

## Research Article

# Numerical Study on Several Stabilized Finite Element Methods for the Steady Incompressible Flow Problem with Damping

Jilian Wu, Pengzhan Huang, and Xinlong Feng

College of Mathematics and System Sciences, Xinjiang University, Urumqi 830046, China

Correspondence should be addressed to Xinlong Feng; [fxlmath@gmail.com](mailto:fxlmath@gmail.com)

Received 24 August 2013; Accepted 19 October 2013

Academic Editor: Suh-Yuh Yang

Copyright © 2013 Jilian Wu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We discuss several stabilized finite element methods, which are penalty, regular, multiscale enrichment, and local Gauss integration method, for the steady incompressible flow problem with damping based on the lowest equal-order finite element space pair. Then we give the numerical comparisons between them in three numerical examples which show that the local Gauss integration method has good stability, efficiency, and accuracy properties and it is better than the others for the steady incompressible flow problem with damping on the whole. However, to our surprise, the regular method spends less CPU-time and has better accuracy properties by using Crout solver.

## 1. Introduction

In this paper, we will consider the following steady incompressible flow problem with damping: seek  $(\mathbf{u}, p)$  such that

$$\begin{aligned} -\nu\Delta\mathbf{u} + \alpha|\mathbf{u}|^{r-2}\mathbf{u} + \nabla p &= \mathbf{f} \quad \text{in } \Omega, \\ \nabla \cdot \mathbf{u} &= 0 \quad \text{in } \Omega, \\ \mathbf{u} &= 0 \quad \text{on } \partial\Omega, \end{aligned} \quad (1)$$

where  $\Omega \subset \mathbb{R}^2$  is a convex polygonal domain with a Lipschitz continuous boundary  $\partial\Omega$  and the symbols  $\Delta$ ,  $\nabla$ , and  $\nabla \cdot$  denote the Laplacian, gradient, and divergence operators, respectively;  $\mathbf{u} = (u_1(x, y), u_2(x, y))$ ,  $p = p(x, y)$ , and  $\mathbf{f} = (f_1(x, y), f_2(x, y))$  represent the velocity vector, the pressure, and the prescribed body force, respectively. Further,  $\alpha|\mathbf{u}|^{r-2}\mathbf{u}$  represents the damping term with two constants  $\alpha > 0$  and  $r > 2$ . In addition,  $|\mathbf{u}| = \sqrt{u_1^2 + u_2^2}$ , and we linearize the nonlinear term  $\alpha|\mathbf{u}|^{r-2}\mathbf{u}$  by allowing it to lag one step behind.

These equations describe various physical situations such as porous media flow, drag or friction effects, and some dissipative mechanisms from the resistance to the motion of the flow. If the damping system is different, energy dissipation is different. So the application of these equations is very extensive in the daily life (see [1–6] and references therein).

It is well known that it is very difficult to compute some PDEs directly while numerical method plays an important role in these problems, so developing an efficient and effective computational method for solving the incompressible flow problem has practical significance and has drawn the attention of many researchers (see [4, 7–12] and the references cited therein). During this time, mixed finite element methods [13] are a natural choice for solving fluid mechanics equations because these equations naturally appear in mixed form in terms of velocity and pressure.

In the analysis and practice of employing mixed finite element methods in solving the incompressible flow problems, the inf-sup condition has played an important role because it ensures stability and accuracy of the underlying numerical schemes. Pairs of finite element spaces that are used to approximate the velocity and the pressure unknown are said to be stable if they satisfy the inf-sup condition. Intuitively speaking, the inf-sup condition is something that enforces a certain correlation between two finite element spaces so that they both have the required properties when employed for the incompressible flow problems. However, due to computational convenience and efficiency in practice, some mixed finite element pairs which do not satisfy the inf-sup condition are also popular. Thus, much attention has been paid to the study of the stabilized method for the Stokes problem [4, 7].

In the present years, studies have focused on stabilization techniques [14, 15], which include penalty method [16, 17], regular method [15], multiscale enrichment method [18], and local Gauss integration method [19]. There exist a lot of theoretical results for the stabilized mixed finite element methods for the Stokes equations and the comparisons between them are also given (see [4, 7, 15, 16, 18–21] and the references cited therein). It is pointed out that authors considered the performance of several stabilized methods for the Stokes equations based on the lowest equal-order pairs in [21]. And authors studied several stabilized methods for the Stokes eigenvalue problem by using different conforming and nonconforming lower-order pairs in [7]. In this report, we will adopt four kinds of stabilized finite element methods but will mainly focus on the incompressible flow problem with damping based on the lowest equal-order pairs. Moreover, we present the comparisons between these methods of the considered problem.

A brief outline of the rest of our paper is organized as follows: in Section 2, we introduce some notations and present some preliminary materials and some well-known results of the steady incompressible flow problem with damping to be used in our subsequent sections; then in Section 3, we review several stabilized mixed finite element methods and recall their key stabilization techniques; in Section 4, comparisons between these stabilized methods are performed numerically; finally, we end with some short conclusions in Section 5.

## 2. Notations and Preliminaries

We will use the following Hilbert spaces:

$$\begin{aligned} \mathbf{X} &= H_0^1(\Omega)^2 \cap L^r(\Omega)^2, \\ \mathbf{Y} &= L^2(\Omega)^2, \\ M &= L_0^2(\Omega) = \left\{ q \in L^2(\Omega) : \int_{\Omega} q dx = 0 \right\}. \end{aligned} \quad (2)$$

Here and in what follows, the spaces  $L^2(\Omega)^m$  ( $m = 1, 2$ ) are equipped with the  $L^2$ -scalar product  $(\cdot, \cdot)$  and  $L^2$ -norm  $\|\cdot\|_{L^2}$  or  $\|\cdot\|_0$ , respectively. Further, we will consider the standard definitions for Sobolev spaces  $W^{m,p}(\Omega)$  which, for any integer  $m > 0$  and any number  $p \geq 1$ , are equipped with the norm

$$\|u\|_{m,p,\Omega} = \|u\|_{m,p}. \quad (3)$$

Notice that

$$W^{m,2}(\Omega) = H^m(\Omega), \quad \|\cdot\|_{m,2} = \|\cdot\|_m. \quad (4)$$

Then the variational formulation of problem (1) is to seek  $(\mathbf{u}, p) \in \mathbf{X} \times M$  such that

$$\begin{aligned} B((\mathbf{u}, p); (\mathbf{v}, q)) + \alpha(|\mathbf{u}|^{r-2}\mathbf{u}, \mathbf{v}) &= (\mathbf{f}, \mathbf{v}), \\ \forall (\mathbf{v}, q) &\in \mathbf{X} \times M, \end{aligned} \quad (5)$$

where

$$\begin{aligned} B((\mathbf{u}, p); (\mathbf{v}, q)) &= a(\mathbf{u}, \mathbf{v}) + d(\mathbf{v}, p) + d(\mathbf{u}, q), \\ \forall (\mathbf{u}, p), (\mathbf{v}, q) &\in \mathbf{X} \times M, \\ a(\mathbf{u}, \mathbf{v}) &= \nu(\nabla \mathbf{u}, \nabla \mathbf{v}), \quad \forall \mathbf{u}, \mathbf{v} \in \mathbf{X}, \\ d(\mathbf{v}, p) &= -(p, \operatorname{div} \mathbf{v}), \quad \forall (\mathbf{v}, p) \in \mathbf{X} \times M. \end{aligned} \quad (6)$$

Now, for convenience, we introduce the finite subspaces  $\mathbf{X}_h \times M_h \subset \mathbf{X} \times M$ , assumed to be uniformly regular in the usual sense. Suppose that  $K_h$  is a triangular decomposition of the domain  $\Omega$  and  $h$  is the maximum mesh size of the partition. Therefore, we define

$$\begin{aligned} \mathbf{X}_h &= \left\{ \mathbf{u} \in C^0(\overline{\Omega})^2 \cap \mathbf{X} : \mathbf{u}|_K \in P_1(K)^2, \forall K \in K_h \right\}, \\ M_h &= \left\{ q \in C^0(\overline{\Omega}) \cap M : q|_K \in P_1(K), \forall K \in K_h \right\}, \end{aligned} \quad (7)$$

where  $P_1(K)$  represents the space of linear functions on  $K$ . And we assume the following basis functions:

$$\begin{aligned} \mathbf{X}_h &= X_h \times X_h, \quad X_h = \operatorname{span} \{ \phi_1^h, \dots, \phi_n^h \}, \\ M_h &= \operatorname{span} \{ \phi_1^h, \dots, \phi_m^h \}, \end{aligned} \quad (8)$$

where  $n$  and  $m$  are the dimensions of  $X_h$  and  $M_h$ , respectively. And the bilinear forms  $a(\mathbf{v}_h, \mathbf{v}_h)$  and  $d(\mathbf{v}_h, q_h)$  satisfy the following conditions [7].

