pmatrix} \mathbf{r}_{1,K} \\ \mathbf{r}_{2,K} \end{pmatrix} \tag{4}$$

chosen $\lambda_{1,K} \geq \lambda_{2,K}$

Let $e_h = u - u_h$ be the error of the approximation solution u_h. Then we have

$$\|\nabla(u - u_h)\|^2_{L^2(\Omega)} = \int_\Omega f(u - u_h)\,\mathrm{d}\Omega - \int_\Omega \nabla u_h \cdot \nabla(u - u_h)\,\mathrm{d}\Omega \tag{5}$$

Let $I_h : H^1(\Omega) \to V_h$ be a clément interpolation operator. Utilizing the Galerkin orthogonality property we have

$$\begin{aligned} \int_\Omega fv\,\mathrm{d}\Omega - \int_\Omega \nabla u_h \cdot \nabla v\,\mathrm{d}\Omega &= \int_\Omega f(v - I_h v)\,\mathrm{d}\Omega - \int_\Omega \nabla u_h \cdot \nabla(v - I_h v)\,\mathrm{d}\Omega \\ &= \sum_{K \in \mathscr{T}_h} \int_K (f + \Delta u_h)(v - I_h v)\,\mathrm{d}\Omega \\ &\quad + \frac{1}{2} \sum_{\gamma \in \epsilon_I} \int_\gamma [\frac{\partial u_h}{\partial n}](v - I_h v)\,\mathrm{d}s \end{aligned} \tag{6}$$

where ϵ_I denotes the set of interior edges of $\mathscr{T}_h$ and $[\cdot]$ defines the jump across edge γ. Using Cauthy-Schwarz ineuqlity and the interpolation estimation results about I_h in [11], we can get

$$\begin{aligned} \int_\Omega fv\,\mathrm{d}\Omega - \int_\Omega \nabla u_h \cdot \nabla v\,\mathrm{d}\Omega \leq\ & C(\hat{K}) \left(\lambda^2_{1,K} \mathbf{r}^T_{1,K} G_K(v) \mathbf{r}_{1,K} + \lambda^2_{2,K} \mathbf{r}^T_{2,K} G_K(v) \mathbf{r}_{2,K}\right)^{1/2} \\ & \times \sum_{K \in \mathscr{T}_h} \left(\|f + \Delta u_h\|_{L^2(K)} + \frac{1}{2} \left(\frac{h_K}{\lambda_{1,K} \lambda_{2,K}} \right)^{\frac{1}{2}} \left\| \frac{\partial u_h}{\partial n} \right\|_{L^2(\partial K)} \right) \end{aligned} \tag{7}$$

where $G_K(v)$ denotes the matrix

$$G_K(v) = \sum_{T \in \Delta_K} \begin{pmatrix} \int_T \left(\frac{\partial v}{\partial x_1}\right)^2 \mathrm{d}x & \int_T \frac{\partial v}{\partial x_1} \frac{\partial v}{\partial x_2}\,\mathrm{d}x \\ \int_T \frac{\partial v}{\partial x_1} \frac{\partial v}{\partial x_2}\,\mathrm{d}x & \int_T \left(\frac{\partial v}{\partial x_2}\right)^2 \mathrm{d}x \end{pmatrix}. \tag{8}$$

and Δ_K is the set of triangles sharing a vertex with triangle K.

Considering the fact [12] that

$$\lambda_{2,K} h_{\hat{K}} \leq h_K \leq \lambda_{1,K} h_{\hat{K}}, \tag{9}$$

we have

$$\begin{aligned} \int_\Omega fv\,\mathrm{d}\Omega - \int_\Omega \nabla u_h \cdot \nabla v\,\mathrm{d}\Omega \leq & C(\hat{K}) \sum_{K \in \mathscr{T}_h} \left(\|f + \Delta u_h\|_{L^2(K)} + \frac{1}{2\lambda^{1/2}_{2,K}} \left\| \frac{\partial u_h}{\partial n} \right\|_{L^2(\partial K)} \right) \\ & \times \left(\lambda^2_{1,K} \mathbf{r}^T_{1,K} G_K(v) \mathbf{r}_{1,K} + \lambda^2_{2,K} \mathbf{r}^T_{2,K} G_K(v) \mathbf{r}_{2,K}\right)^{1/2} \end{aligned} \tag{10}$$

Substituting e_h in place of v, we can obtain the estimation result

$$\begin{aligned}&\|\nabla(u-u_h)\|^2_{L^2(\Omega)}\\ \le& C\sum_{K\in\mathscr{T}_h}\rho_K(u_h)\left(\lambda^2_{1,K}\mathbf{r}^T_{1,K}G_K(u-u_h)\mathbf{r}_{1,K}+\lambda^2_{2,K}\mathbf{r}^T_{2,K}G_K(u-u_h)\mathbf{r}_{2,K}\right)^{1/2}\end{aligned}\tag{11}$$

where $\rho_K(u_h)=\|f+\Delta u_h\|_{L^2(K)}+\frac{1}{2\lambda^{1/2}_{2,K}}\left\|\frac{\partial u_h}{\partial n}\right\|_{L^2(\partial K)}$.

Since u still exists in right hand side, the right hand of inequality (11) can't be used as a posteriori error estimator directly . The recovery technique is employed here to calculate the quantity $\frac{\partial(u-u_h)}{\partial x_i}$. The simplest Z-Z recovery method defined in [34, 35] is considered, and $\frac{\partial(u-u_h)}{\partial x_i}$ is approximated by

$$\Pi_h\frac{\partial u_h}{\partial x_i}-\frac{\partial u_h}{\partial x_i}$$

where Π_h at vertex P is defined as

$$\Pi_h\left(\frac{\partial u_h}{\partial x_i}\right)(P)=\frac{1}{\sum_{\substack{K\in\mathscr{T}_h\\P\in K}}|K|}\sum_{\substack{K\in\mathscr{T}_h\\P\in K}}|K|\left(\frac{\partial u_h}{\partial x_i}\right)_{|K}$$

Therefore, the matrix $G_K(u-u_h)$ can be approximated by $\tilde{G}_K(u_h)$ defined as

(12)

$$\tilde{G}_K(u_h)=\begin{pmatrix}\int_T\left(\Pi_h\frac{\partial u_h}{\partial x_1}-\frac{\partial u_h}{\partial x_1}\right)\left(\Pi_h\frac{\partial u_h}{\partial x_1}-\frac{\partial u_h}{\partial x_1}\right)\mathrm{d}x & \int_T\left(\Pi_h\frac{\partial u_h}{\partial x_1}-\frac{\partial u_h}{\partial x_1}\right)\left(\Pi_h\frac{\partial u_h}{\partial x_2}-\frac{\partial u_h}{\partial x_2}\right)\mathrm{d}x\\ \int_T\left(\Pi_h\frac{\partial u_h}{\partial x_1}-\frac{\partial u_h}{\partial x_1}\right)\left(\Pi_h\frac{\partial u_h}{\partial x_2}-\frac{\partial u_h}{\partial x_2}\right)\mathrm{d}x & \int_T\left(\Pi_h\frac{\partial u_h}{\partial x_2}-\frac{\partial u_h}{\partial x_2}\right)\left(\Pi_h\frac{\partial u_h}{\partial x_2}-\frac{\partial u_h}{\partial x_2}\right)\mathrm{d}x\end{pmatrix}$$

Finally we can get the anisotropic a posteriori error estimator defined as

$$(\eta^A_K)^2=\rho_K(u_h)\left(\lambda^2_{1,K}\mathbf{r}^T_{1,K}\tilde{G}_K(u_h)\mathbf{r}_{1,K}+\lambda^2_{2,K}\mathbf{r}^T_{2,K}\tilde{G}_K(u_h)\mathbf{r}_{2,K}\right)^{1/2}\tag{13}$$

Remark 1. In isotropic case, the error estimate is shown as follows

$$\|\nabla(u-u_h)\|^2_{L^2(\Omega)}\le C\left(\sum_{T\in\mathscr{T}_h}h^2_K\|f+\Delta u_h\|^2_{L^2(K)}+\frac{1}{2}\left\|\left[\frac{\partial u_h}{\partial n}\right]\right\|^2_{L^2(\partial\Omega)}\right)^{1/2}\tag{14}$$

and the corresponding isotropic a posteriori error estimator is

$$(\eta^I_K)^2=h^2_K\|f+\Delta u_h\|^2_{L^2(K)}+\frac{1}{2}h_K\left\|\left[\frac{\partial u_h}{\partial n}\right]\right\|^2_{L^2(\partial K)}\tag{15}$$

It is worth noting that C in Equation (14) depends on aspect ratio of the element and this is the reason why the isotropic a posteriori error estimator will blow up on anisotropic mesh.

