INTERNATIONAL JOURNAL OF
NUMERICAL ANALYSIS AND MODELING
Volume 21, Number 2, Pages 181–200

doi: 10.4208/ijnam2024-1007

AN H^1-CONFORMING SOLENOIDAL BASIS FOR VELOCITY COMPUTATION ON POWELL-SABIN SPLITS FOR THE STOKES PROBLEM

JEFFREY M. CONNORS AND MICHAEL GAIEWSKI

Abstract. A solenoidal basis is constructed to compute velocities using a certain finite element method for the Stokes problem. The method is conforming, with piecewise linear velocity and piecewise constant pressure on the Powell-Sabin split of a triangulation. Inhomogeneous Dirichlet conditions are supported by constructing an interpolating operator into the solenoidal velocity space. The solenoidal basis reduces the problem size and eliminates the pressure variable from the linear system for the velocity. A basis of the pressure space is also constructed that can be used to compute the pressure after the velocity, if it is desired to compute the pressure. All basis functions have local support and lead to sparse linear systems. The basis construction is confirmed through rigorous analysis. Velocity and pressure system matrices are both symmetric, positive definite, which can be exploited to solve their corresponding linear systems. Significant efficiency gains over the usual saddle-point formulation are demonstrated computationally.

Key words. Divergence free, Powell-Sabin, Stokes, finite element, saddle-point.

1. Introduction

This paper relates to finite element computations for the incompressible Stokes problem in two dimensions. Given the real, simply connected and polygonal domain $\Omega \subset \mathbb{R}^2$, we consider the Dirichlet problem to solve for velocity $\mathbf{u} : \overline{\Omega} \to \mathbb{R}^2$ and pressure $p : \Omega \to \mathbb{R}$ such that

$$-\nu \Delta \mathbf{u} + \nabla p = \mathbf{f} \quad \text{on } \Omega, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0 \quad \text{on } \Omega, \tag{2}$$

$$\mathbf{u} = \mathbf{g} \quad \text{on } \partial\Omega, \tag{3}$$

$$\text{and } \int_\Omega p \, d\mathbf{x} = 0, \tag{4}$$

where $\nu > 0$ is a constant viscosity parameter and $\mathbf{g}$ is a target set of boundary values for the velocity field. Bold font will be reserved for vectors and spaces of vector-valued functions.

There are countless research expositions related to the Stokes problem since it connects to many scientific models and problems in mathematics. Here, we are primarily interested in the computation of velocity and pressure variables using a certain finite element method where the incompressibility constraint (2) is ultimately satisfied pointwise over the domain, yielding a true solenoidal velocity field. In contrast, many methods only satisfy this condition in some weak (integrated) sense, and special mixed pairings of elements for the velocity and pressure are generally needed for strong incompressibility. A motivation is that solenoidal velocities break a coupling in the consistency error between velocity and pressure variables, so that velocity computation is not polluted by errors that should only affect the pressure

Received by the editors on August 14, 2023 and, accepted on December 20, 2023.
2000 *Mathematics Subject Classification.* 65N30.

accuracy. In the literature, this property is called *pressure robustness.* A review of methods may be found in [8].

These methods turn out to carry an additional potential benefit: to construct a locally-supported and solenoidal basis directly for the velocity space. Whereas a certain saddle-point problem is typically solved for the velocity and pressure variables, a solenoidal basis can be used to express the linear system in a block-triangular fashion that allows the velocity to be computed without computing the pressure. If the pressure is desired, it can then be calculated after via a separate solve, but in both cases the saddle-point solve is replaced by smaller, symmetric positive-definite systems. Few methods of this type exist at present, and it is not clear that a solenoidal basis can always have a local support. There are methods of discontinuous-Galerkin (DG) type [7, 10, 11]. Also, for Raviart-Thomas (RT), Brezzi-Douglass-Marini (BDM) and hybridized locally-DG mixed elements that possess a weak divergence but not a weak gradient, see [2, 15, 1, 16]. In the H^1-conforming case there is method with fourth-order polynomial velocities [12] and a method with first-order velocities [13].

The purpose of this paper is to develop the decoupled velocity and pressure computations for the mixed pair in [4]. The method enforces solenoidal, first-order polynomial velocities and is H^1-conforming. The element order is the same as the method in [13], but the finite element mesh is quite different. We use Powell-Sabin splits that subdivide triangles of a fairly general mesh into six subtriangles (detailed later), whereas the meshes of [13] use rectangular meshes and subdivide each rectangle into four triangles. This latter meshing approach may be less convenient for some applications. Besides the geometry, the method of this paper allows for a local macro-element assembly requiring only the six local triangles of the Powell-Sabin split at once, grouping nodal basis functions for four nodes. The method of [13] refers to macro-element basis functions over a nine-rectangle grid of four triangles per rectangle, hence thirty-six triangles, and ultimately groups thirteen nodal functions together.

We summarize the paper contents as follows. Details of the Powell-Sabin mesh and finite element method for Stokes are given in Section 2, along with some preliminary technical lemmas. This includes a discussion about handling Dirichlet boundary conditions. In Section 3 we construct a solenoidal basis for the finite element velocity space with local support that can accommodate Dirichlet conditions. The properties are proved, and we include details of the local construction for implementation. In Section 4 we construct a local basis of the pressure space such that each basis function is the divergence of a known basis function in the (non-solenoidal) discrete velocity space, with proof. This pressure basis requires a subset of vertices to be marked using a graph-theoretic *spanning tree*, for which purpose there is already an inexpensive algorithm due to Kruskal [9]. The pressure basis is fairly simple to implement once this is done. Computational examples are given in Section 5 comparing the velocity and pressure computations to the classical saddle-point system, the latter using the usual non-solenoidal basis for velocity. A summary discussion is provided in Section 6.

2. A finite element method on Powell-Sabin splits

This section provides some mathematical preliminaries and then outlines the finite element spaces studied, focusing on a specific instance from amongst those discussed in [4].

2.1. Preliminaries. Given any countable set S, we denote by $|S|$ its cardinality, whereas for sets $S \subset \Omega$, $|S|$ denotes the geometric area. Also, for $S \subset \Omega$ we refer to the function spaces

$$\begin{aligned}
L^2(S) &:= \left\{ f : S \to \mathbb{R} \,|\, \|f\|^2 = \int_S |f|^2 \, d\mathbf{x} < \infty \right\}, \\
L^2_0(S) &:= \left\{ f \in L^2(S) \,|\, \int_S f \, d\mathbf{x} = 0 \right\}, \\
H^k(S) &:= \left\{ f : S \to \mathbb{R} \,|\, \|f\|^2_k = \sum_{|\alpha| \le k} \int_S |D^\alpha f|^2 \, d\mathbf{x} < \infty \right\}, \\
H^k_0(S) &:= \left\{ f \in H^k(S) \,|\, D^\alpha f|_{\partial S} = 0, \ \forall \alpha \, \text{with} \, |\alpha| < k \right\}.
\end{aligned}$$

Here, $\mathbf{x} = (x, y)$ are coordinates in $S \subset \Omega$. Generally, vectors and their components have the form $\mathbf{f} = (f_1, f_2)$. Vector-valued spaces use bold font as well:

$$\mathbf{L}^2(S) = \left[L^2(S)\right]^2, \ \mathbf{H}^k(S) = \left[H^k(S)\right]^2, \ \text{and} \ \mathbf{H}^k_0(S) = \left[H^k_0(S)\right]^2.$$

In order to treat the Dirichlet boundary conditions (3), let $\gamma_0 : \mathbf{H}^1(\Omega) \to \mathbf{H}^{1/2}(\partial\Omega)$ be the classical trace operator. We also use the divergence-free subspaces

$$\begin{aligned}
\mathbf{V}(S) &:= \left\{ \mathbf{v} \in \mathbf{H}^1(S) \,|\, \nabla \cdot \mathbf{v} = 0 \, \text{on} \, S \right\}, \\
\mathbf{V}_0(S) &:= \left\{ \mathbf{v} \in \mathbf{H}^1_0(S) \,|\, \nabla \cdot \mathbf{v} = 0 \, \text{on} \, S \right\}.
\end{aligned}$$

Dot-product and inner-product notations:

$$\begin{aligned}
\mathbf{f} \cdot \mathbf{g} &= f_1 g_1 + f_2 g_2, \ \forall \mathbf{f} \in \mathbb{R}^2, \ \mathbf{g} \in \mathbb{R}^2, \\
(f, g) &= \int_\Omega f(\mathbf{x}) g(\mathbf{x}) \, d\mathbf{x}, \ \forall f \in L^2(\Omega), \ g \in L^2(\Omega), \\
(\mathbf{f}, \mathbf{g}) &= \int_\Omega \mathbf{f}(\mathbf{x}) \cdot \mathbf{g}(\mathbf{x}) \, d\mathbf{x}, \ \forall \mathbf{f} \in \mathbf{L}^2(\Omega), \ \mathbf{g} \in \mathbf{L}^2(\Omega), \\
\nabla \mathbf{f}(\mathbf{x}) : \nabla \mathbf{g}(\mathbf{x}) &= \sum_{i=1,2} \nabla f_i(\mathbf{x}) \cdot \nabla g_i(\mathbf{x}), \ \forall \mathbf{f} \in \mathbf{H}^1(\Omega), \ \mathbf{g} \in \mathbf{H}^1(\Omega), \\
(\nabla \mathbf{f}, \nabla \mathbf{g}) &= \int_\Omega \nabla \mathbf{f}(\mathbf{x}) : \nabla \mathbf{g}(\mathbf{x}) \, d\mathbf{x}, \ \forall \mathbf{f} \in \mathbf{H}^1(\Omega), \ \mathbf{g} \in \mathbf{H}^1(\Omega).
\end{aligned}$$

2.2. Finite element method. Let $\tau_h^M = \{T_k^M\}$ be a family of conforming triangulations of Ω, denoting the triangular mesh elements by T_k for $1 \le k \le |\tau_h^M|$. The domain is polygonal (Section 1), and we assume the triangulation represents the domain boundary exactly. Here, $h = \max_k\{h_k\}$ where $h_k = \operatorname{diam}(T_k^M)$ is the maximum diameter of a triangle in the mesh. The superscript M is used to distinguish this *macro-mesh* from its Powell-Sabin split, the latter denoted here by τ_h. The mesh τ_h is formed via a subdivision of each triangle T_k^M into six smaller triangles.