(i) There is a constant  $c_1 > 0$  such that

$$a(\mathbf{v}_h, \mathbf{v}_h) \geq c_1 \|\mathbf{v}_h\|_{1,\Omega}^2, \quad \forall \mathbf{v}_h \in \mathbf{Z}_h, \quad (9)$$

where  $\mathbf{Z}_h = \{ \mathbf{v}_h \in \mathbf{X}_h \mid d(\mathbf{v}_h, q_h) = 0, \forall q_h \in M_h \}$ .

(ii) There is a constant  $c_2 > 0$  independent of  $h$  such that

$$\sup_{\mathbf{v}_h \in \mathbf{X}_h} \frac{|d(\mathbf{v}_h, q_h)|}{\|\mathbf{v}_h\|_0} \geq c_2 \|q_h\|, \quad \forall (\mathbf{v}_h, q_h) \in \mathbf{X}_h \times M_h. \quad (10)$$

Then we have a unique solution  $(\mathbf{u}_h, p_h)$  of (5) satisfying

$$\begin{aligned} \|\mathbf{u} - \mathbf{u}_h\|_{1,\Omega} + \|p - p_h\|_{0,\Omega} \\ \leq c_3 \left( \inf_{\mathbf{v}_h \in \mathbf{X}_h} \|\mathbf{u} - \mathbf{v}_h\|_{1,\Omega} + \inf_{q_h \in M_h} \|p - q_h\|_{0,\Omega} \right), \end{aligned} \quad (11)$$

where  $c_3$  is a positive constant.

## 3. Stabilized Mixed Finite Element Methods

In this section, we will give several stabilized mixed finite element algorithms to show their different aspects and several ways have been used to stabilize the lowest equal-order finite element space pair as follows (see [7, 21]). First we introduce the following classical Uzawa iterative algorithm.

Let  $\mathbf{H}_1$  and  $\mathbf{H}_2$  be finite dimensional spaces. We consider the following saddle point problems:

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^T \\ \mathbf{B} & -\mathbf{C} \end{bmatrix} \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}, \quad (12)$$

where  $\mathbf{X} \in \mathbf{H}_1$  and  $\mathbf{Y} \in \mathbf{H}_2$  are unknown variables;  $\mathbf{f} \in \mathbf{H}_1$ ,  $\mathbf{g} \in \mathbf{H}_1$ ;  $\mathbf{A} : \mathbf{H}_1 \rightarrow \mathbf{H}_1$  is symmetric positive definite operator,  $\mathbf{B} : \mathbf{H}_1 \rightarrow \mathbf{H}_2$  is linear map,  $\mathbf{B}^T : \mathbf{H}_2 \rightarrow \mathbf{H}_1$  is the transpose operator of  $\mathbf{B}$ , and  $\mathbf{C} : \mathbf{H}_2 \rightarrow \mathbf{H}_2$  is symmetric semidefinite operator. If the initial values  $\mathbf{X}_0 \in \mathbf{H}_1$  and  $\mathbf{Y}_0 \in \mathbf{H}_2$  are given, then  $\mathbf{X}_i$  and  $\mathbf{Y}_i$  ( $i = 1, 2, \dots$ ) are defined by

$$\begin{aligned}\mathbf{X}_i &= \mathbf{X}_{i-1} + \mathbf{A}^{-1} (\mathbf{f} - \mathbf{A}\mathbf{X}_{i-1} - \mathbf{B}^T\mathbf{Y}_{i-1}), \\ \mathbf{Y}_i &= \mathbf{Y}_{i-1} + \tau (\mathbf{B}\mathbf{X}_i - \mathbf{C}\mathbf{Y}_{i-1} - \mathbf{g}),\end{aligned}\quad (13)$$

where  $\tau$  is a given real number.

Let  $e_i^{\mathbf{Y}} = \mathbf{Y} - \mathbf{Y}_i$  be the iteration error generated by the above method. It is easy to show that

$$e_i^{\mathbf{Y}} = (\mathbf{I} - \tau (\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{C})) e_{i-1}^{\mathbf{Y}}. \quad (14)$$

Let  $\lambda_1$  denote the largest eigenvalue of matrix  $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T + \mathbf{C}$ . Then  $\mathbf{Y}_i$  converges to  $\mathbf{Y}$  if  $\tau$  is chosen such that  $0 < \tau \leq 2/\lambda_1$ . In this case,  $\mathbf{X}_i$  and  $\mathbf{Y}_i$  converge, respectively, to  $\mathbf{X}$  and  $\mathbf{Y}$  with a rate of convergence bounded by the absolute value of  $1 - \tau\lambda_1$ . For more details about the saddle point problems, please see [22–24] and the references cited therein.

Next we present four kinds of stabilized finite element method for the steady incompressible flow problem with damping.

*Remark 1* (nonconforming finite element space). For convenience, we let  $h$  be a positive parameter and  $K_h = \{K_j\}$  a regular triangulation of  $\Omega$ . Denote by  $\Gamma_j = \partial\Omega \cap \partial K_j$  the boundary edge and by  $\Gamma_{jk} = \Gamma_{kj} = \partial K_j \cap \partial K_k$  the interior boundary. Set the centers of  $\Gamma_j$  and  $\Gamma_{jk}$  by  $\zeta_j$  and  $\zeta_{jk}$ , respectively. The nonconforming finite element space can be defined as

$$\begin{aligned}\mathbf{NC}_h &= \left\{ \mathbf{v} : \mathbf{v}_j = \mathbf{v}|_{K_j} \in P_1(K_j)^2, \mathbf{v}_j(\zeta_{jk}) = \mathbf{v}_k(\zeta_{kj}), \right. \\ &\quad \left. \mathbf{v}(\zeta_j) = 0, K_j \in K_h, \forall j, k \right\},\end{aligned}\quad (15)$$

where  $P_1(K_j)$  is the set of all polynomials on  $K_j$  of degree less than 1. Note that  $\mathbf{NC}_h$  is not a subspace of  $\mathbf{X}$ . However, in this nonconforming case, the pair of finite element spaces is  $\mathbf{NC}_h \times M_h$ ; that is, the conforming space is still used for pressure.