3. Anisotropic adaptive mesh refinement

A good error estimator, a reliable metric tensor and an appropriate mesh refinement strategy are three important factors influencing the efficiency of adaptive mesh refinement method. In this study, the estimator defined in Equation (13) is used. The metric determination method is then discussed in detail, and the anisotropic adaptive mesh refinement method is introduced in the following.

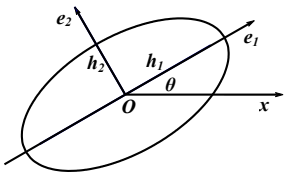

FIGURE 1. Elliptical bubble controlled by Riemannian metric tensor.

3.1. Determination of metric tensor. For two dimensional anisotropic mesh generation, the element size, shape and orientation are described by a 2×2 Riemannian metric tensor $M(\mathbf{x})$ which is a symmetric definite positive matrix function. Let $\mu_1(\mathbf{x})$ and $\mu_2(\mathbf{x})$ define the eigenvalue functions for $M(\mathbf{x})$, $e_1(\mathbf{x})$ and $e_2(\mathbf{x})$ are the corresponding unit orthogonal eigenvector functions, thus the metric tensor can be decomposed as

$$
\begin{aligned}
M(\mathbf{x}) &= \begin{bmatrix} e_1(\mathbf{x}) & e_2(\mathbf{x}) \end{bmatrix} \begin{bmatrix} \mu_1(\mathbf{x}) & 0 \\ 0 & \mu_2(\mathbf{x}) \end{bmatrix} \begin{bmatrix} e_1(\mathbf{x}) & e_2(\mathbf{x}) \end{bmatrix}^T \\
&= \begin{bmatrix} \cos\theta(\mathbf{x}) & -\sin\theta(\mathbf{x}) \\ \sin\theta(\mathbf{x}) & \cos\theta(\mathbf{x}) \end{bmatrix} \begin{bmatrix} \frac{1}{h_1^2(\mathbf{x})} & 0 \\ 0 & \frac{1}{h_2^2(\mathbf{x})} \end{bmatrix} \begin{bmatrix} \cos\theta(\mathbf{x}) & \sin\theta(\mathbf{x}) \\ -\sin\theta(\mathbf{x}) & \cos\theta(\mathbf{x}) \end{bmatrix}
\end{aligned} \tag{16}
$$

where $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$ are the mesh sizes along the direction $e_1(\mathbf{x})$ and $e_2(\mathbf{x})$ at any point $\mathbf{x}$. For node placement method by bubble simulation, $h_1(\mathbf{x})$ and $h_2(\mathbf{x})$ denote the lengths of two radii, and $\theta(\mathbf{x})$ describes the angle of the elliptical bubble with the Ox direction, shown in Figure 1.

The bubble placement method requires the metric at each mesh vertex to be given, namely for each vertex $\mathbf{x}_i$, the elliptical radii $h_{i,1}$, $h_{i,2}$ and the angle θ_i shoule be calculated. For simplicity the aspect ratio $s_i = \frac{h_{i,1}}{h_{i,2}}$ is introduced. So there are three quantities at vertex $\mathbf{x}_i$ need to be determined $h_{i,1}$, s_i and θ_i.

Let $\mathscr{T}^{(l)}$ denote the mesh of Ω with vertices $\left\{\mathbf{x}_i^{(l)}\right\}_{i=1}^{N^{(l)}}$ at the lth refinement level, and $N^{(l)}$ is the number of vertices. Let $\eta_K^{(l)}$ denote the anisotropic a posteriori error estimator on triangle K defined in Equation (13). According to the estimator, the metric tensor of the current mesh will be modified and the new metric function for the $(l+1)$th refinement level is denoted as $M^{(l+1)}$.

Firstly the strategy for determining the orientation is introduced. For estimator (13), Formaggia and Perotto[11] have proved that the estimator is minimum when the element is aligned with the eigenvectors of matrix $\tilde{G}(u_h)$, i.e. the maximum streching direction is taken as the eigenvetor of $\tilde{G}(u_h)$ corresponding to the minimal eigenvalue. So the metric is determined such that the element is aligned with the eigenvectors of $\tilde{G}(u_h)$. We compute $\tilde{G}_i(u_h)$, the average value of $\tilde{G}_K(u_h)$ on the triangles surrounding the vertex $\mathbf{x}_i$

$$
\tilde{G}_i(u_h) = \frac{\sum\limits_{T \in \tilde{T}_i^{(l)}} \tilde{G}_T^{(l)}}{\sum\limits_{T \in \tilde{T}_i^{(l)}} 1}
$$

where $\tilde{T}_i^{(l)}$ represents the union of elements that share the common vertex $\mathbf{x}_i^{(l)}$ of $\mathscr{T}^{(l)}$. We aim to align the triangles around vertex $\mathbf{x}_i^{(l)}$ with the eigenvectors

of $\tilde{G}_i(u_h)$, $e_{i,1}^{(l+1)}$ and $e_{i,2}^{(l+1)}$ where $e_{i,1}^{(l+1)}$ is the eigenvector corresponding to the maximum eigenvalue of $\tilde{G}_K(u_h)$. So the eigenvectors of metric tensor at vertex $\mathbf{x}_i^{(l)}$ is $e_{i,2}^{(l+1)}$ and $e_{i,1}^{(l+1)}$.

Then the way of determining the aspect ratio is considered. Both the analysis in [23] and numerical results in [21] verify that the estimator (13) is reliable and efficient on anisotropic meshes provided that

$$\lambda_{1,K}^2 \mathbf{r}_{1,K}^T \tilde{G}_K(u_h) \mathbf{r}_{1,K} = \lambda_{2,K}^2 \mathbf{r}_{2,K}^T \tilde{G}_K(u_h) \mathbf{r}_{2,K} \tag{17}$$

In other words, the adaptive algorithm should guarantee that the error in minimal and maximal stretching directions is equidistributed. So the aspect ratio for element K is defined as

$$s_K = \sqrt{\frac{\mathbf{r}_{2,K}^T \tilde{G}_K(u_h) \mathbf{r}_{2,K}}{\mathbf{r}_{1,K}^T \tilde{G}_K(u_h) \mathbf{r}_{1,K}}}$$

The aspect ratio on vertex $\mathbf{x}_i^{(l)}$ is defined as,

$$s_i^{(l+1)} = \frac{\sum\limits_{T \in \tilde{T}_i^{(l)}} s_K}{\sum\limits_{T \in \tilde{T}_i^{(l)}} 1}. \tag{18}$$

Finally the strategy for determining mesh size is introduced. The anisotropic estimator at vertex $\mathbf{x}_i$ is defined by area weighted average

$$\eta_i^{(l)} = \frac{\sum\limits_{K \in \tilde{T}_i^{(l)}} \eta_K^{(l)} |T|}{\sum\limits_{K \in \tilde{T}_i^{(l)}} |T|}$$

For adaptive bubble placement, the bubble area should be reduced in regions with large error estimates, and enlarged in regions with small error estimates. The area of elliptic bubble $\mathbf{x}_i$ is

$$r_i^{(l)} = \pi h_{i,1}^{(l)} h_{i,2}^{(l)} = \pi \frac{\left(h_{i,1}^{(l)}\right)^2}{s_i^l} \tag{19}$$

