Electronic Research Archive

https://www.aimspress.com/journal/era

ERA, 32(10): 5633–5648.
DOI: 10.3934/era.2024261
Received: 24 July 2024
Revised: 16 September 2024
Accepted: 07 October 2024
Published: 16 October 2024

Research article

A pressure-robust divergence free finite element basis for the Stokes equations

Jay Chu[1], **Xiaozhe Hu**[2] **and Lin Mu**[3,*]

[1] Departmant of Mathematics, National Tsing Hua University, Taiwan
[2] Department of Mathematics, Tufts University, Medford, MA 02155, USA
[3] Department of Mathematics, University of Georgia, Athens, GA 30602, USA

* **Correspondence:** Email: linmu@uga.edu.

Abstract: This paper considered divergence-free basis methods to solve the viscous Stokes equations. A discrete divergence-free subspace was constructed to reduce the saddle point problem of the Stokes problem to a smaller-sized symmetric and positive definite system solely depending on the velocity components. Then, the system could decouple the unknowns in velocity and pressure and solve them independently. However, such a scheme may not ensure an accurate numerical solution to the velocity. In order to obtain satisfactory accuracy, we used a velocity reconstruction technique to enhance the divergence-free scheme to achieve the desired pressure and viscosity robustness. Numerical results were presented to demonstrate the robustness and accuracy of this discrete divergence-free method.

Keywords: finite element methods; the viscous Stokes equations; divergence-free basis; pressure robustness

1. Introduction

The viscous Stokes problem seeks unknown functions $\mathbf{u}$ and p that fulfill the following equations,

$$-\nu\Delta\mathbf{u} + \nabla p = \mathbf{f} \quad \text{in } \Omega, \tag{1.1}$$
$$\nabla \cdot \mathbf{u} = 0 \quad \text{in } \Omega, \tag{1.2}$$
$$\mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega, \tag{1.3}$$

where Ω is a polygonal domain in $\mathbb{R}^2$. For the nonhomogeneous boundary condition $\mathbf{u} = \mathbf{g}$ on $\partial\Omega$, one can use the standard procedure by letting $\mathbf{u} = \mathbf{u}_0 + \mathbf{u}_g$. $\mathbf{u}_g$ is a known function satisfying $\mathbf{u}_g = \mathbf{g}$ on $\partial\Omega$ and $\mathbf{u}_0$ is zero at $\partial\Omega$ and satisfies (1.1) and (1.2) with different righthand sides. For the sake of simplicity, we only consider the homogeneous boundary condition, i.e., $\mathbf{g} = \mathbf{0}$. The scheme can

be extended to the nonhomogeneous boundary condition. Using the standard notation for the Sobolev spaces, the weak formulation for the Stokes problems (1.1)–(1.3), in the primary velocity-pressure form, we seek $\mathbf{u} \in [H_0^1(\Omega)]^2$ and $p \in L_0^2(\Omega)$ such that,

$$\begin{cases} (\nu \nabla \mathbf{u}, \nabla \mathbf{v}) - (\nabla \cdot \mathbf{v}, p) &= (\mathbf{f}, \mathbf{v}), \quad \forall\, \mathbf{v} \in [H_0^1(\Omega)]^2, \\ (\nabla \cdot \mathbf{u}, q) &= 0, \qquad \forall\, q \in L_0^2(\Omega). \end{cases}$$

In the standard finite element discretization schemes, pressure and velocity unknowns are approximated simultaneously via a saddle-point system. To avoid solving such an indefinite system, the divergence-free finite element methods have been proposed to compute the numerical velocity by solving a symmetric positive-definite system in a divergence-free subspace. Due to the discrete or exact divergence-free property, such a method eliminates the pressure from the coupled systems, resulting in a symmetric positive definite system with a smaller size. Previously, a divergence-free basis was constructed for different finite element methods, e.g., [1–4]. The original divergence-free weak Galerkin (WG) method was proposed in [3].

Unlike most existing divergence-free finite element methods, the discrete divergence-free WG method considered in this paper allows the meshes to consist of a mix of general polygons and hanging nodes. However, although the basis functions are discrete divergence-free, they may not guarantee good velocity approximation since the velocity error may depend on viscosity and pressure. This is because the div-free scheme is non-pressure-robust; thus, the velocity error bound depends on viscosity and pressure. Small viscosity values or inaccurate pressure approximations may produce an incorrect velocity solution to ruin the simulation.

This paper shows that the numerical pollution mentioned above, caused by small viscosities or large pressure errors, also appears for the previous discrete divergence-free WG method. In this paper, we contribute to modify the original scheme and investigate the technique to remove viscosity and pressure effects in the velocity approximations with minimal effort. The technique follows the previous work of the authors and employs the velocity-reconstruction operator to modify the load term. This reconstruction technique was first proposed by Linke [5, 6] and was then widely used to modify the existing finite element scheme for Stokes problems [7–16] and other incompressible fluid problems due to the minimal efforts required to achieve the desired good quality in numerical solutions [17–22]. Unlike using the H(div) basis functions in H(div) finite element methods, the velocity reconstruction operator is designed to map the original velocity basis functions to a suitable subspace of the H(div) space. Then, this modification only changes the load term assembly, but the stiffness matrix remains the same. In addition to the velocity reconstruction operator, we also mention that there are other advanced approaches to achieve the desired pressure-robustness [17,23,24]. Due to the page limitation, we only cite an incomplete list of the previous schemes featuring pressure-robustness. For example, Zhang [25, 26] constructed divergence-free pairs of finite element spaces and used it to solve incompressible fluid problems [27, 28]. Another successful strategy is to employ the Stokes complex of the lowest regularity [29] and the approximate velocity in the H(div) space [30]. A similar approach has been used in hybrid discontinuous Galerkin methods (HDG) to achieve the desired pressure-robustness [31–33]. More details on divergence-free and pressure-robust schemes can be found in the review paper [34]. Recently, there is another approach to achieve the desired robustness by enriching the Raviart-Thomas (RT) basis functions into the H^1 -finite element spaces [35, 36]. In this paper, we focus on designing the proper velocity-reconstruction operator and modifying the source term assembly to achieve robust-

ness. The advantages of this modification lie in the potential to recycle the researchers' previous codes and enhance the former work with minimal changes to the reliable numerical approximation. We also demonstrate the pressure recovery procedure for the case that requires a pressure approximation.

The rest of this paper is organized as follows. In Section 2, we first introduce the notation and two existing numerical algorithms and then propose the robust pressure algorithm and the pressure recovery scheme. In Section 3, we demonstrate the main error estimates for the Stokes problem. Several numerical experiments are presented in Section 4. We conclude this paper in Section 5.

2. Numerical scheme

This section recalls the standard WG method and proposes our new divergence-free and pressure-robust WG methods. Let $\mathcal{T}_h$ be a partition of the domain Ω consisting of a mix of polygons satisfying the set of conditions specified in [37]. Let $\mathcal{E}_h$ denote the set of all edges in $\mathcal{T}_h$ and $\mathcal{E}_h^0 = \mathcal{E}_h \backslash \partial\Omega$ be the set of all interior edges. Based on the partition $\mathcal{T}_h$, we introduce the following finite element spaces W_h and V_h for the pressure and velocity variables, respectively,

$$\begin{aligned} W_h &= \left\{ q : \ q \in L_0^2(\Omega), \ q|_T \in P_0(T) \right\}, \\ V_h &= \left\{ \mathbf{v} = \{\mathbf{v}_0, \mathbf{v}_b\} : \ \{\mathbf{v}_0, \mathbf{v}_b\}|_T \in [P_1(T)]^2 \times [P_0(e)]^2, \ e \subset \partial T, \ \mathbf{v}_b = 0 \text{ on } \partial\Omega \right\}, \end{aligned}$$

where $P_k(\omega)$ denotes the space of polynomials of degree at most k restricted to $\omega = e$ or T.

The discrete weak gradient and divergence operators are defined locally on each $T \in \mathcal{T}_h$ as follows.

