SIAM J. SCI. COMPUT.
Vol. 41, No. 3, pp. B425–B447

ALGEBRAIC HYBRIDIZATION AND STATIC CONDENSATION WITH APPLICATION TO SCALABLE $H(\mathrm{div})$ PRECONDITIONING*

V. DOBREV†, T. KOLEV†, C. S. LEE†, V. TOMOV†, AND P. S. VASSILEVSKI†

Abstract. We propose a unified algebraic approach for the practical application and preconditioning of static condensation and hybridization, two popular techniques in finite element discretizations. We demonstrate the use of this algebraic framework for the construction of scalable solvers for problems involving $H(\mathrm{div})$-spaces discretized by conforming (Raviart–Thomas) elements of arbitrary order. We illustrate through numerical experiments the relative performance of the two (in some sense dual) techniques in comparison with a state-of-the-art parallel solver, ADS [T. V. Kolev and P. S. Vassilevski, *SIAM J. Sci. Comput.*, 34 (2012), pp. A3079–A3098], available at http://www.llnl.gov/casc/hypre and http://mfem.org. Based on these results, we recommend the use of the hybridization technique in practice, due to its clearly demonstrated superior performance with increased benefit for higher-order elements.

Key words. static condensation, hybridization, algebraic multigrid, ADS, $H(\mathrm{div})$ solvers, radiation-diffusion transport

AMS subject classifications. 65F10, 65N20, 65N30

DOI. 10.1137/17M1132562

1. Introduction.

This paper proposes a unified algebraic approach of constructing reduced problems by two popular techniques in the finite element method: static condensation and hybridization. It also discusses approaches for constructing scalable preconditioners for these reduced problems. The present work is motivated by the solution of problems arising in a variety of finite element simulations in practice. Of particular interest is the case when the problems involve the function space $H(\mathrm{div})$ consisting of L_2-vector functions that have their weak divergence in L_2. Such problems naturally arise in the mixed finite element method for the Darcy equation [5], the Brinkman equation [41], as well as in the FOSLS (first-order system least-squares) discretization approach [12]. Also, certain formulations of radiation-diffusion transport [10] lead to problems involving the space $H(\mathrm{div})$. We consider a practical example of such a simulation in our numerical tests.

Highly scalable solvers for $H(\mathrm{div})$ problems have been designed in the past. The most successful ones, of ADS type [33], are based on the *regular decomposition* theory developed by Hiptmair and Xu [28]. Although of optimal complexity, the current state-of-the-art ADS algorithms are quite involved and require additionally solvers for $H(\mathrm{curl})$, such as AMS [32]. Both solvers, ADS and AMS, require some additional user input (e.g., the *discrete gradient* matrix), which makes them not completely *algebraic*. The results of the present paper can be viewed as a further step toward the design of fully algebraic solvers for problems involving $H(\mathrm{div})$ spaces.

*Submitted to the journal's Computational Methods in Science and Engineering section May 31, 2017; accepted for publication (in revised form) January 30, 2019; published electronically May 7, 2019.

http://www.siam.org/journals/sisc/41-3/M113256.html

Funding: This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344 and sponsored by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program, LLNL-JRNL-732140.

†Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551 (dobrev1@llnl.gov, tzanio@llnl.gov, cslee@llnl.gov, tomov2@llnl.gov, panayot@llnl.gov).

To design a faster, yet still scalable, alternative to ADS, we employ a traditional finite element technique commonly used in mixed finite elements referred to as *hybridization*. This is perhaps the first technique proposed to solve the saddle-point problems arising in the mixed methods, since it leads to symmetric and positive-definite (SPD) reduced problems. Its early form appeared in a paper by Fraeijis de Veubeke in 1965 [23] as an efficient solution strategy for elasticity problems. In [2], Arnold and Brezzi studied the method from a theoretical perspective and obtained error estimates for all the unknowns involved; by making use of the mixed hybrid formulation, they also derived the equivalence between certain mixed and nonconforming methods for even-order polynomial approximations over triangles. A more complete study on such equivalence can be found in [1]. The technique was further developed by Cockburn and Gopalakrishnan to allow approximations by polynomials of different degree in different subregions; see, for example, [17] and [18] for applications to mixed finite element approximations of second-order scalar elliptic problems and Stokes flow. More references to the subject can be found in [9, 19, 20, 22, 24, 36, 40, 42, 43].

Another very popular technique in finite elements is *static condensation*. It is a reduction technique intended to reduce the size of a linear system of finite element equations by eliminating at the element or subdomain level internal degrees of freedom (dofs). It was also first introduced in the structural analysis literature in 1965 (cf. [26, 30, 44]) and has been widely used since then. Static condensation neglects the dynamic effect in the reduction, hence the name. On element level, one can eliminate locally dofs that are not shared by neighboring elements (a typical case for high-order elements). Thus, the reduced, Schur complement, problem contains only shared (interface) dofs. If the elimination is performed on a subdomain level, the resulting reduced system coincides with the one in the balancing domain decomposition by constraints (BDDC) algorithm [21]. Recently, BDDC preconditioners with deluxe scaling and adaptive selection of primal constraints for $H(\mathrm{div})$ problems have been developed; see, for instance, [38, 45, 46]. The approach in the present paper, on the other hand, solves the reduced system by algebraic multigrid (AMG).

For $H(\mathrm{div})$-conforming elements (i.e., elements that have only shared dofs through common faces), both static condensation and hybridization lead to reduced problems of the same size (equal to the number of shared interface dofs). If one applies these two approaches not on an element level but for subdomains (union of elements), substantial reduction in size can be achieved. Thus, these two approaches are of great practical interest and the goal of the present paper is to study them in a common framework, emphasize their similarities and differences, and, most importantly, design new or modify existing solution techniques that are efficient and scalable, achieving substantial savings compared to the state-of-the-art solvers for the original problems. For problems obtained by static condensation, there has been success in modifying the state-of-the-art solvers (AMS and ADS) to be directly applicable to the reduced problem; cf. [11, 4]. In this paper, we focus on the design of scalable solvers for problems involving $H(\mathrm{div})$ in a general algebraic setting, extending the preliminary results in [34].

The remainder of the paper is organized as follows. In section 2, we describe the algebraic hybridization approach and show how it can serve as an alternative to traditional finite element assembly. Next, we deduce its relation to static condensation and the mixed hybridized finite element method in section 3. Preconditioning strategies for the reduced systems obtained from hybridization and static condensation are discussed in section 4. Then we describe the implementation details of the methods and present some numerical examples in sections 5 and 6, respectively. Finally, we summarize our results and draw conclusions in section 7.

2. Two perspectives on finite element assembly. In this section we describe the traditional process of finite element assembly and introduce algebraic hybridization as an alternative process to obtain a global linear system describing the same finite element discretization. While the discussion can be more general, in what follows, we focus the presentation on our target example of linear systems coming from finite element discretizations of the variational formulation on the $H(\mathrm{div})$ space: given a linear form $f : H(\mathrm{div};\Omega) \to \mathbb{R}$, find $\mathbf{u} \in H(\mathrm{div};\Omega)$ such that

$$a_{\mathrm{div},\Omega}(\mathbf{u},\mathbf{v}) := \int_\Omega \alpha \,\mathrm{div}\,\mathbf{u}\,\mathrm{div}\,\mathbf{v} + \beta\mathbf{u}\cdot\mathbf{v}\,dx = f(\mathbf{v}) \qquad \forall\,\mathbf{v} \in H(\mathrm{div};\Omega), \tag{2.1}$$

where Ω is some open, polygonal, bounded, and connected domain in $\mathbb{R}^d, d = 2$ or 3. The scalar coefficient $\alpha > 0$ and the tensor-valued coefficient β are bounded and in general heterogeneous. Moreover, we assume β is SPD.

Remark 2.1. Although our presentation focuses on variational problems of the form (2.1), hybridization can be applied to other problems where face-based approximation is involved. Indeed, hybridization is a classic approach for solving saddle-point problems arising from mixed finite element discretizations. We remark here that the discussion and the proposed algebraic solver in section 4 are equally valid for hybridization of mixed finite element problems; see Remark 4.6 for more details.

2.1. Finite element assembly components. In the finite element method, we introduce a triangulation $\mathcal{T}_h$ of the given computational domain and decompose the forms in (2.1) into sums of their local element versions, i.e., $a_{\mathrm{div},\Omega}(\cdot,\cdot) = \sum_{\tau\in\mathcal{T}_h} a_{\mathrm{div},\tau}(\cdot,\cdot)$ and $f(\cdot) = \sum_{\tau\in\mathcal{T}_h} f_\tau(\cdot)$. The triangulation $\mathcal{T}_h$ considered in this paper includes general unstructured simplicial meshes and meshes that consist of parallelograms or parallelepipeds. We further replace $H(\mathrm{div};\Omega)$ with a finite dimension space $\mathcal{H}_h$ induced by $\mathcal{T}_h$. $\mathcal{H}_h$ can be taken as the Raviart–Thomas space [39] or the Brezzi–Douglas–Marini space [7], for instance. Then the discrete problem imposed on the finite element space $\mathcal{H}_h$ is to find $\mathbf{u}_h \in \mathcal{H}_h$ such that

$$\sum_{\tau\in\mathcal{T}_h} a_{\mathrm{div},\tau}(\mathbf{u}_h,\mathbf{v}_h) = \sum_{\tau\in\mathcal{T}_h} f_\tau(\mathbf{v}_h) \qquad \forall\,\mathbf{v}_h \in \mathcal{H}_h. \tag{2.2}$$

On each element $\tau \in \mathcal{T}_h$ we compute a local *element stiffness matrix* A_τ and a *load vector* $\mathbf{f}_\tau$ by using the element basis functions in the local bilinear and linear forms. Note that A_τ is positive-definite because of the mass term $\int_\Omega \beta\mathbf{u}\cdot\mathbf{v}$.