The Powell-Sabin mesh τ_h is constructed via the following procedure. Find the incenter of each T_k^M. This is connected via line segments to the vertices of T_k^M. Given any two adjacent triangles T_k^M and T_j^M, their incenters are then connected via a line segment, which necessarily intersects their shared edge. If a triangle T_k^M has an exterior edge (on $\partial\Omega$), the midpoint of the exterior edge is connected via a line segment to the incenter. An important feature of the Powell-Sabin split is the creation of *singular points* in the mesh. A singular point is any vertex such that all edges meeting at that vertex fall on precisely two straight lines. By construction, the mesh τ_h has precisely one singular vertex along each edge of τ_h^M, and no others.

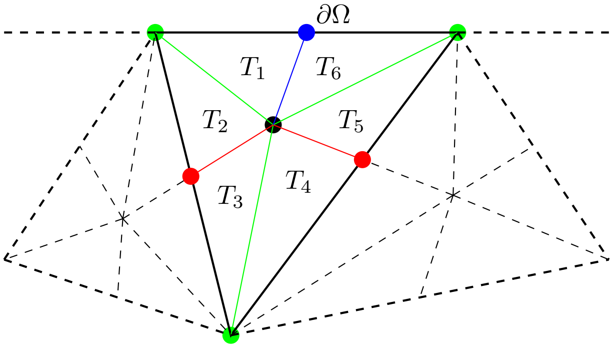

FIGURE 1. The Powell-Sabin split illustrated for a triangle of the macro-mesh τ_h^M. The incenter $\mathbf{z}$ (black dot) is connected to the vertices (green dots, edges) and neighboring incenters (red edges). In case of an edge on the boundary, the incenter is connected to the midpoint of that edge (blue edge, dot). The split results in six triangles, $\tau_{\mathbf{z}} = \{T_1, \dots, T_6\}$, and a set of singular vertices in correspondence with the edges of τ_h^M (red and blue dots).

Given a vertex $\mathbf{z}$, it is convenient for us to use a local indexing for the set of triangles that share this vertex. Such a set is denoted by

$$\tau_{\mathbf{z}}^M = \left\{T_1^M, \dots, T_{|\tau_{\mathbf{z}}^M|}^M\right\} \subset \tau_h^M$$

for triangles in the macro-mesh, or else

$$\tau_{\mathbf{z}} = \left\{T_1, \dots, T_{|\tau_{\mathbf{z}}|}\right\} \subset \tau_h$$

for the Powell-Sabin mesh. Throughout this paper we assume an ordering of the local triangles so that T_j^M and T_{j+1}^M (or T_j and T_{j+1}) share an edge. Moreover, we assume that a point rotated counter-clockwise around $\mathbf{z}$ that crosses the shared edge will move out of triangle j and into $j+1$. Graphically, the Powell-Sabin splitting is summarized in Figure 1, along with an example of the set $\tau_{\mathbf{z}}$ with $\mathbf{z}$ an incenter of a triangle in τ_h^M.

Some notation of use follows here, where an edge or vertex is called *exterior* if it lies on $\partial\Omega$, otherwise it is *interior*.

- E_{int}^M (and E_{ext}^M) – interior (and exterior) edges of τ_h^M
- V_{int}^M (and V_{ext}^M) – interior (and exterior) vertices of τ_h^M
- S_{int} (and S_{ext}) – interior (and exterior) singular vertices of τ_h
- $E^M = E_{int}^M \cup E_{ext}^M$, $V^M = V_{int}^M \cup V_{ext}^M$, $S = S_{int} \cup S_{ext}$,

Note the cardinality relationships

$$|E_{int}^M| = |S_{int}| \quad \text{and} \quad |E_{ext}^M| = |S_{ext}|. \tag{5}$$

Given any region $S \subset \Omega$, let $\mathbb{P}_k(S)$ be the space of polynomials of order k or less on S, with corresponding vector-valued space $\mathbf{P}_k(S) = [\mathbb{P}_k(S)]^2$. Spaces needed to discuss the discrete velocity are

$$\begin{aligned}
\mathbf{X}_h &= \left\{\mathbf{v}_h \in [\mathcal{C}(\overline{\Omega})]^2 \mid \mathbf{v}_h \in \mathbf{P}_1(T),\ \forall T \in \tau_h\right\} \subset \mathbf{H}^1(\Omega),\\
\mathbf{X}_h^0 &= \{\mathbf{v}_h \in \mathbf{X}_h \mid \mathbf{v}_h = \mathbf{0} \text{ on } \partial\Omega\} \subset \mathbf{H}_0^1(\Omega),\\
\mathbf{V}_h &= \mathbf{X}_h \cap \mathbf{V}(\Omega),\\
\mathbf{V}_h^0 &= \mathbf{V}_h \cap \mathbf{V}_0(\Omega).
\end{aligned}$$

The corresponding pressure space is identified after observing a key property of the divergence of the discrete velocity (*cf.* [4]; see [14] for a proof).

Lemma 1. *Let $q \in \mathcal{C}(T)$ for all $T \in \tau_h$ and define a function*

$$\theta_{\boldsymbol{z}}(q) := \begin{cases} q|_{T_1}(\boldsymbol{z}) - q|_{T_2}(\boldsymbol{z}) + q|_{T_3}(\boldsymbol{z}) - q|_{T_4}(\boldsymbol{z}), & \text{if } \boldsymbol{z} \in S_{int}, \\ q|_{T_1}(\boldsymbol{z}) - q|_{T_2}(\boldsymbol{z}), & \text{if } \boldsymbol{z} \in S_{ext}, \end{cases}$$

where $\tau_{\boldsymbol{z}} = \{T_1, \ldots, T_4\}$ if $\boldsymbol{z} \in S_{int}$ or $\tau_{\boldsymbol{z}} = \{T_1, T_2\}$ if $\boldsymbol{z} \in S_{ext}$. Then for any $\boldsymbol{v}_h \in \boldsymbol{X}_h^0$ and singular vertex $\boldsymbol{z} \in S$ there holds $\theta_{\boldsymbol{z}}(\nabla \cdot \boldsymbol{v}_h) = 0$.

The pressure space is

$$P_h = \left\{ q_h \in L_0^2(\Omega) \mid q_h|_T \in \mathbb{P}_0(T),\, \forall T \in \tau_h,\, \theta_{\mathbf{z}}(q_h) = 0,\, \forall \mathbf{z} \in S \right\}.$$

Before the finite element problem can be stated formally, we must address a critical property of the discrete, solenoidal velocity space $\mathbf{V}_h$. This affects the boundary conditions and is also referenced later to construct a solenoidal basis in Section 3. First, we require a technical lemma.

Lemma 2. *Let $T \subset \mathbb{R}^2$ be a triangle with vertices $\boldsymbol{z}_i$, $1 \le i \le 3$, ordered counterclockwise going from $\boldsymbol{z}_i$ to $\boldsymbol{z}_{i+1}$ along ∂T. Each triangle edge, e_i, lies opposite the vertex $\boldsymbol{z}_i$, with outward unit normal vector $\boldsymbol{n}_i$. Note the identity*

$$\sum_{1 \le i \le 3} |e_i| \boldsymbol{n}_i = \mathbf{0}. \tag{6}$$

Given any $\boldsymbol{v} \in \boldsymbol{P}_1(T)$, it holds that

$$\int_T \nabla \cdot \boldsymbol{v}\, d\boldsymbol{x} = -\frac{1}{2} \sum_{1 \le i \le 3} |e_i| \boldsymbol{v}(\boldsymbol{z}_i) \cdot \boldsymbol{n}_i. \tag{7}$$

Remark 1. *The negative sign on the right side of* (7) *is a result of the labeling of edge e_i as being opposite the vertex $\boldsymbol{z}_i$.*

Proof. Let $\mathcal{A} : \mathbb{R}^2 \to \mathbb{R}^2$ be the linear operator that rotates a vector clockwise 90 degrees. Then

$$\sum_{1 \le i \le 3} |e_i| \mathbf{n}_i = |e_1| \mathcal{A}\left(\frac{\mathbf{z}_3 - \mathbf{z}_2}{|e_1|}\right) + |e_2| \mathcal{A}\left(\frac{\mathbf{z}_1 - \mathbf{z}_3}{|e_2|}\right) + |e_3| \mathcal{A}\left(\frac{\mathbf{z}_2 - \mathbf{z}_1}{|e_3|}\right) = \mathbf{0}.$$

Then (7) follows from (6) after applying the divergence form of Green's Theorem. □

We will use ds for a measure along edges.

Lemma 3. *Let $e^M \in E^M$ have endpoints $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$. The trace of any $\boldsymbol{v}_h \in \boldsymbol{V}_h$ along e^M may be uniquely determined from the values*

$$\boldsymbol{v}_h(\boldsymbol{z}_1),\ \boldsymbol{v}_h(\boldsymbol{z}_2),\ \text{and} \int_{e^M} \boldsymbol{v}_h \cdot \boldsymbol{n}\, ds, \tag{8}$$

where $\boldsymbol{n}$ is a unit normal along e^M.

Proof. Refer to Figure 2 to visualize the vertices, edges and normal vectors used in this proof. Let $\mathbf{z}_s \in S$ be the singular vertex of τ_h on e^M. Since $\mathbf{v}_h$ is piecewise linear and continuous along e^M, it suffices to show that if the values in (8) are all zero, then also $\mathbf{v}_h(\mathbf{z}_s) = \mathbf{0}$. Direct integration shows that if $\mathbf{v}_h(\mathbf{z}_i) = \mathbf{0}$, $i = 1, 2$, then

$$\int_{e^M} \mathbf{v}_h \cdot \mathbf{n}\, ds = 0 \Rightarrow \mathbf{v}_h(\mathbf{z}_s) \cdot \mathbf{n} = 0. \tag{9}$$

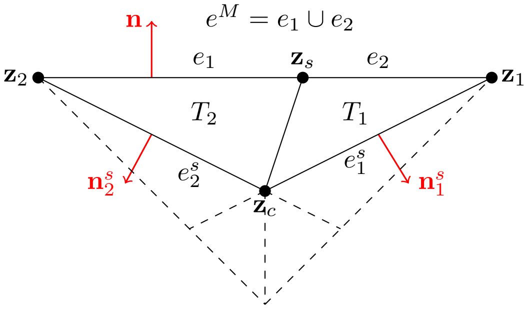

FIGURE 2. Labels referenced in the proof of Lemma 3.