Definition 2.1. *The discrete weak gradient $\nabla_w : V_h \mapsto [P_0(T)]^{2\times 2}$ and weak divergence operator $\nabla_w \cdot : V_h \mapsto P_0(T)$ are defined as follows,*

$$\begin{aligned} (\nabla_w \mathbf{v}, q)_T &= \langle \mathbf{v}_b, q \cdot \mathbf{n} \rangle_{\partial T}, \qquad \forall q \in [P_0(T)]^{2\times 2}, \\ (\nabla_w \cdot \mathbf{v}, \varphi)_T &= \langle \mathbf{v}_b \cdot \mathbf{n}, \varphi \rangle_{\partial T}, \qquad \forall \varphi \in P_0(T). \end{aligned}$$

For each edge $e \in \mathcal{E}_h$, let Q_b be the L^2 projection from $[L^2(e)]^2$ onto $[P_0(e)]^2$. Then, we define

$$\begin{aligned} a(\mathbf{v}, \mathbf{w}) &:= \sum_{T\in\mathcal{T}_h} (\nu \nabla_w \mathbf{v}, \ \nabla_w \mathbf{w})_T + \sum_{T\in\mathcal{T}_h} \frac{\nu}{h_T} \langle Q_b \mathbf{v}_0 - \mathbf{v}_b, \ \ Q_b \mathbf{w}_0 - \mathbf{w}_b \rangle_{\partial T}, \\ b(\mathbf{v}, q) &:= \sum_{T\in\mathcal{T}_h} (\nabla_w \cdot \mathbf{v}, \ q)_T. \end{aligned}$$

Here, h_T denotes the mesh size for the element T. Then, a standard WG algorithm (see [38]) is as follows.

Algorithm 2.1. ***Standard weak Galerkin algorithm (SWG)*** *A numerical approximation for (1.1)–(1.3) is to seek $\mathbf{u}_h = \{\mathbf{u}_0, \mathbf{u}_b\} \in V_h$ and $p_h \in W_h$ such that*

$$\begin{aligned} a(\mathbf{u}_h, \mathbf{v}) - b(\mathbf{v}, p_h) &= (\mathbf{f}, \mathbf{v}_0), \quad \forall \ \mathbf{v} = \{\mathbf{v}_0, \mathbf{v}_b\} \in V_h, \\ b(\mathbf{u}_h, q) &= 0, \qquad \forall \ q \in W_h. \end{aligned}$$

Algorithm 2.1 produces a saddle system, which can be challenging due to its indefiniteness, strong coupling between velocity and pressure, and large size. In some cases, linear solvers for this large system involving both velocity and pressure may not be effective. Instead, we use the divergence-free basis to decouple the velocity and pressure and solve a smaller system, which is symmetric positive definite.

2.1. Divergence-free finite element basis

In this section, we introduce the divergence-free basis. First, we define the discrete divergence-free subspace D_h of V_h in the usual way (see [1–3]) as follows,

$$D_h = \{\mathbf{v} \in V_h;\ b(\mathbf{v}, q) = 0, \quad \forall q \in W_h\}. \tag{2.1}$$

Following the techniques introduced in [3] and using the definition (2.1), we explicitly construct the basis functions as the following three types.

$$D_h = \text{span}\{\underbrace{\Phi_1, \cdots, \Phi_{6N_K}}_{\Psi_0}, \underbrace{\Upsilon_1, \cdots, \Upsilon_{N_E}}_{\Psi_t}, \underbrace{\Lambda_1, \cdots, \Lambda_{N_V}}_{\Psi_V}\}. \tag{2.2}$$

1) Type 1 (Ψ_0): For each $T_i \in \mathcal{T}_h$, $i = 1, \cdots, N_K$ with N_K being the number of elements, all the six linearly independent linear functions $\Phi_{6(i-1)+1}, \Phi_{6(i-1)+2}, \cdots, \Phi_{6(i-1)+6}$ in V_h are discrete divergence-free since they are nonzero only in the interior of element T_i.
2) Type 2 (Ψ_t): For each $e_i \in \mathcal{E}_h^0$, $i = 1, \cdots, N_E$ with N_E being the number of interior edges, let $\mathbf{t}_{e_i}$ be its tangential vector and $\Psi_{e_i,1}$ and $\Psi_{e_i,2}$ be the two basis functions of V_h that are nonzero only on e_i. Define $\Upsilon_i := C_1\Psi_{e_i,1} + C_2\Psi_{e_i,2}$ such that $\Upsilon_i|_{e_i} = \mathbf{t}_{e_i}$. It is easy to verify that Υ_i is discrete divergence-free using the divergence theorem. Note that Υ_i is nonzero only on e_i.
3) Type 3 (Ψ_V): For each interior vertex $P_i \in \mathcal{V}_h$, $i = 1, \cdots, N_V$ with N_V being the number of interior vertices, there are r elements sharing P_i which form a hull $\mathcal{H}_{P_i}$ as shown in Figure 1. Consequently, there are r interior edges e_j ($j = 1, \cdots, r$) incident with P_i. Let $\mathbf{n}_{e_j}$ be a normal vector on e_j, and we assume that the normal vectors $\mathbf{n}_{e_j}$ $j = 1, \cdots, r$ are counterclockwise around the vertex P_i as shown in Figure 1. For each e_j, let $\Psi_{e_j,1}$ and $\Psi_{e_j,2}$ be the two basis functions of V_h, which are nonzero only on e_j. Define $\Theta_j = C_1\Psi_{e_j,1} + C_2\Psi_{e_j,2} \in V_h$ such that $\Theta_j|_{e_j} = \mathbf{n}_{e_j}$ and then define $\Lambda_i = \sum_{j=1}^{r} \frac{1}{|e_j|}\Theta_j$, which is discrete divergence-free by the divergence theorem. The construction can be applied to triangular and polygonal grids, as shown in Figure 1.

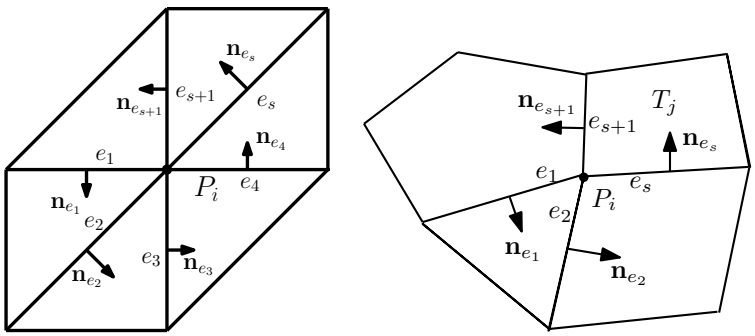

Figure 1. Hull $\mathcal{H}_{P_i}$ for triangular grids and polygonal grids.

The dimension of V_h is $6N_K + 2N_E$. Since we use a piecewise constant space W_h for the pressure, there are $N_K - 1$ divergence-free constraints. Subtracting the number of divergence-free constraints from the total degrees of freedom (DoFs), $(6N_K + 2N_E) - (N_K - 1) = 6N_K + N_E + N_V$ (where we use the

Euler's formula in 2D), we get the dimension of the discrete divergence-free subspace D_h. Note that the total number of the three types of divergence-free basis functions is exactly $6N_K + N_E + N_V$, indicating that we found all the basis functions that are supported locally. Specifically, the above basis functions correspond to the components of $\mathbf{u}_0$ and $\mathbf{u}_b = \{\mathbf{u}_t, \mathbf{u}_V\}$. The basis functions for $\mathbf{u}_0$, i.e., $\{\Psi_0\}$, are defined only on the interior of each element T, which is the same as the previous SWG element. The basis functions for $\mathbf{u}_t$, that is, $\{\Psi_t\}$, are defined only on each edge $e \in \mathcal{E}_h^0$ along the tangential direction, and the basis functions for $\mathbf{u}_V$, i.e., $\{\Psi_V\}$, are defined only on the edges incident with the vertex V.