Remark 2.2. Although the focus of the current paper is on $H(\mathrm{div})$ problems, the discussion in the next two sections (sections 2.2 and 2.3) is on a purely algebraic level and thus is equally valid for discrete problems coming from various finite element discretizations of H^1, $H(\mathrm{curl})$ and other problems.

Remark 2.3. Due to some theoretical considerations, we made the assumption that the mesh elements in $\mathcal{T}_h$ are obtained by linear transformations from a reference element. However, from a practical point of view, such restriction is not needed, and the preconditioners we consider can be applied in problems posed on general quadrilateral and hexahedral meshes, as well as on high-order curvilinear meshes; see sections 6.2 and 6.3.

2.2. Traditional finite element assembly. A central concept in the finite element method is that of *assembly*—the accumulation of local (element) stiffness matrices A_τ and load vectors $\mathbf{f}_\tau$ into a global matrix A and a global vector $\mathbf{f}$ which form the (global) linear system

$$A\mathbf{x} = \mathbf{f} \tag{2.3}$$

for the unknown vector of finite element dofs $\mathbf{x}$. The $n \times n$ matrix A is SPD and generally sparse, i.e., the number of its nonzero entries, $nnz(A)$, is $\mathcal{O}(n)$.

Let $\widehat{A} = blockdiag(A_\tau)$ be the $\hat{n} \times \hat{n}$ block-diagonal matrix with all element stiffness matrices $\{A_\tau\}$ on its main diagonal, and let $\widehat{\mathbf{f}}$ be similarly the $\hat{n} \times 1$ vector containing all element load vectors $\{\mathbf{f}_\tau\}$. The finite element method produces a global-to-local mapping, P, which maps global dofs into local ones for each element. That is, there is an $n \times \hat{n}$ *interpolation* matrix P, for which the traditional assembly process can be written as the triple matrix product that relates A, $\widehat{A}$, and P, and a matrix-vector product that builds the global right-hand side,

$$A = P^T \widehat{A} P \quad \text{and} \quad \mathbf{f} = P^T \widehat{\mathbf{f}}\,. \tag{2.4}$$

In the simplest case the action $\widehat{\mathbf{x}} = P\mathbf{x}$ simply means copying the values of the global dof vector $\mathbf{x}$ to the corresponding local values, independently in each element. More complicated P can account for nonconforming mesh refinement as we describe in some detail in a following section. In both of these, as well as in more general cases, (2.4) can be viewed as a formalized expression for the traditional process of finite element assembly.

2.3. Algebraic hybridization. It is classical in the finite element literature to introduce (2.3) as the global linear system that corresponds to the finite element discretization specified by $\widehat{A}$, $\widehat{\mathbf{f}}$, and P. As we describe next, there is an alternative global linear system corresponding to the same solution which, in certain cases, is advantageous from the perspective of linear solvers. To introduce this alternative, let C be another matrix such that

$$\text{Null}(C) = \text{Range}(P),$$

i.e., a vector z satisfies $Cz = 0$, if and only if there is a vector y, such that $z = Py$. In particular, we have the matrix equality $CP = 0$.

We assume that P is full column rank and that C is full row rank. Similarly to P, the matrix C could also be provided algebraically by the finite element space, e.g., the columns of C^T can be constructed by orthogonally completing the basis given by the columns of P. (The matrix C is not unique, and there are other ways to construct it, as illustrated later in the paper.)

The idea of algebraic hybridization is that instead of adding together the element contributions of $\widehat{A}$, $\widehat{\mathbf{f}}$, we use C to enforce equality constraints in the decoupled vector $\widehat{\mathbf{x}}$ through a new Lagrange multiplier variable $\boldsymbol{\lambda}$. This process is known as *hybridization* in the context of the mixed finite element method [8], but in this section we consider it on a purely algebraic level, independent of the particular continuous problem or finite element discretization approach. In fact, the C-based reformulation can be described in more general terms than classical assembly, which we do in the following main linear algebra result.

THEOREM 2.4. *Let $\widehat{A}$, P, and C be matrices of size $\hat{n} \times \hat{n}$, $\hat{n} \times n$, and $m \times \hat{n}$ ($m = \hat{n} - n$), respectively, such that $\widehat{A}$ is SPD, P is full column rank, C is full row rank, and* $\text{Null}(C) = \text{Range}(P)$. *Also, let $\widehat{\mathbf{f}}$ be a column vector of size $\hat{n}$. Then the problem*

$$P^T \widehat{A} P\, \mathbf{x} = P^T \widehat{\mathbf{f}} \tag{2.5}$$

is equivalent to the following, hybridized*, saddle-point system:*

$$\begin{bmatrix} \widehat{A} & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{x}} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{f}} \\ 0 \end{bmatrix}. \tag{2.6}$$

Specifically, the saddle-point system (2.6) *is uniquely solvable and the solution* $\widehat{\mathbf{x}}$ *of* (2.6) *and* $\mathbf{x}$ *(the solution of the original system) are related as*

$$\widehat{\mathbf{x}} = P\mathbf{x}.$$

The solution $\widehat{\mathbf{x}}$ *is computable via standard (block-)Gaussian elimination: we first compute the Schur complement system for the Lagrange multiplier*

$$H\boldsymbol{\lambda} \equiv C\widehat{A}^{-1}C^T\boldsymbol{\lambda} = C\widehat{A}^{-1}\widehat{\mathbf{f}}; \tag{2.7}$$

then, by back substitution, we have

$$\widehat{\mathbf{x}} = \widehat{A}^{-1}(\widehat{\mathbf{f}} - C^T\boldsymbol{\lambda}).$$

Proof. The saddle-point system is invertible due to our assumptions, namely, $\widehat{A}$ is SPD (hence invertible), and C^T is full column rank, hence the (negative) Schur complement $H = C\widehat{A}^{-1}C^T$ is also SPD and invertible. From $\widehat{A}\widehat{\mathbf{x}} + C^T\boldsymbol{\lambda} = \widehat{\mathbf{f}}$ and $CP = 0$, we obtain

$$P^T\widehat{A}\widehat{\mathbf{x}} = P^T\widehat{\mathbf{f}} =: \mathbf{f}.$$

Now, since $C\widehat{\mathbf{x}} = 0$ (the second equation of (2.6)), and from the assumption $\mathrm{Null}(C) = \mathrm{Range}(P)$, it follows that there is an $\mathbf{x}$ such that $\widehat{\mathbf{x}} = P\mathbf{x}$. Then from $\mathbf{f} = P^T\widehat{A}\widehat{\mathbf{x}} = P^T\widehat{A}P\mathbf{x}$, we see that this $\mathbf{x}$ is unique, namely, $\mathbf{x}$ is the unique solution of the original problem (2.5). ∎

We can summarize the results of the theorem and the previous section by stating that $P^T\widehat{A}P$ and $C\widehat{A}^{-1}C^T$ are equally valid global matrices for the same problem. In the case of finite elements, the first one corresponds to assembly, while the second one is the hybridized system for Lagrange multipliers.

Remark 2.5. Note that the condition that P has full column rank is equivalent to the existence of a left inverse of P: there is an $n \times \hat{n}$ matrix R such that $RP = I$, e.g., $R = (P^TP)^{-1}P^T$. We can use this restriction matrix R to compute the solution $\mathbf{x}$ of (2.5) from the solution $\widehat{\mathbf{x}}$ of (2.6) as $\mathbf{x} = R\widehat{\mathbf{x}}$. Furthermore, if instead of $\widehat{\mathbf{f}}$ we have access only to the assembled right-hand side vector $\mathbf{f} = P^T\widehat{\mathbf{f}}$, we can use $\widetilde{\mathbf{f}} = R^T\mathbf{f}$ in (2.6) instead of $\widehat{\mathbf{f}}$ since $P^T\widetilde{\mathbf{f}} = P^T\widehat{\mathbf{f}} = \mathbf{f}$.

Remark 2.6. Note also that the first set of equations in (2.6),

$$\widehat{A}\widehat{\mathbf{x}} = \widehat{\mathbf{f}} - C^T\boldsymbol{\lambda},$$

can be interpreted as a local version of the problem (in a weak form), where the term $C^T\boldsymbol{\lambda}$ plays the role of a dual vector for Neumann-type boundary conditions. Indeed, in the finite element case, if we integrate $a_{\mathrm{div},\tau}$ by parts locally,

$$\widehat{A}\widehat{\mathbf{x}} = \widehat{\mathbf{f}} - \widehat{F},$$

where $\widehat{F}$ is the Neumann data for $\widehat{\mathbf{x}}$ tested against the test functions in each element. Since the exact solution satisfies (2.5), we have $P^T\widehat{F} = 0$, i.e., $\widehat{F} = C^T\boldsymbol{\lambda}$ for some

$\boldsymbol{\lambda}$ by $\mathrm{Null}(P^T) = \mathrm{Range}(C^T)$. One can then think of hybridization as prescribing Neumann data for local problems from one set of $\boldsymbol{\lambda}$ values and then imposing the fact that the local solutions (fluxes) should match on their shared dofs to derive equations for $\boldsymbol{\lambda}$; for more details, see [31].

The result of Theorem 2.4 can be extended to more general settings that relax the conditions that $\widehat{A}$ and P are full column rank. While Theorem 2.4 is sufficient for the developments in the present paper, we include this more general result below for completeness.

