

An Adaptive Discretization of Incompressible Flow using Node-Based Local Meshes

Weiwei Zhang¹, Yufeng Nie¹, Li Cai¹ and Nan Qi²

Abstract: In this paper, we derive an adaptive mesh generation method for discretizing the incompressible flow using node-based local grids. The flow problem is described by the Stokes equations which are solved by a stabilized low-order P1-P1 (linear velocity, linear pressure) mixed finite element method. The proposed node-based adaptive mesh generation method consists of four components: mesh size modification, a node placement procedure, a node-based local mesh generation strategy and an error estimation technique, which are combined so as to guarantee obtaining a conforming refined/coarsened mesh. The nodes are considered as particles with interaction forces, which are generated by dynamic simulation according to Newton's second law of motion. Then the successive meshes in adaptive procedure are obtained by using Bubble-type Local Mesh Generation (BLMG) method. At each refinement level, the refining and coarsening are archived simultaneously by appropriately modifying the mesh size function, such that the resulting meshes can be refined in regions where the errors are relatively large and coarsened in regions where the errors are relatively small. Numerical results show that the node-based adaptive strategy is applicable and efficient to approximate the true solution and detect local singularities in the flow problems.

Keywords: mesh size function, error estimator, stokes equations, local mesh generation, adaptive analysis.

1 Introduction

Unstructured mesh adaptation is now widely used in numerical simulations to improve the accuracy of solutions as well as to capture the behavior of physical phenomena. Specifically, for the Stokes problem, the adaptive analysis has two procedures: a posteriori error estimation and mesh adaptivity based on the result of

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

² School of Mathematics and Statistics, University of Glasgow, Glasgow G12 8QW, UK.

error estimation. In the last few decades, how to derive a posteriori error estimates for the Stokes equations has received much attention, and many researchers have acquired quantities of good results, see Refs. [Babuvška and Rheinboldt (1978); Verfürth (1989, 1991); Kay and Silvester (1999); Ervin and Phillips (2006); Liao and Silvester (2012)] and references therein.

The performance of adaptive methods for PDEs depends not only on the error estimators, but also on techniques used for constructing adaptive meshes. Successful implementation of the adaptive strategy can improve the accuracy of numerical approximations and decrease the computational cost. In the past two decades, a large number of research works relevant to the adaptive mesh generation in different research and application fields occurred, see for instance [Ju, Gunzburger, and Zhao (2006); Huang, Qin, Wang, and Du (2011); Biboulet, Gravouil, Dureisseix, Lubrecht, and Combescure (2013)] for elliptic problems and [Picasso (2003)] for parabolic problems. In addition, the authors of Refs. [Boussetta, Coupez, and Fourment (2006); Labergère, Rassinoux, and Saanouni (2011)] have proposed an adaptive meshing frame for forging simulation which is based on an iterative mesh improvement by local changes. Although adaptive mesh refinement techniques are available for the Stokes equations [Zheng, Hou, and Shi (2010); He, Xie, and Zheng (2010); Araya, Barrenechea, and Poza (2008)], the mesh adaptation algorithms for the Stokes problems has not been widely studied. In Refs. [Zheng, Hou, and Shi (2010); He, Xie, and Zheng (2010)], the finite element software FreeFem++ [Hecht (2012)] is applied by using the mesh generator *bamg* [Hecht (1998)] to create successively refined meshes. Based on the bisection technique, the *bamg* can subdivide some selected elements to obtain a new mesh quickly. However, it needs to take appropriate measures to remove hanging nodes, coordinating the new mesh. Besides, the smoothness procedure is necessary for the *bamg* between the refined and non-refined areas. In addition, a sequence of refined meshes are provided by using the mesh generator *Triangle* [Shewchuk (2005)] for the generalized Stokes problems [Araya, Barrenechea, and Poza (2008)]. In general, the mesh adaptation methods can be classified into three categories:

- (i) p -refinement in which the same elements are used but the order of interpolation is increased.
- (ii) r -refinement in which the connectivity of the mesh remains unchanged but the nodes are moved to increase mesh density in certain locations.
- (iii) h -refinement in which the element sizes are varied. It can be further divided into two categories:
 - Mesh subdivision where individual elements are subdivided without chan-

ging their original position.

- Complete regeneration by adaptive remeshing.

For the above three procedures, Nithiarasu et al. [Nithiarasu and Zienkiewicz (2000)] point out that for the method (i), as most fluid mechanics problems involve an explicit time marching algorithm, higher order element is not popular. Since new nodes are not added in the method (ii), the accuracy of the solution derived by this method is limited by the initial number of nodes and elements. As to the method (iii), both mesh subdivision and adaptive remeshing are suitable for most of the fluid mechanics problems. While in the mesh subdivision method, it is difficult to take into account simultaneously the bisection and the need of anisotropy of the mesh when generating anisotropic mesh, and further difficulty arises in the coupling of refined and original elements [Nicolas and Fouquet (2013)]. However, the above difficulties can be avoided by using the adaptive remeshing method, in which the whole domain is remeshed based on the a posteriori error estimation computed from the previous solution.

Motivated by the works [Chen, Nie, Zhang, and Wang (2012); Nie, Zhang, Qi, and Li (2014)], in this paper we present the first effort in designing an adaptive node-based local mesh refinement method-BLMG to solve the Stokes problems based on the error estimation. In previous work, the node placement method [Liu, Nie, Zhang, and Wang (2010); Zhang, Nie, and Li (2012); Nie, Zhang, Qi, and Li (2014)] and its acceleration strategies [Qi, Nie, and Zhang (2014)] have been studied, where the node size function is usually given as input information. This is inadvisable to predefine the size functions to generate variable density meshes for the solution of flow problems with local singularities. In this study, through adopting suitable a posteriori error estimator and mesh size modification technique, together with the node-based local mesh generation strategy, the BLMG method can be used to deal with the Stokes problem with the low-order mixed FEM [Bochev, Dohrmann, and Gunzburger (2006); Li and He (2008); He, Xie, and Zheng (2010); Zheng, Hou, and Shi (2010)].

It's well known that the lowest equal-order finite element pairs do not satisfy the inf-sup condition which guarantees the convergence of the FE-discretization [Cuvellier, Segal, and Van Steenhoven (1986)]. Here a stabilized approach [Bochev, Dohrmann, and Gunzburger (2006)] to the Stokes equations is used by adding a stabilized term $G(p, q)$ to the variational formulation. The obtained discretization form can satisfy the inf-sup condition and thus has a unique solution. Next, a brief outline of the BLMG-based adaptive stabilized method is given. Initially, a coarse Delaunay triangular mesh with a uniform size is constructed. Then, the iterative loops of finite element solution, error estimate, modification of mesh size function

and mesh refinement are executed. At each refinement level, a new mesh size field is firstly defined based on the obtained error estimation. Then according to Newton's second law of motion, the nodes are moved into a near-optimal configuration with dynamic simulation. Finally, the patch of elements around each node is built via the BLMG method, and the union of the patches is a Delaunay triangulation. The combination of adaptive BLMG with the low-order mixed FEM is particularly efficient and combines some best algorithmic features of each. In addition, the a posteriori error indicator reported in Ref. [He, Xie, and Zheng (2010)] is used and verified to be equivalent to the discretization error.

The paper is organized as follows. In Section 2, the adaptive BLMG method is described in detail, consisting of how to modify the mesh size based on a posteriori error estimation, how to optimize nodal distribution, and how to generate the node-based local meshes. In Section 3, the adaptive mesh method is applied to the Stokes problems, some well-known results of the stabilized mixed low-order FEMs, and the error estimates used in this study are introduced. In Section 4, numerical experiments are carried out to verify the good stability and accuracy of our method. Finally, some conclusions are obtained in Section 5.

2 Adaptive mesh strategy

For any adaptive finite element method, there are two key points, namely the error estimator of the computed finite element solution and the mesh adaptive strategy to adjust the mesh based on the estimated errors. In this study, two types of error estimators are used which will be briefly described in Section 3.1. As for the mesh adaptation strategy, we mainly focus on the node-based local mesh generation method BLMG at each refinement level.