It remains only to show that the tangential component is zero; equivalently $\mathbf{v}_h(\mathbf{z}_s) \cdot (\mathbf{z}_2 - \mathbf{z}_1) = 0$. Let $\mathbf{z}_c$ be the incenter of a triangle in τ_h^M that has edge e^M. There are two triangles, say T_1 and T_2, in $\tau_{\mathbf{z}_c}$ with edges e_1 and e_2, respectively, that lie on e^M, opposite $\mathbf{z}_c$. They have a shared edge lying opposite $\mathbf{z}_i$ on T_i. Let the edge of T_i opposite $\mathbf{z}_s$ be e_i^s and have outward unit normal $\mathbf{n}_i^s$ for $i = 1, 2$.

Since $\mathbf{v}_h$ is solenoidal, we may apply (7) on each triangle T_i to find

$$|e_1^s|\mathbf{v}_h(\mathbf{z}_s) \cdot \mathbf{n}_1^s + |e_1|\mathbf{v}_h(\mathbf{z}_c) \cdot \mathbf{n} = 0 \tag{10}$$

$$\text{and} \quad |e_2^s|\mathbf{v}_h(\mathbf{z}_s) \cdot \mathbf{n}_2^s + |e_2|\mathbf{v}_h(\mathbf{z}_c) \cdot \mathbf{n} = 0. \tag{11}$$

Note that edges e_1^s, e_2^s and e^M form a perimeter of triangle $T_1 \cup T_2$, hence the vectors $\mathbf{n}_1^s$ and $\mathbf{n}_2^s$ are equivalent to a certain basis of $\mathbb{R}^2$. Then (10)-(11) show that $\mathbf{v}_h(\mathbf{z}_s) = \mathbf{0}$ if $\mathbf{v}_h(\mathbf{z}_c) \cdot \mathbf{n} = 0$. Indeed, now sum (10)-(11) and apply (6), (9) to show that

$$\begin{aligned}(|e_1| + |e_2|)\mathbf{v}_h(\mathbf{z}_c) \cdot \mathbf{n} = -|e_1^s|\mathbf{v}_h(\mathbf{z}_s) \cdot \mathbf{n}_1^s - |e_2^s|\mathbf{v}_h(\mathbf{z}_s) \cdot \mathbf{n}_2^s \\ = (|e_1| + |e_2|)\mathbf{v}_h(\mathbf{z}_s) \cdot \mathbf{n} = 0.\end{aligned}$$

□

A consequence of Lemma 3 is that we cannot generally interpolate boundary data $\mathbf{g} \in [\mathcal{C}(\partial\Omega)]^2$ to compute a solenoidal Stokes velocity $\mathbf{u}_h \in \mathbf{V}_h$ by setting $\mathbf{u}_h(\mathbf{z}) = \mathbf{g}(\mathbf{z})$ at all the exterior nodes $\mathbf{z}$ of the Powell-Sabin mesh. The following definition provides an appropriate interpolant, where the function $\mathbf{G}_h$ will be explicitly constructed in Section 3.

Definition 1 (Boundary data interpolation)**.** *Let $\boldsymbol{g} \in \boldsymbol{H}^{1/2}(\partial\Omega) \cap [\mathcal{C}(\partial\Omega)]^2$. We denote by $\boldsymbol{g}_h$ the interpolant of $\boldsymbol{g}$, where $\boldsymbol{g}_h = \gamma_0(\boldsymbol{G}_h)$ for some $\boldsymbol{G}_h \in \boldsymbol{V}_h$ and*

$$\boldsymbol{g}_h(\boldsymbol{z}) = \boldsymbol{g}(\boldsymbol{z}), \ \forall \boldsymbol{z} \in V_{ext}^M \tag{12}$$

$$\textit{and} \quad \int_e \boldsymbol{g}_h \cdot \boldsymbol{n}\, ds = \int_e \boldsymbol{g} \cdot \boldsymbol{n}\, ds, \ \forall e \in E_{ext}^M, \tag{13}$$

where $\boldsymbol{n}$ is the outward unit normal vector on $\partial\Omega$.

We may now state the finite element problem. We solve for $\mathbf{u}_h \in \mathbf{X}_h$ and $p_h \in P_h$ by decomposing $\mathbf{u}_h = \mathbf{w}_h + \mathbf{G}_h$, $\mathbf{G}_h \in \mathbf{V}_h$ as in Definition 1 and $\mathbf{w}_h \in \mathbf{X}_h^0$ such that

$$a(\mathbf{w}_h, \mathbf{v}_h) - (p_h, \nabla \cdot \mathbf{v}_h) = L(\mathbf{v}_h),\ \forall \mathbf{v}_h \in \mathbf{X}_h^0, \tag{14}$$

$$(\nabla \cdot \mathbf{w}_h, q_h) = 0,\ \forall q_h \in P_h, \tag{15}$$

$$a(\mathbf{w}_h, \mathbf{v}_h) := \nu\left(\nabla \mathbf{w}_h, \nabla \mathbf{v}_h\right),$$

$$L(\mathbf{v}_h) := (\mathbf{f}, \mathbf{v}_h) - a(\mathbf{G}_h, \mathbf{v}_h).$$

However, it is shown in [4] that the solution of (14)-(15) will in fact have $\mathbf{w}_h \in \mathbf{V}_h^0$. Thus $\mathbf{u}_h \in \mathbf{V}_h$, and we focus in this paper on directly solving the problem on the solenoidal subspace; find $\mathbf{w}_h \in \mathbf{V}_h^0$ satisfying

$$a(\mathbf{w}_h, \mathbf{v}_h) = L(\mathbf{v}_h),\ \forall \mathbf{v}_h \in \mathbf{V}_h^0. \tag{16}$$

This is accomplished in Section 3. Note that the pressure variable need not be computed, but if it is desired we also provide a way to compute it after the velocity is found. One solves the problem: find $p_h \in P_h$ such that

$$(p_h, \nabla \cdot \mathbf{v}_h) = a(\mathbf{u}_h, \mathbf{v}_h) - (\mathbf{f}, \mathbf{v}_h),\ \forall \mathbf{v}_h \in \mathbf{X}_h^0 \setminus \mathbf{V}_h^0. \tag{17}$$

This method is detailed in Section 4.

3. A solenoidal velocity basis

Explicit constructions of basis functions are provided here. We also show how to implement boundary conditions, and a localized implementation of the basis using a macro-element technique.

3.1. Bases of $\mathbf{V}_h$ and $\mathbf{V}_h^0$. Given a vertex $\mathbf{z} \in V^M$, the support of each basis function will be $\tau_{\mathbf{z}}^M$. Consider the subspaces with this support:

$$\mathbf{V}_{\mathbf{z}} = \left\{\mathbf{v}_h \in \mathbf{V}_h \mid \mathbf{v}_h(\mathbf{x}) = \mathbf{0}\ \forall \mathbf{x} \in T^M,\ \forall T^M \subset \tau_h^M \setminus \tau_{\mathbf{z}}^M\right\}. \tag{18}$$

We will first construct a basis of $\mathbf{V}_{\mathbf{z}}$. A certain union of these across vertices $\mathbf{z}$ will later form the desired basis for $\mathbf{V}_h$.

Lemma 4. *Given any $\boldsymbol{z} \in V^M$, it holds that $dim(\boldsymbol{V_z}) = 3$. Moreover, a basis $\{\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2, \boldsymbol{\Phi}_3\}$ is found by taking $(\alpha_1, \beta_1, \delta_1) = (1,0,0)$, $(\alpha_2, \beta_2, \delta_2) = (0,1,0)$, $(\alpha_3, \beta_3, \delta_3) = (0,0,1)$, and setting*

$$\boldsymbol{\Phi}_i(\boldsymbol{z}) = (\alpha_i, \beta_i) \tag{19}$$

$$\textit{and} \quad \int_{e_j^M} \boldsymbol{\Phi}_i \cdot \boldsymbol{n}_j \, ds = \delta_i, \tag{20}$$

for $1 \leq i \leq 3$, where, for each $T_j^M \in \tau_{\boldsymbol{z}}^M$, $\{e_{j-1}^M, e_j^M\} \subset E^M$ are its edges that have $\boldsymbol{z}$ as one endpoint, and $\boldsymbol{n}_j$ is the unit normal on e_j^M that points in the direction of counter-clockwise rotation around $\boldsymbol{z}$.

Proof. Note that we must have

$$0 = \int_{T_j^M} \nabla \cdot \boldsymbol{\Phi}_i \, d\mathbf{x} = \pm\left(\int_{e_j^M} \boldsymbol{\Phi}_i \cdot \mathbf{n}_j \, ds - \int_{e_{j-1}^M} \boldsymbol{\Phi}_i \cdot \mathbf{n}_{j-1} \, ds\right),$$

so that the value of δ_i is j-independent in (20). Since $\mathbf{z}$ is the common vertex of all $T_j^M \in \tau_{\mathbf{z}}^M$, the conditions (19)-(20) impose the same constraints on each T_j^M, so it suffices to look at the restriction to one T_j^M. Denote by $\mathbf{z}_c$ the incenter of T_j^M and e^M the edge that lies opposite $\mathbf{z}$.

Consider the vector space

$$\mathbf{W}_j = \left\{\mathbf{v}_h \in \mathcal{C}(T_j^M) \mid \mathbf{v}_h \in \mathbf{P}_1(T), \forall T \in \tau_{\mathbf{z}_c},\ \text{and } \mathbf{v}_h = \mathbf{0} \text{ on } e^M\right\}.$$

The six triangles in $\tau_{\mathbf{z}_c}$ together have seven distinct vertices (see *e.g.* Figure 1, with $\mathbf{v}_h = \mathbf{0}$ at the three vertices on e^M. Any $\mathbf{v}_h \in \mathbf{W}_j$ is uniquely determined by its values at the other four vertices, so $\dim(\mathbf{W}_j) = 8$. Two triangles $T_i \in \tau_{\mathbf{z}_c}$ have an edge lying on e^M, say T_1 and T_2 (see Figure 2). Eight linear constraints may be introduced by requiring

$$\mathbf{v}_h(\mathbf{z}) = (\alpha, \beta),$$
$$\int_{e_j^M} \mathbf{v}_h \cdot \mathbf{n}_j \, ds = \delta,$$

for any desired values α, β and δ, and also setting $\nabla \cdot \mathbf{v}_h = 0$ on triangles T_i, $2 \le i \le 6$. In fact, then $\nabla \cdot \mathbf{v}_h = \mathbf{0}$ will hold also on T_1. This is verified as follows. Note that T_1 and T_2 each have two vertices on e^M, where $\mathbf{v}_h = \mathbf{0}$, and their third vertex is $\mathbf{z}_c$. Apply Lemma 2 on T_2 (where we impose $\nabla \cdot \mathbf{v}_h = \mathbf{0}$) first, which reduces to

$$0 = \mathbf{v}_h(\mathbf{z}_c) \cdot \mathbf{n},$$

where $\mathbf{n}$ is the outward unit normal for T_j^M along e^M. Application of Lemma 2 on T_1 shows that $\nabla \cdot \mathbf{v}_h$ is proportional to $\mathbf{v}_h(\mathbf{z}_c) \cdot \mathbf{n}$, thus $\nabla \cdot \mathbf{v}_h = \mathbf{0}$ on T_1.