THEOREM 2.7. *Assume that $\widehat{A}$ is a square matrix (without assuming that it is SPD, or even invertible) and that* $\mathrm{Null}(C) = \mathrm{Range}(P)$ *(without assuming that P has full column rank and C has full row rank). Then* (2.5) *and* (2.6) *are equivalent in the following sense:*

- *If* $\mathbf{x}$ *solves* (2.5)*, then there exists* $\boldsymbol{\lambda}$ *such that the pair* $(\widehat{\mathbf{x}} = P\mathbf{x}, \boldsymbol{\lambda})$ *solves* (2.6).
- *If* $(\widehat{\mathbf{x}}, \boldsymbol{\lambda})$ *solves* (2.6)*, then there exists* $\mathbf{x}$ *that solves* (2.5) *such that* $P\mathbf{x} = \widehat{\mathbf{x}}$.

If, in addition, we assume that $\widehat{A}$ is invertible, then (2.6) *and* (2.7) *are also equivalent in the following sense:*

- *If* $(\widehat{\mathbf{x}}, \boldsymbol{\lambda})$ *solves* (2.6)*, then* $\boldsymbol{\lambda}$ *solves* (2.7).
- *If* $\boldsymbol{\lambda}$ *solves* (2.7)*, then* $(\widehat{\mathbf{x}} = \widehat{A}^{-1}(\widehat{\mathbf{f}} - C^T\boldsymbol{\lambda}), \boldsymbol{\lambda})$ *solves* (2.6).

Proof. Assume that $\mathbf{x}$ solves (2.5) and set $\widehat{\mathbf{x}} = P\mathbf{x}$. We need to show that there exists $\boldsymbol{\lambda}$ such that $(\widehat{\mathbf{x}}, \boldsymbol{\lambda})$ solves (2.6). The second equation in (2.6) follows from $C\widehat{\mathbf{x}} = CP\mathbf{x} = 0$. The defining condition for $\boldsymbol{\lambda}$ is

$$\widehat{A}\widehat{\mathbf{x}} + C^T\boldsymbol{\lambda} = \widehat{\mathbf{f}} \qquad \text{or} \qquad C^T\boldsymbol{\lambda} = \widehat{\mathbf{f}} - \widehat{A}\widehat{\mathbf{x}} =: \widehat{\mathbf{g}}\,.$$

By (2.5), we have that $P^T\widehat{\mathbf{g}} = 0$, i.e., $\widehat{\mathbf{g}} \in \mathrm{Null}(P^T)$. Since we have

$$\mathrm{Null}(P^T) = \mathrm{Range}(P)^\perp \subseteq \mathrm{Null}(C)^\perp = \mathrm{Range}(C^T)\,,$$

the existence of the required Lagrange multiplier $\boldsymbol{\lambda}$ is proven.

Assume now that $(\widehat{\mathbf{x}}, \boldsymbol{\lambda})$ solves (2.6). On one hand, the second equation in (2.6) is $C\widehat{\mathbf{x}} = 0$, i.e., $\widehat{\mathbf{x}} \in \mathrm{Null}(C)$, and since $\mathrm{Null}(C) \subseteq \mathrm{Range}(P)$, there exists $\mathbf{x}$ such that $P\mathbf{x} = \widehat{\mathbf{x}}$. On the other hand, the first equation in (2.6) implies

$$P^T(\widehat{A}\widehat{\mathbf{x}} + C^T\boldsymbol{\lambda}) = P^T\widehat{\mathbf{f}} \qquad \text{or} \qquad P^T\widehat{A}\widehat{\mathbf{x}} = P^T\widehat{\mathbf{f}}\,,$$

showing that $\mathbf{x}$ solves (2.5). The proof of the second part of the theorem is straightforward. ∎

Note that the uniqueness of $\mathbf{x}$ is not guaranteed in the settings of Theorem 2.7, even if we assume that P has full column rank and $\widehat{A}$ is invertible. Similarly, the uniqueness of $\boldsymbol{\lambda}$ does not follow from the additional assumption that C has full row rank and $\widehat{A}$ is invertible.

2.4. Interpolation and constraint matrices. In this section we provide two general examples for the interpolation and constraint matrices P and C which will be used in the remainder of the paper. The first one is the main example we have in mind for the application to $H(\mathrm{div})$ problems considered in the following sections.

Example 2.1. In this example we consider a general finite element discretization on a nonconforming mesh (i.e., a mesh with constrained "hanging nodes"; see, e.g.,

[13]). We assume that we are starting from mesh elements (or more generally, subdomains that are unions of elements) and that we have identified the interfaces between these substructures (original elements or subdomains). There are dofs interior, "i," to the substructures. We also have dofs on the interfaces. In this setting we assume that on the interfaces we have dofs that are *master*, "m," and *slave*, "s."

Introduce the matrices P and C with the described blocks of i-, m-, and s-dofs,

$$P = \begin{bmatrix} I_i & 0 \\ 0 & I_m \\ 0 & W_{sm} \end{bmatrix} \qquad \text{and} \qquad C = \begin{bmatrix} 0 & -W_{sm} & I_s \end{bmatrix}.$$

The 3-by-2 block form of P above corresponds to the interior and master dofs (for its columns) and interior, master, and slave dofs (for its rows). The block W represents the mapping that determines the slave dofs in terms of the master ones. Finally, the matrix $\widehat{A}$ is the block-diagonal matrix, with blocks assembled for the individual substructures (or simply the element matrices). As before, we assume that $\widehat{A}$ is invertible, which is the case if we have a mass term added to a semidefinite term, like the $H(\mathrm{div})$ bilinear form we consider in this paper.

Note that for conforming meshes, the matrix W_{sm} will have a much simpler structure, typically being a square matrix with one nonzero entry per row. For example, for the low-order Raviart–Thomas elements of interest in this paper, $W_{sm} = -I$ with proper ordering of the dofs and choice of element face normals.

The definition of the constraint matrix C in Example 2.1 was purely algebraic and simply designed to ensure $\mathrm{Null}(C) = \mathrm{Range}(P)$. It may be more desirable to define C as a bilinear form, incorporating finite element information. This is illustrated for $H(\mathrm{div})$ problems in the following example.

Example 2.2. Consider the case where the solution space is $H(\mathrm{div})$-conforming, and introduce a multiplier finite element space $\mathcal{M} \subset L^2(\mathcal{I})$ on the interface $\mathcal{I}$ between elements (or substructures); see Figure 5.1 for illustration.

If we denote the *broken* solution space by $\widehat{\mathcal{V}}$, then the matrix C can be derived from the bilinear form:

$$c(\hat{\mathbf{u}}, \lambda) = \int_{\mathcal{I}} [\hat{\mathbf{u}} \cdot \mathbf{n}]\, \lambda\,, \qquad \hat{\mathbf{u}} \in \widehat{\mathcal{V}}, \lambda \in \mathcal{M}\,.$$

Above, $[\hat{\mathbf{u}} \cdot \mathbf{n}]$ stands for the jump of the normal component of $\hat{\mathbf{u}}$ across the interface $\mathcal{I}$. Similarly to Example 2.1, for the same P, the matrix C now has the M_s-weighted form

$$C = \begin{bmatrix} 0 & -M_s W_{sm} & M_s \end{bmatrix},$$

where M_s is a mass matrix between the multiplier space $\mathcal{M}$ and the normal trace space of $\widehat{\mathcal{V}}$.

3. Relation to static condensation and hybridized mixed finite elements. In this section we show that the hybridized matrix H from (2.7) is closely related to the matrices obtained with two other classical finite element approaches: that of algebraic *static condensation* and that of traditional hybridization used in the mixed finite element method for second-order elliptic equations.

3.1. Relation to static condensation. In the setting of Example 2.1, the matrix $\widehat{A}$ can be partitioned into a two-by-two block form with its first block corresponding to the interior dofs (with respect to the elements/substructures). The second block corresponds to the master and slave dofs combined, i.e., we have

$$\widehat{A} = \begin{bmatrix} \widehat{A}_{ii} & \widehat{A}_{ib} \\ \widehat{A}_{bi} & \widehat{A}_{bb} \end{bmatrix}. \tag{3.1}$$

We notice that $\widehat{A}_{ii}$ is block-diagonal since the interior dofs in one substructure do not couple with interior dofs in any other substructure. The Schur complement

$$\widehat{S} = \widehat{A}_{bb} - \widehat{A}_{bi}\widehat{A}_{ii}^{-1}\widehat{A}_{ib}$$

is SPD, hence invertible. It acts on vectors corresponding to the interface dofs "b" (combined slave and master ones). Now, consider the Schur complement $H = C\widehat{A}^{-1}C^T$ from (2.7) for the hybridized system (2.6). Since $C = [0,\ -W,\ I]$, and

$$\widehat{A}^{-1} = \begin{bmatrix} * & * \\ * & \widehat{S}^{-1} \end{bmatrix},$$

we easily see that

$$H = C\widehat{A}^{-1}C^T = [-W,\ I]\widehat{S}^{-1}[-W,\ I]^T,$$

while the *static condensation* matrix, i.e., the reduced matrix for the master dofs is

$$S = \begin{bmatrix} I \\ W \end{bmatrix}^T \widehat{S} \begin{bmatrix} I \\ W \end{bmatrix}. \tag{3.2}$$

In the case of finite element matrices, where coupling across elements (or more generally, across substructures) occurs only via interface dofs (no vertex and edge dofs), one can decouple each interface dof into exactly two copies and let $W = -I$. Examples of such elements are the $H(\mathrm{div})$-conforming Raviart–Thomas elements and the H^1-nonconforming Crouzeix–Raviart elements.

In this setting $\widehat{A}$ is block-diagonal, and also $\widehat{S}$ is block-diagonal (substructure-by-substructure). Thus, to form the hybridized Schur complement H, we need to assemble the inverses of the local substructure Schur complements, whereas to form the static condensation matrix S, we need to assemble the local substructure Schur complements.