The new area of bubble $\mathbf{x}_i$ is $r_i^{(l+1)}$ which is determined as follows

$$r_i^{(l+1)} = \frac{r_i^{(l)}}{f(\eta_i^{(l)})} \tag{20}$$

where $f(\eta_i^{(l)})$ is a user function to control the extent of mesh refinement. Utilizing the notation of aspect ratio, we can get the new size of $\mathbf{x}_i$ is

$$h_{i,1}^{(l+1)} = \sqrt{\frac{s_i^{(l+1)}}{s_i^{(l)} f(\eta_i^{(l)})}} h_{i,1}^{(l)} \tag{21}$$

$$h_{i,2}^{(l+1)} = \sqrt{\frac{1}{s_i^{(l+1)} s_i^{(l)} f(\eta_i^{(l)})}} h_{i,1}^{(l)} \tag{22}$$

$f(\eta_i^{(l)})$ is a user function defined by

$$f(\eta_i^{(l)}) = \max\left(\min\left(\frac{\eta_i^{(l)}}{c^{(l)}\bar{\eta}^{(l)}}, a\right), b\right) \tag{23}$$

where $\bar{\eta}^{(l)}$ denotes the mean of the anisotropic a posteriori error estimator over all the vertex at the lth refinement level, $c^{(l)}$ is a coefficient close to 1.0, and the parameter a and b denote the bubble size refinement and coarsening factor thresholds, respectively.

The ratio $\frac{\eta_i^{(l)}}{\bar{\eta}^{(l)}}$ between the error at the vertex $\mathbf{x}_i$ and the avarage error reflects the distribution of errors. To improve the efficiency of adaptive mesh refinement, the refinement region is enlarged aritificially through dividing the ratio by $c^{(l)}$, where $c^{(l)} < 1.0$ such that the triangles in the neighbor of the vertex with large errors are refined.

In order to guarantee that the size function changes smoothly, and avoid excessive refining/coarsening in adaptive mesh refinement, a is usually chosen as 4.0 and b is set to be 0.5. From Equation (20) it can be seen that if $f = 4.0$ then the bubble area is decreased and the mesh around the vertex is refined. If $f = 0.5$ holds, then the bubble area is increased and the triangles around the vertex will be coarsened. This concides with the principle of refining mesh in regions with large errors and coarsening the mesh with small errors.

The new metric for the $(l+1)$th refinement level at vertex $\mathbf{x}_i$ is

$$M_i^{(l+1)} = \begin{bmatrix} e_{i,2}^{(l+1)} & e_{i,1}^{(l+1)} \end{bmatrix} \begin{bmatrix} h_{i,1}^{(l+1)} & 0 \\ 0 & h_{i,2}^{(l+1)} \end{bmatrix} \begin{bmatrix} e_{i,2}^{(l+1)} & e_{i,1}^{(l+1)} \end{bmatrix}^T \tag{24}$$

The metric function for any point $\mathbf{x} \in \Omega$ is defined by linear interpolation with respect to the current mesh $\mathscr{T}^{(l)}$. The linear interpolation method of metric in triangle can be seen in [1]. In addition, when calculating the metric function of the point $\mathbf{x}$, it is required to find out the element where the bubble $\mathbf{x}$ lies. The process is called localization operation [17]. In our study, the modified bucket searching method is used to improve the efficiency.

3.2. Node placement by bubble simulation. For the mesh based numerical methods, the accuracy and convergence of the solution depend on the mesh shape, size and orientation, which rely on the node distribution. Many strategies have been proposed for optimizing the node distribution. The NPBS can adjust the node distribution automatically according to the given metric tensor to obtain the high-quality node set. For the NPBS method, the computational domain is viewed as force field and the nodes are considered as the centers of bubbles. Driven by the interaction forces and the damping force, the bubbles are moved according to the Newton's second law of motion, until a force-balance configuration is obtained. Finally, the centers of bubbles form a well-designed node set with high quality. The flowchart is given in Figure 2, and the key procedures are discussed in the following.

The current mesh vertices $\{\mathbf{x}_i\}_{i=1}^{N^{(l)}}$ are viewed as the centers of the bubbles, $N^{(l)}$ is the bubble number at the lth refinement level. For anisotropic mesh generation the elliptic bubble is adopted. The bubble motion is governed by the equation

$$m\ddot{\mathbf{x}}_i + c\dot{\mathbf{x}}_i = \tilde{f}_i, \qquad i = 1 \cdots N^{(l)} \tag{25}$$

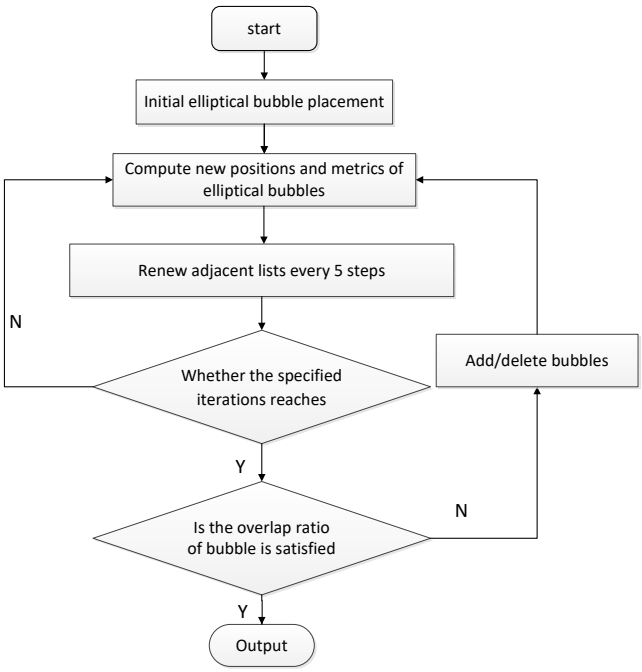

FIGURE 2. The flowchart of the node placement process.

where m is the mass of bubble, c is the damping coefficient, $N^{(l)}$ is the number of bubbles, $\mathbf{x}_i$ is the center of bubble i, and $\tilde{f}_i$ is the resultant force exerted on bubble i by its surrounding bubbles. For any two neighboring bubbles i, j, the interaction force is

$$\tilde{f}(w) = \begin{cases} l_0\left(1.25w^3 - 2.375w^2 + 1.125\right) & 0 \le w \le 1.5 \\ 0 & 1.5 < w \end{cases} \tag{26}$$

where w is the ratio of the real distance l and the ideal distance l_0 between the centers of elliptic bubbles i and j, i.e. $w = l/l_0$. The modified interaction force is adopted here proposed by Zhou et al.[33]. Compared with the traditional interaction force, the modified one adapts to the problems with strong singularity better. The ideal distance l_0 for elliptic bubble i and j is defined as

$$l_0 = l_{ij} + l_{ji} \tag{27}$$

Here l_{ij} is the radius of elliptic bubble i in the direction from center of bubble i to center of bubble j. l_{ji} is defined in the similar way. From the definition of interaction force, a repulsive force is applied when the distance of two elliptic bubbles is smaller than the ideal distance l_0 ($l/l_0 < 1$) shown in Figure 3(a), when $l/l_0 = 1$, the interaction force is zero shown in Figure 3(b), or an attractive force is adopted when the bubbles are located farther than l_0 ($1 < l/l_0 < 1.5$), shown in Figure 3(c), until the attractive force is zero if $l/l_0 > 1.5$.