So the corresponding discrete variational formulation of (5) for the Stokes equations with damping reads as follows. Seek  $(\mathbf{u}_h, p_h) \in \mathbf{X}_h \times M_h$  such that

$$\begin{aligned}B((\mathbf{u}_h, p_h); (\mathbf{v}, q)) + \alpha (|\mathbf{u}_h|^{r-2} \mathbf{u}_h, \mathbf{v}) &= (\mathbf{f}, \mathbf{v}), \\ \forall (\mathbf{v}, q) \in \mathbf{X}_h \times M_h.\end{aligned}\quad (16)$$

Then we can get the following equations from (16):

$$\begin{aligned}A_1 \{u_{1,h}\} + \alpha A_2 \{u_{1,h}\} + B_x^T \{p_h\} &= \left\{ \int_{\Omega} \phi_i f_1 \right\}, \\ A_1 \{u_{2,h}\} + \alpha A_2 \{u_{2,h}\} + B_y^T \{p_h\} &= \left\{ \int_{\Omega} \phi_i f_2 \right\}, \\ B_x \{u_{1,h}\} + B_y \{u_{2,h}\} &= 0.\end{aligned}\quad (17)$$

Next, let  $\mathbf{U}_h$  and  $\mathbf{P}_h$  be the array of the velocity and pressure, respectively. Then it is easy to see that (17) can be written in matrix form:

$$\begin{bmatrix} \mathbf{A}_1 + \alpha \mathbf{A}_2 & \mathbf{B}_1^T \\ \mathbf{B}_1 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{U}_h \\ \mathbf{P}_h \end{bmatrix} = \begin{bmatrix} \mathbf{F}_h \\ 0 \end{bmatrix}, \quad (18)$$

where  $\mathbf{A}_1 = \text{diag}(A_1, A_1)$ ,  $A_1 = (A_{1i,j})$ , and  $A_{1i,j} = \nu(\nabla\phi_j^h, \nabla\phi_i^h)$ ,  $i, j = 1, \dots, n$ ;  $\mathbf{A}_2 = \text{diag}(A_2, A_2)$ ,  $A_2 = (A_{2i,j})$ , and  $A_{2i,j} = \mathcal{U}(\phi_j^h, \phi_i^h)$ ,  $i, j = 1, 2, \dots, n$ ;  $\mathbf{B}_1 = (B_x, B_y)$ ,  $B_x = (B_{xij})$ ,  $B_y = (B_{yij})$ ,  $B_{xij} = -\int_{\Omega} (\partial\phi_j^h/\partial x)\phi_i^h$ , and  $B_{yij} = -\int_{\Omega} (\partial\phi_j^h/\partial y)\phi_i^h$ ,  $i, j = 1, \dots, n$ ;  $\mathbf{U}_h = [\{u_{1,h}\}, \{u_{2,h}\}]^T$ ;  $\mathbf{F}_h = [\{\int_{\Omega} \phi_i^h f_1\}, \{\int_{\Omega} \phi_i^h f_2\}]^T$ ; and  $\mathbf{P}_h = [p_1, \dots, p_m]^T$ .

*Remark 2.* Here,  $\mathcal{U} = (\sqrt{\{u_{1,h}\}^2 + \{u_{2,h}\}^2})^{r-2}$ . Because we linearized the nonlinear term  $\alpha|\mathbf{u}|^{r-2}\mathbf{u}$ , then  $\mathcal{U}$  is a known-term in the process of solving (18). And for (16), the process of linearizing is as follows.

Given  $(\mathbf{u}_h^n, p_h^n)$ , seek  $(\mathbf{u}_h^{n+1}, p_h^{n+1}) \in \mathbf{X}_h \times M_h$  such that

$$B((\mathbf{u}_h^{n+1}, p_h^{n+1}); (\mathbf{v}, q)) + \alpha (|\mathbf{u}_h^n|^{r-2} \mathbf{u}_h^{n+1}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}). \quad (19)$$

In the nonconforming case, the discrete nonconforming formulation for the steady incompressible flow problem with damping is to seek  $(\mathbf{u}_h, p_h) \in \mathbf{NC}_h \times M_h$  such that

$$B'((\mathbf{u}_h, p_h); (\mathbf{v}, q)) + \alpha \sum_{K \in K_h} (|\mathbf{u}_h|^{r-2} \mathbf{u}_h, \mathbf{v})_K = \sum_{K \in K_h} (\mathbf{f}, \mathbf{v})_K, \quad (20)$$

where

$$\begin{aligned}B'((\mathbf{u}, p); (\mathbf{v}, q)) &= a'(\mathbf{u}, \mathbf{v}) + d'(\mathbf{v}, p) + d'(\mathbf{u}, q), \\ a'(\mathbf{u}, \mathbf{v}) &= \nu \sum_{K \in K_h} (\nabla \mathbf{u}, \nabla \mathbf{v})_K, \\ d'(\mathbf{v}, p) &= - \sum_{K \in K_h} (\text{div } \mathbf{v}, p)_K.\end{aligned}\quad (21)$$

Note that (17) is a saddle point problem and the lowest equal-order pair does not satisfy the discrete inf-sup condition

$$\sup_{\mathbf{v}_h \in \mathbf{X}_h} \frac{d(\mathbf{v}_h, p_h)}{\|\nabla \mathbf{v}_h\|_0} \geq c_4 \|p_h\|_0 \quad (22)$$

$$\text{or } \sup_{\mathbf{v}_h \in \mathbf{NC}_h} \frac{d'(\mathbf{v}_h, p_h)}{\|\nabla \mathbf{v}_h\|_{0,h}} \geq c_4 \|p_h\|_0 \quad \forall q_h \in M_h,$$

where the constant  $c_4 > 0$  is independent of  $h$  and  $\|\nabla \mathbf{v}_h\|_{0,h} = (\sum_j |\mathbf{v}|_{1,K_j}^2)^{1/2}$ , for all  $\mathbf{v} \in \mathbf{NC}_h$ .

*Algorithm 1* (Penalty method). The penalty method compensates for the inf-sup condition deficiency by adding the penalty term as follows.

Seek  $(\mathbf{u}_h, p_h) \in \mathbf{X}_h \times M_h$  such that

$$B((\mathbf{u}_h, p_h); (\mathbf{v}, q)) + \alpha (|\mathbf{u}_h|^{r-2} \mathbf{u}_h, \mathbf{v}) + \frac{\varepsilon}{\nu} (p_h, q) = (\mathbf{f}, \mathbf{v}),$$

$$\forall (\mathbf{v}, q) \in \mathbf{X}_h \times M_h, \quad (23)$$

where  $\varepsilon > 0$  is a penalty parameter. The performance of this method obviously depends on the choice of the penalty parameter  $\varepsilon$ . Then the matrix form of (23) can be expressed as

$$\begin{bmatrix} \mathbf{A}_1 + \alpha \mathbf{A}_2 & \mathbf{B}_1^T \\ \mathbf{B}_1 & \frac{\varepsilon}{\nu} \mathbf{D}_1 \end{bmatrix} \begin{bmatrix} \mathbf{U}^h \\ \mathbf{P}^h \end{bmatrix} = \begin{bmatrix} \mathbf{F}^h \\ 0 \end{bmatrix}, \quad (24)$$

where the matrixes  $\mathbf{A}_1$  and  $\mathbf{B}_1$  are presented and  $\mathbf{D}_1$  is deduced from  $(p_h, q)$ , using the base for  $M_h$  in the usual manner; that is,

$$\mathbf{D}_1 = (D_{1i,j}), \quad D_{1i,j} = \int_{\Omega} \varphi_i^h \varphi_j^h dx dy, \quad i, j = 1, 2, \dots, m. \quad (25)$$

Let  $e_i^{\mathbf{P}^h} = \mathbf{P}^h - \mathbf{P}_i^h$  ( $i = 1, 2, \dots$ ); we use the above Uzawa iterative algorithm and get

$$e_{i+1}^{\mathbf{P}^h} = \left( \mathbf{I} - \tau \left( \mathbf{B}_1 (\mathbf{A}_1 + \alpha \mathbf{A}_2)^{-1} \mathbf{B}_1^T - \frac{\varepsilon}{\nu} \mathbf{D}_1 \right) \right) e_i^{\mathbf{P}^h}. \quad (26)$$