Using the divergence-free basis (2.2), the decoupled algorithm can be proposed to solely solve the velocity $\mathbf{u}_h$ as follows; see [3] for more details.

Algorithm 2.2. ***Divergence-free WG algorithm*** *A discrete divergence free approximation for (1.1)–(1.3) is to find* $\mathbf{u}_h = \{\mathbf{u}_0, \mathbf{u}_b\} \in D_h$ *such that*

$$a(\mathbf{u}_h,\ \mathbf{v}) \quad = \quad (\mathbf{f}, \mathbf{v}_0), \quad \forall\, \mathbf{v} = \{\mathbf{v}_0, \mathbf{v}_b\} \in D_h.$$

Although this algorithm decouples the unknown variables in $\mathbf{u}$ and p, it is essentially equivalent to the SWG Algorithm 2.1. Thus, the velocity error still depends on the pressure error (see Theorem 3.1 and Table 2). This may cause inaccuracy and instability when problems occur with a low viscosity and a pressure singularity. This computational challenge can be resolved using the pressure-robust enhancement, which will be discussed in the next section.

2.2. Pressure robust enhancement

We shall employ the velocity reconstruction operator to enhance Algorithm 2.2. The reconstruction operator $\Pi_h\mathbf{v} : D_h \to \tilde{D}_h \subset H(\mathrm{div}; \Omega)$ is defined as

$$\int_e \Pi_h\mathbf{v} \cdot \mathbf{n} ds = \int_e \mathbf{v} \cdot \mathbf{n} ds. \tag{2.3}$$

Let $\tilde{D}_h|_T = \mathrm{RT}_0(T) \cap H(\mathrm{div}; \Omega)$. As the fact that Ψ_t is aligning the tangential direction on each edge, we only need to compute the reconstruction operator corresponding to $\Psi_V = \{\Lambda_1 \cdots \Lambda_{N_V}\}$. It gives

$$\Pi_h\mathbf{v} = \begin{cases} 0, & \text{if } \mathbf{v} = \mathbf{v}_0 \in \Psi_0, \\ 0, & \text{if } \mathbf{v} = \mathbf{v}_b \in \Psi_t, \\ \Pi_h\Lambda_j, & \text{if } \mathbf{v} = \mathbf{v}_b \in \Psi_V = \mathrm{span}\{\Lambda_i, i = 1, \cdots, \mathrm{N}_V\}. \end{cases}$$

Here, it is easy to verify that in (2.3): $\Pi_h\Lambda_i = \sum_{j=1}^{r} \mathrm{sign}_{e_j} \phi_{e_j}^{\mathrm{RT}_0}$, where $\phi_{e_j}^{\mathrm{RT}_0}$ is the corresponding RT_0 basis on the edge e. We associate a unit normal vector $\mathbf{n}_e$ with $e \in \mathcal{E}_h^0$, which is assumed to be oriented from T^+ to T^-. If e is a boundary edge/face, then $\mathbf{n}_e$ is the unit outward normal vector to $\partial\Omega$. For the outer normal $\mathbf{n}$, if $[\mathbf{n}|_T]_{e_j} = \mathbf{n}_{e_j}$, we assign $\mathrm{sign}_{e_j} = 1$; if $[\mathbf{n}|_T]_{e_j} = -\mathbf{n}_{e_j}$, we assign $\mathrm{sign}_{e_j} = -1$. Thus, by employing Π_h, we propose the following pressure-robust scheme.

Algorithm 2.3. ***Pressure-robust divergence-free WG algorithm*** *A pressure-robust divergence-free approximation for (1.1)–(1.3) is to find* $\mathbf{u}_h = \{\mathbf{u}_0, \mathbf{u}_b\} \in D_h$ *satisfying*

$$a(\mathbf{u}_h, \mathbf{v}) \quad = \quad (\mathbf{f}, \Pi_h\mathbf{v}), \quad \forall\, \mathbf{v} = \{\mathbf{v}_0, \mathbf{v}_b\} \in D_h.$$

As we can see from the discretization, the stiffness matrix is the same as Algorithm 2.2, but only the load vector changes. By this minor modification, the desired pressure-robustness can be achieved. The results will be demonstrated in Theorem 3.2 and validated in numerical experiments.

Remark 2.1. *For triangular, rectangular, tetrahedral, and cubic meshes, we can directly employ the associated RT_0 or $RT_{[0]}$ basis functions to perform the velocity reconstruction. For polygonal / polyhedral meshes, the techniques in [7, 8] can be used to build the operator $\Pi_h\mathbf{v}$.*

2.3. Pressure recovering

In Algorithms 2.2 and 2.3, we decouple the unknowns and only compute the velocity solution $\mathbf{u}_h$. In some cases, the pressure variable is also needed. In this section, we propose the following procedure that computes the pressure after obtaining the velocity $\mathbf{u}_h$.

Algorithm 2.4. ***Pressure recovering algorithm*** *The pressure can be obtained by solving the following equation: find $p_h \in W_h$ such that*

$$b(\mathbf{v}, p_h) = \ell(\mathbf{v}) - a(\mathbf{u}_h, \mathbf{v}),\ \forall\, \mathbf{v} \in \mathbf{V}_h \backslash D_h.$$

Here, $\ell(\mathbf{v}) = (\mathbf{f}, \mathbf{v}_0)$ for Algorithm 2.2 and $\ell(\mathbf{v}) = (\mathbf{f}, \Pi_h\mathbf{v})$ for Algorithm 2.3. As $\mathbf{v} \in \mathbf{V}_h \backslash D_h$, let us assume $p_h^- = p_h|_T$ is already known, and we can choose $\mathbf{v} = \{\mathbf{v}_0 = 0, \mathbf{v}_b = \mathbf{n}_e\}$ with $e \in \partial T$ and the value of $p_h^+ = p_h$ on the adjacent element sharing the edge e is not computed. Then, the definition of $b(\cdot,\cdot)$ implies $b(\mathbf{v}, p_h) = (\nabla_w \cdot \mathbf{v}, p_h) = \sum_T \langle \mathbf{v}_b \cdot \mathbf{n}_e, p_h \rangle_{\partial T} = \langle \mathbf{v}_b \cdot \mathbf{n}_e, [\![p_h]\!] \rangle_e = |e| \left(p_h^+ - p_h^- \right)$. Here, $\mathbf{n}_e$ denotes the normal direction from the current element T to its adjacent element that shares the edge e. In the implementation, we can assume $p_h|_{T_1} = 0$ to start and compute all the values in $p_h|_T$ as above sequentially and locally. There is no need to form the global matrix explicitly.

2.4. Further DoFs reduction by eliminating $\mathbf{u}_0$ unknowns

In the above proposed algorithms, we can do further DoFs enhancement by eliminating the unknowns corresponding to $\mathbf{u}_0$ to obtain a smaller system. This elimination can be done locally when the global matrix is assembled via static condensation. To state the local elimination procedure, denote by $D_h(T)$ the restriction of D_h on T, i.e.,

$$D_h(T) = \{\mathbf{v} = \{\mathbf{v}_0, \mathbf{v}_b\} \in D_h, \mathbf{v}(\mathbf{x}) = 0, \text{ for } \mathbf{x} \notin T\}.$$

Algorithm 2.5. *An approximation to the problem (1.1)–(1.3) is given by seeking $\mathbf{u}_h = \{\mathbf{u}_0, \mathbf{u}_t, \mathbf{u}_b\} \in D_h$ satisfying a global equation*

$$a(\mathbf{u}_h, \mathbf{v}) = 0,\ \forall \mathbf{v} = \{0, \Psi_t, \Psi_b\} \in D_h,$$

and a local system on each element $T \in \mathcal{T}_h$,

$$a(\mathbf{u}_h, \mathbf{v}) = (\mathbf{f}, \mathbf{v}_0),\ \forall \mathbf{v} = \{\Psi_0, 0, 0\} \in D_h(T).$$