For solving the Equation (25), the second-order Euler predictor-corrector method is used until the force balance configuration is obtained. For adaptive mesh refinement method, the displacements of bubbles in different regions differ a lot. For the bubbles with large errors, called *moving* bubbles, the position should be calculated

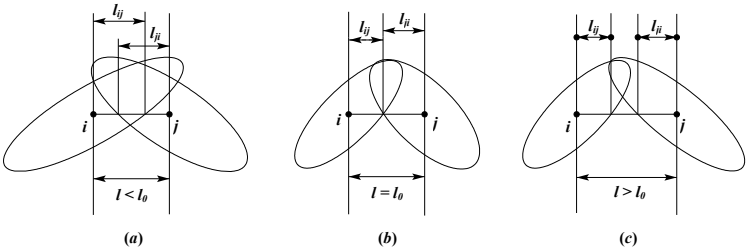

FIGURE 3. The relationship between force and positions of elliptical bubbles.(a) Repulsive force. (b) Stable. (c) Attractive force.

precisely with small time step, while for other bubbles, called *oscillating* bubbles, the calculation should be rough to save the computation, i.e., the multilevel approach for time step is applied. To be specific, the motion equations are solved numerically with small time step for moving bubbles while large time step for oscillating bubbles. Suppose the time step for moving bubbles is Δt, then the time step for oscillating bubbles is set as $\kappa\Delta t$ ($\kappa > 1$), where κ is the parameter to control the time step size for oscillating bubbles. The oscillating bubbles step forward for once with time step $\kappa\Delta t$, then the moving bubbles advance forward κ times with time step Δt. The iterative process are repeated until the balanced configuration is obtained.

Since the interaction forces between bubbles are short-range, only the forces exserted by adjacent bubbles need to be considered. In order to search adjacent bubbles around bubble i quickly, an adjacent list including its adjacent bubbles located within distance $r = 1.7\sigma$ is built, where σ is the ideal distance between two bubbles, i.e., $\sigma = l_{ij} + l_{ji}$.

The bubble number can be adjusted automatically through deleting the bubbles whose overlap ratios are too large or adding bubbles whose overlap ratios are too small during dynamic simulation. In anisotropic case, the overlap ratio is defined as

$$\alpha_i = \sum_{j=0}^{N_i} \frac{1}{l_{ij}}(2l_{ij} + l_{ji} - d_{ij}) \tag{28}$$

where l_{ij} and l_{ji} are shown in Figure 3, d_{ij} is the real distance between the centers of bubble i and bubble j, N_i is the length of the adjacent list corresponding to bubble i. Actually, the overlap ratio describes the number of adjacent bubbles. In the ideal case, the overlap ratios of bubbles on a line, on a surface, or in the internal volume are 2, 6 and 12, respectively. Through the bubble insertion/deletion, the approximate bubble number can be obtained. Finally a node set meeting the requirement of the metric tensor can be obtained.

3.3. Anisotropic bubble-type local mesh generation(ABLMG). The node placement by bubble simulation provides a high-quality node set, as well as the adjacent list which is conductive to generate local mesh. The new local mesh generation method is called Anisotropic Bubble-type Local Mesh generation (ABLMG).

Some notations are firstly introduced as follows. As shown in Figure 4, the element *patch* is composed of the elements which share the common node, and the common node is called *central* node. The node in the patch other than the central node is named *satellite* node, and the the element is referred to a *satellite* element of the *central* node.

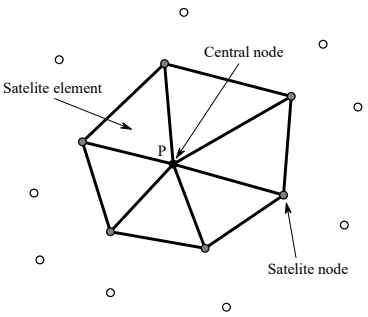

FIGURE 4. The patch associated with a central node P.

The adjacent list of each node contains its neighboring nodes including all satellite nodes and a small number of non-satellite nodes. For Delaunay mesh, the local mesh corresponding to the central node can be obtained through removing all the non-satellite nodes and connecting the satellite nodes. The union of the local mesh is the global Delaunay mesh. Taking the central node P as an example, the strategy of removing non-satellite nodes is shown as follows:

(1) Connecting the central node P with all of the adjacent nodes in its adjacent list.
(2) Sorting all the adjacent nodes in counterclockwise order, we get the sequence $\cdots P_{j-1}, P_j, P_{j+1} \cdots$.
(3) Checking node $P_j (j = 1, 2, \cdots, N_j)$ whether a satellite node of the central node P.

If the adjacent list of node P_{j-1} doesn't contain P_{j+1}, then node P_j is a satellite node of the central node P. Otherwise, the segment $P_{j-1}P_{j+1}$ intersects with the segment PP_j. The intersection test between segment $P_{j-1}P_{j+1}$ and segment PP_j is applied.

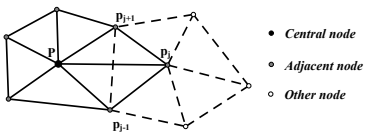

FIGURE 5. Intersection test for the patch of the central node P.

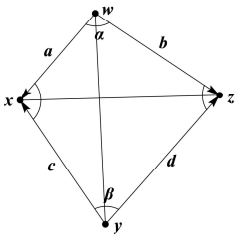

FIGURE 6. Edge swap.

For isotropic case, Delaunay criteria can be used as intersection test to check the relationship between the node P_j and the circumscribed circle of $\triangle PP_{j-1}P_{j+1}$, however, it is disable in anisotropic case. The edge swap criteria [5] is used here to remove the non-satelite nodes. In Figure 6, if $\angle\alpha + \angle\beta > 180°$, i.e., $\sin(\alpha + \beta) < 0$, the edge xz is non-Delaunay edge and should be removed. The condition can be simplified as follows

$$
\begin{aligned}
\sin(\alpha+\beta) &= \sin\alpha\cos\beta + \sin\beta\cos\alpha \\
&= \frac{a\times b}{|a|\cdot|b|}\cdot\frac{c\cdot d}{|c|\cdot|d|} + \frac{c\times d}{|c|\cdot|d|}\cdot\frac{a\cdot b}{|a|\cdot|b|} \\
&\propto (a\times b)(c\cdot d) + (c\times d)(a\cdot b) < 0.
\end{aligned} \tag{29}
$$

In Riemannian space it is equivalent to

$$
\sqrt{\det \mathbf{M}_{avg}}(a\times b)(c^T\mathbf{M}_{avg}d) + \sqrt{\det \mathbf{M}_{avg}}(c\times d)(a^T\mathbf{M}_{avg}b) < 0 \tag{30}
$$

where $\mathbf{M}_{avg} = (\mathbf{M}(w) + \mathbf{M}(x) + \mathbf{M}(y) + \mathbf{M}(z))/4$.

Since $\sqrt{\det \mathbf{M}_{avg}}$ is positive, so the edge swap criteria is obtained

$$
(a\times b)(c^T\mathbf{M}_{avg}d) + (c\times d)(a^T\mathbf{M}_{avg}b) < 0 \tag{31}
$$

If Equation (31) holds, the edge xz should bd removed, otherwise wy will be removed and the nodes $\mathbf{y}$ and $\mathbf{w}$ will be deleted from each other's adjacent lists respectively.

After removing all the non-satellite nodes for each central node, only the satellite nodes are left in the adjacent list and the local mesh can be obtained. The ABLMG method is very easy to implement, and the process is suitable for parallel environment since it can be executed simultaneously.

4. ABLMG-based adaptive mesh refinement method

Let Ω denote the given domain. The flowchart of the adaptive mesh refinement algorithm based on ABLMG method is given as follows:

(1) Generate an initial coarse mesh $\mathscr{T}^{(0)}$ on Ω and define the refinement level $l = 0$.
(2) Solve the model problem (1) by linear finite element method on $\mathscr{T}^{(l)}$. If l is more than the prescribed refinement level, stop, otherwise go to step 3.
(3) Compute the anisotropic error estimator (13) for each element $K \in \mathscr{T}^{(l)}$. Determine the metric tensor for each mesh vertex utilizing Equations (24), and the metric function $\mathbf{M}^{(l+1)}$ can be obtained through linear interpolation.
(4) Generate the node distribution satisfying the new metric function $\mathbf{M}^{(l+1)}$ using node placement method by bubble simulation.
(5) Build the local mesh for each node with ABLMG method, and the union of the local mesh is the global Delaunay mesh $\mathscr{T}^{(l+1)}$. Set $l = l + 1$, then go to step 2.