*Algorithm II* (Regular method). This method uses a simple way to stabilize the mixed finite element approximation without a loss of accuracy, that is, to seek  $(\mathbf{u}_h, p_h) \in \mathbf{X}_h \times M_h$  such that

$$B((\mathbf{u}_h, p_h); (\mathbf{v}, q)) + \alpha (|\mathbf{u}_h|^{r-2} \mathbf{u}_h, \mathbf{v}) - \delta \sum_{K \in \mathcal{K}_h} (\nabla p_h - \mathbf{f}, \nabla q)_K = (\mathbf{f}, \mathbf{v}), \quad (27)$$

for all  $(\mathbf{v}, q) \in \mathbf{X}_h \times M_h$ , where  $\delta = h^2/(\beta\nu)$  is a stabilization parameter and  $\beta > 0$ . The matrix form of the above stabilized version can be expressed as

$$\begin{bmatrix} \mathbf{A}_1 + \alpha \mathbf{A}_2 & \mathbf{B}_1^T \\ \mathbf{B}_1 & \delta \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{U}^h \\ \mathbf{P}^h \end{bmatrix} = \begin{bmatrix} \mathbf{F}^h \\ \mathbf{C}_1^h \end{bmatrix}, \quad (28)$$

where additional blocks  $\mathbf{D}_2$  and  $\mathbf{C}_1^h$  correspond to

$$- \sum_{K \in \mathcal{K}_h} (\nabla p_h, \nabla q)_K,$$

$$- \delta \sum_{K \in \mathcal{K}_h} (\mathbf{f}, \nabla q)_K, \quad (29)$$

respectively; that is,

$$\mathbf{D}_2 = (D_{2i,j}), \quad D_{2i,j} = - \sum_{K \in \mathcal{K}_h} \int_K \nabla \varphi_i^h \nabla \varphi_j^h dx dy, \quad (30)$$

$$i, j = 1, \dots, m,$$

and  $\mathbf{C}_1^h = [-\delta \{ \sum_K \int_K (\partial \varphi_i^h / \partial x) f_1 \}, -\delta \{ \sum_K \int_K (\partial \varphi_i^h / \partial y) f_2 \}]^T$ , where

$$\left\{ \sum_K \int_K \frac{\partial \varphi_i^h}{\partial x} f_1 \right\} = \sum \left[ \int_K \frac{\partial \varphi_1^h}{\partial x} f_1, \dots, \int_K \frac{\partial \varphi_m^h}{\partial x} f_1 \right]^T,$$

$$\left\{ \sum_K \int_K \frac{\partial \varphi_i^h}{\partial y} f_2 \right\} = \sum \left[ \int_K \frac{\partial \varphi_1^h}{\partial y} f_2, \dots, \int_K \frac{\partial \varphi_m^h}{\partial y} f_2 \right]^T. \quad (31)$$

Similar to (26), we also have

$$e_{i+1}^{\mathbf{P}^h} = \left( \mathbf{I} - \tau \left( \mathbf{B}_1 (\mathbf{A}_1 + \alpha \mathbf{A}_2)^{-1} \mathbf{B}_1^T - \delta \mathbf{D}_2 \right) \right) e_i^{\mathbf{P}^h}. \quad (32)$$

*Algorithm III* (Multiscale enrichment method). Another stabilized way is the multiscale enrichment approach which includes the usual Galerkin least squares stabilized terms on each finite element and positive jump terms at interelement boundaries. Namely, seek  $(\mathbf{u}_h, p_h) \in \mathbf{X}_h \times M_h$  such that

$$B((\mathbf{u}_h, p_h); (\mathbf{v}, q)) + \alpha (|\mathbf{u}_h|^{r-2} \mathbf{u}_h, \mathbf{v}) - \delta_1 \sum_{K \in \mathcal{K}_h} (\nabla p_h - \mathbf{f}, \nabla q)_K + \delta_2 \sum_{\Gamma_{jk}} ([\nu \partial_n \mathbf{u}_h], [\nu \partial_n \mathbf{v}])_{\Gamma_{jk}} = (\mathbf{f}, \mathbf{v}),$$

$$\forall (\mathbf{v}, q) \in \mathbf{X}_h \times M_h, \quad (33)$$

where  $\delta_1 = h^2/(\beta_1\nu)$  and  $\delta_2 = h/(\beta_2\nu)$  are the positive stabilization parameters,  $n$  is the normal outward vector,  $\partial_n$  is normal derivative operator, and  $[\mathbf{v}]$  denotes the jump of  $\mathbf{v}$  across  $e$ . Moreover, a direct algebraic manipulation leads to the matrix form

$$\begin{bmatrix} \mathbf{A}_1 + \alpha \mathbf{A}_2 & \mathbf{B}_1^T \\ \mathbf{B}_1 + \delta_2 \mathbf{D}_3 & \delta_1 \mathbf{D}_2 \end{bmatrix} \begin{bmatrix} \mathbf{U}^h \\ \mathbf{P}^h \end{bmatrix} = \begin{bmatrix} \mathbf{F}^h \\ \mathbf{C}_2^h \end{bmatrix}, \quad (34)$$

where the matrix  $\mathbf{D}_3$  is deduced from the term  $\delta_2 \sum_{\Gamma_{jk}} ([\nu \partial_n \mathbf{u}_h], [\nu \partial_n \mathbf{v}])_{\Gamma_{jk}}$ .

Similar to (26), we have

$$e_{i+1}^{\mathbf{P}^h} = \left( \mathbf{I} - \tau \left( \mathbf{B}_1 + \delta_2 \mathbf{D}_3 \right) (\mathbf{A}_1 + \alpha \mathbf{A}_2)^{-1} \mathbf{B}_1^T - \delta_1 \mathbf{D}_2 \right) e_i^{\mathbf{P}^h}. \quad (35)$$

*Algorithm IV* (Local Gauss integration method). The local Gauss integration method is to add two Gauss integrals rather than any stabilization parameter to the original discrete formulation (16) as follows. Seek  $(\mathbf{u}_h, p_h) \in \mathbf{X}_h \times M_h$  such that

$$B((\mathbf{u}_h, p_h); (\mathbf{v}, q)) + \alpha (|\mathbf{u}_h|^{r-2} \mathbf{u}_h, \mathbf{v}) - G(p_h, q) = (\mathbf{f}, \mathbf{v}), \quad (\mathbf{v}, q) \in \mathbf{X}_h \times M_h, \quad (36)$$

where  $G(p_h, q)$  is defined by

$$G(p_h, q) = \delta_3 \sum_{K \in \mathcal{K}_h} \left\{ \int_{K,2} p_h q dx - \int_{K,1} p_h q dx \right\}, \quad \forall p_h, q \in M_h, \quad (37)$$

TABLE 1: Numerical results for the penalty method with  $\nu = 1.0e - 4$ ,  $\alpha = 1.0e - 4$ , and  $r = 3$ .

$1/h$	CPU-time	$\ u - u_h\ _1 / \ u\ _1$	$\ P - P_h\ _0 / \ P\ _0$	$u_{H^1}$ -rate	$p_{L^2}$ -rate
12	0.375	$2.3745E - 1$	$1.3649E - 1$	—	—
24	3.265	$1.1800E - 1$	$6.8136E - 2$	1.0088	1.0023
36	10.39	$7.8528E - 2$	$4.5442E - 2$	1.0045	0.9990
48	27.562	$5.8844E - 2$	$3.4113E - 2$	1.0031	0.9968
60	61.25	$4.7051E - 2$	$2.7326E - 2$	1.0023	0.9942

TABLE 2: Results got from the regular method with  $\nu = 1.0e - 4$ ,  $\alpha = 1.0e - 4$ , and  $r = 3$ .