The proof reduces to showing that if $\alpha = \beta = \delta = 0$ then $\mathbf{v}_h = \mathbf{0}$ at all seven vertices. We already have $\mathbf{v}_h = \mathbf{0}$ at the three vertices on e^M, plus at $\mathbf{z}$ as well under these conditions. Since $\delta = 0$, we may apply Lemma 3 to conclude that $\mathbf{v}_h = \mathbf{0}$ along the other two edges of the macro-triangle T_j^M, hence at the two singular vertices on those edges, leaving only the incenter vertex $\mathbf{z}_c$. Above we showed $\mathbf{v}_h(\mathbf{z}_c) \cdot \mathbf{n} = 0$; repeating the arguments (apply Lemma 2) on any of the triangles T_i with $3 \le i \le 6$ shows that $\mathbf{v}_h(\mathbf{z}_c) \cdot \mathbf{n}^* = 0$ for $\mathbf{n}^* = \mathbf{n}_j$ or $\mathbf{n}^* = \mathbf{n}_{j-1}$. Since $\mathbf{n}$ and $\mathbf{n}^*$ cannot be parallel, it follows $\mathbf{v}_h(\mathbf{z}_c) = 0$. □

The solenoidal basis is discussed in the next result. It requires the exclusion of a single function to form a basis, but the association made with the boundary by choosing a vertex $\mathbf{z}_0 \in V_{ext}^M$ is for convenience in implementing boundary conditions; see Section 3.2. Also, in Section 4 a single boundary vertex must be chosen to extract a special basis for the pressure space, so in practice it is natural to use the same vertex $\mathbf{z}_0$.

Theorem 1. *Given an indexing $\boldsymbol{z}_k \in V^M$, $0 \le k \le |V^M| - 1$ such that $\boldsymbol{z}_0 \in V_{ext}^M$ (lies on $\partial\Omega$), we denote by $\boldsymbol{\Phi}_{\boldsymbol{i}}^{(\boldsymbol{k})}$, $1 \le i \le 3$, the basis functions of $\boldsymbol{V}_{\boldsymbol{z}}$ defined in Lemma 4 for $\boldsymbol{z} = \boldsymbol{z}_k$. We define sets*

$$\mathcal{B} = \left[\bigcup_{i=1}^{3} \bigcup_{k=0}^{|V^M|-1} \boldsymbol{\Phi}_{\boldsymbol{i}}^{(\boldsymbol{k})} \right] \setminus \left\{ \boldsymbol{\Phi}_{\mathbf{3}}^{(\mathbf{0})} \right\},$$
$$\mathcal{B}^0 = \left\{ \boldsymbol{\Phi}_{\boldsymbol{i}}^{(\boldsymbol{k})} \in \mathcal{B} \,|\, \boldsymbol{z}_k \in V_{int}^M \right\}.$$

It holds that $\mathcal{B}$ is a basis of $\mathbf{V}_h$ and $\mathcal{B}^0$ is a basis of $\mathbf{V}_h^0$.

Proof. We prove that $\mathcal{B}$ is a basis of $\mathbf{V}_h$ by showing that $\dim(\mathbf{V}_h) = 3|V^M| - 1$ along with the linear independence of $\mathcal{B}$. The former claim is a counting argument. Since $\mathbf{V}_h$ is the kernel of the divergence operator on $\mathbf{X}_h$, we may use the rank-nullity theorem. It is proved in ([6], Theorem 4) that $\dim(\nabla \cdot \mathbf{X}_h) = |E^M| + 3|\tau_h^M|$.

Thus

$$\begin{aligned}(21)\quad \dim(\mathbf{V}_h) = \dim(\mathbf{X}_h) - |E^M| - 3|\tau_h^M| &= 2|V| - |E^M| - 3|\tau_h^M| \\ &= 2(|V^M| + |E^M| + |\tau_h^M|) - |E^M| - 3|\tau_h^M| \\ &= 2|V^M| + |E^M| - |\tau_h^M|,\end{aligned}$$

and we apply the Euler identity $|E^M| - |\tau_h^M| = |V^M| - 1$ to conclude $\dim(\mathbf{V}_h) = 3|V^M| - 1 = |\mathcal{B}|$. Next, assume $\mathbf{v} \in \operatorname{span}(\mathcal{B})$ satisfies

$$\mathbf{v} = \sum_{i=1}^{3} \sum_{k=0}^{|V^M|-1} c_i^{(k)} \mathbf{\Phi}_i^{(k)} = \mathbf{0} = (0,0),$$

where $c_3^{(0)} = 0$ (recall this basis function is excluded from $\mathcal{B}$). By construction, we have at the vertices in $\mathbf{z}_j \in V^M$

$$\mathbf{v}(\mathbf{z}_j) = (c_1^{(j)}, c_2^{(j)}) = (0,0),\ 0 \le j \le |V^M| - 1,$$

and we may reduce

$$\mathbf{v} = \sum_{k=0}^{|V^M|-1} c_3^{(k)} \mathbf{\Phi}_3^{(k)} = \mathbf{0} = (0,0).$$

Note that each coefficient $c_3^{(k)}$ may be associated with a point $\mathbf{z}_k \in V^M$; these are joined by the edges in E^M to form a connected graph. Given any edge $e \in E^M$ with unit normal $\mathbf{n}$ and endpoints $\mathbf{z}_n$ and $\mathbf{z}_m$, we have

$$\int_e \mathbf{v} \cdot \mathbf{n}\, ds = \pm \left(c_3^{(n)} - c_3^{(m)} \right) = 0,$$

and therefore $c_3^{(n)} = c_3^{(m)}$ holds for all values n, m. Since $c_3^{(0)} = 0$, we have $c_3^{(k)} = 0$ for all k. This proves linear independence of $\mathcal{B}$. Finally, the fact that $\mathcal{B}^0$ is a basis of $\mathbf{V}_h^0$ now follows by simply restricting the boundary values to zero. □

3.2. Implementation details. The correct representation of Dirichlet conditions is now easy to show. Recall the decomposition $\mathbf{u}_h = \mathbf{G}_h + \mathbf{w}_h$, where $\mathbf{w}_h \in \mathbf{V}_h^0$ solves (16) and $\mathbf{G}_h \in \mathbf{V}_h$ satisfies Definition 1. The component $\mathbf{w}_h$ is computed using the basis $\mathcal{B}^0$. We may expand out $\mathbf{G}_h$ using $\mathcal{B} \setminus \mathcal{B}^0$; for convenience in exposition, assume the exterior edge ordering $e_k^M \in E_{ext}^M$, $1 \le k \le |E_{ext}^M|$, such that e_k^M and e_{k+1}^M share a vertex. This supports a vertex labelling $\mathbf{z}_k \in V^M$, $0 \le k \le |V^M| - 1$, such that each edge $e_k^M \in E_{ext}^M$ has endpoints $\mathbf{z}_{k-1}$ and $\mathbf{z}_k$.

We may expand out

$$\mathbf{G}_h = \sum_{i=1}^{3} \sum_{k=0}^{|V_{ext}^M|-1} c_i^{(k)} \mathbf{\Phi}_i^{(k)}.$$

Boundary values are still interpolated at the points $\mathbf{z}_k \in V_{ext}^M$, corresponding to (12):

$$\mathbf{G}_h(\mathbf{z}_k) = (c_1^{(k)}, c_2^{(k)}) = \mathbf{g}(\mathbf{z}_k),\ 0 \le k \le |V_{ext}^M| - 1.$$

At split points in the Powell-Sabin mesh along the boundary, the values of $\mathbf{G}_h$ are determined implicitly through the normal moments (13). Specifically, we have the system of equations

$$\int_{e_k^M} \mathbf{G}_h \cdot \mathbf{n}\, ds = \pm(c_3^{(k)} - c_3^{(k-1)}) = \int_{e_k^M} \mathbf{g} \cdot \mathbf{n}\, ds,\ 1 \le k \le |V_{ext}^M| - 1,$$

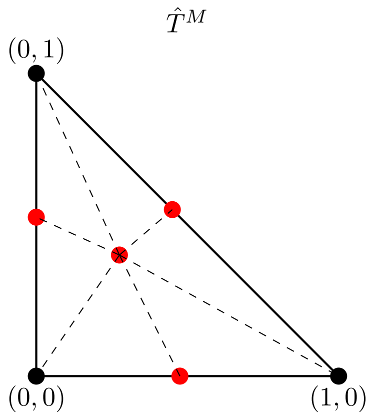

FIGURE 3. A localized construction of basis functions can be made explicit using a reference mapping $F_k : \hat{T}^M \to T_k^M$. The mapping F_k is defined as an affine function that maps the vertices of $\hat{T}^M$ to those of T_k^M (black). The additional points used to subdivide $\hat{T}^M$ are the preimages under F_k of the singular vertices and incenter on T_k^M (red).

where $\mathbf{n}$ is the outward unit normal on $\partial\Omega$ and the sign depends on whether the edges are ordered clockwise or counter-clockwise around the boundary. It is not necessary to include case $k = |V_{ext}^M| (= |E_{ext}^M|)$ here because the boundary data must already satisfy the compatability condition

$$\nabla \cdot \mathbf{u}_h = 0 \Rightarrow \int_{\partial\Omega} \mathbf{G}_h \cdot \mathbf{n}\, ds = 0.$$

We have $c_3^{(0)} = 0$ by convention in Theorem 1, resulting in a square, lower-triangular system of equations here that is trivial to solve.

A localized construction of basis functions is possible using a reference map. The reference triangle is denoted by $\hat{T}^M$, which has vertices $\hat{\mathbf{z}}_1 = (0,0)$, $\hat{\mathbf{z}}_2 = (1,0)$ and $\hat{\mathbf{z}}_3 = (0,1)$. Let $T_k^M \in \tau_{\mathbf{z}}^M$. We denote by $F_k : \hat{T}^M \to T_k^M$ an affine map onto T_k^M with Jacobian J_k. The reference triangle is then divided into six subtriangles, say $\hat{T}_i$ for $1 \leq i \leq 6$. This is done by adding four points that are the preimages under F_k of the singular nodes and incenter on T_k^M; see Figure 3.