Let $\mathcal{T}_h^{(l)}$ denote the triangulation of the computational domain Ω with vertices $\{\mathbf{x}_i^{(l)}\}_{i=1}^{N^{(l)}}$ at the refinement level l , $N^{(l)}$ is the corresponding number of vertices. Suppose that a finite element solution $u_h^{(l+1)}$ is obtained in the adaptation process. Now the question is how to obtain a new mesh $\mathcal{T}_h^{(l+1)}$ using this new solution and the previous mesh $\mathcal{T}_h^{(l)}$. The mesh generation algorithm BLMG requires a mesh size function to decide the distribution of grid points. Thus, roughly speaking, the adaptive BLMG method consists of three steps:

Step 1: Develop a new mesh size $h^{(l+1)}$ according to a posteriori error estimators and the current mesh size $h^{(l)}$;

Step 2: Generate the new mesh vertices $\{\mathbf{x}_i^{(l+1)}\}_{i=1}^{N^{(l+1)}}$ using the node placement method by bubble simulation;

Step 3: Connect the vertices $\{\mathbf{x}_i^{(l+1)}\}_{i=1}^{N^{(l+1)}}$ to generate global meshes $\mathcal{T}_h^{(l+1)}$ with the BLMG method. Details will be discussed in Section 2.3.

2.1 Relationship between a posteriori error estimator and mesh size

In this subsection, how to modify the current mesh size based on a posteriori error estimate is discussed. Let the patch $\tilde{T}_i^{(l)}$ represent the set of the elements that share the common mesh vertex $\mathbf{x}_i^{(l)}$ of $\mathcal{T}_h^{(l)}$. $\eta_T^{(l)}$ denote the obtained error estimators on element $T \in \mathcal{T}_h^{(l)}$. Then based on the error estimator $\eta_T^{(l)}$ at the element level, the a posteriori error estimate at the vertex $\mathbf{x}_i^{(l)}$ is defined as the area-weighted average in $\tilde{T}_i^{(l)}$

$$\eta_i^{(l)} = \frac{\sum_{T \in \tilde{T}_i^{(l)}} \eta_T^{(l)} |T|}{\sum_{T \in \tilde{T}_i^{(l)}} |T|}, \quad (1)$$

where $|T|$ denote the area of element T . After obtaining the error estimate at each mesh vertex, the new size function can be uniquely determined by a piecewise linear function $h^{(l+1)}$ on Ω , such that for any vertex $\mathbf{x}_i^{(l)}$ of $\mathcal{T}_h^{(l)}$,

$$h^{(l+1)}(\mathbf{x}_i^{(l)}) = \frac{h^{(l)}(\mathbf{x}_i^{(l)})}{f(\eta_i^{(l)}), \quad (2)$$

where $h^{(l)}(\mathbf{x}_i^{(l)})$ is the average length of the sides sharing the vertex $\mathbf{x}_i^{(l)}$, and $f(\eta_i^{(l)})$ is defined by

$$f(\eta_i^{(l)}) = \max \left(\min \left(\frac{\eta_i^{(l)}}{\bar{\eta}^{(l)}}, a \right), b \right) \quad (3)$$

where $\bar{\eta}^{(l)}$ denote the mean value of the a posteriori error estimate over all the elements, and the parameters a and b denote the thresholds of mesh refinement and coarsening factors, respectively, i.e., $a > 1$, $0 < b < 1$. The values of a and b play an important role in controlling the mesh density during adaptive procedure. If the value of a is set to be relatively large, for example, 4.0 or 5.0, it will cause that the grid is refined excessively in some locations at initial refinement levels. Then the coarsening operation in those locations may be necessary at the later refinement levels. If the value of a is set slightly larger than 1.0, it may need a lot of iterations to achieve the required accuracy, increasing the time consumption for

mesh refinement. Similarly, the coarsening factor threshold b also must be chosen appropriately. In the ideal case, through the setting of appropriate values of a and b , the mesh size can be suitably decreased in large-error regions and increased in small-error regions. In practice, we usually set $a = 2.0$ and $b = 0.7$. Then by moving the mesh points, one can refine the elements in some locations and simultaneously coarsen elements elsewhere. Thus, the refinement and coarsening of the mesh is achieved in the same framework which is very different from other refinement strategies of additionally designing a coarsening step (for example, Binev et al. [Binev, Dahmen, and DeVore (2004)] improve the adaptive strategy proposed by Morin et al. [Morin, Nochetto, and Siebert (2000)] by adding a coarsening step). In addition, the size function for any point $\mathbf{x} \in \Omega$ is defined by interpolation with respect to the current mesh $\mathcal{T}_h^{(l)}$. When calculating the value of the size function at $\mathbf{x} \in \Omega$, it is required to search the nearest neighbor, and the operation of search is always very time-consuming. In practice, this work is efficiently accomplished using the software package "ANN" [Arya, Mount, Netanyahu, Silverman, and Wu (1998)] which is based on kd -trees search strategies.

2.2 Node placement by bubble simulation

After determining the new size function, the current nodes will be optimized by bubble simulation [Nie, Zhang, Liu, and Wang (2010); Qi, Nie, and Zhang (2014)] via inserting or deleting nodes iteratively, such that the refined or coarsened nodes satisfy the requirement of the new size function. The process of the node placement is shown in Algorithm 1, and a brief description is given in the following.

Algorithm 1 Node placement by bubble simulation

Require: The desired mesh size function $h(x, y)$, (x, y) is the position coordinate of nodes.

Require: The current mesh vertices $\{\mathbf{x}_i\}_{i=1}^N$.

- 1: Establish an adjacency list for each bubble which stores the information of its neighboring bubbles.
 - 2: Solve the motion equations of the bubbles iteratively using the Euler's predictor-corrector numerical method.
 - 3: Update the adjacency lists every k iterative steps ($k = 5$).
 - 4: Adjust the number of the bubbles based on the overlap ratios per a fixed iterative step.
 - 5: Check if the termination condition is satisfied. If so, terminate; otherwise, go back to step 2.
-

At the beginning of the simulation, the current mesh vertices $\{\mathbf{x}_i^{(l)}\}_{i=1}^{N^{(l)}}$ are considered as the centers of the bubbles, and the radius of the bubble centered at \mathbf{x}_i is $\frac{h(\mathbf{x}_i)}{2}$. Here each bubble is dragged by two types of forces, i.e., the interaction force from neighboring bubbles and the viscous damping force from the system.

The interaction force for two neighboring bubbles, much like the van der Waals force, is approximated by [Shimada and Gossard (1998)]

$$f(w) = \begin{cases} k_0 (1.25w^3 - 2.375w^2 + 1.125), & 0 \leq w \leq 1.5, \\ 0, & 1.5 < w, \end{cases} \quad (4)$$

which tries to maintain the ideal distance between two bubbles by exerting a repelling force when they are too close, or an attracting force when they are not too distant, if two bubbles are far enough, there will be no interaction between them. Here w is the ratio of the real distance and the desired distance between two bubbles, and the desired distance for two bubbles i and j is defined as $\frac{h(\mathbf{x}_i)}{2} + \frac{h(\mathbf{x}_j)}{2}$ which is the sum of the radii of the two bubbles.

Meanwhile, the damping force from the system is denoted by $-\tilde{c}\dot{\mathbf{x}}_i$, which makes the bubble system converge to a stable configuration, where \tilde{c} is the damping coefficient. According to Newton's second law of motion, the motion equation of each bubble is

$$\tilde{m}\ddot{\mathbf{x}}_i + \tilde{c}\dot{\mathbf{x}}_i = \tilde{f}_i \quad (i = 1, 2, \dots, N^{(l)}), \quad (5)$$

where \tilde{m} is the mass of the bubble, and \tilde{f}_i is the resultant force exerting on bubble i by its surrounding bubbles

$$\tilde{f}_i = \sum_{j=1, j \neq i}^{n^{(l)}} f_{ij}, \quad (6)$$

where f_{ij} is the interaction force from the neighboring bubble j . The information of neighboring bubbles is stored in an adjacency list related to bubble i , and $n^{(l)}$ denote the size of the adjacency list. About the establishment and update of the adjacency list associated with each bubble, we can refer to [Nie, Zhang, Liu, and Wang (2010)] for details.

To solve the second-order differential equation system (5), adopting high-precision numerical methods (such as fourth-order Runge-Kutta method) will result in a high-quality node set. However, this will inevitably lead to large time consumption due to the dynamic simulation. Thus, it is necessary to choose a lower complexity numerical method which also ensures the quality of node sets simultaneously. Instead of the fourth-order Runge-Kutta method, a second-order Euler's predictor-corrector method is used here to obtain the position coordinates of bubbles at the next step.