5. Numerical examples

In this section, several numerical tests with an anisotropic behaviour are presented to illustrate the effectiveness of the ABLMG-based adaptive mesh refinement method. The numerical experiments contain different forms of singularities which can be encountered in practical applications. We compare our method with the adaptive mesh refinement method based on the software BAMG. The comparative study about anisotropic and isotropic adaptive mesh refinement method is also carried out to verify the advantage of the anisotropic adaptation.

The convergence rate CR[30] with respect to the norm $|\cdot|_{H^1}$ at the refinement level l is computed by

$$CR = \frac{2\log\left(|e_{h,l}|_{H^1}/|e_{h,l-1}|_{H^1}\right)}{\log\left(N_{l-1}/N_l\right)} \tag{32}$$

where N_l is the number of nodes and $e_{h,l}$ denotes the error $u - u_h$ at the refinement level l.

Let η^A be the isotropic, anisotropic error estimates defined by

$$\eta^A = \left(\sum_{K\in\mathscr{T}_h} (\eta_K^A)^2\right)^{1/2}$$

where η_K^A are defined in Equation (13). We have computed the corresponding effectivity indices[22] namely

$$ei^A = \frac{\eta^A}{\|\nabla(u-u_h)\|_{L^2(\Omega)}}$$

to demonstrate that only the anisotropic a posteriori error estimator is effective on anisotropic mesh.

In order to test the anisotropic mesh quality, the formula discussed in [4] is used. Mesh quality α for $\triangle P_1P_2P_3$ is defined as

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

where

$$\hat{\alpha}(P_i) = 2\sqrt{3}\frac{\sqrt{\det(\mathbf{M}_i)}\cdot\det[P_2-P_1, P_3-P_1]}{d(\mathbf{M}_i, P_1P_2)^2 + d(\mathbf{M}_i, P_2P_3)^2 + d(\mathbf{M}_i, P_1P_3)^2} \tag{33}$$

In Equation (33) $\mathbf{M}_i$ is the metric tensor at P_i, $\det(\mathbf{M})$ is the determinant of $\mathbf{M}$, and $d(\mathbf{M},\cdot)$ describes the distance under metric tensor $\mathbf{M}$.

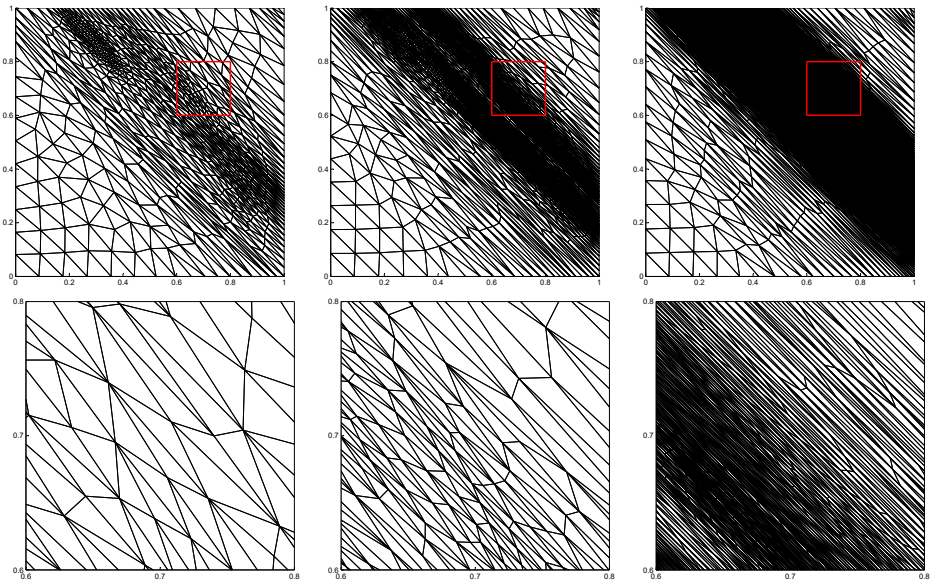

FIGURE 7. Adaptive ABLMG meshes for Example 1. Top: adaptive meshes at the 2*nd*, 5th and 7th refinement level. Bottom: The close-up views at $(0.7, 0.7)$.

5.1. Example 1. The first example is the boundary value problem on $\Omega = [0,1]^2$ in the form of Equation (1) with the right side f and the Dirichlet boundary condition being chosen such that the exact solution is

$$u = \frac{1}{1 + e^{20(x+y)-25}} \tag{34}$$

The solution function represents a shock wave along the line $y = 1.25 - x$[15]. The ABLMG-based adaptive finite element method is carried out to obtain a convergent solution.

The initial coarse mesh is uniform with mesh size $h^{(0)} = 0.1$, and then the anisotropic adaptive meshes are generated by the ABLMG adaptive mesh refinement method. The meshes at the 2nd, 5th and 7th refinement level and the close-ups of the corresponding mesh are shown in Figure 7. The meshes at different levels all are concentrated in the correct position and aligned along the shock wave. The mesh size and orientation confirm that our metric determination method is reliable.

Table 1 gives the information about mesh quality, maximum value of aspect ratio $ratio_{max}$, solution errors and effectivity indices of anisotropic and isotropic estimators at all refinement levels. From Table 1, it can be seen that the meshes at each refinement level remain well shaped supported by the average quality q_{ave}. The high quality mesh conduces to solve PDEs with finite element methods. The maximum value of the aspect ratio $ratio_{max}$ implies that the mesh has strong anisotropic feature. The effectivity indices of anisotropic estimator for different aspect ratios being constant value indicates that the anisotropic error estimator (13) is independent of the aspect ratio. It demonstrates the anisotropic error estimator (13) is equivalent to the true error.

TABLE 1. Mesh quality, effectivity indices and solution errors for Example 1.

l	N	q_{ave}	$ratio_{max}$	$\|e\|_{H^1}$	CR	ei^A
0	131	0.954	2.5	$6.441e-01$	0.0	4.065
1	309	0.868	9.8	$2.393e-01$	3.481	5.036
2	496	0.853	11.9	$1.229e-01$	2.816	5.535
3	591	0.880	15.6	$8.684e-02$	3.968	5.020
4	861	0.887	32.3	$5.116e-02$	2.813	5.079
5	1211	0.895	61.3	$3.458e-02$	2.295	5.105
6	1786	0.908	74.3	$2.311e-02$	2.076	5.131
7	2611	0.922	81.9	$1.573e-02$	2.024	5.142
8	3622	0.936	90.2	$1.165e-02$	1.837	5.061
9	5085	0.950	103.6	$8.726e-03$	1.703	4.973
10	7241	0.958	121.5	$6.779e-03$	1.429	4.983
11	10458	0.961	125.8	$5.364e-03$	1.274	4.928
12	15264	0.967	127.1	$4.380e-03$	1.072	4.847

A comparison study about convergence rate for four mesh refinement schemes is conducted. The isotropic mesh refinement scheme applies the BLMG-based adaptive mesh refinement [30] based on the estimator (15). The $BAMG$-based anisotropic adaptive mesh refinement method employs the software $BAMG$ according to our metric tensor (24) to refine the mesh. The plots of the error norm ($|e|_{H^1(\Omega)}$) versus the number of nodes for different mesh refinement strategies are

shown in Figure 8. It can be seen that both isotropic and anisotropic adaptive mesh refinement methods lead to much smaller errors than the uniform mesh although they have different levels of errors. The isotropic adaptive method requires 10 times more DOFs than an anisotropic one to obtain the same level of error. This demonstrates that the anisotropic adaptive mesh refinement method performs better than isotropic one by adapting not only mesh size but also shape and orientation. For the two anisotropic mesh refinement schemes, the ABLMG-based adaptive mesh refinement method can obtain much better approximation than the *BAMG* at almost the same number of DOFs.