$1/h$	CPU-time	$\ u - u_h\ _1 / \ u\ _1$	$\ P - P_h\ _0 / \ P\ _0$	$u_{H^1}$ -rate	$p_{L^2}$ -rate
12	0.282	$1.1945E - 0$	$1.5179E - 2$	—	—
24	1.641	$4.4196E - 1$	$4.7747E - 3$	1.4344	1.6686
36	4.657	$2.4423E - 1$	$2.5361E - 3$	1.4628	1.5605
48	10.313	$1.6070E - 1$	$1.6313E - 3$	1.4550	1.5337
60	21.094	$1.1646E - 1$	$1.1619E - 3$	1.4431	1.5207

$\delta_3 > 0$ , and  $\int_{K,i} g(x)dx$  indicates a local Gauss integral over  $K$  that is exact for polynomials of degree  $i$  ( $i = 1, 2$ ). Then the corresponding matrix form of this stabilized method is

$$\begin{bmatrix} \mathbf{A}_1 + \alpha \mathbf{A}_2 & \mathbf{B}_1^T \\ \mathbf{B}_1 & \delta_3 \mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{U}^h \\ \mathbf{P}^h \end{bmatrix} = \begin{bmatrix} \mathbf{F}^h \\ 0 \end{bmatrix}, \quad (38)$$

where

$$\begin{aligned} \mathbf{G} &= - \sum_{K \in \mathcal{K}_h} (G_2 - G_1), \\ G_2 &= \left( \int_{K,2} \varphi_i^h \varphi_j^h \right), \\ G_1 &= \left( \int_{K,1} \varphi_i^h \varphi_j^h \right), \quad i, j = 1, \dots, m. \end{aligned} \quad (39)$$

As (26), we get

$$e_{i+1}^{\mathbf{P}^h} = (\mathbf{I} - \tau (\mathbf{B}_1 (\mathbf{A}_1 + \alpha \mathbf{A}_2)^{-1} \mathbf{B}_1^T - \delta_3 \mathbf{G})) e_i^{\mathbf{P}^h}. \quad (40)$$

It is well known that, if  $\tau$  is well chosen, then  $U_i$  and  $P_i$  converge, respectively, to  $U$  and  $P$  with a rate of convergence based on (26), (32), (35), and (40). From these equations, we can find that the penalty method converges faster than the local Gauss integration method. We can obtain that coefficient matrices of penalty algorithm, regular algorithm, and local Gauss integration algorithm are all symmetric; however, the multiscale enrichment method's coefficient matrices are not symmetric from (10), (28), (34), and (38). What is more, it is easy to see that the matrix calculations of multiscale enrichment algorithm are more complex and cumbersome, so this method maybe costs more time.

By using the regularity assumptions and well-established techniques for velocity and pressure [4, 7], the theoretical convergence rates should be of order  $O(h)$  for the velocity in the  $L^2$ -norm and of order  $O(h^2)$  for the pressure in the  $H^1$ -norm, respectively, by using all these stabilized methods.

## 4. Numerical Experiments

In this section, we will give three numerical tests to confirm the numerical theory developed in the previous section. In the given experiments, the pressure and velocity are approximated by the lowest equal-order finite element pairs defined with respect to the same uniform triangulation; that is, the mesh consists of triangular elements that are obtained by dividing  $\Omega$  into subsquares of equal size and then drawing the diagonal in each subsquare.

**4.1. Numerical Test 1.** In this example, we consider the exact solution problem firstly. Let the domain  $\Omega$  be the unit square  $\Omega = (0, 1) \times (0, 1) \subset \mathbb{R}^2$ . The exact solution for the velocity  $\mathbf{u} = (u_1, u_2)$  and pressure  $p$  is given as follows:

$$p(x, y) = \cos(\pi x) \cos(\pi y),$$

$$u_1(x, y) = 2\pi \sin^2(\pi x) \sin(\pi y) \cos(\pi y), \quad (41)$$

$$u_2(x, y) = -2\pi \sin(\pi x) \cos(\pi x) \sin^2(\pi y),$$

and the right-hand side  $\mathbf{f} = (f_1(x, y), f_2(x, y))$  is determined by the original problem (1). Our goal in this test is to compare CPU-time, the  $L^2$ -error of the pressure, and  $H^1$ -error of the velocity; the experimental rates of convergence for these methods with different values of  $h$  are tabulated in Tables 1, 2, 3, 4, and 5. What is more, the rates of convergence are calculated by the formula  $\log(E_i/E_{i+1})/\log(h_i/h_{i+1})$ , where  $E_i$  and  $E_{i+1}$  are the relative errors corresponding to the meshes of sizes  $h_i$  and  $h_{i+1}$ , respectively.

From this experiment, we can learn the following several points. (1) For penalty method, the result of this method is well in which parameter "r" can be 2, 3, and 4 and we can use UMFPAK or default solver in the process of calculation. (2) Results got from the penalty, regular, multiscale enrichment, and local Gauss integration methods by using conforming and nonconforming elements are presented in Tables 1–5, respectively. Here, we choose  $\varepsilon = 1.0e - 6$ ,  $\alpha = 1.0e - 4$ ,  $r = 3$ ,  $\text{Re} = 10000$ , and  $\beta = 160$ ,  $\beta_1 = 160$ ,  $\beta_2 = 100$ ,

TABLE 3: Results got from the multiscale enrichment method with  $\nu = 1.0e - 4$ ,  $\alpha = 1.0e - 4$ , and  $r = 3$ .

$1/h$	CPU-time	$\ u - u_h\ _1 / \ u\ _1$	$\ p - p_h\ _0 / \ p\ _0$	$u_{H^1}$ -rate	$p_{L^2}$ -rate
12	2.496	0.8924	1.1544	—	—
24	12.746	0.6523	0.2823	0.4520	2.0319
36	37.175	0.4853	0.2094	0.7293	0.7370
48	83.648	0.3737	0.1713	0.9091	0.6977
60	161.32	0.2996	0.1445	0.9900	0.7624

TABLE 4: Results got from the local Gauss integration method with  $\nu = 1.0e - 4$ ,  $\alpha = 1.0e - 4$ , and  $r = 3$ .

$1/h$	CPU-time	$\ u - u_h\ _1 / \ u\ _1$	$\ p - p_h\ _0 / \ p\ _0$	$u_{H^1}$ -rate	$p_{L^2}$ -rate
12	0.406	2.2348E - 1	8.9420E - 3	—	—
24	4.578	1.1458E - 1	2.8981E - 3	1.0353	1.6255
36	8.5	7.4890E - 2	1.8494E - 3	1.0488	1.1078
48	16.328	5.5377E - 2	1.3620E - 3	1.0493	1.0635
60	27.39	4.3894E - 2	1.0468E - 3	1.0414	1.1797

TABLE 5: Results got from the local Gauss integration method with the nonconforming element with  $\nu = 1.0e - 4$ ,  $\alpha = 1.0e - 4$ , and  $r = 3$ .

$1/h$	CPU-time	$\ u - u_h\ _1 / \ u\ _1$	$\ p - p_h\ _0 / \ p\ _0$	$u_{H^1}$ -rate	$p_{L^2}$ -rate
12	0.297	3.6007	5.8335E - 2	—	—
24	2.500	1.8851	1.6524E - 2	0.9336	1.8198
36	5.735	1.2682	7.5810E - 3	0.9776	1.9217
48	11.469	0.9543	4.3219E - 3	0.9887	1.9534
60	21.719	0.7646	2.7858E - 3	0.9932	1.9681

because they can deal with the considered problem well. (3) For regular method, it works well only we choose Crout solver; however, for other methods, several solvers can be used and their difference is not big. What is more, the value of “ $r$ ” has little influence on the results within a certain range; in this paper, we give the results of  $r = 3$  presented in tables. From Tables 1–5, we can see that these methods work well and keep the convergence rates just like the theoretical analysis except for the multiscale enrichment method. Meanwhile, it can be seen that the penalty method requires the least CPU-time, which validates the analysis in Section 3. As expected, we have an interesting observation that the error of the nonconforming local Gauss integration method is better than that of the conforming version, which is not surprising since the degree of freedom of the nonconforming method is nearly three times than that of the conforming one on uniform mesh. Hence, it is natural that the nonconforming local Gauss integration method is more accurate and costs more CPU-time. The penalty method should use less time than the regular method and local Gauss integration method theoretically but in fact not; maybe this is caused by the different solver.

Besides, from the convergence results on this example, we can see that regular and multiscale enrichment methods are not better than other methods. And the nonconforming local Gauss integration method shows the best numerical stability.

**4.2. Numerical Test 2.** In this test, we test a popular benchmark problem, the lid-driven flow. Let the computation be carried out in the region  $\Omega = \{(x, y) \mid 0 < x, y < 1\}$ . We assume the normal component of the velocity to be zero on  $\partial\Omega$  and the tangential component to be zero except along  $y = 1$ , where it is set to one.