During dynamic simulation, the overlap ratio [Shimada and Gossard (1998); Nie, Zhang, Liu, and Wang (2010)] is defined for each bubble to control the number of bubbles,

$$\alpha_i = \frac{1}{r_i} \sum_{j=1}^{n^{(l)}} (2r_i + r_j - l_{ij}), \quad (7)$$

where r_i and r_j denote the ideal radii of bubbles i and j , respectively, i.e., $r_i = \frac{h(\mathbf{x}_i)}{2}$ and $r_j = \frac{h(\mathbf{x}_j)}{2}$; l_{ij} is the real distance between the centers of bubbles i and j . In fact, the overlap ratio indicates the number of neighboring bubbles of each bubble. Hence, through setting threshold values of the overlap ratio for the bubbles on a line, on a surface or in an internal volume, we compute the overlap ratio per a fixed iterative step. The bubbles with large overlap ratios will be deleted, while the ones with small overlap ratios will have new bubbles added in its neighborhood. In this manner, the population of bubbles can be controlled dynamically.

The termination condition in previous study [Nie, Zhang, Liu, and Wang (2010)] is usually set to be that the iterative process reaches a prescribed round number, which is usually an empirical value. In this study, the termination condition is logically set such that $N_1 - N_2 \leq 0.5\% \times N_1$ is satisfied for twice, where N_1 and N_2 are the number of nodes before and after this round of simulation, respectively. By adopting the less complexity numerical method and the improved termination condition mentioned above, the computing cost decrease significantly by 50%, meanwhile, the average quality of the corresponding triangulation is maintained 90% substantially [Qi, Nie, and Zhang (2014)].

2.3 Bubble-type local mesh generation method

Based on the node set generated by the bubble simulation, how to connect these nodes to construct a triangulation of the analysis domain is discussed in this subsection. Firstly, some notations are introduced in the following. In a Delaunay triangulation, the elements which share one common node P compose an element patch denoted by \tilde{T}_P , and the common node P is viewed as a *central node*. Each element in the patch \tilde{T}_P is called a *satellite element* of the central node P , and a node of a satellite element, other than the central node is referred to as a *satellite node*, which is illustrated in Fig. 1. For each central node, the corresponding element patch can be generated by the BLMG method described in the following.

The node placement method mentioned above provides not only a high-quality node set, but also the adjacency list of each node which stores the information of the neighboring nodes. Meanwhile, the adjacency list related to each node includes [Chen, Nie, Zhang, and Wang (2012)]:

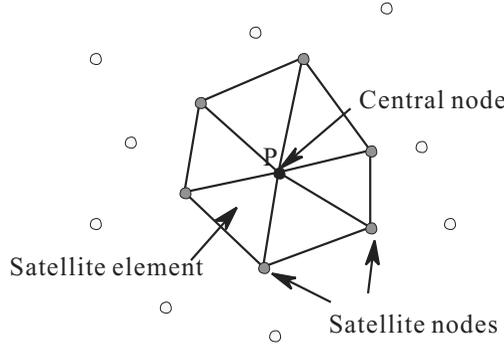


Figure 1: Element patch \tilde{T}_P associated to a central node P .

- all of the satellite nodes in the corresponding Delaunay triangulation;
- a small number of non-satellite nodes (no more than 2).

In fact, in a Delaunay triangulation, mesh edges are built by connecting the central nodes and their satellite nodes. Since both the satellite and non-satellite nodes lie in the adjacency list, the question is how to delete the non-satellite nodes in the adjacency list of each node. In the following, a simple strategy of removing the non-satellite nodes is given as follows:

- Connect every central node with all its adjacent nodes from its corresponding adjacency list.
- Sort the nodes from the adjacency list in a counterclockwise order. Taking the central node P as an example, we can get the sequence like $\cdots P_{j-1}, P_j, P_{j+1} \cdots$ which is shown in Fig. 2.
- Check each node in the sequence whether a satellite node of the central node P in turns.

When checking the node P_j whether a satellite node of P , we firstly need to judge whether or not the nodes P_{j-1} and P_{j+1} lie in each other's adjacency lists. If not, the node P_j is a satellite node of the central node P . Otherwise, according to step (i), the nodes P_{j-1} and P_{j+1} are connected into the edge $\overline{P_{j-1}P_{j+1}}$ which intersects with the edge $\overline{PP_j}$, then Delaunay criteria can be used to check the position relationship between the node P_j and the circumscribed circle of $\Delta PP_{j-1}P_{j+1}$. If P_j lies outside the circumscribed circle of $\Delta PP_{j-1}P_{j+1}$, then P_j is a non-satellite node of the central node P , deleting P_j from the adjacency list; otherwise, P_j is a satellite node of P .

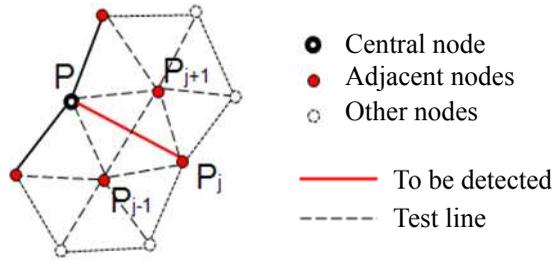


Figure 2: Deleting non-satellite nodes around a central node P .

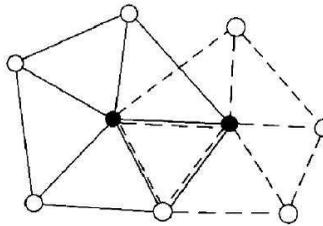


Figure 3: Inconsistency of local mesh.

After deleting the non-satellite nodes of each central node, the adjacency list corresponding to one central node only consists of all its satellite nodes. Then the set of the satellite elements associated with each central node is built which can be seen in Fig. 1. Local meshes around each node are generated under the constraints that the edges of satellite elements do not cross each other. If this condition is not satisfied, the satellite elements are referred to as inconsistent, as shown in Fig. 3. However, the inconsistency phenomenon which probably occurring in the BLMG process is circumvented in virtue of the uniqueness of the Delaunay partition which has been verified in Ref. [Nie and Chang (2006)]. This ensures that the union of the satellite elements is a Delaunay triangulation of the domain.

The most significant difference between the BLMG and the local mesh generation techniques in Ref. [Fujisawa, Inaba, and Yagawa (2003)] is that the latter begins by appropriately distributing the nodes in the computational domain, i.e., nodal coordinates and nodal density information are given as input information. However, how to generate suitable nodes in Ref. [Fujisawa, Inaba, and Yagawa (2003)] is not mentioned. Moreover, the authors of Ref. [Fujisawa, Inaba, and Yagawa (2003)] take much effort to search satellite nodes for local mesh generation, nevertheless, the searching process is completely avoidable in the BLMG method because the adjacency list of each node contains the total information of the neighboring nodes.

Thus, the BLMG is efficient and robust to conduct research for node-based local mesh generation in complex computational domain.

T. Coupez et al. [Boussetta, Coupez, and Fourment (2006)] also adopt local mesh refinement technique which is considered as the improvement of an existing mesh rather than a complete rebuilding process. In Ref. [Boussetta, Coupez, and Fourment (2006)], some elements are iteratively selected and removed by using appropriate local criteria, and local refinement is implemented in the current visited subdomain. This indicates that the selection of remeshing areas is partially affected by manual intervention, and a special treatment may be needed for the transition between two different levels of refinement. Also, the transition region must be carefully set to avoid the sharp jump of mesh size or occurring excessive refinement. Compared to T. Coupez's approach, the BLMG-based adaptive method uses a strategy based on particles to move mesh points and reconstruct a new triangulation. The refining/coarsening of mesh is completely determined by the size function based on a posteriori error estimation. Meanwhile, this technique works identical for 1D, 2D and 3D problems. For any dimension, the rules for moves of particles and the node connection scheme are the same, and 3D BLMG-based adaptive method is a future research topic.

3 Application to Stokes problems

In this study, the BLMG-based adaptive method is applied to solve the homogeneous Dirichlet boundary value problem for the Stokes equations

$$\begin{cases} -\Delta \mathbf{u} + \nabla p = \mathbf{f} & \text{in } \Omega, \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega, \\ \mathbf{u} = 0 & \text{on } \partial\Omega, \end{cases} \quad (8)$$

where Ω represent a polyhedral domain in R^2 with a Lipschitz continuous boundary $\partial\Omega$, \mathbf{u} is the velocity vector, p is the pressure, and \mathbf{f} is the prescribed body force.