Functions $\mathbf{v}_h \in \mathbf{V}_h$ will be represented locally on T_k^M by leveraging the Piola transformation

$$\mathbf{v}_h = \frac{1}{|\det(J_k)|} J_k \hat{\mathbf{v}}_h. \tag{22}$$

As shown in the proof of Lemma 4, $\mathbf{v}_h$ can be determined uniquely on T_k^M via three degrees of freedom using the basis (19)-(20). Given that F_k maps reference vertex $\hat{\mathbf{z}}_j$ to $\mathbf{z}$, we can associate three basis functions on the reference element, say $\hat{\mathbf{\Phi}}_i^{(j)}$, $1 \leq i \leq 3$, such that the divergence (in reference coordinates) is $\hat{\nabla} \cdot \hat{\mathbf{\Phi}}_i^{(j)} = 0$, with

$$\hat{\mathbf{\Phi}}_i^{(j)}(\hat{\mathbf{z}}_j) = (\hat{\alpha}_i, \hat{\beta}_i) \tag{23}$$

$$\text{and} \quad \int_{\hat{e}_j^M} \hat{\mathbf{\Phi}}_i^{(j)} \cdot \hat{\mathbf{n}}_j \, d\hat{s} = \hat{\delta}_i. \tag{24}$$

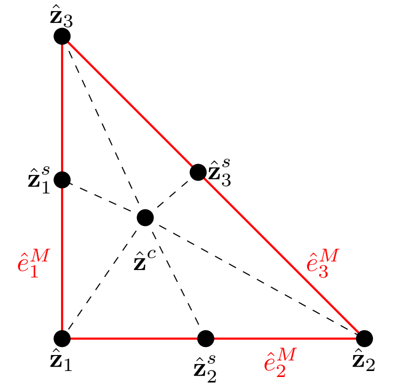

FIGURE 4. Labels for some edges (red) and points on $\hat{T}^M$.

Here, $\hat{e}_j^M$ is the edge of $\hat{T}^M$ between $\hat{\mathbf{z}}_{j-1}$ and $\hat{\mathbf{z}}_j$ (modulo 3), and $\hat{\mathbf{n}}_j$ is the outward-pointing unit normal vector along that edge (see Figure 4). We choose $(\hat{\alpha}_1, \hat{\beta}_1, \hat{\delta}_1) = (1, 0, 0)$, $(\hat{\alpha}_2, \hat{\beta}_2, \hat{\delta}_2) = (0, 1, 0)$, and $(\hat{\alpha}_3, \hat{\beta}_3, \hat{\delta}_3) = (0, 0, 1)$.

Since there are three such functions for each reference vertex, we have a total of nine local basis functions on $\hat{T}^M$. These correspond to the three vertices of T_k^M. However, care is needed to map between the basis functions when changing variables. Given the values

$$(a_i, b_i) = |\det(J_k)|(J_k)^{-1}\mathbf{\Phi}_i(\mathbf{z}),\ i = 1,\, 2,$$

the correct mappings are

$$\begin{aligned}
\mathbf{\Phi}_1|_{T_k^M} &= \frac{a_1}{|\det(J_k)|} J_k \hat{\mathbf{\Phi}}_1^{(j)} + \frac{b_1}{|\det(J_k)|} J_k \hat{\mathbf{\Phi}}_2^{(j)},\\
\mathbf{\Phi}_2|_{T_k^M} &= \frac{a_2}{|\det(J_k)|} J_k \hat{\mathbf{\Phi}}_1^{(j)} + \frac{b_2}{|\det(J_k)|} J_k \hat{\mathbf{\Phi}}_2^{(j)},\\
\text{and } \mathbf{\Phi}_3|_{T_k^M} &= \frac{1}{|\det(J_k)|} J_k \hat{\mathbf{\Phi}}_3^{(j)}.
\end{aligned}$$

The Piola transformation is well-known to preserve the divergence and the integral of the normal component along edges, so it is easy to verify the relationships above.

Let $\hat{\mathbf{z}}_i^s$, $1 \le i \le 3$, satisfy the requirements that $\hat{\mathbf{z}}_i^s$ lies on $\hat{e}_i^M$ and $\hat{\mathbf{z}}_i^s = (F_k)^{-1}(\mathbf{z}_i^s)$ for some singular vertex $\mathbf{z}_i^s \in S$ that lies on an edge of T_k^M. Also, let $\hat{\mathbf{z}}^c = (F_k)^{-1}(\mathbf{z}^c)$, where $\mathbf{z}^c$ is the incenter for T_k^M. Labels for points on $\hat{T}^M$ are shown in Figure 4. In order to describe each $\hat{\mathbf{\Phi}}_i^{(j)}$ explicitly, it suffices to list its values at the seven points $\hat{\mathbf{z}}_i$ and $\hat{\mathbf{z}}_i^s$, $1 \le i \le 3$, and $\hat{\mathbf{z}}^c$. Given that $\hat{\mathbf{\Phi}}_i^{(j)} = (\hat{u}_i^{(j)}, \hat{v}_i^{(j)})$, $\hat{\mathbf{z}}^c = (\hat{x}^c, \hat{y}^c)$ and $\hat{\mathbf{z}}_i^s = (\hat{x}_i^s, \hat{y}_i^s)$, these values are provided in Table 1 ($j = 1$), Table 2 ($j = 2$), and Table 3 ($j = 3$).

While the construction has been localized to T_k^M, it is important to note that the basis functions $\hat{\mathbf{\Phi}}_i^{(j)}$ depend on the locations of the singular vertices and incenter of T_k^M. In implementation, assembly may be performed by looping over triangles in τ_h^M with a subloop over the six subtriangles of the Powell-Sabin split, computing nodal values locally for all basis functions. In turn, once these nodal values are found, each $\mathbf{\Phi}_i$ has an obvious representation in the standard nodal basis used in continuous, piecewise linear finite element methods, so integration on each triangle

TABLE 1. Tabulated values of $(\hat{u}_i^{(1)}, \hat{v}_i^{(1)}) = \hat{\mathbf{\Phi}}_i^{(1)}$, $1 \leq i \leq 3$, at the points shown in Figure 4.

	$\hat{\mathbf{z}}^c$	$\hat{\mathbf{z}}_1^s$	$\hat{\mathbf{z}}_2^s$	$\hat{\mathbf{z}}_3^s$	$\hat{\mathbf{z}}_1$	$\hat{\mathbf{z}}_2$	$\hat{\mathbf{z}}_3$
$\hat{u}_1^{(1)}$	$-\hat{y}^c$	$-\hat{y}_1^s$	$1 - \hat{x}_2^s$	0	1	0	0
$\hat{v}_1^{(1)}$	$\hat{y}^c$	$\frac{\hat{y}_1^s - \hat{y}^c}{\hat{x}^c}$	0	0	0	0	0
$\hat{u}_2^{(1)}$	$\hat{x}^c$	0	$\frac{\hat{x}_2^s - \hat{x}^c}{\hat{y}^c}$	0	0	0	0
$\hat{v}_2^{(1)}$	$-\hat{x}^c$	$1 - \hat{y}_1^s$	$-\hat{x}_2^s$	0	1	0	0
$\hat{u}_3^{(1)}$	-2	-2	$2\frac{\hat{x}^c - \hat{x}_2^s}{\hat{y}^c}$	0	0	0	0
$\hat{v}_3^{(1)}$	2	$2\frac{\hat{y}_1^s - \hat{y}^c}{\hat{x}^c}$	2	0	0	0	0

TABLE 2. Tabulated values of $(\hat{u}_i^{(2)}, \hat{v}_i^{(2)}) = \hat{\mathbf{\Phi}}_i^{(2)}$, $1 \leq i \leq 3$, at the points shown in Figure 4.

	$\hat{\mathbf{z}}^c$	$\hat{\mathbf{z}}_1^s$	$\hat{\mathbf{z}}_2^s$	$\hat{\mathbf{z}}_3^s$	$\hat{\mathbf{z}}_1$	$\hat{\mathbf{z}}_2$	$\hat{\mathbf{z}}_3$
$\hat{u}_1^{(1)}$	0	0	$\hat{x}_2^s$	$\hat{x}_3^s + \frac{\hat{x}^c - \hat{x}_3^s}{1 - \hat{x}^c - \hat{y}^c}$	0	1	0
$\hat{v}_1^{(1)}$	$-\hat{y}^c$	0	0	$\frac{\hat{y}^c - \hat{y}_3^s}{1 - \hat{x}^c - \hat{y}^c}$	0	0	0
$\hat{u}_2^{(1)}$	0	0	$\frac{\hat{x}_2^s - \hat{x}^c}{\hat{y}^c}$	$\frac{\hat{x}^c - \hat{x}_3^s}{1 - \hat{x}^c - \hat{y}^c}$	0	0	0
$\hat{v}_2^{(1)}$	$\hat{x}^c - 1$	0	$\hat{x}_2^s - 1$	$\hat{x}_3^s + \frac{\hat{y}^c - \hat{y}_3^s}{1 - \hat{x}^c - \hat{y}^c}$	0	1	0
$\hat{u}_3^{(1)}$	0	0	$2\frac{\hat{x}_2^s - \hat{x}^c}{\hat{y}^c}$	$2\frac{\hat{x}^c - \hat{x}_3^s}{1 - \hat{x}^c - \hat{y}^c}$	0	0	0
$\hat{v}_3^{(1)}$	-2	0	-2	$2\frac{\hat{y}^c - \hat{y}_3^s}{1 - \hat{x}^c - \hat{y}^c}$	0	0	0

TABLE 3. Tabulated values of $(\hat{u}_i^{(3)}, \hat{v}_i^{(3)}) = \hat{\mathbf{\Phi}}_i^{(3)}$, $1 \leq i \leq 3$, at the points shown in Figure 4.