In other words, we may view the hybridization approach as a *dual* technique to the static condensation approach. This has important consequences for the construction of solvers for the respective problems. For example, we can expect that solvers for S will have the same nature as solvers for the original matrix A, whereas solvers for H will need preconditioners that are effective in the dual of the original trace space. This is the case for the $H(\mathrm{div})$ problem, as observed in [34].

3.2. Relation to hybridized mixed finite elements. Next, we show that under certain conditions the hybridized system H in (2.7) resulting from algebraic hybridization of (2.2) is equivalent to the discrete system obtained by the hybridized mixed finite element discretization [16] for the scalar second-order elliptic problem

$$\begin{aligned} -\operatorname{div}(\beta^{-1}\nabla q) + \alpha^{-1}q &= -g \qquad \text{in } \Omega, \\ q &= 0 \qquad \text{on } \partial\Omega. \end{aligned} \tag{3.3}$$

This observation is noteworthy because it motivates the use of AMG to solve (2.7); cf. [34].

In order to make a more precise statement of this equivalence, we first derive the mixed discretization of the above problem. By introducing an additional unknown $\mathbf{u} = \beta^{-1}\nabla q$, (3.3) is turned into the mixed problem

$$\begin{aligned} \beta\mathbf{u} - \nabla q &= 0 \qquad && \text{in } \Omega, \\ \operatorname{div}\mathbf{u} - \alpha^{-1} q &= g \qquad && \text{in } \Omega, \\ q &= 0 \qquad && \text{on } \partial\Omega. \end{aligned} \tag{3.4}$$

In the following, we take the discrete approximation space for $H(\operatorname{div};\Omega)$ to be the "broken" Raviart–Thomas space of order k, denoted by $\widehat{RT}_h^k$, which is just the Raviart–Thomas space without normal continuity requirement across interelement interface. Also, let $RT_h^k(\tau)$ be the local Raviart–Thomas space on an element $\tau \in \mathcal{T}_h$. Then we take $Q_h^k(\tau) = \operatorname{div}\left(RT_h^k(\tau)\right)$ and $M_h^k(e) = (RT_h^k \cdot \mathbf{n}_e)|_e$, where $\mathbf{n}_e$ is a unit normal on the interelement interface e. Define

$$Q_h^k := \{p_h \in L^2(\Omega) : p_h|_\tau \in Q_h^k(\tau)\} \quad \text{and} \quad M_h^k := \{\mu_h \in L^2(\mathcal{E}_h) : \mu_h|_e \in M_h^k(e)\},$$

where $\mathcal{E}_h$ is the union of interelement interfaces of the triangulation $\mathcal{T}_h$. The hybridized mixed discretization [16] for (3.3) is to find $(\widehat{\mathbf{u}}_h, q_h, \lambda_h) \in \widehat{RT}_h^k \times Q_h^k \times M_h^k$ such that

$$\begin{aligned} &\sum_{\tau\in\mathcal{T}_h}\int_\tau \beta\widehat{\mathbf{u}}_h\cdot\widehat{\mathbf{v}}_h + \sum_{\tau\in\mathcal{T}_h}\int_\tau q_h \operatorname{div}\widehat{\mathbf{v}}_h - \sum_{\tau\in\mathcal{T}_h}\int_{\partial\tau\setminus\partial\Omega}\lambda_h(\widehat{\mathbf{v}}_h\cdot\mathbf{n}_\tau) = 0 && \forall\, \widehat{\mathbf{v}}_h \in \widehat{RT}_h^k, \\ &\sum_{\tau\in\mathcal{T}_h}\int_\tau \operatorname{div}\widehat{\mathbf{u}}_h p_h - \sum_{\tau\in\mathcal{T}_h}\int_\tau \alpha^{-1} q_h p_h = \int_\Omega g p_h && \forall\, p_h \in Q_h^k, \\ &\sum_{\tau\in\mathcal{T}_h}\int_{\partial\tau\setminus\partial\Omega}(\widehat{\mathbf{u}}_h\cdot\mathbf{n}_\tau)\mu_h = 0 && \forall\, \mu_h \in M_h^k, \end{aligned} \tag{3.5}$$

where $\mathbf{n}_\tau$ is the unit outward normal of τ. Using matrix-vector notation, the above system takes the form

$$\begin{bmatrix} \widehat{M}_\beta & \widehat{B}^T & C^T \\ \widehat{B} & -\widehat{W}_{\alpha^{-1}} & 0 \\ C & 0 & 0 \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{x}} \\ \mathbf{q} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{g} \\ 0 \end{bmatrix}.$$

The corresponding reduced problem for the Lagrange multipliers reads

$$\widetilde{H}\boldsymbol{\lambda} := \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \widehat{M}_\beta & \widehat{B}^T \\ \widehat{B} & -\widehat{W}_{\alpha^{-1}} \end{bmatrix}^{-1} \begin{bmatrix} C^T \\ 0 \end{bmatrix} \boldsymbol{\lambda} = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \widehat{M}_\beta & \widehat{B}^T \\ \widehat{B} & -\widehat{W}_{\alpha^{-1}} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ \mathbf{g} \end{bmatrix}. \tag{3.6}$$

Next, we state the equivalence result of our main interest. That result allows us to characterize the energy norm associated with the hybridized system (2.7), namely, that in the $H(\operatorname{div})$-case, this norm is equivalent to a discrete H^1-norm. The latter is our main motivation for utilizing AMG solvers which are designed for H^1-like discrete systems and proven in practice to be highly scalable.

PROPOSITION 3.1. *Suppose α is piecewise-constant on the triangulation $\mathcal{T}_h$ and $\mathcal{H}_h$ is taken to be RT_h^k. Then, by using the same constraint matrix C as in* (3.6)*, the hybridized system H in* (2.7) *resulting from algebraic hybridization of* (2.2) *is identical to $\widetilde{H}$ in* (3.6) *obtained from the hybridized mixed discretization of* (3.3)*, i.e.,*

$$H = \widetilde{H}.$$

Proof. Note that $\widehat{W}_{\alpha^{-1}}$ is a mass matrix scaled by α^{-1}, so $\widehat{W}_{\alpha^{-1}}$ is invertible and consequently

$$\widetilde{H} := \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} \widehat{M}_\beta & \widehat{B}^T \\ \widehat{B} & -\widehat{W}_{\alpha^{-1}} \end{bmatrix}^{-1} \begin{bmatrix} C^T \\ 0 \end{bmatrix} = C \left(\widehat{M}_\beta + \widehat{B}^T (\widehat{W}_{\alpha^{-1}})^{-1} \widehat{B} \right)^{-1} C^T.$$

Therefore, $H = \widetilde{H}$ if and only if $\widehat{A} = \widehat{M}_\beta + \widehat{B}^T (\widehat{W}_{\alpha^{-1}})^{-1} \widehat{B}$. Moreover, since these matrices are block-diagonal (each block corresponding to a single element), it suffices to show the above equality on each element τ, that is,

$$A_\tau = M_{\beta,\tau} + B_\tau^T (W_{\alpha^{-1},\tau})^{-1} B_\tau. \tag{3.7}$$

To see the connection between the two, we notice that the element matrix A_τ comes from the element bilinear form

$$a_{\mathrm{div},\tau}(\widehat{\mathbf{u}}_h, \widehat{\mathbf{v}}_h) := \alpha_\tau \int_\tau \operatorname{div} \widehat{\mathbf{u}}_h \operatorname{div} \widehat{\mathbf{v}}_h + \int_\tau \beta \widehat{\mathbf{u}}_h \widehat{\mathbf{v}}_h,$$

where we have used the fact that α is piecewise-constant: $\alpha \equiv \alpha_\tau$ on τ. By the property of the discrete de Rham sequence, we have $\operatorname{div}\left(RT_h^k(\tau)\right) = Q_h^k(\tau)$; cf. [5, section 2.5.1]. Thus, the integral $\int_\tau \operatorname{div} \widehat{\mathbf{u}}_h \operatorname{div} \widehat{\mathbf{v}}_h$ is an $L^2(\tau)$ inner product between two functions $\operatorname{div} \widehat{\mathbf{u}}_h$ and $\operatorname{div} \widehat{\mathbf{v}}_h$ in Q_h^k. Hence, the element matrix A_τ can be decomposed as

$$A_\tau = \alpha_\tau D_\tau^T W_\tau D_\tau + M_{\beta,\tau}, \tag{3.8}$$

where D_τ is the matrix representation of the map $\operatorname{div} : RT_h^k(\tau) \to Q_h^k(\tau)$. Similarly, the matrix B_τ, which is the matrix representation of the integral $\int_\tau \operatorname{div} \widehat{\mathbf{u}}_h p_h$, can be decomposed as

$$B_\tau = W_\tau D_\tau \tag{3.9}$$

Now, since α is piecewise-constant,

$$(W_{\alpha^{-1},\tau})^{-1} = (\alpha_\tau^{-1} W_\tau)^{-1} = \alpha_\tau (W_\tau)^{-1}. \tag{3.10}$$

Finally, we conclude the proof by observing that (3.7) is a direct consequence of (3.8), (3.9), and (3.10). ∎

Remark 3.2. If we use in the hybridized mixed discretization of (3.6) the "broken" Brezzi–Douglas–Marini space of order $k+1$, $\widehat{BDM}_h^{k+1}$, Proposition 3.1 is still valid by taking $\mathcal{H}_h$ as BDM_h^{k+1} instead of RT_h^k. This is because the compatibility condition $\operatorname{div}\left(BDM_h^{k+1}(\tau)\right) = Q_h^k(\tau)$ also holds for Brezzi–Douglas–Marini elements; see, for example, [5, section 2.5.1].