The standard variational formulation of (8) is given by: find $(\mathbf{u}, p) \in (\mathbf{V}, S)$ satisfying

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) &= (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}, \\ b(\mathbf{u}, q) &= 0, \quad \forall q \in S, \end{aligned} \quad (9)$$

where

$$\mathbf{V} = (H_0^1(\Omega))^2 \quad \text{and} \quad S = L_0^2(\Omega) = \{\varphi \in L^2(\Omega); \int_{\Omega} \varphi d\mathbf{x} = 0\},$$

and

$$a(\mathbf{u}, \mathbf{v}) = \int_{\Omega} \nabla \mathbf{u} : \nabla \mathbf{v} d\Omega, \quad b(\mathbf{u}, q) = - \int_{\Omega} q \nabla \cdot \mathbf{u} d\Omega,$$

with the inner product (\cdot, \cdot) in $L^2(\Omega)$. The norm and seminorm in $H^k(\Omega)$ are denoted by $\|\cdot\|_k$ and $|\cdot|_k$, respectively.

Let $\mathcal{P}_0, \mathcal{P}_1$ denote the constant polynomials space and the linear polynomials space, we put

$$\begin{aligned} R_0 &= \{\varphi \in L^2(\Omega) : \forall T \in \mathcal{T}_h, \varphi|_T \in \mathcal{P}_0\}, \\ P_1 &= \{\varphi \in C(\bar{\Omega}) : \forall T \in \mathcal{T}_h, \varphi|_T \in \mathcal{P}_1\}. \end{aligned} \quad (10)$$

The lowest equal order C^0 velocity and pressure pair is

$$\mathbf{V}_h = \{P_1 \cap H_0^1(\Omega)\}^2 \text{ and } S_h = P_1 \cap L_0^2(\Omega). \quad (11)$$

As we know, the above finite element pair does not satisfy the discrete inf-sup condition.

Now, the stabilized low-order mixed finite element method for Stokes equations in [Bochev, Dohrmann, and Gunzburger (2006)] is: find $(\mathbf{u}, p) \in (\mathbf{V}, S)$ such that

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) &= (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{V}, \\ b(\mathbf{u}, q) - G(p, q) &= 0, \quad \forall q \in S, \end{aligned} \quad (12)$$

where

$$G(p, q) = \int_{\Omega} (p - \Pi_0 p)(q - \Pi_0 q) d\Omega. \quad (13)$$

Here $\Pi_0 : L^2(\Omega) \mapsto R_0$ is the orthogonal projection operator. By restricting (12) to the finite element spaces we can obtain the stabilization method. That is: find $(\mathbf{u}_h, p_h) \in (\mathbf{V}_h, S_h)$ such that

$$\begin{aligned} a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) &= (\mathbf{f}, \mathbf{v}_h), \quad \forall \mathbf{v}_h \in \mathbf{V}_h, \\ b(\mathbf{u}_h, q_h) - G(p_h, q_h) &= 0, \quad \forall q_h \in S_h, \end{aligned} \quad (14)$$

It has been proved in Ref. [Bochev, Dohrmann, and Gunzburger (2006)] that (14) is a stable variational problem.

3.1 Error estimator

Two residual a posteriori error estimators at the element level are defined, see [He, Xie, and Zheng (2010)]. For simplicity, let $e_h = \mathbf{u} - \mathbf{u}_h$ and $\varepsilon_h = p - p_h$, where (\mathbf{u}, p) is the solution of the Stokes equations (8), and (\mathbf{u}_h, p_h) is the solution of the stabilized mixed problem (14). In the residual error estimators, one is related to the stabilization term, the other is based on the residue of the finite element discretization.

(i) The first kind of indicator

$$\eta_{\Pi} = \|(I - \Pi_0)p_h\|_0. \quad (15)$$

(ii) The second kind of indicator

$$\eta_{R,T} = \left\{ h_T^2 \|P_0 \mathbf{f} - \nabla p_h\|_{0,T}^2 + \frac{1}{2} \sum_{e \subset \partial T \cap \Omega} h_e \left\| \left[\frac{\partial \mathbf{u}_h}{\partial n} - p_h \cdot \mathbf{n} \right]_J \right\|_{0,e}^2 + \|\nabla \cdot \mathbf{u}_h\|_{0,T}^2 \right\}^{\frac{1}{2}}, \quad (16)$$

where I is the identity operator, P_0 denote the L^2 -projection onto P_1 , and the $[\cdot]_J$ denote the jump in the normal flux across the edge e . Then the total a posteriori error is

$$\eta_R = \eta_{\Pi} + \left(\sum_{T \in \mathcal{T}_h} \eta_{R,T}^2 \right)^{\frac{1}{2}}. \quad (17)$$

In order to illustrate that the mesh refinement strategy presented in this study is independent of the choice of the specific error estimator, the projection error estimator η_P is also used in comparison with the residual error estimator η_R , where η_P is defined by $\eta_P = \left(\sum_{T \in \mathcal{T}} \eta_{P,T}^2 \right)^{\frac{1}{2}}$ with

$$\eta_{P,T} = \|(I - \Pi_1)\nabla \mathbf{u}_h\|_{0,T} + \|(I - \Pi_0)p_h\|_{0,T}. \quad (18)$$

Note that the projection operator $\Pi_1 : L^2(\Omega) \rightarrow R_1$ has the same properties as Π_0 , see Ref. [Zheng, Hou, and Shi (2010)].

3.2 BLMG-based adaptive mixed finite element method

The low-order mixed finite element methods are applied to simulate the Stokes problem because of their advantages in computation. However, it is well known that the lowest equal-order triangular elements based on linear approximation of the velocity field are not stable. There are many stabilized finite element methods available to fix this deficiency [Becker and Braack (2001); Dohrmann and Bochev (2004)]. In this study, a stabilized approach [Bochev, Dohrmann, and Gunzburger (2006)] by adding an stabilized term $G(p, q)$ to the variational formulation is applied, which ensures the discretization form to be unconditionally stable. Next, a tolerance η^* is set, and the BLMG-based adaptive finite element strategy is carried out as follows.

Step 1: Generate an initial coarse triangulation $\mathcal{T}^{(0)}$ of Ω and set $l = 0$. Solve the problem (8) on $\mathcal{T}^{(0)}$.

Step 2: Compute

$$\eta_R = \eta_\Pi + \left(\sum_{T \in \mathcal{T}_h} \eta_{R,T}^2 \right)^{\frac{1}{2}} \text{ (or } \eta_P = \left(\sum_{T \in \mathcal{T}_h} \eta_{P,T}^2 \right)^{\frac{1}{2}} \text{)}.$$

If $\eta_R \leq \eta^*$ (or $\eta_P \leq \eta^*$), stop, then we obtain the final finite element solution. Otherwise, go to Step 3.

Step 3: Compute the local error estimators $\eta_{\Pi,T} + \eta_{R,T}$ (or $\eta_{P,T}$) for each element $T \in \mathcal{T}^{(l)}$. Determine a new size function $h^{(l+1)}$ using Eq. (2).

Step 4: Optimize the node placement according to $h^{(l+1)}$, and generate local meshes with the BLMG method. Set $l = l + 1$, and then go back to Step 2.

4 Numerical results

Four experiments are executed to verify the efficiency and reliability of the BLMG-based adaptive mixed finite element method. In all cases, we set the coefficient $k_0 = 1.0$, the mass coefficient $\tilde{m} = 1.0$ and the damping coefficient $\tilde{c} = 3.8429$ in (4) and (5), respectively.

For convenience of presentation, we introduce the following notations:

- E^l ::= number of elements in $\mathcal{T}^{(l)}$;
- Rate means experimental order of convergence. $Rate := \frac{2 \log(e_h^l/e_h^{l-1})}{\log(E^{l-1}/E^l)}$,
- $q(T) := \frac{2R_T}{r_T} = \frac{(a+b-c)(b+c-a)(a+c-b)}{abc}$ is the quality for any triangle T , where R_T and r_T are the radii of the largest inscribed circle and the smallest circumscribed circle, a , b and c are side lengths of T , respectively. Let q_{avg} denote the average values of the quality of all elements.

4.1 A singular problem

In this example, based on the residual error estimator η_R mentioned in Section 2.1, the numerical results calculated by the BLMG-based adaptive method are compared with the results using public domain finite element software FreeFem++ [Hecht (2012)], where adaptive meshes are generated successively with the mesh generator *bamg* [Hecht (1998)].