Remark 2.2. *The above algorithm consists of a local system solved on each element $T \in \mathcal{T}_h$ to eliminate $\mathbf{u}_0$ and a global system for $\mathbf{u}_b$, and a global system has $\mathbf{u}_b$ as its only unknowns that will reduce the number of unknowns of the WG system. The comparison of DoFs is shown in Figure 2.*

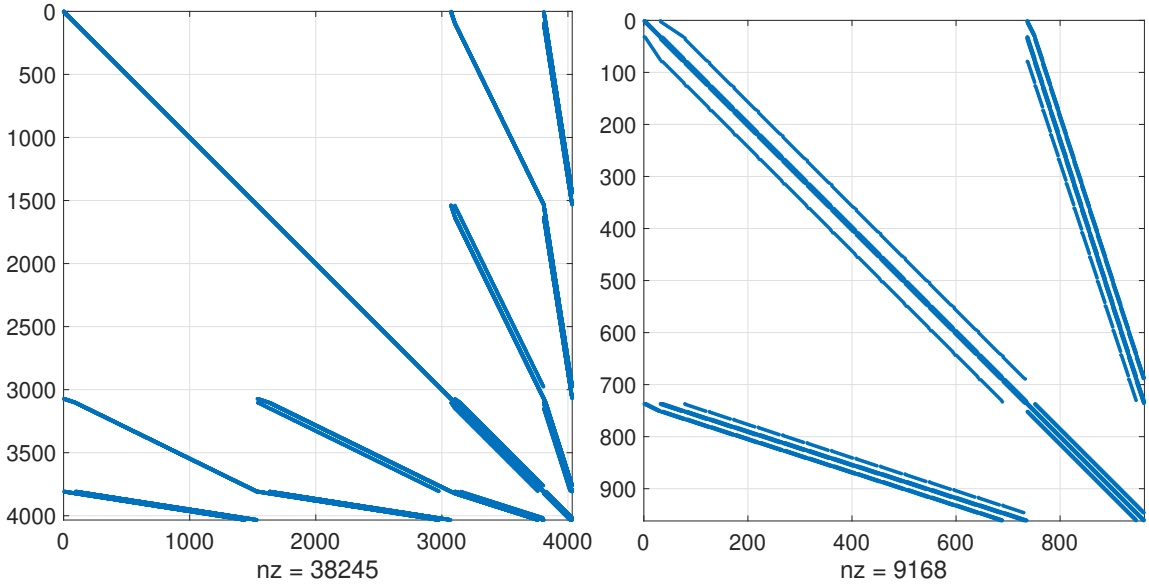

Figure 2. Sparsity pattern for Algorithm 2.2/2.3 (left) and Algorithm 2.5 (right) for uniform mesh with $h = 1/16$.

3. Convergence results

In this section, we present the error analysis of Algorithm 2.3 to demonstrate its advantages. Denote by Q_0 the L^2 projection operator from $[L^2(T)]^2$ onto $[P_1(T)]^2$. Define $Q_h\mathbf{u} = \{Q_0\mathbf{u}, Q_b\mathbf{u}\} \in V_h$ and let $\mathbb{Q}_h p$ be the local L^2 projections onto $P_0(T)$. Furthermore, we define the following norm corresponding to the WG finite element methods:

$$|\!|\!|\mathbf{v}|\!|\!| := \left(\sum_{T\in\mathcal{T}_h} \left(\|\nabla_w \mathbf{v}\|_T^2 + h_T^{-1} \|Q_b\mathbf{v}_0 - \mathbf{v}_b\|_{\partial T}^2 \right) \right)^{1/2}.$$

The following optimal error estimates have been derived in [3, 38].

Theorem 3.1. *(Non-pressure robust scheme) Let* $(\mathbf{u}; p) \in [H_0^1(\Omega) \cap H^2(\Omega)]^2 \times (L_0^2(\Omega) \cap H^1(\Omega))$ *be the solution of (1.1)–(1.3) and* $(\mathbf{u}_h; p_h) \in V_h \times W_h$ *be the solutions of (1.1)–(1.3) and Algorithm 2.1 or Algorithms 2.2–2.4, respectively. Then, the following error estimate holds true,*

$$|\!|\!|Q_h\mathbf{u} - \mathbf{u}_h|\!|\!| \le Ch(\|\mathbf{u}\|_2 + \frac{1}{\nu}\|p\|_1), \quad \|\mathbb{Q}_h p - p_h\| \le Ch(\nu\|\mathbf{u}\|_2 + \|p\|_1). \tag{3.1}$$

Proof. The proofs can be found in [3].

Theorem 3.2. *(Pressure-robust scheme) Let* $(\mathbf{u}; p) \in [H_0^1(\Omega) \cap H^2(\Omega)]^2 \times (L_0^2(\Omega) \cap H^1(\Omega))$ *be the solution of (1.1)–(1.3) and* $(\mathbf{u}_h; p_h) \in V_h \times W_h$ *be the solution of (1.1)–(1.3) and Algorithms 2.3 and 2.4, respectively. Then, the following error estimate holds true,*

$$|\!|\!|Q_h\mathbf{u} - \mathbf{u}_h|\!|\!| \le Ch\|\mathbf{u}\|_2, \quad \|\mathbb{Q}_h p - p_h\| \le Ch\nu\|\mathbf{u}\|_2. \tag{3.2}$$

Proof. By estimating the inconsistent errors caused by changing the righthand load vector and following the techniques in [3], the theorem can be proved.

Remark 3.1. *Although the divergence-free scheme only needs to solve the velocity component, the velocity error may still depend on the pressure, which is a non-pressure-robust scheme as shown in Theorem 3.1. By modifying the load vector, we completely remove the pressure dependence in the error estimate to achieve the desired pressure robustness. Besides, the pressure-robust error analysis can be obtained similarly to the rigorous analysis in [7].*

4. Numerical experiments

In this section, we test several benchmark problems to report numerical performance and validate the convergence results shown in Theorems 3.1 and 3.2. In all numerical tests, triangular meshes have been used.

4.1. Example 1. Homogeneous Dirichlet boundary condition.

Let $\Omega = (0, 1) \times (0, 1)$ and the exact solution $\mathbf{u}$ and p be,

$$\mathbf{u} = \begin{pmatrix} 10x^2y(x-1)^2(2y-1)(y-1) \\ -10xy^2(2x-1)(x-1)(y-1)^2 \end{pmatrix} \text{ and } p = 10(2x-1)(2y-1).$$

Denote the errors $\mathbf{e} = \{(Q_0\mathbf{u} - \mathbf{u}_0, Q_b\mathbf{u} - \mathbf{u}_b)\}$ and $\epsilon = \mathbb{Q}_h p - p_h$. We first compare the computational cost corresponding to Algorithm 2.1, Algorithm 2.2/2.3, and Algorithm 2.5. Since the stiffness matrix corresponding to Algorithms 2.2 and 2.3 is identical, we only compute the DoFs for Algorithm 2.2. The profiles for DoFs are reported in Table 1. Here, we exclude the unknowns for $\mathbf{u}_b$ for the Dirichlet boundary as computing the required DoFs.

Table 1. Example 4.1: Comparison of DoFs for different algorithms. Here, "DoFs" denotes the degrees of freedom and "nnz" denotes the number of nonzeros of the stiffness matrix.

	Algorithm 2.1			Algorithm 2.2/2.3		Algorithm 2.5	
N	DoFs_u	DoFs_p	DoFs	DoFs	nnz	DoFs	nnz
4	272	32	304	241	1861	49	360
8	1120	128	1248	993	8805	225	1984
16	4544	512	5056	4033	38,245	961	9168
32	18,304	2048	20,352	16,257	159,333	3969	39,280
64	73,472	8192	81,664	65,281	650,341	16,129	162,480
128	294,400	32,768	327,168	261,633	2,627,685	65,025	660,784
256	1,178,624	131,072	1,309,696	1,047,553	10,563,685	261,121	2,665,008

In this table, we first observe that the required DoFs can be significantly reduced by employing the divergence-free basis. Since we only modify the assembly of the source term in Algorithm 2.3, the required DoFs in Algorithms 2.2 and 2.3 remain the same. As static condensation is employed (Algorithm 2.5), the DoFs of the global system can be further reduced. For example, when $N = 256$, the size of the global system is reduced from 1 M to 0.2 M, while the density of the matrix increased from 1E-5 to 4E-5.