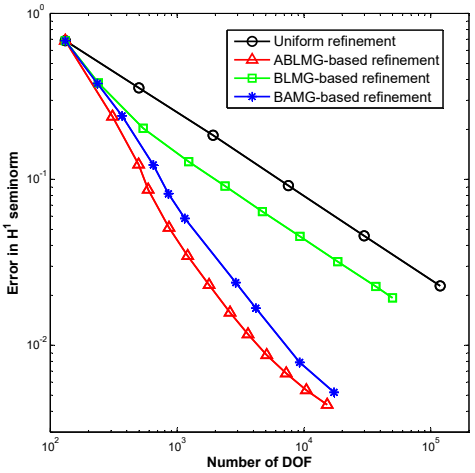

FIGURE 8. Error comparison for different adaptive mesh refinement methods.

TABLE 2. Comparison of the errors between the method in [15] and the ABLMG-based mesh refinement method.

	Edge-based estimator	Hessian recovery estimator	Full error estimator	The present method
N	684	693	714	648
$\|e\|_{L^2(\Omega)}$	1.4×10^{-3}	3.5×10^{-4}	3.4×10^{-4}	3.5×10^{-4}

Meanwhile, we also compare the result with other methods in [15] for the same PDE and the same exact solution shown in Table 2. The L^2 norm of error with three anisotropic error estimator are 1.4×10^{-3}, 3.5×10^{-4} and 3.4×10^{-4} on meshes with 684, 693 and 714 vertices. For our ABLMG-based adaptive method we have $|e|_{L^2(\Omega)} = 3.5 \times 10^{-4}$ on the mesh with 648 vertices; this means higher accuracy can be attained with less vertices with the help of our adaptive mesh refinement method.

5.2. Example 2. This example is to generate adaptive meshes for the problem in which the exact solution is

$$u(x,y) = \tanh\left(-100\left(y - \frac{1}{2} - \frac{1}{4}\sin(2\pi x)\right)^2\right), \qquad \forall (x,y) \in \Omega \equiv [0,1]^2. \tag{35}$$

A solution plot and the corresponding adaptive anisotropic mesh are shown in Figure 9. Compared with Example 1, this function exhibits a weaker anisotropic feature.

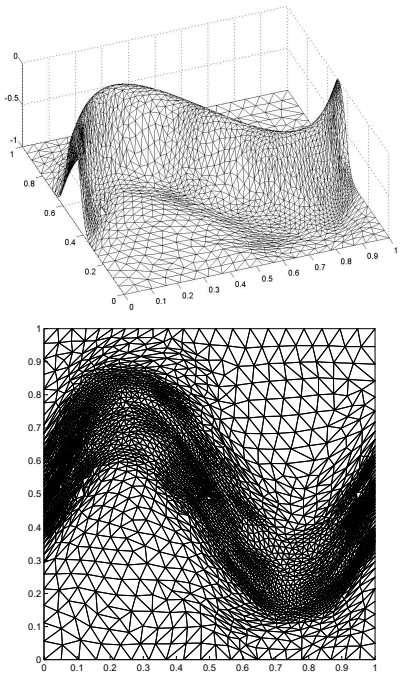

FIGURE 9. Surface plot of the Example 2 and the corresponding adaptive mesh.

As in Example 1, the anisotropic mesh generated by ABLMG-based adaptive mesh refinement method have high quality, which is demonstrated by the value q_{ave} in Table 3. Table 3 also includes the information about aspect ratios, solution errors and convergence rates at each refinement level. It can be seen that the mesh anisotropy is weak supported by the value $ratio_{max}$.

The plots of the error norms ($|e_h|_{H^1(\Omega)}$) versus the number of nodes for different refinement strategies is depicted in Figure 10. From Figure 10, we can observe that the anisotropic adaptive mesh refinement method is better than the isotropic one although the solution anisotropy is not significant. To achieve the similar accuracy, the number of DOFs for ABLMG-based refinement method is smaller than the number for the $BAMG$-based refinement strategy. For the problem with weak anisotropy, the numerical result of anisotropic refinement is close to the isotropic one. Thus we can conclude that better approximation can be obtained for anisotropic adaptive mesh refinement method for stronger anisotropic problems.

5.3. Example 3. This example is to generate an adaptive mesh for the model problem (1) with the true solution being chosen as

$$u = \tanh(60y) - \tanh(60x - 60y - 30), \qquad \forall (x, y) \in \Omega \equiv [0, 1]^2 \tag{36}$$

The function exhibits a strong anisotropic feature along the boundary layer $y = 0$ and along the line $y = x - 0.5$[15].

TABLE 3. Mesh quality, effectivity indices and solution errors for Example 2.

l	N	q_{ave}	$ratio_{max}$	$\|e\|_{H^1}$	CR	ei^A
0	131	0.887	2.3	$4.344e-00$	0.0	10.409
1	353	0.862	5.6	$2.848e-00$	0.851	5.775
2	663	0.932	6.1	$1.221e-01$	2.687	3.701
3	944	0.890	5.3	$7.387e-01$	2.846	2.727
4	1412	0.915	4.7	$4.992e-01$	1.945	2.083
5	2147	0.923	4.6	$3.590e-01$	1.573	1.586
6	3446	0.930	3.5	$2.641e-01$	1.298	2.157
7	5206	0.947	3.9	$2.011e-01$	1.321	2.583
8	7593	0.952	4.2	$1.626e-01$	1.125	2.624
9	10965	0.931	4.6	$1.307e-01$	1.190	1.951
10	17532	0.938	5.6	$9.863e-02$	1.199	2.238

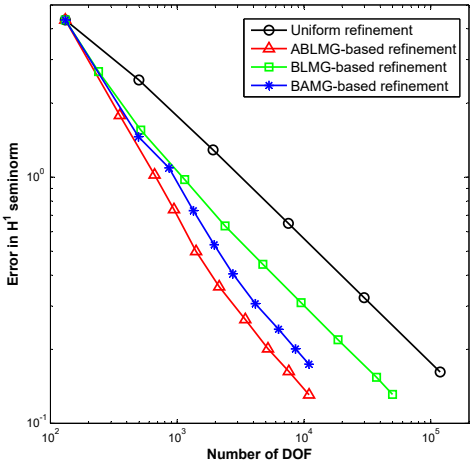

FIGURE 10. Error comparison for different adaptive mesh refinement methods.

Figure 11 displays the refined meshes by the ABLMG-based adaptive mesh refinement method. One can observe that all the meshes produce correct concentrations and are properly aligned with the fronts of the shock wave and the boundary layer. The absolute error distribution of the corresponding refinement level are shown in Figure 12. It can be clearly seen that the absolute error for mesh refinement method is reduced and almost equally distributed on the elements. Similar to that in first two examples, the same conclusion about the solution erros and effectivity indices can be obtained in Table 4. For this example, the mesh anisotropy is strong concluded by the value $ratio_{max}$. The anisotropic error estimator is also reliable and efficient.

The plots of the error norms ($|e_h|_{H^1(\Omega)}$) versus the number of nodes for different refinement strategies are depicted in Figure 13. From Figure 13, we can observe that the ABLMG-based adaptive mesh refinement method is much better than the isotropic one for the problems with strong anisotropy.