As mentioned earlier in this section, the relation in Proposition 3.1 suggests that AMG is likely to be a good solver for (2.7). We will discuss this in more detail in the following section.

4. Scalable algebraic $H(\mathrm{div})$ preconditioning. In this section we propose several scalable AMG approaches for finite element discretizations of the $H(\mathrm{div})$ form $a_{\mathrm{div}}(\cdot,\cdot)$ based on the methods discussed in the previous sections. Due to their scalability on challenging practical problems, we base our preconditioners on the parallel

AMG approaches available in the *hypre* library [3]. Specifically, we take advantage of the BoomerAMG [27], AMS [32], and ADS [33] methods in *hypre*, targeting H^1, $H(\mathrm{curl})$, and $H(\mathrm{div})$ discretization problems, respectively.

We first consider the static condensation Schur complement S, (3.2), and discuss approaches for its preconditioning. Since S acts in general on traces of functions from a given space ($H(\mathrm{div})$, $H(\mathrm{curl})$, or H^1), a suitable preconditioner for S will be some approximation to the Schur complement of an (optimal) preconditioner for the original matrix A. Of course, for efficiency, we would like to build a preconditioner directly for the (sparse) matrix S and not for the original matrix A. Such preconditioners are discussed in [11], where it is shown that BoomerAMG and AMS work well on Schur complements of H^1 and $H(\mathrm{curl})$ discretizations, respectively. Similar analysis for the $H(\mathrm{div})$ case was recently performed in [4]. In all three cases, the summary of the stable decomposition theory is that if BoomerAMG/AMS/ADS works for a particular problem, it will also work, purely algebraically, for a Schur complement of the problem. Thus, the $H(\mathrm{div})$ Schur complement S considered in this paper can be efficiently preconditioned with a direct application of ADS as discussed and demonstrated in [4].

We next consider preconditioning approaches for the hybridized matrix H. Based on the connection with mixed methods from the previous section we argue that one suitable preconditioner for the hybridized Schur complement H in the case of the $H(\mathrm{div})$ problem is an H^1-based AMG built from the matrix H. The rest of this section provides arguments for this choice, by exploring the "near-nullspace" of H, i.e., the set of its low-frequency eigenmodes, which is critical in AMG theory.

Consider the Lagrangian multiplier space $\lambda_h \in M_h^k$ associated with the given finite element mesh $\mathcal{T}_h$. Let $m_\tau(\lambda_h) = \frac{1}{|\partial\tau|}\int_{\partial\tau} \lambda_h \, ds$; then the expressions

$$|\lambda_h|^2 = \sum_{\tau\in\mathcal{T}_h} \left(\frac{1}{h_\tau} \|\lambda_h - m_\tau(\lambda_h)\|_{\partial\tau}^2 \right), \quad |||\lambda_h|||^2 = \sum_{\tau\in\mathcal{T}_h} \left(h_\tau \|\lambda_h\|_{\partial\tau}^2 \right) + |\lambda_h|^2$$

define a seminorm and a norm on M_h^k. The coefficient vector for $\lambda_h \in M_h^k$ is denoted by $\boldsymbol{\lambda}$. In [17], the following spectral equivalence is established:

$$\boldsymbol{\lambda}^T H \boldsymbol{\lambda} \simeq |||\lambda_h|||^2.$$

This result implies that H has a small near-nullspace containing only the constant functions, as opposed to the large near-nullspace of the original system A. Therefore, the hybridized Schur complement H is "H^1-like," which motivates the use of H^1-based AMG for preconditioning.

Remark 4.1. For lowest-order Raviart–Thomas elements, it is well-known (cf. [2, 35, 14]) that the hybridized system of mixed method $\widetilde{H}$ is equivalent to the algebraic system arising from H^1-nonconforming discretization of second-order scalar elliptic problems. In particular, in the case when β is piecewise-constant and $\widehat{W}_{\alpha^{-1}} = 0$ in (3.6), it was shown in [14] that the two systems are identical: there exists an H^1-nonconforming space S_h such that for any $\lambda_h \in M_h^k$, there exists $\psi_h \in S_h$ such that

$$\boldsymbol{\lambda}^T \widetilde{H} \boldsymbol{\lambda} = \sum_{\tau\in\mathcal{T}_h} (\beta^{-1}\nabla\psi_h, \nabla\psi_h)_\tau.$$

This also suggests the H^1-like nature of the hybridized system H. For more general higher-order discretizations, the equivalence between hybridization of mixed methods

and H^1-nonconforming discretizations was shown in [2, 1]. As an application of the equivalence (see, e.g., [6, 14, 1]), one solution strategy for the hybridized mixed finite element problems is to first solve the equivalent nonconforming discrete problems using geometric multigrid algorithms, and then map the solutions of the latter to the solutions of the original hybridized mixed finite element problems using the equivalence established. In contrast, in the current paper we examine the near-nullspace of $\widetilde{H}$ from an algebraic perspective (see Lemma 4.2 and Proposition 4.3 below), and our proposed solver is an algebraic multigrid solver that solves the hybridized system directly.

In the remainder of this section, we show that H has a small near-nullspace from a more algebraic point of view.

First, by exploiting $\widetilde{H}$, one can see that H has the explicit form

$$H = \widetilde{H} = C(\widehat{M}_\beta^{-1} - \widehat{M}_\beta^{-1}\widehat{B}^T(\widehat{W}_{\alpha^{-1}} + \widehat{B}\widehat{M}_\beta^{-1}\widehat{B}^T)^{-1}\widehat{B}\widehat{M}_\beta^{-1})C^T. \tag{4.1}$$

By Theorem 3.6 in [17] and a Poincaré type of inequality (cf. proof of Theorem 2.3 in [25]), there exist mesh-independent constants K_1, K_2, and K_3 such that

$$\boldsymbol{\lambda}^T C\widehat{Y}C^T\boldsymbol{\lambda} \le \boldsymbol{\lambda}^T H\boldsymbol{\lambda} \le K_1|||\lambda_h|||^2 \le K_2|\lambda_h|^2 \le K_3\big(\boldsymbol{\lambda}^T C\widehat{Y}C^T\boldsymbol{\lambda}\big) \qquad \forall \boldsymbol{\lambda}, \tag{4.2}$$

where $\widehat{Y} = \widehat{M}_\beta^{-1} - \widehat{M}_\beta^{-1}\widehat{B}^T(\widehat{B}\widehat{M}_\beta^{-1}\widehat{B}^T)^{-1}\widehat{B}\widehat{M}_\beta^{-1}$.

That is, H is spectrally equivalent to $C\widehat{Y}C^T$. In what follows, we first study the nullspace of $\widehat{Y}$.

LEMMA 4.2. *The nullspace of $\widehat{Y}$ coincides with the range space of $\widehat{B}^T$, i.e.,*

$$\mathrm{Null}(\widehat{Y}) = \mathrm{Range}(\widehat{B}^T).$$

Proof. Note that for any $\widehat{\mathbf{x}} \in \mathrm{Range}(\widehat{B}^T)$, there exists $\widehat{\mathbf{y}}$ such that $\widehat{\mathbf{x}} = \widehat{B}^T\widehat{\mathbf{y}}$, so

$$\widehat{Y}\widehat{\mathbf{x}} = \widehat{M}_\beta^{-1}\widehat{\mathbf{x}} - \widehat{M}_\beta^{-1}\widehat{B}^T(\widehat{B}\widehat{M}_\beta^{-1}\widehat{B}^T)^{-1}\widehat{B}\widehat{M}_\beta^{-1}\widehat{B}^T\widehat{\mathbf{y}} = \widehat{M}_\beta^{-1}\widehat{\mathbf{x}} - \widehat{M}_\beta^{-1}\widehat{B}^T\widehat{\mathbf{y}} = 0.$$

Thus, we have

$$\mathrm{Range}(\widehat{B}^T) \subseteq \mathrm{Null}(\widehat{Y}). \tag{4.3}$$

On the other hand, let $\widehat{\mathbf{x}} \in \mathrm{Null}(\widehat{Y})$, then

$$\widehat{M}_\beta^{-1}\widehat{\mathbf{x}} - \widehat{M}_\beta^{-1}\widehat{B}^T(\widehat{B}\widehat{M}_\beta^{-1}\widehat{B}^T)^{-1}\widehat{B}\widehat{M}_\beta^{-1}\widehat{\mathbf{x}} = 0.$$

Consequently, $\widehat{\mathbf{x}} = \widehat{B}^T(\widehat{B}\widehat{M}_\beta^{-1}\widehat{B}^T)^{-1}\widehat{B}\widehat{M}_\beta^{-1}\widehat{\mathbf{x}}$. So for any $\widehat{\mathbf{y}} \in \mathrm{Null}(\widehat{B})$, we have

$$\begin{aligned}\langle\widehat{\mathbf{x}}, \widehat{\mathbf{y}}\rangle &= \Big\langle \widehat{B}^T(\widehat{B}\widehat{M}_\beta^{-1}\widehat{B}^T)^{-1}\widehat{B}\widehat{M}_\beta^{-1}\widehat{\mathbf{x}}, \widehat{\mathbf{y}}\Big\rangle \\ &= \Big\langle (\widehat{B}\widehat{M}_\beta^{-1}\widehat{B}^T)^{-1}\widehat{B}\widehat{M}_\beta^{-1}\widehat{\mathbf{x}}, \widehat{B}\widehat{\mathbf{y}}\Big\rangle = 0.\end{aligned}$$

Hence,

$$\mathrm{Null}(\widehat{Y}) \subseteq \mathrm{Null}(\widehat{B})^\perp = \mathrm{Range}(\widehat{B}^T).$$

The above inclusion relation and (4.3) imply that

$$\mathrm{Null}(\widehat{Y}) = \mathrm{Range}(\widehat{B}^T). \qquad \square$$

The final result in this section characterizes the "near-nullspace" of $C\widehat{Y}C^T$. That is, by modifying C to $\widetilde{C}$ to include boundary faces, we end up with a modified hybridized matrix $\widetilde{C}\widehat{Y}\widetilde{C}^T$ which happens to have a nontrivial nullspace. To be precise, the modified matrix $\widetilde{C}$ corresponds to the bilinear form

$$\widetilde{C}_h(\widehat{\mathbf{u}}_h, \mu_h) = \sum_{\tau \in \mathcal{T}_h} \int_{\partial\tau} (\widehat{\mathbf{u}}_h \cdot \mathbf{n}_\tau)\mu_h. \tag{4.4}$$

If we compare with the original C (cf. (3.5)), $\widetilde{C}$ includes extra Lagrange multiplier dofs from the domain boundary. Note that $C\widehat{Y}C^T$ and $\widetilde{C}\widehat{Y}\widetilde{C}^T$ are the same away from the domain boundary.