Next, we will test the performance corresponding to non-pressure-robust scheme, Algorithm 2.2/Algorithm 2.4, and pressure-robust scheme, Algorithm 2.3/Algorithm 2.4, for a sequence of meshes and

varying values in ν. In Table 2, we report the error profiles and the convergence results. We observed that:

- All the error profiles produced by Algorithm 2.2/Algorithm 2.4 and Algorithm 2.3/Algorithm 2.4 result in the optimal convergence rate: the velocity error measured in H^1-norm and pressure error measured in L^2-norm are of order $O(h)$; the velocity error measured in L^2-norm is of order $O(h^2)$. The convergence rates stay the same for different viscosity values ν.
- The velocity errors produced by the non-pressure-robust scheme Algorithms 2.2–2.4 depend on the viscosity ν. When ν decreases, the velocity error measured in the norms H^1- and L^2- increases on the order of $\frac{1}{\nu}$. In contrast, the pressure errors measured in L^2-norm stay the same as ν varies. These observations agree with (3.1).
- The velocity errors produced by pressure-robust scheme Algorithms 2.3 and 2.4 do not depend on the viscosity ν. When ν decreases, the velocity error measured in the norms H^1- and L^2- remains the same. In contrast, the pressure error measured in L^2-norm decreases at the order ν. These observations agree with (3.2).
- The above observations validate that, although the scheme can decouple the unknowns in velocity and pressure and solve them independently, the div-free finite element space is sometimes insufficient to ensure the accuracy of numerical solutions with satisfaction.

4.2. Example 2 - Nonhomogeneous Dirichlet boundary condition

In this test, we shall consider the nonhomogeneous Dirichlet boundary conditions. Let $\Omega = (0, 1) \times (0, 1)$ and the exact solution is taken as

$$\mathbf{u} = \begin{pmatrix} \sin(\pi x)\sin(\pi y) \\ \cos(\pi x)\cos(\pi y) \end{pmatrix}, \text{ and } p = \sin x.$$

It is easy to see that $\mathbf{u}|_{\partial\Omega} \neq 0$. Thus, one needs to modify the method in order to deal with nonhomogeneous Dirichlet boundary conditions.

Table 3 reports the error profiles and convergence results. We compare the performance of Algorithm 2.2/2.4 and Algorithm 2.3/2.4 on a sequence of meshes with different values of ν. To start, non-pressure-robust scheme Algorithm 2.2 and pressure-robust scheme Algorithm 2.3 have been employed to simulate the numerical velocity component. Then, when the velocity approximation is available, Algorithm 2.4 is used to recover the unknown pressure. As in the above test, though Algorithm 2.2 can decouple velocity/pressure and solely solve the unknown velocity, Algorithm 2.2 fails to produce reliable numerical solutions when the viscosity values are small. In contrast, the pressure-robust scheme Algorithm 2.3 is able to produce a viscosity-independent simulation for the velocity. As viscosity values vary, velocity errors (measured in the L^2-norm and the H^1-norm) remain the same. Furthermore, reducing viscosity values ν produces a more accurate numerical pressure, which gives a convergence rate for the pressure measured in the L^2-norm as $O(h)$. These numerical results confirm the theoretical conclusions in the above section.

Table 2. Example 4.1: Error profiles and convergence results.

$1/h$	$\|\mathbf{e}_0\|$	order	$\|\nabla\mathbf{e}_0\|$	order	$\|\epsilon\|$	order	$\|\mathbf{e}_0\|$	order	$\|\nabla\mathbf{e}_0\|$	order	$\|\epsilon\|$	order
	Algorithm 2.2/Algorithm 2.4						Algorithm 2.3/Algorithm 2.4					
ν = 1												
8	6.30E-2		5.23E-1		1.29E+0		8.27E-3		1.70E-1		2.71E-2	
16	1.72E-2	1.9	2.84E-1	0.9	6.31E-1	1.0	2.17E-3	1.9	8.74E-2	1.0	1.17E-2	1.2
32	4.47E-3	1.9	1.47E-1	0.9	3.00E-1	1.1	5.50E-4	2.0	4.41E-2	1.0	5.32E-3	1.1
64	1.13E-3	2.0	7.44E-2	1.0	1.44E-1	1.1	1.38E-4	2.0	2.21E-2	1.0	2.57E-3	1.0
128	2.84E-4	2.0	3.74E-2	1.0	6.98E-2	1.0	3.46E-5	2.0	1.10E-2	1.0	1.27E-3	1.0
ν = 1E-2												
8	6.21E+0		5.10E+1		1.29E+0		8.27E-3		1.70E-1		2.71E-4	
16	1.70E+0	1.9	2.79E+1	0.9	6.31E-1	1.0	2.17E-3	1.9	8.74E-2	1.0	1.17E-4	1.2
32	4.41E-1	1.9	1.45E+1	0.9	3.00E-1	1.1	5.50E-4	2.0	4.41E-2	1.0	5.32E-5	1.1
64	1.12E-1	2.0	7.32E+0	1.0	1.44E-1	1.1	1.38E-4	2.0	2.21E-2	1.0	2.57E-5	1.0
128	2.80E-2	2.0	3.68E+0	1.0	6.98E-2	1.0	3.46E-5	2.0	1.10E-2	1.0	1.27E-5	1.0
ν = 1E-4												
8	6.21E+2		5.10E+3		1.29E+0		8.27E-3		1.70E-1		2.71E-6	
16	1.70E+2	1.9	2.79E+3	0.9	6.31E-1	1.0	2.17E-3	1.9	8.74E-2	1.0	1.17E-6	1.2
32	4.41E+1	1.9	1.45E+3	0.9	3.00E-1	1.1	5.50E-4	2.0	4.41E-2	1.0	5.32E-7	1.1
64	1.12E+1	2.0	7.32E+2	1.0	1.44E-1	1.1	1.38E-4	2.0	2.21E-2	1.0	2.57E-7	1.0
128	2.80E+0	2.0	3.68E+2	1.0	6.98E-2	1.0	3.46E-5	2.0	1.10E-2	1.0	1.27E-7	1.0
ν = 1E-6												
8	6.21E+4		5.10E+5		1.29E+0		8.27E-3		1.70E-1		2.71E-8	
16	1.70E+4	1.9	2.79E+5	0.9	6.31E-1	1.0	2.17E-3	1.9	8.74E-2	1.0	1.17E-8	1.2
32	4.41E+3	1.9	1.45E+5	0.9	3.00E-1	1.1	5.50E-4	2.0	4.41E-2	1.0	5.32E-9	1.1
64	1.12E+3	2.0	7.32E+4	1.0	1.44E-1	1.1	1.38E-4	2.0	2.21E-2	1.0	2.57E-9	1.0
128	2.80E+2	2.0	3.68E+4	1.0	6.98E-2	1.0	3.46E-5	2.0	1.10E-2	1.0	1.27E-9	1.0
ν = 1E-8												
8	6.21E+6		5.10E+7		1.29E+0		8.27E-3		1.70E-1		2.71E-10	
16	1.70E+6	1.9	2.79E+7	0.9	6.31E-1	1.0	2.17E-3	1.9	8.74E-2	1.0	1.17E-10	1.2
32	4.41E+5	1.9	1.45E+7	0.9	3.00E-1	1.1	5.50E-4	2.0	4.41E-2	1.0	5.32E-11	1.1
64	1.12E+5	2.0	7.32E+6	1.0	1.44E-1	1.1	1.38E-4	2.0	2.21E-2	1.0	2.57E-11	1.0
128	2.80E+4	2.0	3.68E+6	1.0	6.98E-2	1.0	3.46E-5	2.0	1.10E-2	1.0	1.27E-11	1.0

Table 3. Example 4.2: Error profiles and convergence results.