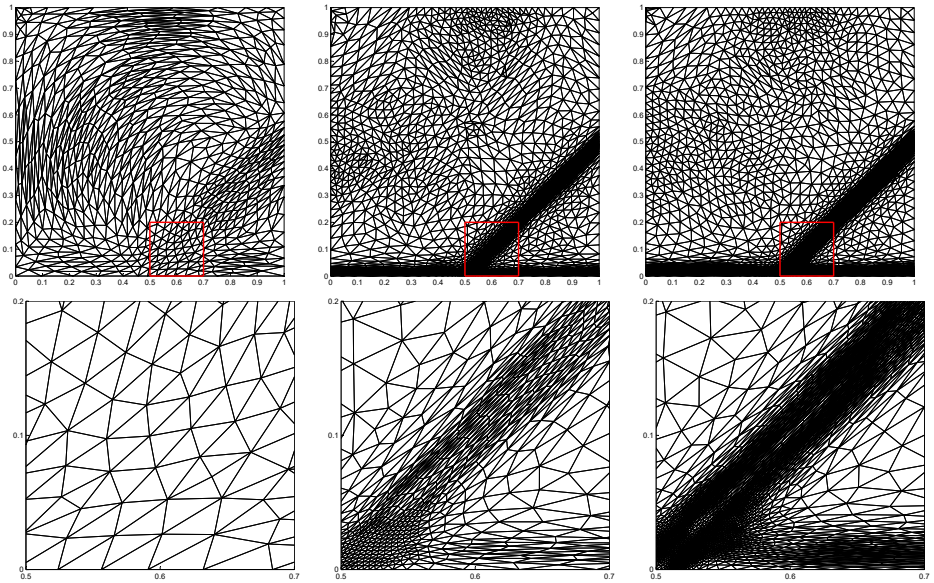

FIGURE 11. ABLMG meshes and corresponding close-ups at the $2nd$, 5th and 7th refinement level for Example 3.

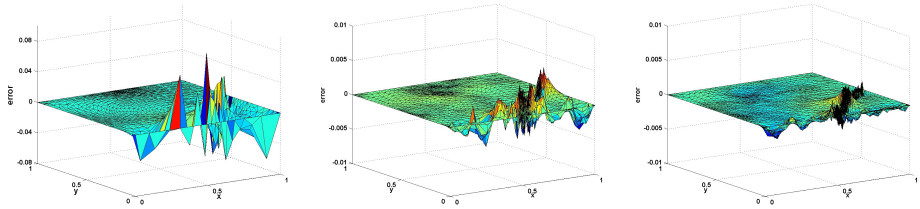

FIGURE 12. Error distribution on mesh at the $2nd$, 5th and 7th refinement level for Example 3.

TABLE 4. Solution errors, convergence ratios and effectivity indices for Example 3.

l	N	$ratio_{max}$	$\|e\|_{H^1}$	CR	ei^A
0	353	2.5	$7.842e-00$	0.0	3.667
1	669	16.4	$3.536e-00$	2.492	4.273
2	1284	21.1	$1.529e-00$	2.572	4.764
3	1628	29.2	$7.350e-01$	6.173	5.003
4	2145	43.6	$4.684e-01$	3.268	5.149
5	3154	57.1	$3.231e-01$	1.927	5.186
6	4818	59.1	$2.262e-01$	1.682	5.238
7	7132	101.8	$1.711e-01$	1.426	5.483
8	9000	125.3	$1.494e-01$	1.162	5.245

5.4. Example 4. In this example, the same PDE as in Example 1 is solved to test the capability of handling the boundary layer singularity using ABLMG-based

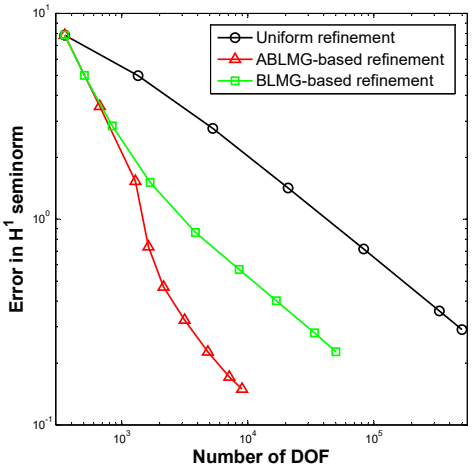

FIGURE 13. Error comparison for different adaptive mesh refinement methods.

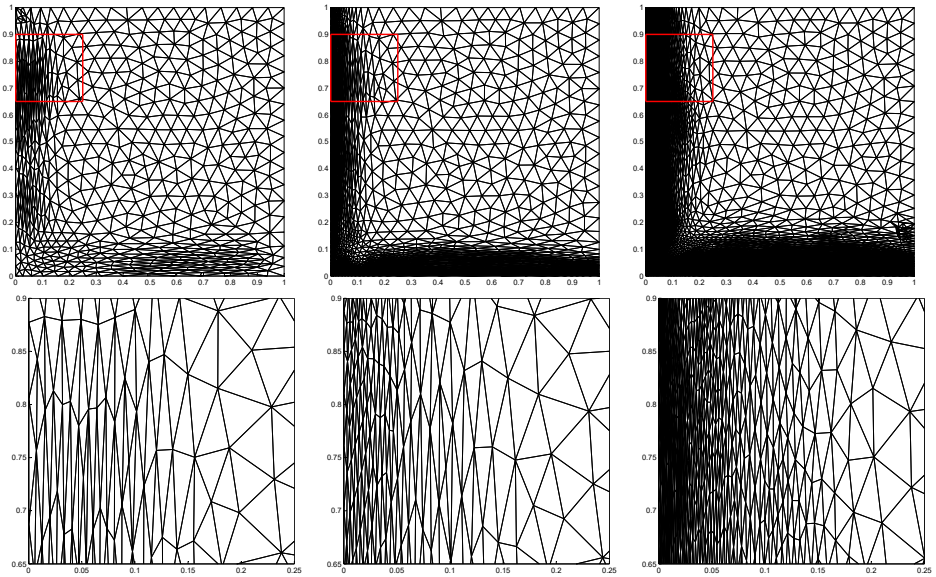

FIGURE 14. ABLMG meshes and corresponding close-ups at the $2nd$, 5th and 7th refinement level for Example 4.

AFEM. The exact solution is defined as

$$u(x, y) = e^{-25x} + e^{-25y}$$

where the function models a boundary layer near the coordinate axes[15].

Figure 14 displays the refined meshes by the ABLMG-based adaptive mesh refinement method. One can observe that the elements of the meshes are concentrated in the correct position and properly aligned in the boundary layers. The absolute error distribution of the corresponding refinement level are also shown in the Figure 15. It can be clearly seen that the absolute error for mesh refinement method is reduced and almost equally distributed on the elements. The plots of the error for

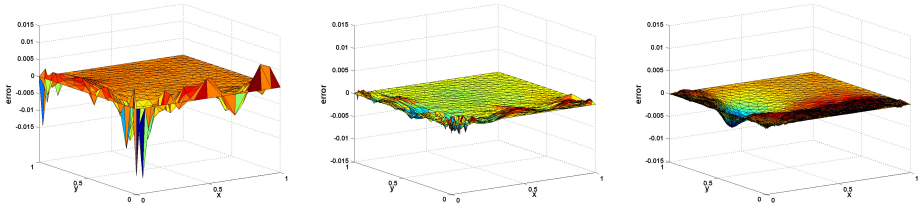

FIGURE 15. Error distribution on mesh at the 2*nd*, 5th and 7th refinement level for Example 4.

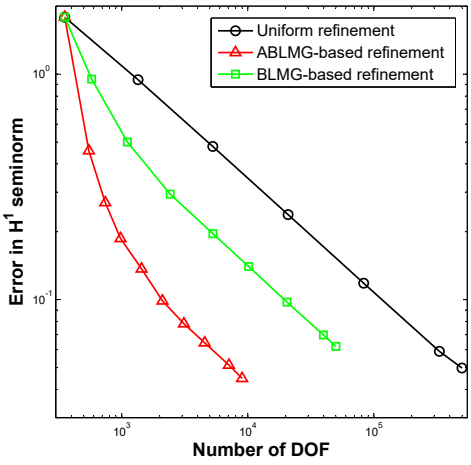

FIGURE 16. Error comparison for different adaptive mesh refinement methods.

different refinement strategies versus the number of nodes are presented in Figure 16. Similar to that in previous examples, the same conclusions about the solution errors can be obtained.