In this example, we simulate the referred physics phenomena. In Figures 1, 2, 3, 4, and 5, we present the velocity streamlines and pressure level lines for  $\varepsilon = 10e - 6$ ,  $\alpha = 100$ ,  $h = 1/30$ , and  $r = 4$  based on these five methods. From Figures 1(b)–5(b), only Gauss methods can obtain resolved pressure. For the velocity, from Figures 1(a)–5(a), we can see that Gauss methods can capture this model better than the other methods.

**4.3. Numerical Test 3.** In this example, we choose the exact solution for the velocity and pressure in the unit square as follows:

$$\begin{aligned}
 p(x, y) &= 10(2x - 1)(2y - 1), \\
 u_1(x, y) &= 10x^2(x - 1)^2y(y - 1)(2y - 1), \\
 u_2(x, y) &= -10x(x - 1)(2x - 1)y^2(y - 1)^2,
 \end{aligned} \tag{42}$$

and the right-hand side is determined just as Test 1. In this example, parameters we choose are  $\nu = 1.0e - 4$ ,  $\alpha = 1.0e - 4$ ,

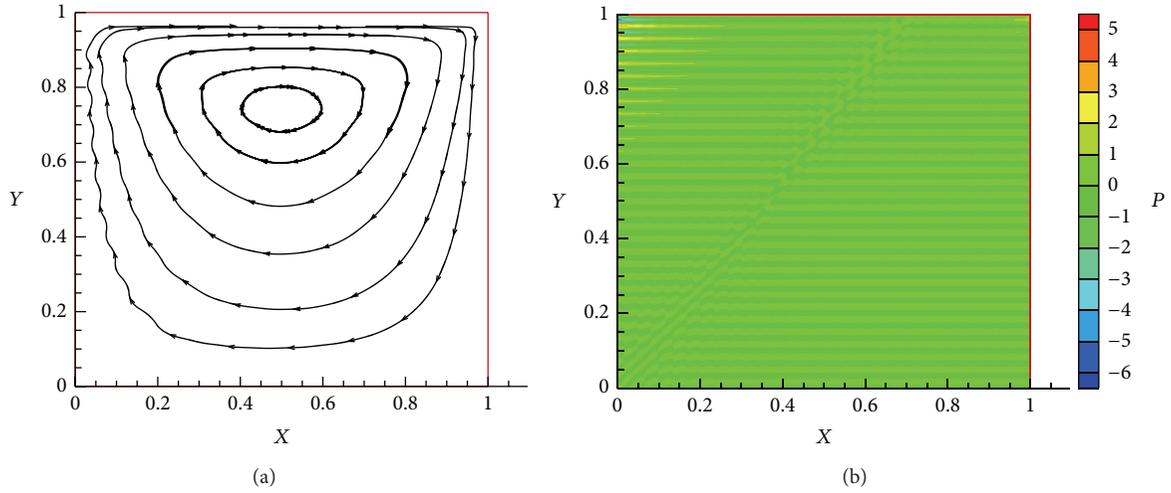


FIGURE 1: Velocity streamlines (a) and pressure level lines (b) for the penalty method with  $\varepsilon = 1.0e - 6$ ,  $\alpha = 100$ , and  $r = 4$ , respectively.

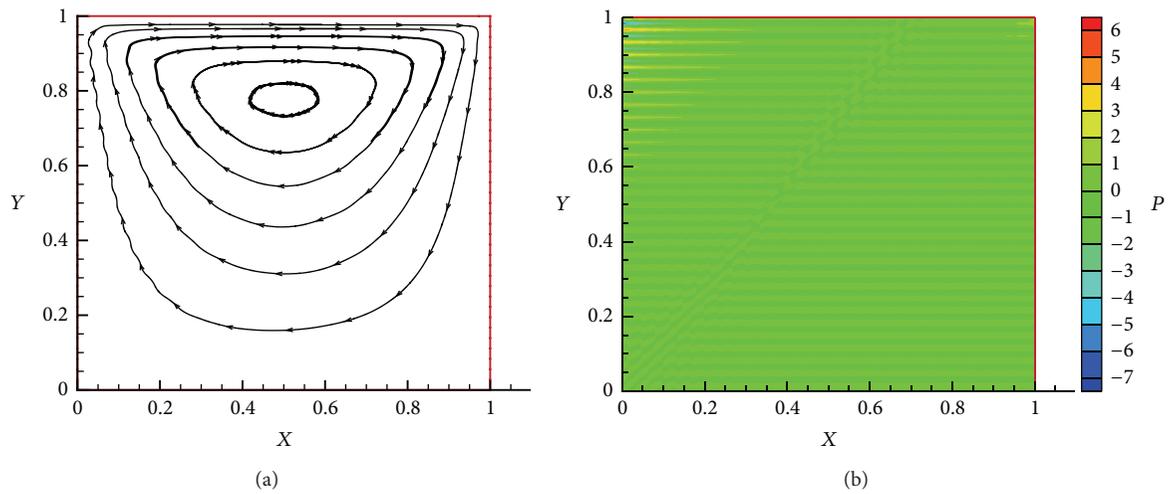


FIGURE 2: Velocity streamlines (a) and pressure level lines (b) for the regular method with  $\varepsilon = 1.0e - 6$ ,  $\alpha = 100$ , and  $r = 4$ , respectively.

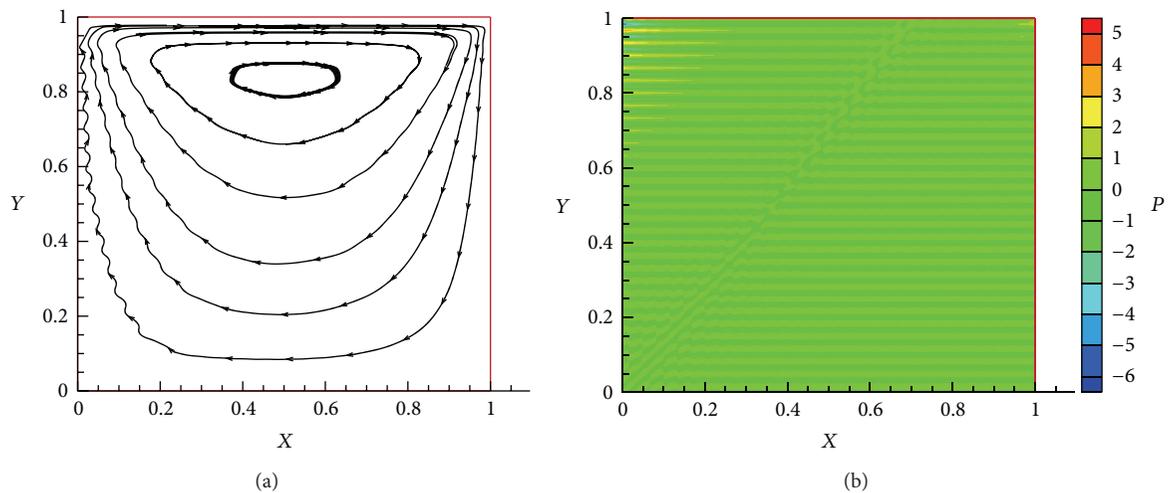


FIGURE 3: Velocity streamlines (a) and pressure level lines (b) for the multiscale enrichment method with  $\varepsilon = 1.0e - 6$ ,  $\alpha = 100$ , and  $r = 4$ , respectively.