We consider the flow problem in a circular segment with radius 1 and angle $3\pi/2$, the boundary conditions are [Bank and Welfert (1990)]

$$\begin{aligned} \mathbf{u}_b = & \frac{1}{2} \{ (\xi^2 - 1) [\cos((2 - \alpha)\theta) - \cos(\alpha\theta) - \xi^{-1} \sin((2 - \alpha)\theta)] \\ & - \xi^{-1} (1 + 3\xi^2) \sin(\alpha\theta), \\ & (\xi^2 - 1) [\xi^{-1} (\cos((2 - \alpha)\theta) - \cos(\alpha\theta)) + \sin((2 - \alpha)\theta)] \\ & - (\xi^2 + 3) \sin(\alpha\theta) \}, \end{aligned}$$

where we have $\alpha = 856399/1572864 \simeq 0.54$ and $\xi = \sqrt{(1 + \alpha)/(1 - \alpha)} \simeq 1.84$. (r, θ) is a polar representation of a point in the circular sector. The exact solution is given by

$$\begin{aligned} \mathbf{u} &= r^\alpha \mathbf{u}_b \\ p &= 2r^{-(1-\alpha)} (\xi^2 - 1) \{ \cos((1 - \alpha)\theta) - \xi^{-1} \sin((1 - \alpha)\theta) \} \end{aligned}$$

which is singular at the center of the disk.

We start with an initial mesh generated with the Delaunay triangulation at $h = 0.2$, which is shown in Fig. 4(a). The successive meshes are obtained by using the FreeFem++ and the BLMG-based adaptive finite element method, respectively, in which the meshes for the 2nd refinement level are shown in Figs. 4(b) and 4(c). It can be seen that both adaptive methods capture the singularity at the origin and produce very small elements.

However, the meshes generated with the BLMG method have good smoothness and transition where the gradient of mesh is changed gradually. To illustrate this point, we choose two adaptive meshes with the approximate number of elements generated by the FreeFem++ and the BLMG-based adaptive method, respectively. The mesh on $\mathcal{T}_h^{(3)}$ with 1070 elements is generated by the FreeFem++ which is shown in Fig. 5, and the mesh on $\mathcal{T}_h^{(4)}$ with 1152 elements is generated by the BLMG-based adaptive method which is shown in Fig. 6. The triangular elements in Fig. 5(a) are then enlarged twice, and the enlarged views are shown in Figs. 5(b) and 5(c). Also, Figs. 6(b) and 6(c) give the enlarged views of Fig. 6(a). From Figs. 5 and 6, we can obtain that the mesh generated by the BLMG method has better gradualness compared to that using the FreeFem++, especially in the corner area.

Let $h_{actual}^{(l)}$ denote the actual size at each mesh vertex which is defined as the average of the size of the edges sharing the vertex. $h^{(l)}$ denotes the desired mesh size at each mesh vertex which is defined in (2). Then $\left| \frac{h_{actual}^{(l)}}{h^{(l)}} - 1 \right|$ denote the deviation between the actual mesh size and the desired mesh size at each vertex, and it's much better

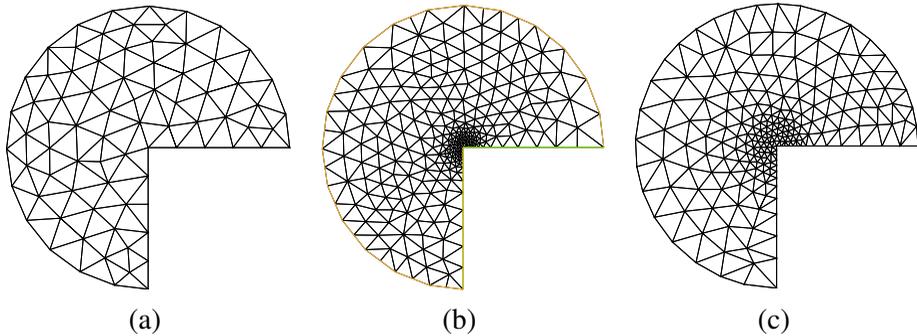


Figure 4: From left to right: (a) initial mesh, (b) adaptive meshes on $\mathcal{T}_h^{(2)}$ generated by FreeFem++ and (c) by the BLMG method.

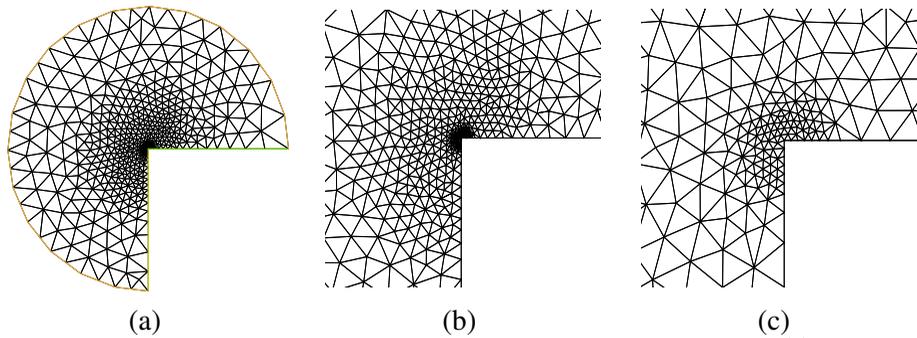


Figure 5: (a) Adaptive mesh generated with the FreeFem++ on $\mathcal{T}_h^{(3)}$, (b) and (c) are the enlarged views.

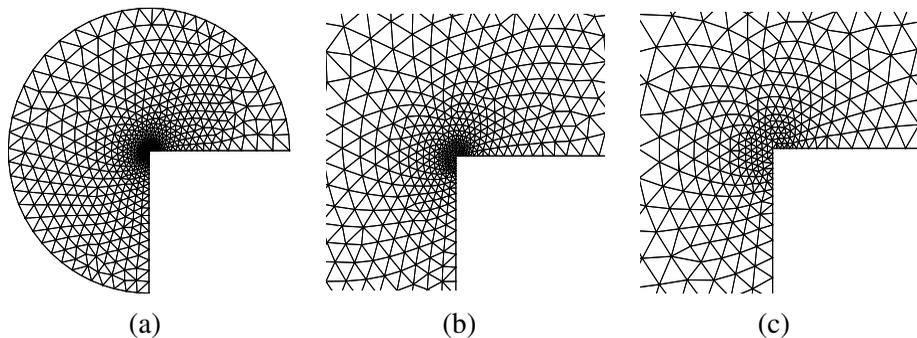


Figure 6: (a) Adaptive mesh generated with the BLMG method on $\mathcal{T}_h^{(4)}$, (b) and (c) are the enlarged views.

Table 1: The convergence analysis for a sequence of adaptive meshes using FreeFem++.

l	E	q_{avg}	$\left \frac{h^{(l)}_{actual}}{h^{(l)}} - 1 \right $		$\frac{\ \nabla(\mathbf{u}-\mathbf{u}_h)\ _0}{\ \nabla\mathbf{u}\ _0}$		$\frac{\ p-p_h\ _0}{\ p\ _0}$	
			Max	Average	Error	Rate	Error	Rate
0	128	0.9216	-	-	0.2024	-	0.7414	-
1	241	0.9216	0.3705	0.08550	0.1383	1.2035	0.5163	1.1440
2	495	0.9293	0.4312	0.07646	0.08013	1.5172	0.3627	0.9809
3	1070	0.9365	0.7167	0.07230	0.05572	0.9424	0.2574	0.8901
4	2122	0.9448	0.7685	0.06497	0.03970	0.9905	0.1803	1.0405

Table 2: The convergence analysis for a sequence of adaptive meshes using BLMG-based adaptive method.

l	E	q_{avg}	$\left \frac{h^{(l)}_{actual}}{h^{(l)}} - 1 \right $		$\frac{\ \nabla(\mathbf{u}-\mathbf{u}_h)\ _0}{\ \nabla\mathbf{u}\ _0}$		$\frac{\ p-p_h\ _0}{\ p\ _0}$	
			Max	Average	Error	Rate	Error	Rate
0	128	0.9216	-	-	0.2024	-	0.7414	-
1	210	0.9169	0.2301	0.08287	0.1408	1.4662	0.5542	1.1752
2	344	0.9380	0.2694	0.05899	0.09904	1.4268	0.4084	1.2374
3	664	0.9590	0.2442	0.05565	0.06290	1.3808	0.2886	1.0776
4	1152	0.9722	0.3196	0.04440	0.04387	1.3081	0.2233	0.9045
5	2209	0.9723	0.2955	0.03350	0.02932	1.2376	0.1512	1.1991

as $\left| \frac{h^{(l)}_{actual}}{h^{(l)}} - 1 \right| \rightarrow 0$, this indicates that the generated mesh satisfies the requirement of the desired mesh size.