6. Conclusions

In this paper, we present an anisotropic adaptive mesh refinement method combining anisotropic bubble-type local mesh generation with the metric determined by the a posteriori error estimator. Our metric determination method use the recovery-based technique to approximate the direction information of the solution. The new metric assures that the elements are aligned with the correct direction and concentrated in the suitable position. The ABLMG-based mesh refinement method assures that the triangles remain very well shaped at all levels of refinement. Numerical results have shown that the new method is robust and effective. Optimal convergence rates with respect to H^1 norm can be obtained through the method.

Though the initial study about ABLMG-based mesh refinement method is carried out here for classical linear second order elliptic equation, it can be also applied to solve more complex system of equations. Since the local mesh for each node can be obtained simultaneously, the node-based local mesh generation method has great potential advantage in parallel computing, and the study of the parallel ABLMG-based adaptive finite element method will be our future work.

Acknowledgments

This research was supported by National Natural Science Foundation of China [No.11471262 and 11501450], China Postdoctoral Science Foundation (No.2018M633568), Natural Science Basic Research Plan in Shaanxi Province of China(No.2018JQ1049) and the Fundamental Research Funds for the Central Universities.

References

[1] F. Alauzet. Size gradation control of anisotropic meshes. *Annals of the University of Oradea Economic Science*, 46(1-2):181–202, 2010.

[2] T. Apel, S. Grosman, P. K. Jimack, and A. Meyer. A new methodology for anisotropic mesh refinement based upon error gradients. *Applied Numerical Mathematics*, 50(3-4):329–341, 2004.

[3] J. D. Boissonnat, C. Wormser, and M. Yvinec. Anisotropic delaunay mesh generation. *SIAM Journal on Computing*, 44(2):467–512, 2015.

[4] H. Borouchaki, P. L. George, F. Hecht, P. Laug, and E. Saltel. Delaunay mesh generation governed by metric specifications. part i. algorithms. *Finite Elements in Analysis and Design*, 25(12):61–83, 1997.

[5] F. J. Bossen and P. S. Heckbert. A pliant method for anisotropic mesh generation. In *5th International Meshing Roundtable*, pages 63–74, 1996.

[6] W. W. Chen, Y. F. Nie, W. W. Zhang, and L. Wang. A fast local mesh generation method about high-quality node set. *Jisuan Lixue Xuebao/Chinese Journal of Computational Mechanics*, 29(5):704–709, 2012.

[7] E. F. D'Azevedo. Optimal triangular mesh generation by coordinate transformation. *SIAM Journal on Scientific and Statistical Computing*, 12(4):755–786, 1991.

[8] C. Dobrzynski. *Adaptation de maillage anisotrope 3d et application à l'aérothermique des bětiments*. PhD thesis, Universitě Pierre et Marie Curie, Paris VI, France, 2005.

[9] Vít Dolejší. Anisotropic hp-adaptive discontinuous galerkin method for the numerical solution of time dependent PDEs. *Applied Mathematics and Computation*, 267(C):682–697, 2015.

[10] Q. Du and D. Wang. Anisotropic centroidal voronoi tessellations and their applications. *SIAM Journal on Scientific Computing*, 26(3):737–761, 2005.

[11] L. Formaggia and S. Perotto. New anisotropic a priori error estimates. *Numerische Mathematik*, 89(4):641–667, 2001.

[12] L. Formaggia and S. Perotto. Anisotropic error estimates for elliptic problems. *Numerische Mathematik*, 94(1):67–92, 2003.

[13] F. Hecht. Bamg: Bidimensional anisotropic mesh generator. *Inria Report*, 1998.

[14] W. Z. Huang. Metric tensors for anisotropic mesh generation. *Journal of Computational Physics*, 204(2):633–665, 2005.

[15] W.Z. Huang, L. Kamenski, and J. Lang. A new anisotropic mesh adaptation method based upon hierarchical a posteriori error estimates. *Journal of Computational Physics*, 229(6):2179–2198, 2010.

[16] Y. Q. Huang, Y. F. Su, H. Y. Wei, and N. Y. Yi. Anisotropic mesh generation methods based on acvt and natural metric for anisotropic elliptic equation. *Science China-Mathematics*, 56(12):2615–2630, 2013.

[17] R. Krause and E. Rank. A fast algorithm for point-location in a finite element mesh. *Computing*, 57(1):49–62, 1996.

[18] R. Kuate. Anisotropic metrics for finite element meshes using a posteriori error estimates: Poisson and stokes equations. *Engineering with Computers*, 29(4):497–505, 2013.

[19] P. Laug and H. Borouchaki. The bl2d mesh generator: Beginner's guide, user's and programmer's manual. Technical Report RT-0275 3, Institut National de Recherche en Informatique et Automatique (INRIA), Rocquencourt, 1996.

[20] Y. Liu, Y. F. Nie, W. W. Zhang, and L. Wang. Node placement method by bubble simulation and its application. *Computer Modeling in Engineering & Sciences*, 55(1):89–109, 2010.

[21] M. Picasso. An anisotropic error indicator based on zienkiewicz-zhu error estimator: Application to elliptic and parabolic problems. *SIAM Journal on Scientific Computing*, 24(4):1328–1355, 2003.

[22] M. Picasso. Numerical study of the effectivity index for an anisotropic error indicator based on zienkiewicz-zhu error estimator. *Communications in Numerical Methods in Engineering*, 19(1):13–23, 2003.

[23] M. Picasso. Adaptive finite elements with large aspect ratio based on an anisotropic error estimator involving first order derivatives. *Computer Methods in Applied Mechanics & Engineering*, 196(1):14–23, 2006.

[24] R. Schneider. A review of anisotropic refinement methods for triangular meshes in FEM. In *Lecture Notes in Applied and Computational Mechanics*, volume 66, pages 133–152. Springer, 2013.

[25] R. Schneider and P. K. Jimack. Anisotropic mesh adaption based on a posteriori estimates and optimisation of node positions. *Journal of Mathematical Sciences*, 125(3):405–418, 2006.

[26] K. Shimada and D. Gossard. Bubble mesh: automated triangular meshing of non-manifold geometry by sphere packing. In *Proceedings of the third ACM symposium on Solid modeling and applications*, pages 409–419. ACM, 1995.

[27] R. B. Simpson. Anisotropic mesh transformations and optimal error control. *Applied Numerical Mathematics*, 14(1-3):183–198, 1994.

[28] D. Wang, R. Li, and N. Yan. An edge-based anisotropic mesh refinement algorithm and its application to interface problems. *Communications in Computational Physics*, 8(3):511–540, 2010.

[29] S. Yamakawa and K. Shimada. Triangular/quadrilateral remeshing of an arbitrary polygonal surface via packing bubbles. In *Geometric Modeling and Processing 2004*, pages 153–162, 2004.

[30] W. W. Zhang, Y. F. Nie, and Y. T. Gu. Adaptive finite element analysis of elliptic problems based on bubble-type local mesh generation. *Journal of Computational and Applied Mathematics*, 280:42–58, 2015.

[31] W. W. Zhang, Y. F. Nie, and Y. Q. Li. A new anisotropic local meshing method and its application in parametric surface triangulation. *Computer Modeling in Engineering & Ences*, 88(6):507–529, 2012.

[32] W. W. Zhang, Y. F. Nie, and L. Wang. Bubble meshing method for two-parametric surface. *Chinese Journal of Computational Physics*, 29(1):43–50, 2012.

[33] Y. Q. Zhou, Y. F. Nie, and W. W. Zhang. A modified bubble placement method and its application in solving elliptic problem with discontinuous coefficients adaptively. *International Journal of Computer Mathematics*, 94(6):1268–1289, 2017.

[34] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33(7):1331–1364, 1992.

[35] O. C. Zienkiewicz and J. Z. Zhu. The superconvergent patch recovery and a posteriori error estimates. part 2: Error estimates and adaptivity. *International Journal for Numerical Methods in Engineering*, 33(7):1365–1382, 2010.

Research Center for Computational Science, Northwestern Polytechnical University, Xi'an 710072, P.R. China

E-mail: guowei@mail.nwpu.edu.cn, yfnie@nwpu.edu.cn and wwzhang@nwpu.edu.cn