PROPOSITION 4.3. *Let* $\mathbf{1}$ *be the coefficient vector of the constant* 1 *function in* Q_h^k *(recall that* Q_h^k *is the discrete subspace of* $L^2(\Omega)$ *in section* 3.2*). Define the constraint matrix* $\widetilde{C}$ *to come from the modified bilinear form* (4.4)*; then*

$$\mathrm{Null}(\widetilde{C}\widehat{Y}\widetilde{C}^T) = \mathrm{span}\left\{(\widetilde{C}\widetilde{C}^T)^{-1}\widetilde{C}\widehat{B}^T\mathbf{1}\right\}. \tag{4.5}$$

Proof. Note that $\widehat{Y}$ is symmetric positive semidefinite, so $\widehat{Y}^{\frac{1}{2}}$ is a well-defined real matrix, and by Lemma 4.2, we have

$$\mathrm{Null}(\widehat{Y}^{\frac{1}{2}}) = \mathrm{Null}(\widehat{Y}) = \mathrm{Range}(\widehat{B}^T).$$

Therefore, we can characterize $\mathrm{Null}(\widetilde{C}\widehat{Y}\widetilde{C}^T)$ as

$$\boldsymbol{\lambda} \in \mathrm{Null}(\widetilde{C}\widehat{Y}\widetilde{C}^T) \quad \Leftrightarrow \quad \widehat{Y}^{\frac{1}{2}}(\widetilde{C}^T\boldsymbol{\lambda}) = 0 \quad \Leftrightarrow \quad \widetilde{C}^T\boldsymbol{\lambda} \in \mathrm{Range}(\widehat{B}^T). \tag{4.6}$$

Obviously, $\widetilde{C}^T\boldsymbol{\lambda} \in \mathrm{Range}(\widetilde{C}^T)$, so (4.6) suggests that

$$\boldsymbol{\lambda} \in \mathrm{Null}(\widetilde{C}\widehat{Y}\widetilde{C}^T) \quad \Leftrightarrow \quad \widetilde{C}^T\boldsymbol{\lambda} \in \mathrm{Range}(\widetilde{C}^T) \cap \mathrm{Range}(\widehat{B}^T). \tag{4.7}$$

Next, we show that $\mathrm{Range}(\widetilde{C}^T) \cap \mathrm{Range}(\widehat{B}^T)$ is one-dimensional. Note that the constraint matrix $\widetilde{C}$ comes from the bilinear form (4.4). Thus, $\widetilde{C}_h(\widehat{\mathbf{u}}_h, \mu_h) = 0$ enforces the continuity of the master and slave dofs of $\widehat{\mathbf{u}}_h$ on the interface and additionally enforces zero values for the dofs on the boundary. So there exists a matrix P such that $\mathrm{Null}(\widetilde{C}) = \mathrm{Range}(P)$. Note that the columns of P are coefficient vectors of the basis chosen for the functions with matching interface values and zero boundary values, (cf. section 2.3). Then, we have

$$\mathrm{Range}(\widetilde{C}^T) \cap \mathrm{Range}(\widehat{B}^T) = \mathrm{Null}(\widetilde{C})^\perp \cap \mathrm{Range}(\widehat{B}^T) = \mathrm{Range}(P)^\perp \cap \mathrm{Range}(\widehat{B}^T).$$

This means that

$$\widehat{B}^T\mathbf{q} \in \mathrm{Range}(\widetilde{C}^T) \cap \mathrm{Range}(\widehat{B}^T) \quad \Leftrightarrow \quad P^T(\widehat{B}^T\mathbf{q}) = 0.$$

Recall that $\widehat{B}$ comes from the elementwise bilinear form

$$\sum_{\tau \in \mathcal{T}_h} \int_\tau (\mathrm{div}\, \widehat{\mathbf{u}}_h)\, p_h.$$

Since P enforces continuity and zero boundary value of $\widehat{\mathbf{u}}_h$, $P^T\widehat{B}^T = (\widehat{B}P)^T$ corresponds to the bilinear form on the conforming space with zero boundary condition

$$B_h(\mathbf{u}_h, p_h) = \int_\Omega (\mathrm{div}\, \mathbf{u}_h)\, p_h.$$

The only functions p_h satisfying $B_h(\mathbf{u}_h, p_h) = 0$ for all $\mathbf{u}_h$ are the constant functions. Hence, we deduce that

$$\text{Range}(\widetilde{C}^T) \cap \text{Range}(\widehat{B}^T) = \text{span}\{\widehat{B}^T\mathbf{1}\}. \tag{4.8}$$

Note that $\text{Null}(\widetilde{C}^T) = \{0\}$ (i.e., $\widetilde{C}^T$ is injective), so

$$\widetilde{C}^T\boldsymbol{\lambda} \in \text{span}\{\widehat{B}^T\mathbf{1}\} \quad \Leftrightarrow \quad \boldsymbol{\lambda} \in \text{span}\left\{(\widetilde{C}\widetilde{C}^T)^{-1}\widetilde{C}\widehat{B}^T\mathbf{1}\right\}. \tag{4.9}$$

Finally, (4.5) is a direct consequence of (4.7), (4.8), and (4.9). ∎

Although it has been shown that the hybridized system H is H^1-like [1, 2, 14, 17], AMG solvers in *hypre* may not work properly if the near-nullspace of H is not spanned by constant vectors. An important consequence of Proposition 4.3 is that it gives an explicit basis vector for the one-dimensional near-nullspace of H, which can be used to diagonally rescale H such that constant vectors are in the near-nullspace of the rescaled system. More precisely, if $\widehat{B}$ is available and $d := (\widetilde{C}\widetilde{C}^T)^{-1}\widetilde{C}\widehat{B}^T\mathbf{1}$ does not have zero entries, we can take D as the diagonal matrix such that $D_{ii} = d_i$; then the rescaled system D^THD will have the proper scaling that AMG is expecting. Thus, $\boldsymbol{\lambda}$ in (2.7) can be solved efficiently by solving

$$(D^THD)\boldsymbol{\lambda}_D = D^TC\widehat{A}^{-1}\widehat{\mathbf{f}}, \quad \boldsymbol{\lambda} = D\boldsymbol{\lambda}_D.$$

Remark 4.4. In the MFEM finite element library [37], the default choice of constraint matrix C gives $(\widetilde{C}\widetilde{C}^T)^{-1}\widetilde{C}\widehat{B}^T\mathbf{1} = (1, 1, \ldots, 1)^T$. Therefore, the hybridized system H in MFEM automatically has the correct scaling that is suitable for AMG.

Remark 4.5. Note that $\widehat{B}$ is readily available from discretization if mixed finite element problem (3.5) is being solved. In the general case where $\widehat{B}$ is not available, the scaling vector can be obtained using the method proposed in [34, section 3].

Remark 4.6. Since the algebraic results Lemma 4.2 and Proposition 4.3 are based on the structure of $\widetilde{H}$ (4.1), obviously they are also valid for hybridization of mixed finite element discretization of (3.3). In particular, they remain valid for the second-order scalar elliptic problems without the zeroth-order term:

$$-\text{div}(\beta^{-1}\nabla q) = -g.$$

In fact, the hybridized system in this case is exactly $C\widehat{Y}C^T$ and so we do not even need to consider the equivalence (4.2). Hence, AMG is also a suitable solver for hybridized system $\widetilde{H}$ of mixed finite element methods.

5. Implementation details. In this section we provide some details for the practical application of the presented approaches as implemented in the finite element library MFEM [37].

The implementation is contained in class `code Hybridization`, which can be used either directly or more conveniently through the `code BilinearForm` class. The prolongation operator P is not directly constructed. Instead, its action can be performed using the methods in class `code FiniteElementSpace`. The constraint matrix C is constructed as in Example 2.2, i.e., based on a specified multiplier finite element space and a given bilinear form.