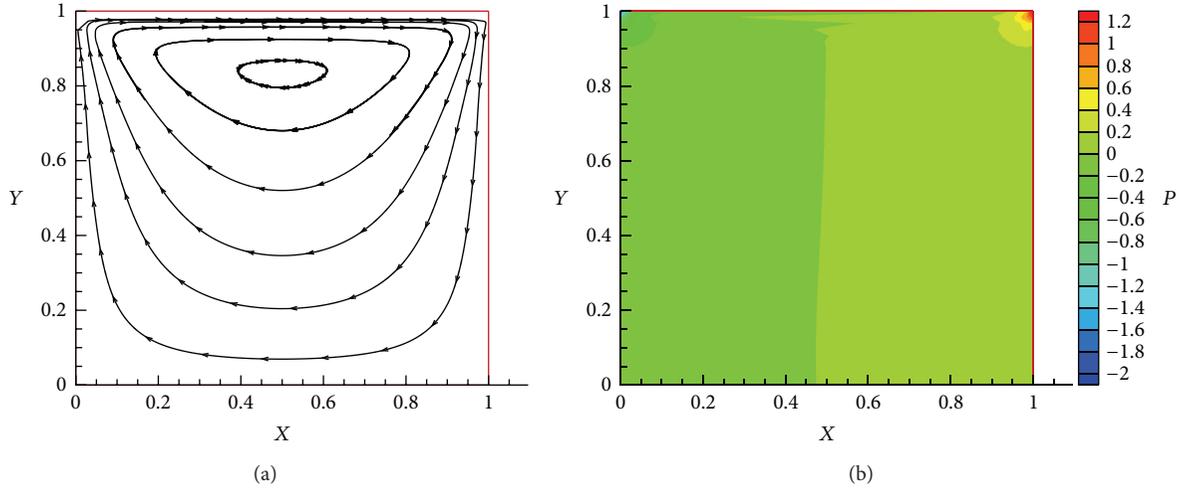


FIGURE 4: Velocity streamlines (a) and pressure level lines (b) for the local Gauss integration method with  $\varepsilon = 1.0e - 6$ ,  $\alpha = 100$ , and  $r = 4$ , respectively.

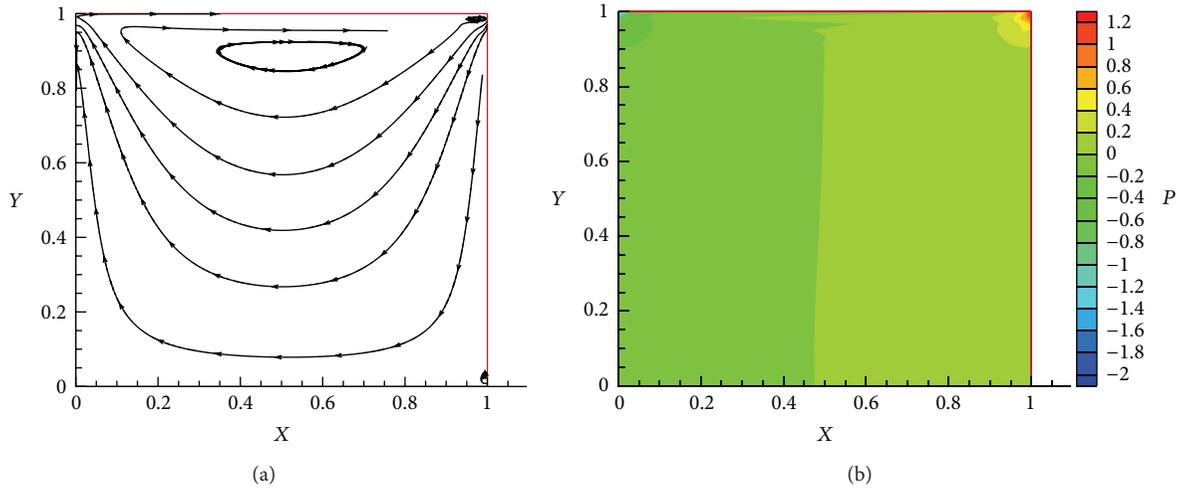


FIGURE 5: Velocity streamlines (a) and pressure level lines (b) for the local Gauss integration method with the nonconforming element with  $\varepsilon = 1.0e - 6$ ,  $\alpha = 100$ , and  $r = 4$ , respectively.

TABLE 6: Numerical results for the penalty method with  $\nu = 1.0e - 4$ ,  $\alpha = 1.0e - 4$ , and  $r = 3$ .

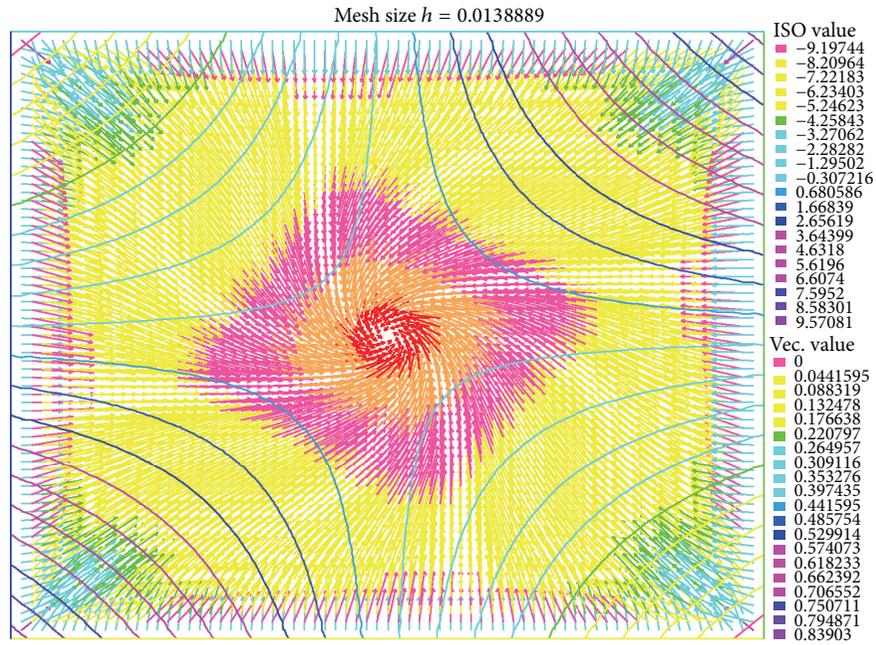
$1/h$	CPU-time	$\ u - u_h\ _1 / \ u\ _1$	$\ p - p_h\ _0 / \ p\ _0$	$u_{H^1}$ -rate	$p_{L^2}$ -rate
12	0.219	$2.8476E - 1$	$1.8917E - 1$	—	—
24	1.625	$1.4126E - 1$	$9.8071E - 2$	1.0114	0.9478
36	5.719	$9.3881E - 2$	$6.6229E - 2$	1.0077	0.9682
48	15.64	$7.0298E - 2$	$5.0001E - 2$	1.0056	0.9771
60	32.281	$5.6183E - 2$	$4.0161E - 2$	1.0044	0.9820

and  $r = 3$ . For the penalty and local Gauss integration methods, CPU-time, the  $L^2$ -error of the pressure,  $H^1$ -error of the velocity, and the experimental rates of convergence for these methods with different values of  $h$  are tabulated in Tables 6 and 7. What is more, the numerical solutions are given in Figure 6. From this test, we can get conclusion that is similar to Test 1. However, the result got from multiscale method is not better.

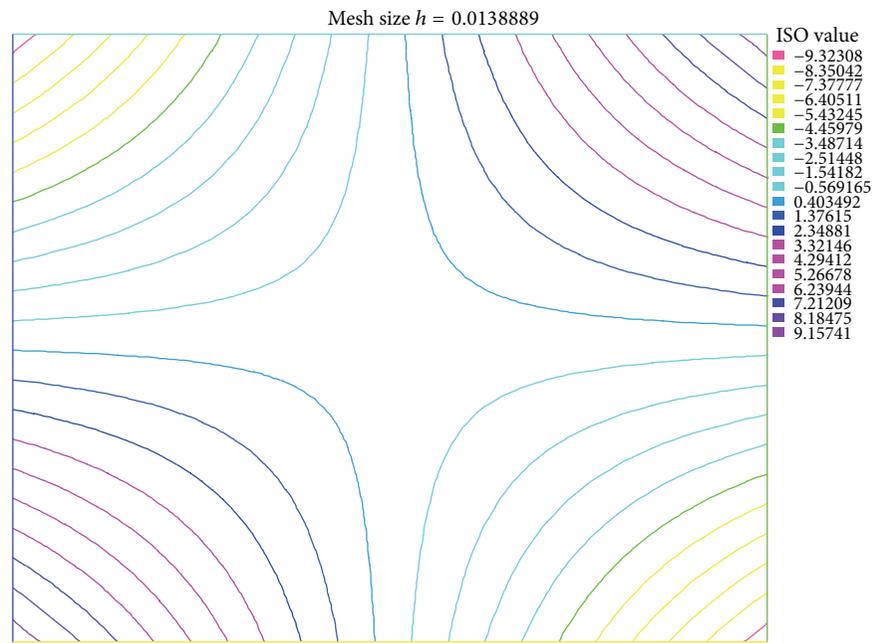
## 5. Conclusions

We have used several stabilized mixed finite element methods in solving the steady incompressible flow problem with damping based on the lowest equal-order pairs in this paper. We give some conclusions by comparing numerically as follows.