	$\hat{\mathbf{z}}^c$	$\hat{\mathbf{z}}_1^s$	$\hat{\mathbf{z}}_2^s$	$\hat{\mathbf{z}}_3^s$	$\hat{\mathbf{z}}_1$	$\hat{\mathbf{z}}_2$	$\hat{\mathbf{z}}_3$
$\hat{u}_1^{(1)}$	$\hat{y}^c - 1$	$\hat{y}_1^s - 1$	0	$\frac{\hat{y}_3^s - \hat{y}^c}{1 - \hat{x}^c - \hat{y}^c} - \hat{x}_3^s$	0	0	1
$\hat{v}_1^{(1)}$	0	$\frac{\hat{y}_1^s - \hat{y}^c}{\hat{x}^c}$	0	$\frac{\hat{y}^c - \hat{y}_3^s}{1 - \hat{x}^c - \hat{y}^c}$	0	0	0
$\hat{u}_2^{(1)}$	$-\hat{x}^c$	0	0	$\frac{\hat{x}^c - \hat{x}_3^s}{1 - \hat{x}^c - \hat{y}^c}$	0	0	0
$\hat{v}_2^{(1)}$	0	$\hat{y}_1^s$	0	$\frac{\hat{x}_3^s - \hat{x}^c}{1 - \hat{x}^c - \hat{y}^c} - \hat{x}_3^s$	0	0	1
$\hat{u}_3^{(1)}$	2	2	0	$2\frac{\hat{x}_3^s - \hat{x}^c}{1 - \hat{x}^c - \hat{y}^c}$	0	0	0
$\hat{v}_3^{(1)}$	0	$2\frac{\hat{y}^c - \hat{y}_1^s}{\hat{x}^c}$	0	$2\frac{\hat{y}_3^s - \hat{y}^c}{1 - \hat{x}^c - \hat{y}^c}$	0	0	0

can leverage classical techniques. Also, since the values of $\mathbf{\Phi}_i$ at $\mathbf{z}$ are defined already by Lemma 4, it is only necessary to compute basis values at incenters and singular vertices. Values at singular vertices can be stored to avoid recomputation when looping over triangles in τ_h^M. The ultimate cost to assemble the linear system and solve is investigated in Section 5 via comparison with the classical saddle-point approach, which does not use the solenoidal basis.

4. Computation of pressure from velocity

It is possible to compute p_h after $\mathbf{u}_h$ has been computed, if desired, by solving the problem posed in (17). Bases for $\mathbf{X}_h^0 \setminus \mathbf{V}_h^0$ and P_h will be constructed for this purpose in this section. The next result is used later to verify the correct number of basis functions, and also shows that the linear system for the pressure solve is square.

Lemma 5. *It holds that* $\dim(\boldsymbol{X}_h^0 \setminus \boldsymbol{V}_h^0) = \dim(P_h) = 2|\tau_h^M| + 2|E_{int}^M| - |V_{int}^M|$.

Proof. The pressure space is piecewise constant on τ_h, but with the constraints imposed by Lemma 1, which remove a degree of freedom for each singular point. As proved in [4], an equivalent way to count is three basis functions per interior singular point and one per exterior singular point. Along with the constraint $P_h \subset L_0^2(\Omega)$, it follows from the correspondence between singular points and edges of τ_h^M that $\dim(P_h) = 3|E_{int}^M| + |E_{ext}^M| - 1$. We count three edges per triangle, so including multiplicity for interior edges we find $3|\tau_h^M| = 2|E_{int}^M| + |E_{ext}^M|$, thus $\dim(P_h) = 3|\tau_h^M| + |E_{int}^M| - 1$. Insert the Euler identity in the form

$$1 = |\tau_h^M| + |V^M| - |E^M| = |\tau_h^M| + |V_{int}^M| - |E_{int}^M|$$

and we find $\dim(P_h) = 2|\tau_h^M| + 2|E_{int}^M| - |V_{int}^M|$.

Now refer to Theorem 1: $|\mathcal{B}^0| = \dim(\mathbf{V}_h^0) = 3|V_{int}^M|$. All interior vertices on the Powell-Sabin split contribute toward the dimension of $\mathbf{X}_h^0$; we may group the vertices as V_{int}^M plus interior singular vertices and incenters of triangles in τ_h^M, so that

$$\dim(\mathbf{X}_h^0) = 2(|V_{int}^M| + |E_{int}^M| + |\tau_h^M|)$$
$$\Rightarrow \dim(\mathbf{X}_h^0 \setminus \mathbf{V}_h^0) = 2|\tau_h^M| + 2|E_{int}^M| - |V_{int}^M| = \dim(P_h).$$

□

The construction of basis functions builds upon the scalar finite element space

$$S_h = \left\{\lambda \in \mathcal{C}(\overline{\Omega}) \,|\, \lambda|_T \in \mathbb{P}_1(T),\ \forall T \in \tau_h\right\}.$$

Let $\{\lambda_{\mathbf{z}}\}$ for all vertices $\mathbf{z} \in V$ be the usual Lagrangian (nodal) basis for S_h, such that for a given indexing $\{\mathbf{z}_i\}$ of V we have $\lambda_{\mathbf{z}_j}(\mathbf{z}_k) = \delta_{jk}$. Next, we denote the singular points on the edges $e_j^M \in E_{int}^M$ as $\mathbf{z}_s^{(j)}$ for $1 \leq j \leq |E_{int}^M|$. On each edge e_j^M, associate a unit normal vector $\mathbf{n}_j$ and a unit tangent vector $\mathbf{t}_j$. The basis construction will make use of some functions

$$\boldsymbol{\psi}_s^{(j)} = \lambda_{\mathbf{z}_s^{(j)}} \mathbf{n}_j \quad \text{and} \quad \boldsymbol{\phi}_s^{(j)} = \lambda_{\mathbf{z}_s^{(j)}} \mathbf{t}_j.$$

The support of these functions is the set of four triangles $T \in \tau_h$ with the common interior singular vertex $\mathbf{z}_s^{(j)}$. If we denote the incenters of each triangle $T_k^M \in \tau_h^M$ by $\mathbf{z}_c^{(k)}$ for $1 \leq k \leq |\tau_h^M|$, then the following are also needed:

$$\boldsymbol{\psi}_c^{(k)} = \lambda_{\mathbf{z}_c^{(k)}} (1, 0) \quad \text{and} \quad \boldsymbol{\phi}_c^{(k)} = \lambda_{\mathbf{z}_c^{(k)}} (0, 1).$$

The support of these functions is the set of six triangles $T \in \tau_h$ with the common vertex $\mathbf{z}_c^{(k)}$.

Some functions above must be excluded to extract a basis. This must be done carefully. First, identify one vertex $\mathbf{z}_0 \in V_{ext}^M$ (that lies on $\partial\Omega$; see also Theorem 1), and group it together with the interior vertices $\mathbf{z}_j \in V_{int}^M$, $1 \leq j \leq |V_{int}^M|$. There exists some subset of edges in E_{int}^M, say e_k^M that we may index (without loss of generality) as $1 \leq k \leq N$, such that these edges form a spanning tree for the graph created by the vertex set $\{\mathbf{z}_0, \ldots, \mathbf{z}_{|V_{int}^M|}\}$ and all of the edges that have both endpoints $\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2 \in \{\mathbf{z}_0, \ldots, \mathbf{z}_{|V_{int}^M|}\}$. A spanning tree of a connected graph is itself a connected subgraph that touches every vertex in the graph without creating any cycles. These particular vertices and edges form a finite connected graph, so a spanning tree exists. It is well-known that $N = |V_{int}^M|$. We compute a spanning tree using Kruskal's algorithm [9] for implementation.

Theorem 2. *Given the indexing above, the following set $\mathcal{S}$ is a basis of $\boldsymbol{X}_h^0 \setminus \boldsymbol{V}_h^0$:*

$$\mathcal{S} = \left\{\boldsymbol{\psi}_s^{(j)}\right\}_{j=|V_{int}^M|+1}^{|E_{int}^M|} \bigcup \left\{\boldsymbol{\phi}_s^{(j)}\right\}_{j=1}^{|E_{int}^M|} \bigcup \left\{\boldsymbol{\psi}_c^{(k)}\right\}_{k=1}^{|\tau_h^M|} \bigcup \left\{\boldsymbol{\phi}_c^{(k)}\right\}_{k=1}^{|\tau_h^M|}.$$

Furthermore, $\nabla \cdot \mathcal{S}$ is a basis of P_h.

Proof. Indeed, by Lemma 5 we have the correct number of basis functions. Note that the functions in $\mathcal{S}$ are equivalent to a subset of the standard nodal basis functions typically used for the globally continuous, piecewise linear velocity space on τ_h except that at singular points the vectors are expressed in the local basis $\{\mathbf{n}_j, \mathbf{t}_j\}$ of $\mathbb{R}^2$. Since the properties of these basis functions are well-known, it is immediately clear that $\mathcal{S} \subset \mathbf{X}_h^0 \setminus \mathbf{V}_h^0$ by construction and that $\mathcal{S}$ is linearly independent. Therefore, it is a basis for $\mathbf{X}_h^0 \setminus \mathbf{V}_h^0$.

Note that each function in $\nabla \cdot \mathcal{S}$ is also in $L_0^2(\Omega)$ (apply the divergence form of Green's theorem), thus each function is in P_h by Lemma 1. It remains to show that if

$$\sum_{j=|V_{int}^M|+1}^{|E_{int}^M|} a_j \nabla \cdot \boldsymbol{\psi}_e^{(j)} + \sum_{j=1}^{|E_{int}^M|} b_j \nabla \cdot \boldsymbol{\phi}_e^{(j)} + \sum_{k=1}^{|\tau_h^M|} c_k \nabla \cdot \boldsymbol{\psi}_c^{(k)} + \sum_{k=1}^{|\tau_h^M|} d_k \nabla \cdot \boldsymbol{\phi}_c^{(k)} = 0,$$

then the coefficients shown are all zero. Equivalently, $0 = \nabla \cdot \boldsymbol{\phi}$ where

$$\boldsymbol{\phi} = \sum_{j=|V_{int}^M|+1}^{|E_{int}^M|} a_j \boldsymbol{\psi}_e^{(j)} + \sum_{j=1}^{|E_{int}^M|} b_j \boldsymbol{\phi}_e^{(j)} + \sum_{k=1}^{|\tau_h^M|} c_k \boldsymbol{\psi}_c^{(k)} + \sum_{k=1}^{|\tau_h^M|} d_k \boldsymbol{\phi}_c^{(k)}.$$

Thus $\boldsymbol{\phi} \in \mathbf{V}_h^0$. Since $\boldsymbol{\phi}$ is already expressed as a linear combination of basis functions, it suffices to show that $\boldsymbol{\phi} = \mathbf{0}$. The basis functions are non-zero at (specific) singular points and incenters but vanish at neighboring vertices, so $\boldsymbol{\phi}(\mathbf{z}) = \mathbf{0}$ for any $\mathbf{z} \in V^M$. Upon changing to the basis of $\mathbf{V}_h^0$ in Theorem 1, this latter property implies

$$\boldsymbol{\phi} = \sum_{k=1}^{|V_{int}^M|} \beta_k \boldsymbol{\Phi}_3^{(k)}.$$