During a construction stage, the element matrices are reordered and stored in the 2×2 block form (3.1); see Figure 5.1 for an illustration. Recall that the i- and b-block

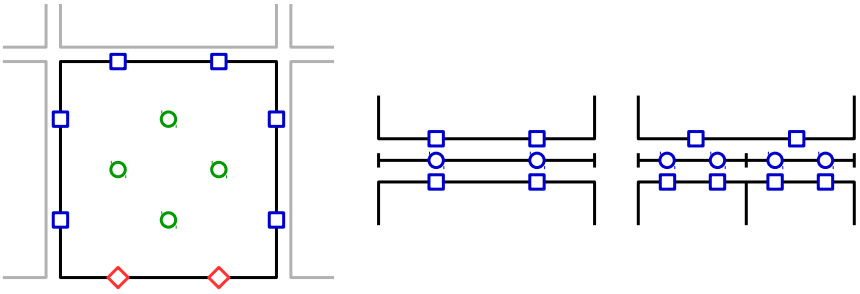

FIG. 5.1. *Left: local RT_1 dofs: i-dofs (green circles), b-dofs (blue squares), and e-dofs (red diamonds). Center/right: dofs associated with a conforming (center) and a nonconforming (right) mesh face: RT_1 dofs to match (blue squares) and multiplier dofs (blue circles).*

unknowns correspond to the element interior and boundary (or interface) dofs. Any rows and columns associated with essential degrees of freedom (e-dofs) are eliminated. Algebraically, the interior dofs can be defined as those corresponding to zero columns in the C matrix. In other words, C has the 1×2 block form

$$C = \begin{bmatrix} 0 & C_b \end{bmatrix}.$$

As a consequence, the matrix H of the hybridized system (2.7) can be computed as

$$H = C\widehat{A}^{-1}C^T = C_b\widehat{S}_b^{-1}C_b^T,$$

where $\widehat{S}_b = \widehat{A}_{bb} - \widehat{A}_{bi}\widehat{A}_{ii}^{-1}\widehat{A}_{ib}$ is the Schur complement of $\widehat{A}$ which can be formed independently for each element in the mesh. In MFEM, this is performed using elementwise block LU factorizations of the form

$$\begin{bmatrix} \widehat{A}_{ii} & \widehat{A}_{ib} \\ \widehat{A}_{bi} & \widehat{A}_{bb} \end{bmatrix} = \begin{bmatrix} \widehat{L}_i & 0 \\ \widehat{L}_{bi} & \widehat{I}_b \end{bmatrix} \begin{bmatrix} \widehat{U}_i & \widehat{U}_{ib} \\ 0 & \widehat{S}_b \end{bmatrix}$$

to compute the Schur complements $\widehat{S}_b$, which are then themselves LU-factorized, allowing simple multiplication of any vector or matrix with $\widehat{S}_b^{-1}$.

Note that for Robin-type boundary conditions, one needs to assemble contributions to the global system matrix A coming from integration over boundary faces. In MFEM, such face matrices are assembled into the element matrix of the element adjacent to the corresponding face. The rest of the hybridization procedure remains the same.

6. Numerical tests. To validate the parallel scalability of the proposed solvers, in this section we present several weak scaling tests, where the number of dofs per processor is held to be about the same, while the number of processors is increased. All the experiments, except the ones in section 6.3, are performed on the cluster Sierra at Lawrence Livermore National Laboratory (LLNL). Sierra has a total of 1944 nodes (Intel Xeon EP X5660 @ 2.80 GHz), each of which has 12 cores and 24 GB of memory. The numerical results are generated using the libraries MFEM [37] and *hypre* [29] developed at LLNL. Furthermore, in all the examples except the ones in section 6.3, we solve (2.1) with right-hand-side $f(\mathbf{v}) := \int_\Omega \mathbf{g}\cdot\mathbf{v}\,d\mathbf{x}$, where $\mathbf{g} = [1,1,1]^T$. The actual domain and values of coefficients α and β will be specified below in each example.

Note that in the setup phase, the main computational cost of both hybridization and static condensation is the inversion of some local matrices, which is intrinsically parallel. More precisely, the reduced system (2.7) in hybridization requires inversion of $\widehat{A}$, while the reduced system (3.2) in static condensation requires inversion of $\widehat{A}_{ii}$. Inversions of both $\widehat{A}$ and $\widehat{A}_{ii}$ are perfectly parallel because they are block-diagonal. When solving reduced systems, we rely on various algebraic solvers in *hypre*, which are highly scalable [3]. So, by and large, both the hybridization and the static condensation solvers are expected to exhibit good parallel scalability.

6.1. Soft/hard materials. Consider solving (2.1) on the unit cube $\Omega = [0,1]^3$. Let $\Omega_i = [\frac{1}{4}, \frac{1}{2}]^3 \cup [\frac{1}{2}, \frac{3}{4}]^3$; see Figure 6.1. We consider $\alpha \equiv 1$ in Ω, whereas

$$\beta = \begin{cases} 1 & \text{in } \Omega \backslash \Omega_i, \\ 10^p & \text{in } \Omega_i, \end{cases} \quad p \in \{-8, -4, 0, 4, 8\}.$$

This example is intended to test the robustness of the solvers against coefficient jumps. We discretize (2.1) by Raviart–Thomas elements (RT_h^k) on hexahedral meshes. In order to better contrast the effect of higher-order discretizations, the initial meshes in the weak scaling tests are chosen so that the problem size is the same for discretizations of different orders; see Table 6.1. The PCG stopping criterion is a reduction in the l_2 residual by a factor of 10^{-12}. The time to solution and number of PCG iterations (in parentheses) for the proposed approaches, as well as ADS, are reported in Tables 6.2–6.4. Here, the time to solution for the hybridization approach includes the formation time of the hybridization Schur complement H, the setup time of AMG (BoomerAMG [27]), PCG solving time, and the time for recovering the original solution by back substitution. Similarly, the time to solution for the static condensation approach includes the formation time of S (cf. section 3), ADS setup time, PCG solving time, and back substitution. Last, the time to solution for "ADS" is simply ADS setup time and PCG solving time. Our results show that all the solvers have good weak scaling and robustness with respect to coefficient jumps. In all the cases, while both the hybridization and static condensation solvers give shorter solving time than ADS, the

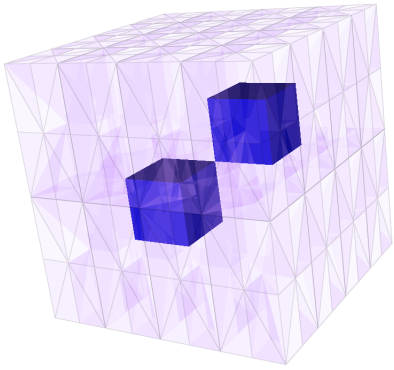

FIG. 6.1. *Domains for the soft/hard materials example. Ω_i is in blue.*

TABLE 6.1
Initial mesh sizes in the weak scaling tests in section 6.1.

Finite element	RT_0	RT_1	RT_3
Initial mesh	$64 \times 64 \times 32$	$32 \times 32 \times 16$	$16 \times 16 \times 8$

TABLE 6.2
Solving time (seconds) and #iter (in parentheses) of different solvers: RT_0 on hexahedral meshes.

#proc	Size	$p=-8$	$p=-4$	$p=0$	$p=4$	$p=8$
Hybridization + AMG						
3	401,408	3.39 (24)	3.5 (26)	3.55 (27)	3.56 (27)	3.44 (26)
24	3,178,496	7.29 (29)	7.43 (31)	7.34 (30)	7.29 (30)	7.35 (30)
192	25,296,896	8.28 (33)	8.54 (37)	8.45 (32)	8.3 (32)	8.3 (33)
1,536	201,850,880	9.79 (37)	10.33 (39)	9.98 (36)	10.28 (38)	10.17 (36)
ADS						
3	401,408	17.35 (16)	17.51 (16)	15.94 (13)	20.54 (19)	20.11 (21)
24	3,178,496	36.94 (18)	36.91 (18)	35.69 (16)	38.76 (21)	40 (23)
192	25,296,896	42.54 (20)	42.39 (20)	40.99 (17)	43.75 (22)	45.7 (25)
1,536	201,850,880	49.91 (21)	49.64 (21)	47.73 (19)	52.93 (25)	52.62 (27)

TABLE 6.3
Solving time (seconds) and #iter (in parentheses) of different solvers: RT_1 on hexahedral meshes.

#proc	Size	$p=-8$	$p=-4$	$p=0$	$p=4$	$p=8$
Hybridization + AMG						
3	401,408	3.35 (27)	3.39 (28)	3.46 (28)	3.43 (29)	3.46 (28)
24	3,178,496	5.97 (30)	6.14 (33)	6.06 (32)	5.95 (31)	6.11 (32)
192	25,296,896	6.85 (36)	7.03 (37)	7.25 (36)	7 (35)	7.43 (35)
1,536	201,850,880	7.84 (39)	9.17 (41)	8.17 (39)	8.83 (41)	9.85 (40)
Static condensation + ADS						
3	401,408	19.69 (15)	19.65 (15)	18.23 (12)	20.92 (17)	21.91 (18)
24	3,178,496	40.5 (19)	40.69 (19)	36.77 (15)	42.42 (21)	43.51 (22)
192	25,296,896	46.48 (21)	46.81 (21)	42.63 (17)	48.68 (23)	51.62 (26)
1,536	201,850,880	54.78 (23)	54.43 (22)	50.38 (19)	59.4 (26)	61.56 (29)
ADS						
3	401,408	34.73 (18)	34.53 (17)	30.66 (14)	38.93 (22)	41.44 (24)
24	3,178,496	60.9 (20)	61.13 (20)	55.19 (16)	68.06 (25)	71.44 (27)
192	25,296,896	68.55 (22)	67.39 (21)	62.5 (18)	77.74 (28)	81.1 (30)
1,536	201,850,880	78.8 (24)	80.99 (24)	72.89 (20)	90.74 (31)	102.93 (37)

TABLE 6.4
Solving time (seconds) and #iter (in parentheses) of different solvers: RT_3 on hexahedral meshes.