$1/h$	$\|\|\mathbf{e}_0\|\|$	order	$\|\|\nabla\mathbf{e}_0\|\|$	order	$\|\|\epsilon\|\|$	order	$\|\|\mathbf{e}_0\|\|$	order	$\|\|\nabla\mathbf{e}_0\|\|$	order	$\|\|\epsilon\|\|$	order
	Algorithm 2.2/Algorithm 2.4						Algorithm 2.3/Algorithm 2.4					
						$\nu = 1$						
8	5.12E-2		5.83E-1		4.57E-1		2.81E-2		8.76E-1		7.62E-1	
16	1.28E-2	2.0	2.91E-1	1.0	2.36E-1	1.0	7.69E-3	1.9	4.54E-1	0.9	4.41E-1	0.8
32	3.18E-3	2.0	1.46E-1	1.0	1.19E-1	1.0	2.00E-3	1.9	2.30E-1	1.0	2.37E-1	0.9
64	7.95E-4	2.0	7.29E-2	1.0	5.95E-2	1.0	5.06E-4	2.0	1.15E-1	1.0	1.23E-1	0.9
128	1.99E-4	2.0	3.65E-2	1.0	2.98E-2	1.0	1.27E-4	2.0	5.78E-2	1.0	6.29E-2	1.0
						ν = 1E-2						
8	3.80E-1		2.87		1.99E-2		2.81E-2		8.76E-1		7.43E-3	
16	9.79E-2	2.0	1.47	1.0	1.01E-2	1.0	7.69E-3	1.9	4.54E-1	0.9	4.36E-3	0.8
32	2.47E-2	2.0	7.44E-1	1.0	5.16E-3	1.0	2.00E-3	1.9	2.30E-1	1.0	2.36E-3	0.9
64	6.20E-3	2.0	3.73E-1	1.0	2.63E-3	1.0	5.06E-4	2.0	1.15E-1	1.0	1.23E-3	0.9
128	1.55E-3	2.0	1.87E-1	1.0	1.33E-3	1.0	1.27E-4	2.0	5.78E-2	1.0	6.28E-4	1.0
						ν = 1E-4						
8	3.51E+1		2.91E+2		2.41E-2		2.81E-2		8.76E-1		1.43E-4	
16	9.06	2.0	1.51E+2	0.9	1.24E-2	1.0	7.69E-3	1.9	4.54E-1	0.9	2.40E-5	2.6
32	2.29	2.0	7.65E+1	1.0	6.36E-3	1.0	2.00E-3	1.9	2.30E-1	1.0	1.29E-5	0.9
64	5.74E-1	2.0	3.84E+1	1.0	3.24E-3	1.0	5.06E-4	2.0	1.15E-1	1.0	9.36E-6	0.5
128	1.44E-1	2.0	1.92E+1	1.0	1.64E-3	1.0	1.27E-4	2.0	5.78E-2	1.0	5.53E-6	0.8
						ν = 1E-6						
8	3.50E+3		2.92E+4		2.41E-2		2.81E-2		8.76E-1		2.08E-4	
16	9.05E+2	2.0	1.51E+4	0.9	1.24E-2	1.0	7.69E-3	1.9	4.54E-1	0.9	5.40E-5	1.9
32	2.29E+2	2.0	7.65E+3	1.0	6.37E-3	1.0	2.00E-3	1.9	2.30E-1	1.0	1.37E-5	2.0
64	5.74E+1	2.0	3.84E+3	1.0	3.25E-3	1.0	5.06E-4	2.0	1.15E-1	1.0	3.40E-6	2.0
128	1.44E+1	2.0	1.92E+3	1.0	1.64E-3	1.0	1.27E-4	2.0	5.78E-2	1.0	8.26E-7	2.0
						ν = 1E-8						
8	3.50E+5		2.92E+6		2.41E-2		2.81E-2		8.76E-1		2.09E-4	
16	9.05E+4	2.0	1.51E+6	0.9	1.24E-2	1.0	7.69E-3	1.9	4.54E-1	0.9	5.44E-5	1.9
32	2.29E+4	2.0	7.65E+5	1.0	6.37E-3	1.0	2.00E-3	1.9	2.30E-1	1.0	1.39E-5	2.0
64	5.74E+3	2.0	3.84E+5	1.0	3.25E-3	1.0	5.06E-4	2.0	1.15E-1	1.0	3.50E-6	2.0
128	1.44E+3	2.0	1.92E+5	1.0	1.64E-3	1.0	1.27E-4	2.0	5.78E-2	1.0	8.80E-7	2.0

4.3. Example 3 - Mixed boundary condition

In this test, let $\Omega = (0, 1) \times (0, 1)$ and the exact solution $\mathbf{u}$ and p be,

$$\mathbf{u} = \begin{pmatrix} 3\pi \sin(\pi x)^3 \sin(\pi y)^2 \cos(\pi y) \\ -3\pi \sin(\pi x)^2 \sin(\pi y)^3 \cos(\pi x) \end{pmatrix} \text{ and } p = \sin(\pi x).$$

In this test, the top ($y = 1$), left ($x = 0$), and bottom ($y = 0$) boundaries are assumed to be the Dirichlet boundary conditions. The right boundary ($x = 1$) employs the Neumann boundary condition.

We perform convergence tests on a sequence of meshes with varying viscosity values ν. The error profiles for the velocity and pressure solutions are plotted in Figures 3–5, in which we vary the mesh size h. The numerical results are similar to the two examples above. Figure 3 demonstrates the velocity errors measured in the L^2-norm. When we use the non-pressure-robust scheme Algorithm 2.2/2.4, the errors converge at the second order for different values of ν. However, the velocity error depends on ν and the pressure. As the pressure error does not dominate the velocity errors, the error increases as a factor 1ν. This means that Algorithm 2.2/2.4 fails to produce reliable numerical solutions for the velocity. In contrast, Algorithm 2.3/2.4 outperforms Algorithm 2.2/2.4 and produces a robust numerical simulation. The velocity errors show an invariant behavior for varying viscosity values ν.

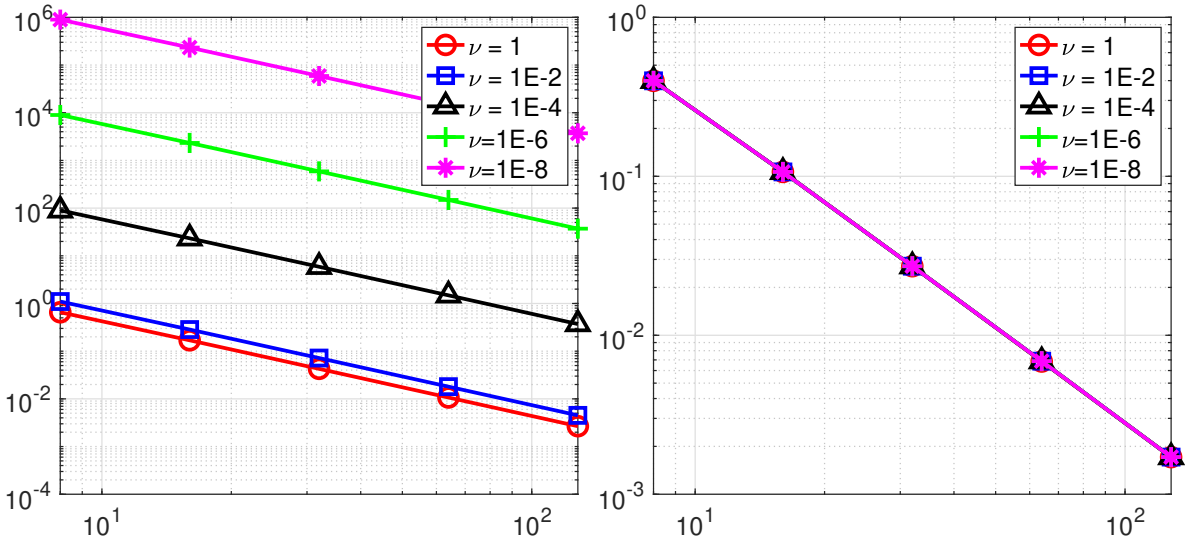

Figure 3. Example 4.3: Convergence results for Algorithm 2.2/2.4 (Left) and Algorithm 2.3/2.4 (Right) for velocity errors measured in L^2-norm.