The accuracy analysis of the two adaptive strategies for the stabilized finite elements is reported in Tables 1 and 2. From these tables, we note that the average quality of meshes generated by the adaptive BLMG method is slightly higher than that obtained using the FreeFem++. A smaller deviation between the actual mesh size and the desired mesh size is obtained in the adaptive meshes generated by our method. In addition, in the case of almost the same number of elements, for example, when $l = 4$ in Table 1 and $l = 5$ in Table 2, the BLMG-based adaptive method can get a better approximation for the singular problem. Therefore, these results demonstrate that the BLMG-based refinement method is applicable and can be able to approximate the solution for the Stokes problem.

Table 3: The convergence analysis for a sequence of adaptive meshes based on η_R .

l	E	q_{avg}	$\bar{\eta}$	$\frac{\ \mathbf{u}-\mathbf{u}_h\ _0}{\ \mathbf{u}\ _0}$		$\frac{\ \nabla(\mathbf{u}-\mathbf{u}_h)\ _0}{\ \nabla\mathbf{u}\ _0}$		$\frac{\ p-p_h\ _0}{\ p\ _0}$	
				Error	Rate	Error	Rate	Error	Rate
0	116	0.9407	1.8522	0.2297	-	0.6667	-	1.3070	-
1	192	0.8980	0.5048	0.07647	4.3649	0.4177	1.8560	0.3507	5.2210
2	459	0.9104	0.1428	0.01687	3.4681	0.1994	1.6969	0.1246	2.3746
3	1034	0.9203	0.05749	0.006003	2.5445	0.09684	1.7784	0.07046	1.4041
4	2292	0.9491	0.02262	0.002742	1.9684	0.06640	0.9483	0.03781	1.5642

4.2 L-shape domain problem

The second example is a flow problem in the L-shape domain $\Omega = (-1, 1)^2 - [0, 1]^2$. The right-hand side is defined by $\mathbf{f} = -\Delta\mathbf{u} + \nabla p$ with the following prescribed exact solution:

$$u_1 = \frac{y-0.1}{\sqrt{(x-0.1)^2+(y-0.1)^2}}$$

$$u_2 = -\frac{x-0.1}{\sqrt{(x-0.1)^2+(y-0.1)^2}}$$

$$p = \frac{1}{y+1.05} - \frac{\log(2.05)+\log(1.05)-2\log(0.05)}{3}$$

It is clear that $\mathbf{u} = (u_1, u_2)$ and p are singular at the point $(0.1, 0.1)$ and along the line $y = -1.05$, respectively. Thus, we expect the refined meshes occur around the origin and along the line $y = -1$.

Firstly, we consider the residual estimator η_R for the BLMG-based adaptive stabilized method and report numerical results in Table 3. The adaptive meshes based on η_R for the 1st, 3rd and 4th refinement levels are shown in Fig. 7. We can see that the refinements get good approximation solution as $h \rightarrow 0$, and this can be verified from the relative error values $\frac{\|\mathbf{u}-\mathbf{u}_h\|_0}{\|\mathbf{u}\|_0}$, $\frac{\|\nabla(\mathbf{u}-\mathbf{u}_h)\|_0}{\|\nabla\mathbf{u}\|_0}$ and $\frac{\|p-p_h\|_0}{\|p\|_0}$, respectively. From Table 3, we also note that the mean value of a posterior error estimate over all the elements $\bar{\eta}$ is strictly reduced and almost equally distributed on the elements at the end.

Secondly, we use the projection estimator η_P for the BLMG-based adaptive stabilized method. The numerical results are given in Table 4, which are consistent with the results in Table 3. Fig. 8 shows the refined meshes at the 1st, 3rd and 4th refinement levels with estimator η_P . From Figs. 7 and 8, the refined meshes occur around the origin and along the line $y = -1$ for the both error estimators as we expected. This confirms that the BLMG refinement strategy is flexible and independent of the choice of the error estimator. Moreover, from Tables 3 and 4, the numerical results show that the BLMG-based adaptive stabilized method works well for the both error estimators, and high convergence rate can be obtained.

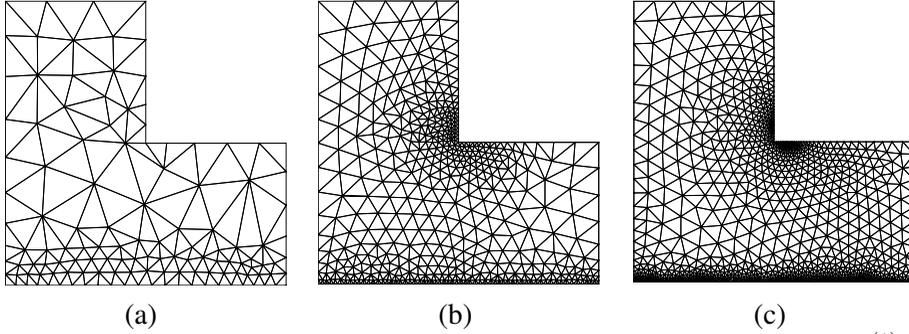


Figure 7: Adaptive meshes generated with the error estimator η_R . (a) $\mathcal{T}_h^{(1)}$, (b) $\mathcal{T}_h^{(3)}$ and (c) $\mathcal{T}_h^{(4)}$.

Table 4: The convergence analysis for a sequence of adaptive meshes based on η_P .

l	E	q_{avg}	$\bar{\eta}$	$\frac{\ \mathbf{u}-\mathbf{u}_h\ _0}{\ \mathbf{u}\ _0}$		$\frac{\ \nabla(\mathbf{u}-\mathbf{u}_h)\ _0}{\ \nabla\mathbf{u}\ _0}$		$\frac{\ p-p_h\ _0}{\ p\ _0}$	
				Error	Rate	Error	Rate	Error	Rate
0	116	0.9407	0.2555	0.2297	-	0.6667	-	1.3070	-
1	260	0.9076	0.09555	0.08213	2.5483	0.3200	1.8189	0.5474	2.1565
2	537	0.9123	0.03884	0.01661	4.4068	0.1675	1.7843	0.1502	3.5655
3	1179	0.9397	0.01688	0.004537	3.3006	0.09964	1.3215	0.04950	2.8234
4	2532	0.9472	0.007857	0.001932	2.2343	0.07294	0.8164	0.02635	1.6503

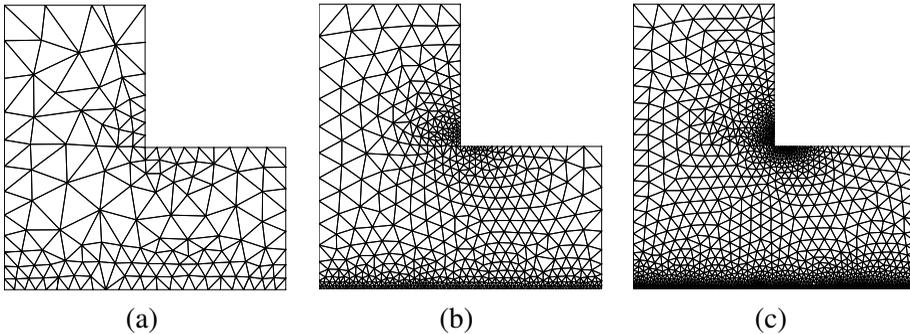


Figure 8: Adaptive meshes generated with the error estimator η_P . (a) $\mathcal{T}_h^{(1)}$, (b) $\mathcal{T}_h^{(3)}$ and (c) $\mathcal{T}_h^{(4)}$.

4.3 The driven cavity flow

The driven cavity flow is a popular benchmark problem for testing numerical schemes. In this test, fluid is enclosed in a square box. There is a unit tangential velocity $\mathbf{u} = (1, 0)$ at the top boundary without any other force source. In addition, the velocity is enforced zero on both lateral sides and the bottom side. It is well studied that there are two singularities arising at the top corners of the square box.

Firstly, we start with an initial mesh generated from the Delaunay triangulation with $h = 0.1$ which is shown in Fig. 9(a). Figs. 9(b) and 9(c) show the results for adaptive refinements with estimator η_R , we note that, in the successive iterations the adaptive strategies generate more triangles in the two upper corners of the cavity due to the two singularities, which meets the requirement of the problem.