Given any edge e_j^M with $1 \le j \le |V_{int}^M|$, note that

$$\begin{aligned}
&\int_{e_j^M} \boldsymbol{\psi}_e^{(m)} \cdot \mathbf{n}_j \, ds = 0, \;\; |V_{int}^M| + 1 \le m \le |E_{int}^M|, \\
&\int_{e_j^M} \boldsymbol{\phi}_e^{(m)} \cdot \mathbf{n}_j \, ds = 0, \;\; 1 \le m \le |E_{int}^M|, \\
&\int_{e_j^M} \boldsymbol{\psi}_c^{(m)} \cdot \mathbf{n}_j \, ds = 0, \;\; 1 \le m \le |\tau_h^M|, \\
&\int_{e_j^M} \boldsymbol{\phi}_c^{(m)} \cdot \mathbf{n}_j \, ds = 0, \;\; 1 \le m \le |\tau_h^M|, \\
&\Rightarrow \int_{e_j^M} \boldsymbol{\phi} \cdot \mathbf{n}_j \, ds = 0.
\end{aligned}$$

In turn, we have a square, homogeneous linear system of equations for the coefficients β_k:

$$\sum_{k=1}^{|V_{int}^M|} \beta_k \int_{e_j^M} \mathbf{\Phi}_3^{(k)} \cdot \mathbf{n}_j \, ds = \sum_{k=1}^{|V_{int}^M|} \beta_k \, (\pm \delta_{kj}) = 0, \;\; 1 \le j \le |V_{int}^M|. \tag{25}$$

Recall that the edges e_j^M form a spanning tree for the vertices $\{\mathbf{z}_0, \ldots, \mathbf{z}_{|V_{int}^M|}\}$. Each basis function $\mathbf{\Phi}_3^{(k)}$ is associated with vertex $\mathbf{z}_k$, and they can be put into one-to-one correspondence with the edges e_j^M, $1 \le j \le |V_{int}^M|$, say $\mathbf{z}_k$ is an endpoint of edge $e_{j_k}^M$. Since the labeling of interior vertices is arbitrary, for convenience we may assume that edge $e_{j_k}^M$ has endpoints $\mathbf{z}_{k-1}$ and $\mathbf{z}_k$. As $\mathbf{z}_0$ lies on $\partial\Omega$, only basis function $\mathbf{\Phi}_3^{(1)}$ is supported on edge $e_{j_1}^M$, so $\beta_1 = 0$ by (25). Inductively, given that $\beta_{k-1} = 0$ we find that only $\mathbf{\Phi}_3^{(k)}$ is supported on edge $e_{j_k}^M$, so $\beta_k = 0$ also for $2 \le k \le |V_{int}^M|$. Thus, $\boldsymbol{\phi} = \mathbf{0}$, completing the proof. □

5. Computations

Two test problems are provided here on the domain $\Omega = (0,1) \times (0,1) \subset \mathbb{R}^2$. The usual norms for the spaces $L^2(\Omega)$ and $[H^1(\Omega)]^2$ are denoted by $\|\cdot\|$ and $\|\cdot\|_1$, respectively. Computations that use the solenoidal basis for velocity are labelled as SOL, as opposed to SP for the saddle-point formulation (without the solenoidal basis). All computations are done in MATLAB using the multifrontal direct solver package invoked by the command `mldivide`, commonly referred to as the "backslash" operator. The Choleski solver is called by `mldivide` for all SOL calculations in order to exploit the symmetric, positive definite matrix properties, whereas for the SP system the solver called is MA57. The MA57 solver is optimized for sparse, symmetric indefinite systems. The macro-meshes τ_h^M were generated using code from [5, 17]. The code for Kruskal's Algorithm comes from [3].

5.1. Manufactured solution test. We will use the manufactured solution

$$\mathbf{u} = \begin{pmatrix} \sin(x)\cos(y) \\ -\cos(x)\sin(y) \end{pmatrix}, \quad p = xy - \frac{1}{4}, \tag{26}$$

with parameter choice $\nu = 1$. Errors for velocity and pressure are shown in Figure 5 in their respective norms. The horizontal axis shows the mesh size h for the Powell-Sabin splits τ_h. The optimal convergence rates were observed for both velocity and pressure variables. Between SOL and SP, the computed states agreed to within the

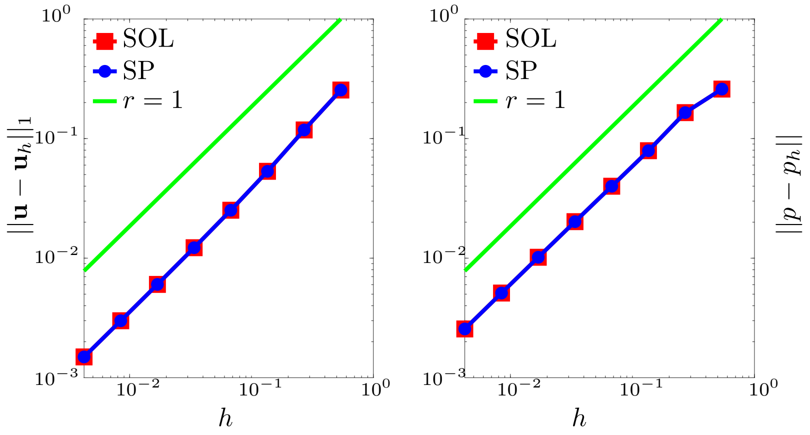

FIGURE 5. Velocity and pressure errors converge with the optimal scaling $\|\mathbf{u} - \mathbf{u}_h\|_1 = \mathcal{O}(h^r)$ and $\|p - p_h\| = \mathcal{O}(h^r)$ with $r = 1$ using the manufactured solution. The SOL and SP velocity calculations agree to round-off precision, so the plotted data overlaps.

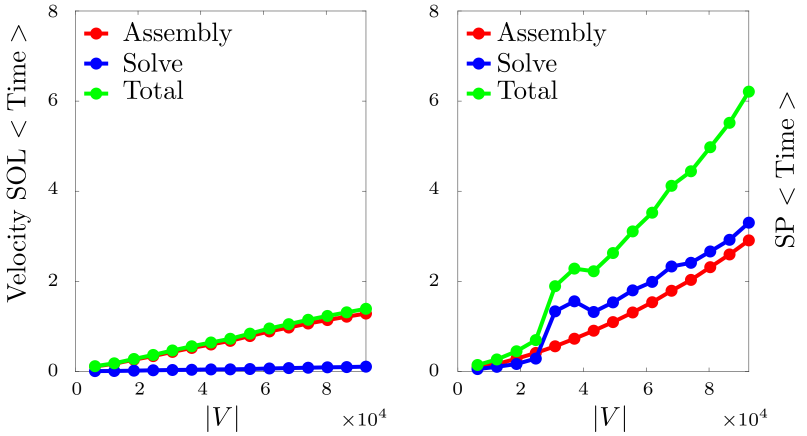

FIGURE 6. Time trials comparing velocity computation with SOL versus SP; manufactured solution problem.

limits of round-off error. We also observed (not shown) that the condition number of the matrix for the SOL velocity system was consistently less than 1% of the corresponding SP condition number.

Run times were computed to assemble and to solve the systems. In Figure 6 we plot $<$ Time $>$ versus $|V|$, where $|V|$ is the number of vertices in the Powell-Sabin mesh and $<$ Time $>$ is the averaged timing over 20 runs. The timings for SOL are only for computing velocity. A jump up in solver time was observed for the SP case around $|V| \approx 3 \times 10^4$; output from the solver package indicated this was due to an increase in the cost to correctly factor the system. Although assembly times using SOL are significant, there is an overall time savings to compute velocity alone using SOL versus SP on fine meshes. Solver timings are much smaller for SOL even on fairly coarse meshes, which would translate to further efficiency gains for applications that require repeated solves without significant reassembly.

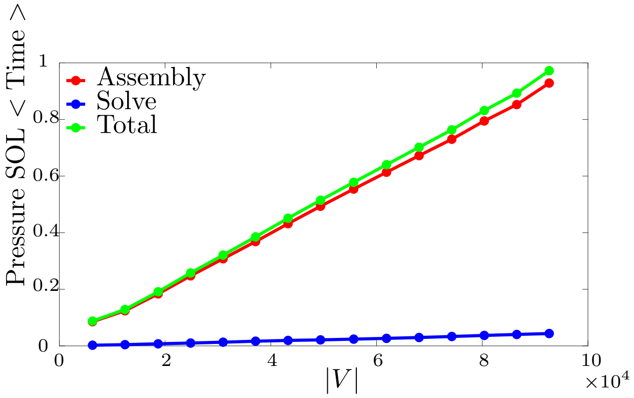

FIGURE 7. SOL pressure timings; manufactured solution problem.

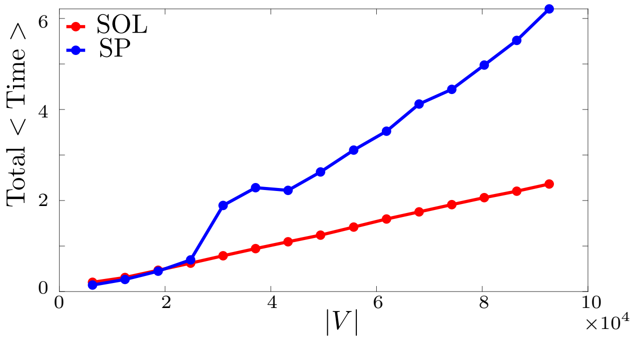

FIGURE 8. Total time to assemble and solve for pressure and velocity; manufactured solution problem.

Timings are displayed for the computation of pressure via the SOL method in Figure 7. These times are somewhat less than the corresponding times for the SOL velocity computations. The total time required to compute both velocity and pressure is compared in Figure 8. In conclusion, the SOL method supports a more efficient system solve on fairly coarse meshes, and may be used to compute velocity alone more quickly. Due to assembly costs, the SOL and SP methods are about even in computing the full velocity-pressure system on relatively coarse meshes (here about $|V| < 3 \times 10^4$), but there is an increasing time advantage using SOL as the meshes become finer.