The H^1-error of the velocity is plotted in Figure 4. We observe the same behavior and can draw the same conclusion as above.

Lastly, the L^2-error of the pressure is plotted in Figure 5. Algorithm 2.2/2.4 first produces a better pressure approximation before the dominance of pressure error $h\|p\|_1$. However, as the viscosity variable ν decreases, the pressure term will dominate the error, i.e., $h\|p\|_1 \gg h\nu\|\mathbf{u}\|_2$. Thus, pressure errors show a constant behavior on the same mesh with decreasing values of ν. On the other hand, Algorithm 2.3/2.4 outperforms Algorithm 2.2/2.4. For small ν, Algorithm 2.3/2.4 produces much better pressure solutions. All of the above tests validate our theoretical conclusions.

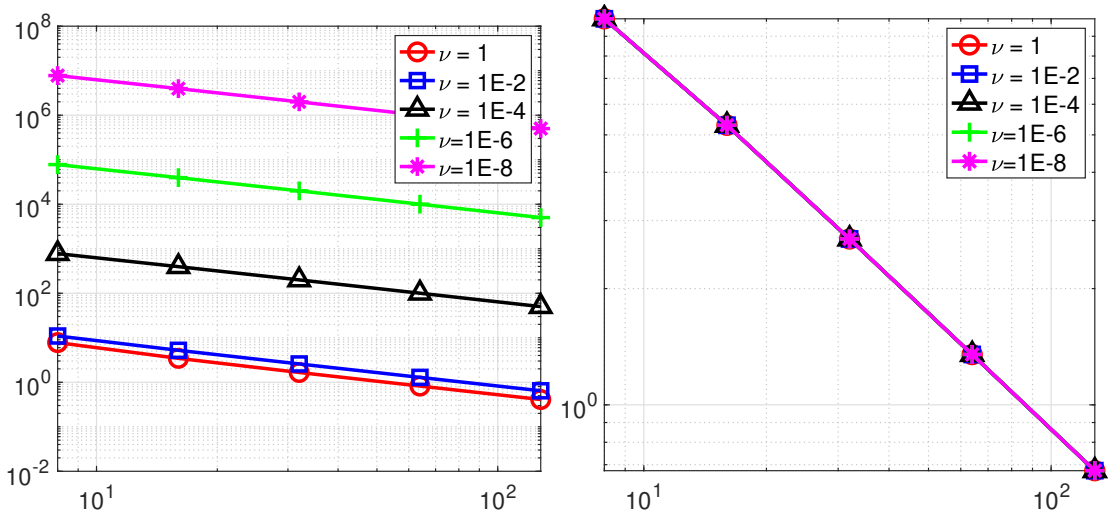

Figure 4. Example 4.3: Convergence results for Algorithm 2.2/2.4 (Left) and Algorithm 2.3/2.4 (Right) for velocity errors measured in H^1-norm.

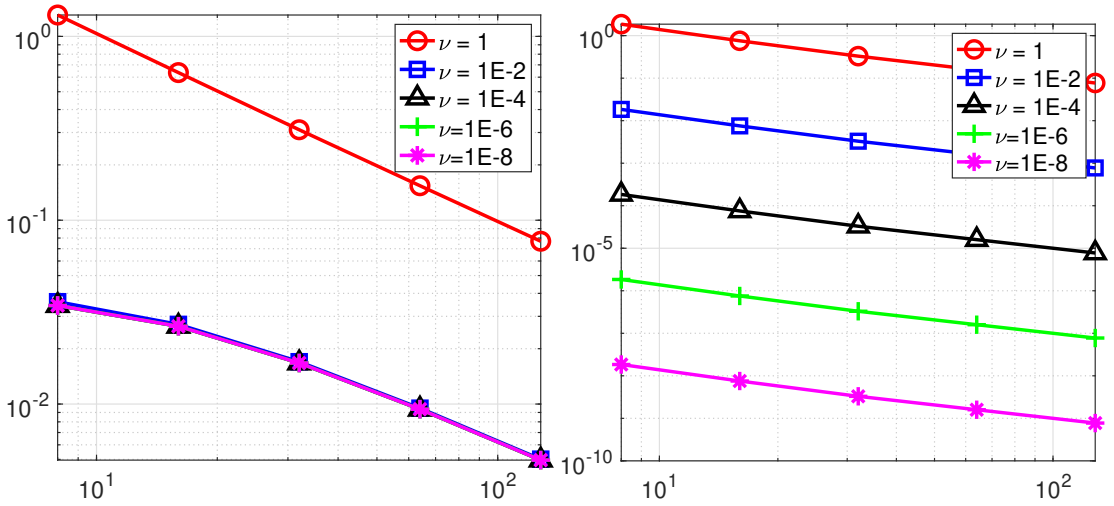

Figure 5. Example 4.3: Convergence results for Algorithm 2.2/2.4 (Left) and Algorithm 2.3/2.4 (Right) for pressure errors measured in L^2-norm.

5. Conclusions

In this paper, we enhanced the divergence-free WG finite element method proposed in [3] by modifying the load function via the velocity reconstruction operator. Our proposed algorithm results in a symmetric positive definite matrix, and the velocity error is pressure-robust. Moreover, we illustrated the procedure for recovering the pressure variables. Numerical experiments are presented to validate the theoretical results. As a future work plan, we will consider the development of an effective preconditioner, which may improve the efficiency further. In addition, the construction of a high-order ($k > 2$) divergence-free basis will be given and analyzed in our future work.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

Jay Chu is partially supported by the Ministry of Science and Technology of Taiwan under the research grant MOST 111-2115-M-007-012. Xiaozhe Hu is partially supported by the National Science Foundation under the grant DMS-2208267. Lin Mu is partially supported by the National Science Foundation under the grant DMS-2309557.

Conflict of interest

The authors declare there are no conflicts of interest.

References

1. S. C. Brenner, A nonconforming multigrid method for the stationary Stokes equations, *Math. Comput.*, **55** (1990), 411–437. https://doi.org/10.1090/S0025-5718-1990-1035927-5
2. F. Thomasset, *Implementation of Finite Element Methods for Navier-Stokes equations*, Springer-Verlag, New York-Berlin, 1981.
3. L. Mu, J. Wang, X. Ye, S. Zhang, A discrete divergence free weak Galerkin finite element method for the Stokes equations, *Appl. Numer. Math.*, **125** (2018), 172–182. https://doi.org/10.1016/j.apnum.2017.11.006
4. J. H. Adler, Y. He, X. Hu, S. MacLachlan, P. Ohm, Monolithic multigrid for a reduced-quadrature discretization of poroelasticity, *SIAM J. Sci. Comput.*, **45** (2023), S54–S81. https://doi.org/10.1137/21M1429072
5. A. Linke, A divergence-free velocity reconstruction for incompressible flows, *C.R. Math.*, **350** (2012), 837–840. https://doi.org/10.1016/j.crma.2012.10.010
6. A. Linke, On the role of the Helmholtz decomposition in mixed methods for incompressible flows and a new variational crime, *Comput. Methods Appl. Mech. Eng.*, **268** (2014), 782–800. https://doi.org/10.1016/j.cma.2013.10.011
7. L. Mu, Pressure robust weak Galerkin finite element methods for Stokes problems, *SIAM J. Sci. Comput.*, **42** (2020), 608–629. https://doi.org/10.1137/19M1266320
8. L. Mu, X. Ye, S. Zhang, A stabilizer-free, pressure-robust, and superconvergence weak Galerkin finite element method for the Stokes equations on polytopal mesh, *SIAM J. Sci. Comput.*, **43** (2021), 2614–2637. https://doi.org/10.1137/20M1380405
9. X. Hu, S. Lee, L. Mu, S. Y. Yi, Pressure-robust enriched Galerkin methods for the Stokes equations, *J. Comput. Appl. Math.*, **436** (2024), 115449. https://doi.org/10.1016/j.cam.2023.115449
10. P. L. Lederer, A. Linke, C. Merdon, J. Schóoberl, Divergence-free reconstruction operators for pressure-robust Stokes discretizations with continuous pressure finite elements, *SIAM J. Numer. Anal.*, **55** (2017), 1291–1314. https://doi.org/10.1137/16M1089964