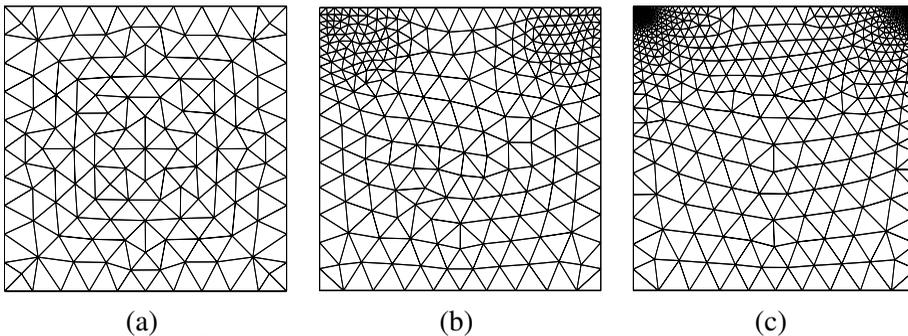


Figure 9: From left to right: (a) the initial mesh, (b) the first adaptive mesh, and (c) the third adaptive mesh.

Then we compare the results of our method based on η_R ($E=1282$) with the discrete solution obtained via the Galerkin method on a much uniform fine mesh ($E=5992$). In Fig. 10, we draw the x component of the velocity $u_1(x, y)$ along the vertical centerline $x = 0.5$, and the y component of the velocity along the horizon centerline. We can see that numerical results of the BLMG-based adaptive mixed finite element method are very close to the ones of Galerkin method on the much fine mesh.

Finally, in order to show the prominent features of our adaptive stabilized methods, the pressure level lines for the driven cavity are shown in Fig. 11 by using our adaptive stabilized method and the Galerkin method. The BLMG-based adaptive stabilized method uses less elements ($E=1282$) to obtain the results, which are similar to that of the Galerkin method on the much fine mesh ($E=80000$) in [Song, Hou, and Zheng (2013)]. This implies that our adaptive algorithm saves lots of computer memories to solve this fluid problem, and we can achieve higher accuracy with less degree of freedom.

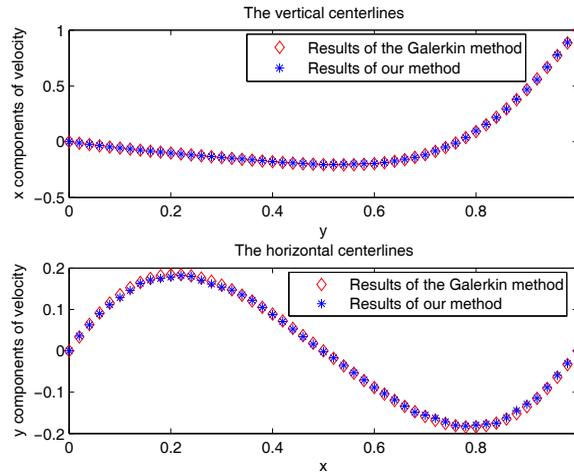


Figure 10: The velocity along the centerlines for the cavity flow.

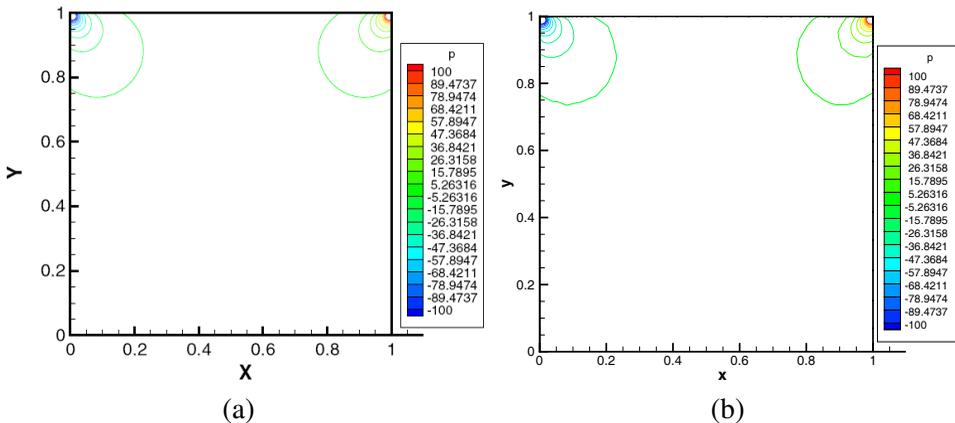


Figure 11: The pressure level lines: (a) with the Galerkin method on the much fine mesh in [Song, Hou, and Zheng (2013)] and (b) with the BLMG-based adaptive stabilized method.

4.4 The stokes flow over a step

The flow over a step is another benchmark problem which possesses the corner singularity. The computational domain is given by $\Omega = [0, 4] \times [0, 1] - [1.2, 1.6] \times [0, 0.4]$, and the flow is constrained with the Dirichlet boundary condition $\mathbf{u} = (0, 0)$ at the upper and lower boundaries. Moreover, the inflow at $x = 0$ is a parabolic flow $\mathbf{u} = (4y(1 - y), 0)$, whereas the outflow is a natural boundary condition.

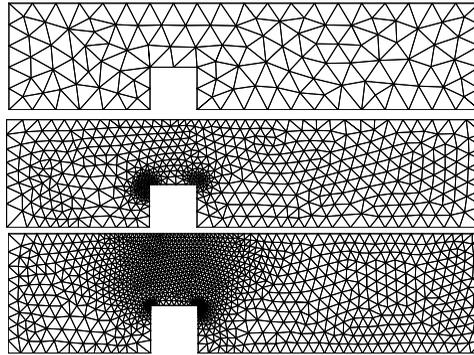


Figure 12: From top to bottom: The initial mesh, the first adaptive mesh, and the second adaptive mesh.

Fig. 12 shows the refined meshes generated by the BLMG method based on the residual error estimator η_R . From Fig. 12, the mesh refinement appears near the two corners of the step after several adaptive iterations. The related contour plots of the pressure based on these meshes are shown in Figs. 13(a)-13(c). We can observe that numerical oscillation appears in the initial grid, while after enough adaptive iterations, the singular nature of the pressure is well captured with less oscillations. Meanwhile, the contour plot of the pressure based on uniform mesh using the Galerkin method ($E=7205$) is also shown in Fig. 13(d), which is similar to that of the BLMG-based adaptive stabilized method ($E=2533$) given in Fig. 13(c). This verifies again that our method can use less elements to achieve good results.

5 Conclusions

In this paper, we develop an adaptive node-based local mesh refining algorithm BLMG for the Stokes equations. Numerical tests show the efficiency and reliability of the adaptive BLMG method. The adaptive meshes generated by our method have good transition among elements, and the refinement or coarsening of the mesh in the adaptive process can be achieved at the same framework through modifying

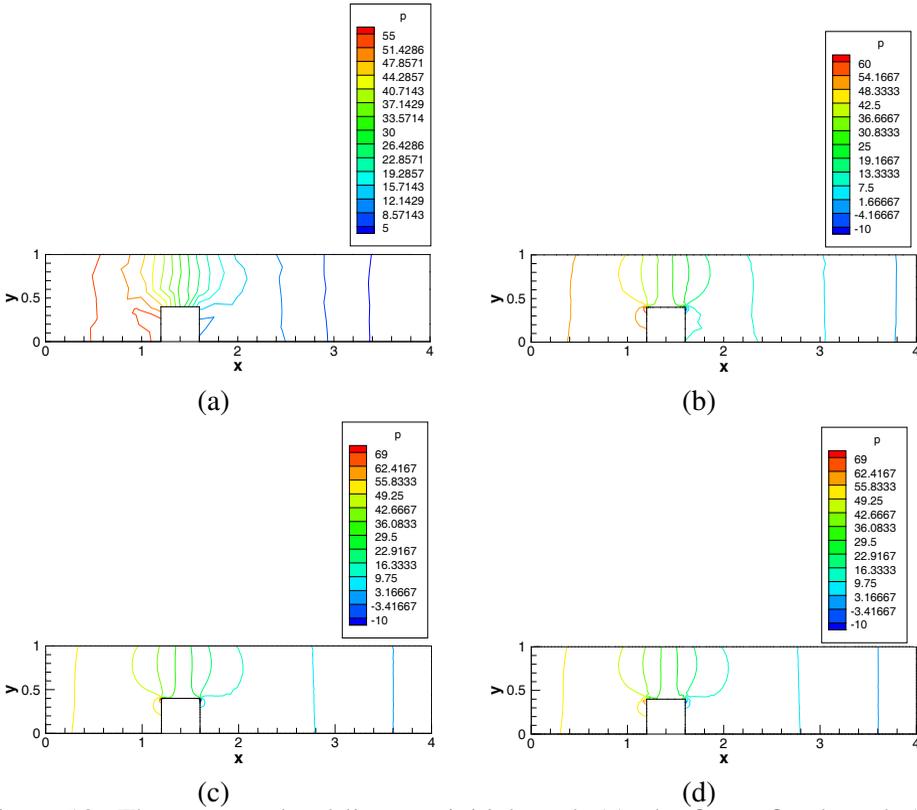


Figure 13: The pressure level lines on initial mesh (a), the first refined mesh (b) and the second refined mesh (c) using the BLMG-based adaptive stabilized method. The pressure level line on uniform mesh using the Galerkin method (d).

the size function and controlling the node distribution. It also shows numerically that the BLMG-based adaptive stabilized method is very flexible which is independent of specific error estimators. Besides, since the satellite elements related to each central node are generated locally, the BLMG method has strong potential of parallelism, and the study of the parallel BLMG-based adaptive finite element method is our future work.