All of those methods' stability and efficiency depends on their parameter values. In terms of the penalty method,



(a)



(b)

FIGURE 6: Numerical solutions for the penalty method (a) and pressure level lines for the local Gauss integration method (b) with  $\varepsilon = 1.0e - 4$ ,  $\alpha = 1.0e - 4$ , and  $r = 4$ , respectively.

TABLE 7: Results got from the local Gauss integration method with  $\nu = 1.0e - 4$ ,  $\alpha = 1.0e - 4$ , and  $r = 3$ .

$1/h$	CPU-time	$\ u - u_h\ _1 / \ u\ _1$	$\ p - p_h\ _0 / \ p\ _0$	$u_{H^1}$ -rate	$p_{L^2}$ -rate
12	0.219	$1.78153E - 1$	$2.01165E - 2$	—	—
24	3.109	$7.64659E - 1$	$4.97672E - 3$	1.22023	2.01511
36	5.437	$4.78738E - 1$	$2.20800E - 3$	1.15491	2.00432
48	9.578	$3.40798E - 1$	$1.24253E - 3$	1.18138	1.99853
60	16.062	$2.58872E - 1$	$7.96681E - 4$	1.23219	1.99177

the smaller its parameter value, the more stable the method. However, it can not be too small, otherwise, the condition number of the system matrix arising from this method will become too large to be solve. For the regular and multiscale enrichment methods whose performance heavily depends on the choice of the stabilization parameters, however, it is difficult to choose fine parameters in fact. What is more, a poor choice of these stabilization parameters can also lead to serious deterioration in the convergence rates. The local Gauss integration method is free of stabilization parameters and shows numerically the best performance among the methods considered for the given problem.

## Acknowledgments

This work is in part supported by the NSF of China (nos. 11271313 and 61163027), the China Postdoctoral Science Foundation (nos. 2012M512056 and 2013M530438), the Key Project of Chinese Ministry of Education (no. 212197), the NSF of Xinjiang Province (no. 2013211B01), and the Doctoral Foundation of Xinjiang University (no. BSI20102).

## References

- [1] X. Cai and L. Lei, " $L^2$  decay of the incompressible Navier-Stokes equations with damping," *Acta Mathematica Scientia B*, vol. 30, no. 4, pp. 1235–1248, 2010.
- [2] X. Cai and Q. Jiu, "Weak and strong solutions for the incompressible Navier-Stokes equations with damping," *Journal of Mathematical Analysis and Applications*, vol. 343, no. 2, pp. 799–809, 2008.
- [3] P. Constantin and F. Ramos, "Inviscid limit for damped and driven incompressible Navier-Stokes equations in  $\mathbb{R}^2$ ," *Communications in Mathematical Physics*, vol. 275, no. 2, pp. 529–551, 2007.
- [4] P. Huang, X. Feng, and D. Liu, "A stabilised nonconforming finite element method for steady incompressible flows," *International Journal of Computational Fluid Dynamics*, vol. 26, no. 2, pp. 133–144, 2012.
- [5] A. A. Ilyin and E. S. Titi, "The damped-driven 2D Navier-Stokes system on large elongated domains," *Journal of Mathematical Fluid Mechanics*, vol. 10, no. 2, pp. 159–175, 2008.
- [6] D. M. Liu and K. T. Li, "Finite element analysis of the Stokes equations with damping," *Mathematica Numerica Sinica*, vol. 32, no. 4, pp. 433–448, 2010 (Chinese).
- [7] P. Huang, Y. He, and X. Feng, "Numerical investigations on several stabilized finite element methods for the Stokes eigenvalue problem," *Mathematical Problems in Engineering*, vol. 2011, Article ID 745908, 14 pages, 2011.
- [8] Q. Liu and Y. Hou, "A postprocessing mixed finite element method for the Navier-Stokes equations," *International Journal of Computational Fluid Dynamics*, vol. 23, no. 6, pp. 461–475, 2009.
- [9] M. Picasso and J. Rappaz, "Stability of time-splitting schemes for the Stokes problem with stabilized finite elements," *Numerical Methods for Partial Differential Equations*, vol. 17, no. 6, pp. 632–656, 2001.
- [10] D. Shi and J. Ren, "Nonconforming mixed finite element approximation to the stationary Navier-Stokes equations on anisotropic meshes," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 71, no. 9, pp. 3842–3852, 2009.
- [11] Z. Si, T. Zhang, and K. Wang, "A Newton iterative mixed finite element method for stationary conduction-convection problems," *International Journal of Computational Fluid Dynamics*, vol. 24, no. 3–4, pp. 135–141, 2010.
- [12] X. Ye, "A discontinuous finite volume method for the Stokes problems," *SIAM Journal on Numerical Analysis*, vol. 44, no. 1, pp. 183–198, 2006.
- [13] V. Girault and P.-A. Raviart, *Finite Element Methods for Navier-Stokes Equations: Theory and Algorithms*, Springer, Berlin, Germany, 1986.
- [14] P. B. Bochev, C. R. Dohrmann, and M. D. Gunzburger, "Stabilization of low-order mixed finite elements for the Stokes equations," *SIAM Journal on Numerical Analysis*, vol. 44, no. 1, pp. 82–101, 2006.
- [15] J. Douglas, Jr. and J. P. Wang, "An absolutely stabilized finite element method for the Stokes problem," *Mathematics of Computation*, vol. 52, no. 186, pp. 495–508, 1989.
- [16] Y. He, "Optimal error estimate of the penalty finite element method for the time-dependent Navier-Stokes equations," *Mathematics of Computation*, vol. 74, no. 251, pp. 1201–1216, 2005.
- [17] J. Li, L. Mei, and Y. He, "A pressure-Poisson stabilized finite element method for the non-stationary Stokes equations to circumvent the inf-sup condition," *Applied Mathematics and Computation*, vol. 182, no. 1, pp. 24–35, 2006.
- [18] R. Araya, G. R. Barrenechea, and F. Valentin, "Stabilized finite element methods based on multiscaled enrichment for the Stokes problem," *SIAM Journal on Numerical Analysis*, vol. 44, no. 1, pp. 322–348, 2006.
- [19] J. Li and Y. He, "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, 2008.
- [20] X. Feng, I. Kim, H. Nam, and D. Sheen, "Locally stabilized  $P_1$ -nonconforming quadrilateral and hexahedral finite element methods for the Stokes equations," *Journal of Computational and Applied Mathematics*, vol. 236, no. 5, pp. 714–727, 2011.
- [21] J. Li, Y. He, and Z. Chen, "Performance of several stabilized finite element methods for the Stokes equations based on the lowest equal-order pairs," *Computing*, vol. 86, no. 1, pp. 37–51, 2009.
- [22] X. Feng and Y. He, "Modified homotopy perturbation method for solving the Stokes equations," *Computers & Mathematics with Applications*, vol. 61, no. 8, pp. 2262–2266, 2011.
- [23] X. Feng, Y. He, and J. Meng, "Application of modified homotopy perturbation method for solving the augmented systems," *Journal of Computational and Applied Mathematics*, vol. 231, no. 1, pp. 288–301, 2009.
- [24] X. Feng and L. Shao, "On the generalized SOR-like methods for saddle point problems," *Journal of Applied Mathematics & Informatics*, vol. 28, no. 3–4, pp. 663–677, 2010.