11. A. Linke, C. Merdon, W. Wollner, Optimal L^2 velocity error estimate for a modified pressure-robust Crouzeix–Raviart Stokes element, *IMA J. Numer. Anal.*, **37** (2017), 354–374. https://doi.org/10.1093/imanum/drw019
12. A. Linke, G. Matthies, L. Tobiska, Robust arbitrary order mixed finite element methods for the incompressible Stokes equations with pressure independent velocity errors, *ESAIM. Math. Model. Numer. Anal.*, **50** (2016), 289–309. https://doi.org/10.1051/m2an/2015044
13. D. Frerichs, C. Merdon, Divergence-preserving reconstructions on polygons and a really pressure-robust virtual element method for the Stokes problem, *IMA J. Numer. Anal.*, **42** (2022), 597–619. https://doi.org/10.1093/imanum/draa073
14. P. L. Lederer, S. Rhebergen, A pressure-robust embedded discontinuous Galerkin method for the Stokes problem by reconstruction operators, *SIAM J. Numer. Anal.*, **58** (2020), 2915–2933. https://doi.org/10.1137/20M1318389
15. G. Wang, L. Mu, Y. Wang, Y. He, A pressure-robust virtual element method for the Stokes problem, *Comput. Methods Appl. Mech. Eng.*, **382** (2021), 113879. https://doi.org/10.1016/j.cma.2021.113879
16. A. Linke, C. Merdon, M. Neilan, F. Neumann, Quasi-optimality of a pressure-robust nonconforming finite element method for the Stokes-problem, *Math. Comput.*, **87** (2018), 1543–1566. https://doi.org/10.1090/mcom/3344
17. K. L. Kirk, S. Rhebergen, Analysis of a pressure-robust hybridized discontinuous Galerkin method for the stationary Navier–Stokes equations, *J. Sci. Comput.*, **81** (2019), 881–897. https://doi.org/10.1007/s10915-019-01040-y
18. D. Yang, Y. He, Y. Zhang, Analysis and computation of a pressure-robust method for the rotation form of the incompressible Navier–Stokes equations with high-order finite elements, *Comput. Math. Appl.*, **112** (2022), 1–22. https://doi.org/10.1016/j.camwa.2022.02.017
19. C. Merdon, W. Wollner, Pressure-robustness in the context of optimal control, *SIAM J. Control Optim.*, **61** (2023), 342–360. https://doi.org/10.1137/22M1482603
20. X. Liu, Y. Nie, Pressure-independent velocity error estimates for (Navier-)Stokes nonconforming virtual element discretization with divergence free, *Numerical Algorithms*, **90** (2022), 477–506. https://doi.org/10.1007/s11075-021-01195-6
21. Y. Wang, G. Wang, Y. Shen, A pressure-robust virtual element method for the Navier-Stokes problem on polygonal mesh, *Comput. Math. Appl.*, **131** (2023), 124–137. https://doi.org/10.1016/j.camwa.2022.12.013
22. S. Lee, L. Mu, A uniform and pressure-robust enriched Galerkin method for the Brinkman equations, *J. Sci. Comput.*, **99** (2024), 39. https://doi.org/10.1007/s10915-024-02503-7
23. S. Rhebergen, G. N. Wells, Preconditioning for a pressure-robust HDG discretization of the Stokes equations, *SIAM J. Sci. Comput.*, **44** (2022), 583–604. https://doi.org/10.1137/21M1420964
24. D. Kim, L. Zhao, E. Chung, E. J. Park, Pressure-robust staggered DG methods for the Navier-Stokes equations on general meshes, preprint, arXiv:2107.09226.
25. S. Zhang, A new family of stable mixed finite elements for the 3D Stokes equations, *Math. Comput.*, **74** (2005), 543–554. https://doi.org/10.1090/S0025-5718-04-01711-9

26. S. Zhang, A family of Qk+1,k × Qk,k+1 divergence-free finite elements on rectangular grids, *SIAM J. Numer. Anal.*, **47** (2009), 2090–2107. https://doi.org/10.1137/080728949
27. J. Guzman, M. Neilan, Conforming and divergence-free Stokes elements on general triangular meshes, *Math. Comput.*, **83** (2014), 15–36. https://doi.org/10.1090/S0025-5718-2013-02753-6
28. J. Guzman, M. Neilan, Inf-sup stable finite elements on barycentric refinements producing divergence-free approximations in arbitrary dimensions, *SIAM J. Numer. Anal.*, **56** (2018), 2826–2844. https://doi.org/10.1137/17M1153467
29. S. H. Christiansen, K. Hu, Generalized finite element systems for smooth differential forms and Stokes' problem, *Numer. Math.*, **140** (2018), 327–371. https://doi.org/10.1007/s00211-018-0970-6
30. J. Wang, X. Ye, New finite element methods in computational fluid dynamics by H(div) elements, *SIAM J. Numer. Anal.*, **45** (2007), 1269–1286. https://doi.org/10.1137/060649227
31. C. Lehrenfeld, J. Schoberl, High order exactly divergence-free hybrid discontinuous Galerkin methods for unsteady incompressible flows, *Comput. Methods Appl. Mech. Eng.*, **307** (2016), 339–361. https://doi.org/10.1016/j.cma.2016.04.025
32. J. Carrero, B. Cockburn, D. Schotzau, Hybridized globally divergence-free LDG methods. Part I: The Stokes problem, *Math. Comput.*, **75** (2006), 533–563. https://doi.org/10.1090/S0025-5718-05-01804-1
33. G. Fu, Y. Jin, W. Qiu, Parameter-free superconvergent H(div)-conforming HDG methods for the Brinkman equations, *IMA J. Numer. Anal.*, **39** (2019), 957–982. https://doi.org/10.1093/imanum/dry001
34. V. John, A. Linke, C. Merdon, M. Neilan, L. G. Rebholz, On the divergence constraint in mixed finite element methods for incompressible flows, *SIAM Rev.*, **59** (2017), 492–544. https://doi.org/10.1137/15M1047696
35. X. Li, H. Rui, A low-order divergence-free H (div)-conforming finite element method for Stokes flows, *IMA J. Numer. Anal.*, **42** (2022), 3711–3734. https://doi.org/10.1093/imanum/drab080
36. V. John, X. Li, C. Merdon, H. Rui, Inf–sup stabilized Scott–Vogelius pairs on general shape-regular simplicial grids by Raviart–Thomas enrichment, *Math. Models Methods Appl. Sci.*, **34** (2024), 919–949. https://doi.org/10.1142/S0218202524500180
37. J. Wang, X. Ye, A weak Galerkin mixed finite element method for second-order elliptic problems, *Math. Comput.*, **83** (2014), 2101–2126. https://doi.org/10.1090/S0025-5718-2014-02852-4
38. J. Wang, X. Ye, A weak Galerkin finite element method for the Stokes equations, *Adv. Comput. Math.*, **42** (2016), 155–174. https://doi.org/10.1007/s10444-015-9415-2

AIMS Press