Acknowledgement: This research was supported by National Natural Science Foundation of China (Grant Nos: 11471262, 11471261, 11101333), the Doctorate Foundation of Northwestern Polytechnical University (Grant No: CX201236), and NPU Foundation for Fundamental Research (Grant No: 3102014JCQ01075).

References

- Araya, R.; Barrenechea, G. R.; Poza, A.** (2008): An adaptive stabilized finite element method for the generalized stokes problem. *Journal of Computational and Applied Mathematics*, vol. 214, no. 2, pp. 457–479.
- Arya, S.; Mount, D. M.; Netanyahu, N. S.; Silverman, R.; Wu, A. Y.** (1998): An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *Journal of the ACM*, vol. 45, no. 6, pp. 891–923.
- Babuvška, I.; Rheinboldt, W. C.** (1978): Error estimates for adaptive finite element computations. *SIAM Journal on Numerical Analysis*, vol. 15, no. 4, pp. 736–754.
- Bank, R. E.; Welfert, B. D.** (1990): A posteriori error estimates for the stokes equations: a comparison. *Computer Methods in Applied Mechanics and Engineering*, vol. 82, no. 1, pp. 323–340.
- Becker, R.; Braack, M.** (2001): A finite element pressure gradient stabilization for the stokes equations based on local projections. *Calcolo*, vol. 38, no. 4, pp. 173–199.
- Biboulet, N.; Gravouil, A.; Dureisseix, D.; Lubrecht, A.; Combescure, A.** (2013): An efficient linear elastic fem solver using automatic local grid refinement and accuracy control. *Finite Elements in Analysis and Design*, vol. 68, pp. 28–38.
- Binev, P.; Dahmen, W.; DeVore, R.** (2004): Adaptive finite element methods with convergence rates. *Numerische Mathematik*, vol. 97, no. 2, pp. 219–268.
- Bochev, P. B.; Dohrmann, C. R.; Gunzburger, M. D.** (2006): Stabilization of low-order mixed finite elements for the stokes equations. *SIAM Journal on Numerical Analysis*, vol. 44, no. 1, pp. 82–101.
- Boussetta, R.; Coupez, T.; Fourment, L.** (2006): Adaptive remeshing based on a posteriori error estimation for forging simulation. *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 48–49, pp. 6626–6645.
- 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.
- Cuvelier, C.; Segal, A.; Van Steenhoven, A. A.** (1986): *Finite element methods and Navier-Stokes equations*, volume 22. Springer.
- Dohrmann, C. R.; Bochev, P. B.** (2004): A stabilized finite element method for the stokes problem based on polynomial pressure projections. *International Journal for Numerical Methods in Fluids*, vol. 46, no. 2, pp. 183–201.

Ervin, V. J.; Phillips, T. N. (2006): Residual a posteriori error estimator for a three-field model of a non-linear generalized stokes problem. *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 19, pp. 2599–2610.

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. 481–511.

He, Y.; Xie, C.; Zheng, H. (2010): A posteriori error estimate for stabilized low-order mixed fem for the stokes equations. *Advances in Applied Mathematics and Mechanics*, vol. 2, no. 6, pp. 798–809.

Hecht, F. (1998): The mesh adapting software: bamg. inria report, 1998.

Hecht, F. (2012): New development in freefem++. *Journal of Numerical Mathematics*, vol. 20, no. 3-4, pp. 251–265.

Huang, Y.; Qin, H.; Wang, D.; Du, Q. (2011): Convergent Adaptive Finite Element Method Based on Centroidal Voronoi Tessellations and Superconvergence. *Communications in Computational Physics*, vol. 10, no. 2, pp. 339–370.

Ju, L.; Gunzburger, M.; Zhao, W. (2006): Adaptive Finite Element Methods for Elliptic PDEs Based on Conforming Centroidal Voronoi-Delaunay Triangulations. *SIAM Journal on Science Computing*, vol. 28, no. 6, pp. 2023–2053.

Kay, D.; Silvester, D. (1999): A posteriori error estimation for stabilized mixed approximations of the stokes equations. *SIAM Journal on Scientific Computing*, vol. 21, no. 4, pp. 1321–1336.

Labergère, C.; Rassineux, A. A.; Saanouni, K. K. (2011): 2d adaptive mesh methodology for the simulation of metal forming processes with damage. *International Journal of Material Forming*, vol. 4, no. 3, pp. 317–328.

Li, J.; He, Y. (2008): A stabilized finite element method based on two local gauss integrations for the stokes equations. *Journal of Computational and Applied Mathematics*, vol. 214, no. 1, pp. 58–65.

Liao, Q.; Silvester, D. (2012): A simple yet effective a posteriori estimator for classical mixed approximation of stokes equations. *Applied Numerical Mathematics*, vol. 62, no. 9, pp. 1242–1256.

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.

Morin, P.; Nochetto, R. H.; Siebert, K. G. (2000): Data oscillation and convergence of adaptive fem. *SIAM Journal on Numerical Analysis*, vol. 38, no. 2, pp. 466–488.

- Nicolas, G.; Fouquet, T.** (2013): Adaptive mesh refinement for conformal hexahedral meshes . *Finite Elements in Analysis and Design*, vol. 67, pp. 1–12.
- Nie, Y. F.; Chang, S.** (2006): Node-based local mesh generation algorithm. *Chinese Journal of Computational Mechanics*, vol. 23, no. 2, pp. 252–256.
- Nie, Y. F.; Zhang, W. W.; Liu, Y.; Wang, L.** (2010): A Node Placement Method with high quality for mesh generation. In *IOP Conference Series: Materials Science and Engineering*, volume 10, pg. 012218. IOP Publishing.
- Nie, Y. F.; Zhang, W. W.; Qi, N.; Li, Y. Q.** (2014): Parallel node placement method by bubble simulation. *Computer Physics Communications*, vol. 185, no. 3, pp. 798–808.
- Nithiarasu, P.; Zienkiewicz, O. C.** (2000): Adaptive mesh generation for fluid mechanics problems. *International Journal for Numerical Methods in Engineering*, vol. 47, no. 1-3, pp. 629–662.
- Picasso, M.** (2003): An anisotropic error indicator based on zienkiewicz–zhu error estimator: Application to elliptic and parabolic problems. *SIAM Journal on Scientific Computing*, vol. 24, no. 4, pp. 1328–1355.
- Qi, N.; Nie, Y. F.; Zhang, W. W.** (2014): Acceleration strategies based on an improved bubble packing method. *Communications in Computational Physics*, vol. 16, no. 1, pp. 115–135.
- Shewchuk, J.** (2005): Triangle: A Two-Dimensional Quality Mesh Generator and Delaunay Triangulator. <http://www.cs.cmu.edu/~quake/triangle.html>, 2005.
- 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.
- Song, L.; Hou, Y.; Zheng, H.** (2013): Adaptive variational multiscale method for the stokes equations. *International Journal for Numerical Methods in Fluids*, vol. 71, no. 11, pp. 1369–1381.
- Verfürth, R.** (1989): A posteriori error estimators for the stokes equations. *Numerische Mathematik*, vol. 55, no. 3, pp. 309–325.
- Verfürth, R.** (1991): A posteriori error estimators for the stokes equations ii non-conforming discretizations. *Numerische Mathematik*, vol. 60, no. 1, pp. 235–249.
- Zhang, W. W.; Nie, Y. F.; Li, Y. Q.** (2012): A new anisotropic local meshing method and its application in parametric surface triangulation . *CMES-Computer Modeling in Engineering & Sciences*, vol. 88, no. 6, pp. 507–529.

Zheng, H.; Hou, Y.; Shi, F. (2010): A posteriori error estimates of stabilization of low-order mixed finite elements for incompressible flow. *SIAM Journal on Scientific Computing*, vol. 32, no. 3, pp. 1346–1360.