5.2. Lid-driven cavity test. Methods SOL and SP were tested on the benchmark lid-driven cavity problem as seen in, *e.g.*, [18]. We again chose the parameter $\nu = 1$. This problem sets the problem body and boundary forcings as

$$\mathbf{f} = \mathbf{0}, \quad \mathbf{g} = \begin{cases} (1,0), & \text{for } (x,y) \text{ such that } y = 1,\ 0 < x < 1 \\ (0,0), & \text{otherwise} \end{cases}.$$

The SOL and SP velocity and pressure values agreed to within machine precision error. The velocity and pressure profiles are presented in Figure 9 using a mesh with 64 macro-triangles. The same timing measurements that were shown for the manufactured solution problem in Figure 6-Figure 8 were repeated for the lid-driven cavity problem. These latter results are shown in Figure 10-Figure 12. Clearly, the

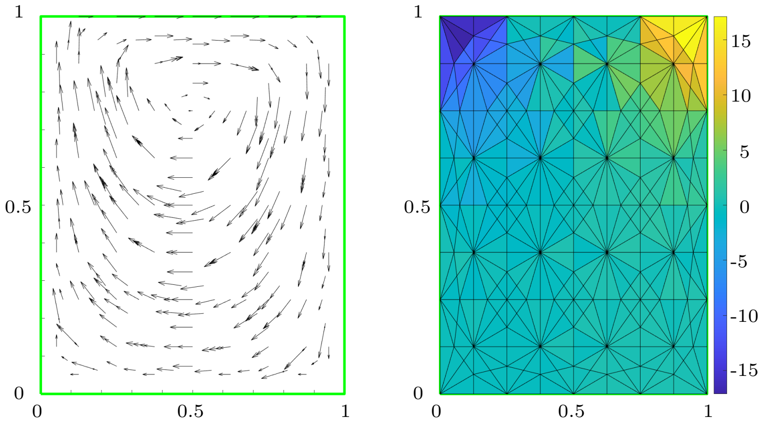

FIGURE 9. Velocity (left) and pressure (right) estimates for SOL for the lid-driven cavity problem.

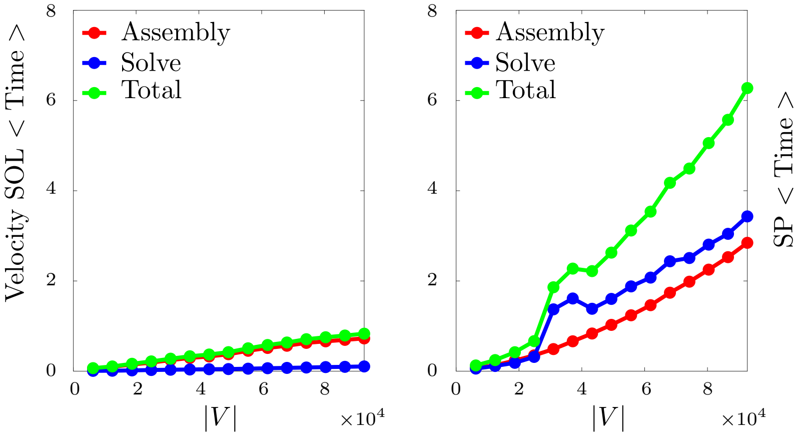

FIGURE 10. Time trials comparing velocity computation with SOL versus SP on the lid driven cavity problem.

results are very similar between the two test problems. There is even slightly more advantage using SOL in the lid-driven cavity problem because of reduced assembly costs compared to the manufactured solution problem.

6. Summary

A solenoidal basis of a first-order polynomial velocity space was explicitly constructed in this paper for the conforming, divergence-free velocity-pressure finite element pair studied in [4] for the Stokes problem. These elements are defined on the Powell-Sabin split of a triangulation for a two-dimensional domain. Inhomogeneous Dirichlet conditions were enabled by constructing an interpolation operator for boundary data into the trace of the solenoidal velocity subspace. A basis of the pressure space was also derived using divergences of non-solenoidal functions from the full discrete velocity space. Its construction used a graph-theoretic argument to

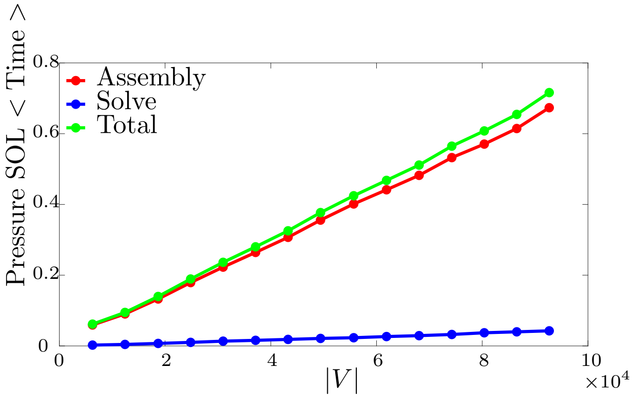

FIGURE 11. SOL pressure timings for lid-driven cavity problem.

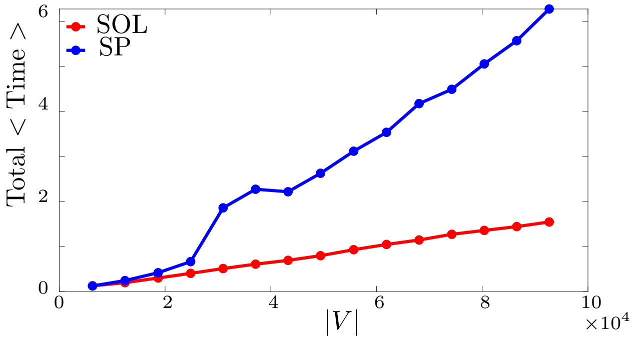

FIGURE 12. Total time to assemble and solve for pressure and velocity; lid-driven cavity problem.

discard functions associated with a certain subset of vertices in the mesh, retaining a spanning set. The implementation can be accomplished by generating a spanning tree related to the mesh, for which purpose their is an efficient algorithm due to Kruskal [9]. All basis functions were shown to have local support. Rigorous proofs were included that verify the construction of bases. Also, some implementation details for the solenoidal velocity basis were discussed that enable a localization of the computations to assemble the linear system.

The use of the bases studied enables a block-triangular structure for the velocity-pressure system, wherein the velocity can be isolated and computed without computing the pressure. The computed velocity can then be used after to compute the pressure, if desired. The non-solenoidal component of the discrete velocity space is eliminated from the velocity computation, which further reduces the problem size.

The system matrices for the velocity and pressure solves are both symmetric, positive-definite and support the use of the Choleski direct solver. Computations using MATLAB verified a significant decrease in time to compute velocity alone compared to solving the usual saddle-point system. Except on fairly coarse meshes, even computing both pressure and velocity together was faster using the solenoidal velocity basis.

Other benefits are expected for situations not included in the scope of this paper. The conjugate gradient method and associated preconditioners could be applied for larger sized problems when a direct solve is not feasible. Also, for applications that require repeated linear solves with a reduced cost of system reassembly compared to the initial assembly, there could be a larger efficiency gain using the solenoidal basis than was demonstrated in this paper.

The paper [4] also includes divergence-free finite element methods for the three-dimensional Stokes problem on Worsey-Farin splits. The solenoidal approach should extend to this method as well. Although the extension and implementation details are likely to require a significant effort, the critical theoretical components of the basis construction still apply, including the graph-theoretic tools used for the pressure basis.

References

[1] A. Alonso Rodríguez, J. Camaño, E. De Los Santos, and F. Rapetti. A graph approach for the construction of high order divergence-free Raviart-Thomas finite elements. Calcolo, 55:1–28, 2018.

[2] J. Carrero, B. Cockburn, and D. Schötzau. Hybridized globally divergence-free LDG methods. Part I: the Stokes problem. Mathematics of Computation, 75:533–563, 2006.

[3] Cheilakos, Nickolas. Kruskal algorithm. https://www.mathworks.com/matlabcentral/file exchange/13457-kruskal-algorithm, 2006.

[4] Maurice Fabien, Johnny Guzmán, Michael Neilan, and Ahmed Zytoon. Low-order divergence-free approximations for the Stokes problem on Worsey–Farin and Powell–Sabin splits. Computer Methods in Applied Mechanics and Engineering, 390:1–24, Feb 2022.

[5] Gockenbach, Mark. Understanding and Implementing the Finite Element Method. 2006.

[6] J. Guzmán, A. Lischke, and M. Neilan. Exact sequences on Powell-Sabin splits. Calcolo, 57: 1–25, 2020.

[7] Peter Hansbo and Mats G. Larson. Piecewise divergence-free discontinuous Galerkin methods for Stokes flow. Communications in Numerical Methods in Engineering, 24(5):355–366, 2008.

[8] Volker John, Alexander Linke, Christian Merdon, Michael Neilan, and Leo G. Rebholz. On the divergence constraint in mixed finite element methods for incompressible flows. SIAM Review, 59(3):492–544, 2017.

[9] Joseph B Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical society, 7(1):48–50, 1956.

[10] A. Montlaur, S. Fernandez-Mendez, and A. Huerta. Discontinuous Galerkin methods for the Stokes equations using divergence-free approximations. International Journal for Numerical Methods in Fluids, 57(9):1071–1092, 2008.

[11] Lin Mu and Xiu Ye. A simple finite element method for the Stokes equations. Advances in Computational Mathematics, 43:1305–1324, 2017.

[12] Chunjae Park. A locally calculable P^3-pressure in a decoupled method for incompressible Stokes equations, arXiv:2107.08472, 2021.

[13] Jinshui Qin and Shangyou Zhang. Stability and approximability of the P1-P0 element for Stokes equations. International Journal for Numerical Methods in Fluids, 54(5):497–515, 2007.

[14] L. R. Scott and M. Vogelius. Norm estimates for a maximal right inverse of the divergence operator in spaces of piecewise polynomials. ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique, 19(1):111–143, 1985.

[15] Junping Wang, Yanqiu Wang, and Xiu Ye. A robust numerical method for stokes equations based on divergence-free H(div) finite element methods. SIAM Journal on Scientific Computing, 31(4):2784–2802, 2009.

[16] Xiu Ye and Shangyou Zhang. A numerical scheme with divergence free H-div triangular finite element for the Stokes equations. Applied Numerical Mathematics, 167:211–217, 2021.

[17] Yonker, Shea. Triangulation manipulation program. https://github.com/syonker/Triangulation-Manipulation-Program, April 2018.

[18] Olek C Zienkiewicz, Robert Leroy Taylor, and Perumal Nithiarasu. The Finite Element Method for Fluid Dynamics. Butterworth-Heinemann, 2013.

Department of Mathematics, University of Connecticut, Storrs, CT 06269, USA
E-mail: jeffrey.connors@uconn.edu and michael.gaiewski@uconn.edu