#proc	Size	$p=-8$	$p=-4$	$p=0$	$p=4$	$p=8$
Hybridization + AMG						
3	401,408	9.27 (33)	9.46 (35)	9.42 (37)	9.62 (38)	9.51 (38)
24	3,178,496	13.1 (41)	12.91 (43)	14 (44)	13.53 (44)	13.25 (44)
192	25,296,896	14.81 (48)	15.58 (50)	15.26 (50)	14.71 (50)	16.4 (50)
1,536	201,850,880	16.8 (53)	16.33 (56)	16.63 (56)	17.67 (56)	19.06 (56)
Static condensation + ADS						
3	401,408	31.38 (12)	31.38 (12)	29.28 (10)	32.65 (13)	32.38 (13)
24	3,178,496	56.09 (15)	55.87 (15)	54.64 (14)	59.06 (17)	59.03 (17)
192	25,296,896	67.01 (18)	67.28 (18)	65.22 (17)	70.43 (20)	70.53 (20)
1,536	201,850,880	83.81 (20)	82.14 (20)	81.69 (19)	85.65 (22)	86.65 (23)
ADS						
3	401,408	184.67 (18)	186.1 (18)	169.27 (14)	188.58 (19)	189.52 (20)
24	3,178,496	253.3 (19)	259.11 (20)	240.87 (16)	269.12 (22)	284.04 (25)
192	25,296,896	288.32 (22)	289.14 (22)	261.95 (17)	305.86 (25)	325.93 (29)
1,536	201,850,880	329.76 (24)	330.46 (23)	301.61 (19)	353.63 (28)	366.94 (31)

hybridization approach is the fastest; see Figure 6.2. Moreover, as the finite element order goes higher, discrepancy in the solution time between the solvers becomes more significant even though the problem size is the same. In particular, for the RT_3 discretization, hybridization is in general 4 times faster than static condensation and 20 times faster than ADS.

6.2. The crooked pipe problem. This example deals with large coefficient jumps and highly stretched elements; see Figure 6.3 for an illustration of the mesh in this problem. The coefficients α and β are both piecewise-constant functions such that

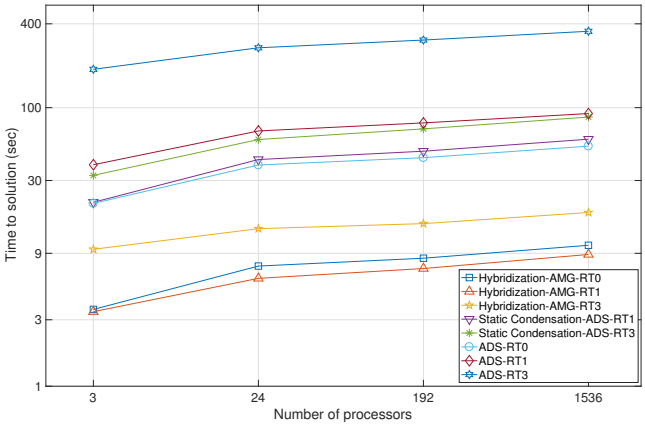

FIG. 6.2. *Weak scaling plots in* log-log *scale for the soft/hard materials example when* $p = 4$.

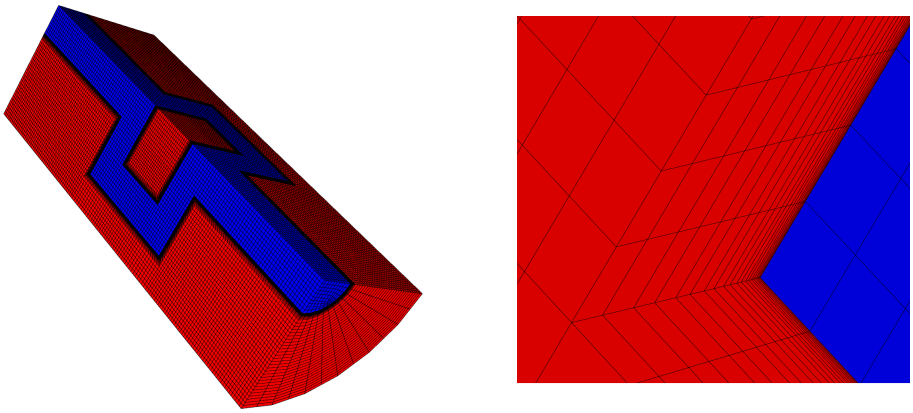

FIG. 6.3. *Geometry and material interface of the crooked pipe example.*

$$(\alpha, \beta) = \begin{cases} (1.641, 0.2) & \text{in red region,} \\ (0.00188, 2000) & \text{in blue region.} \end{cases}$$

Again, the initial meshes for lower-order approximations are refined so that the problem size is the same for discretizations of different approximation orders, and the weak scaling results are reported in Table 6.5. Note that we were running out of memory when setting up ADS for RT_3 (indicated by "$*$"), which shows that the hybridization and static condensation approaches are more memory-friendly for high-order discretizations. The results in Table 6.5 again suggest that hybridization with AMG is the most efficient solver among the three for this particular problem; it is also observed that the iteration counts in the hybridization approach are more steady than the other two.

6.3. Application to radiation diffusion. In this section we compare the proposed approaches when they are used for solving the radiation diffusion equations in flux form; see [10]. This model is often used in astrophysical and inertial confinement fusion simulations. We consider a problem with two materials ($k = 1, 2$ below),

TABLE 6.5
Solving time (seconds) and #iter (in parentheses) for the crooked pipe example.

#proc	Size	RT_0	RT_1	RT_3
Hybridization + AMG				
12	2,805,520	8.56 (34)	8.44 (33)	22.41 (42)
96	22,258,240	13.4 (37)	12.45 (36)	29.97 (44)
768	177,322,240	17.82 (45)	15.28 (45)	37.13 (49)
6,144	1,415,603,200	23.73 (54)	21.36 (55)	43.01 (60)
Static condensation + ADS				
12	2,805,520	73.76 (52)	98.73 (43)	190.53 (40)
96	22,258,240	126.87 (80)	177.71 (74)	305.51 (67)
768	177,322,240	191.67 (117)	291.03 (122)	481.28 (107)
6,144	1,415,603,200	274.09 (168)	463.95 (190)	715.79 (158)
ADS				
12	2,805,520	73.76 (52)	174.55 (59)	$*$ $(*)$
96	22,258,240	126.87 (80)	291.4 (93)	$*$ $(*)$
768	177,322,240	191.67 (117)	442 (139)	$*$ $(*)$
6,144	1,415,603,200	274.09 (168)	706.5 (212)	$*$ $(*)$

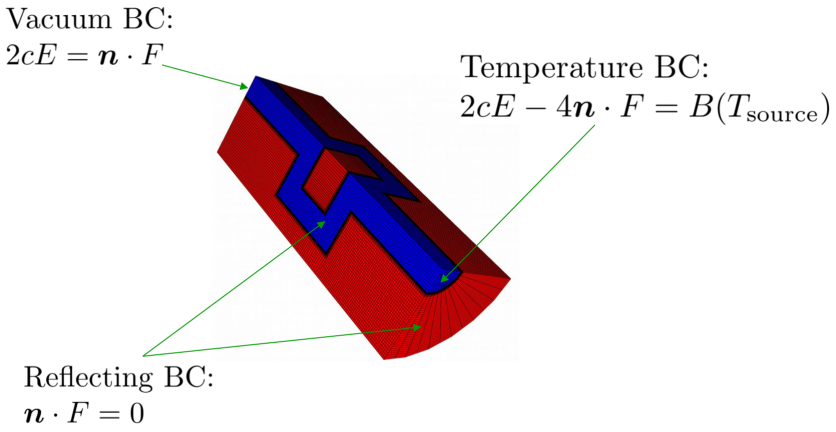

FIG. 6.4. *Boundary conditions for the radiation diffusion problem.*

indicated by blue and red in Figure 6.4. There are no mixed zones. The governing equations are

$$\begin{aligned} \rho_k \frac{de_k}{dt} &= c\sigma_k(E - B(T_k)), \\ \frac{dE}{dt} + \nabla \cdot F &= -c\sum_k \sigma_k(E - B(T_k)), \\ \frac{1}{3}\nabla E &= -\sum_k \frac{\sigma_k}{c} F, \end{aligned} \tag{6.1}$$

subject to the boundary conditions specified in Figure 6.4. The system (6.1) is solved for the materials' specific internal energies e_k, the radiation energy E, and the radiation flux F. The material densities ρ_k are given functions. The constant c represents the speed of light. The energy exchange between materials and radiation field is controlled by the material opacities σ_k and the Planck's function for black-body radiation B, which are functions of the material temperatures $T_k(\rho_k, e_k)$. The temperature boundary condition (see Figure 6.4) heats up one of the ends of the pipe, causing an increase of the radiation field E and its diffusion throughout the domain.

TABLE 6.6
2D problem with #dof = 324712.

	RT_0		RT_1			RT_3		
Method	HB	AMS	HB	SC	AMS	HB	SC	AMS
Time	3.05	15.35	1.11	7.29	18.63	1.35	6.21	35.68
GMRES iterations	100	245	50	129	200	73	130	190
Final residual × 1e-5	2.09	5.34	9.65	3.76	6.64	3.61	3.58	2.01

TABLE 6.7
3D problem with #dof = 124208.

	RT_0		RT_1			RT_3		
Method	HB	ADS	HB	SC	ADS	HB	SC	ADS
Time	0.39	3.86	0.58	4.39	10.06	3.02	12.44	67.68
GMRES iterations	24	42	27	23	43	28	18	46
Final residual × 1e-5	8.23	3.87	1.48	7.00	2.54	1.01	0.882	1.19

The discrete approximations of E and e_k are in the finite element space $Q_r \subset L^2$, consisting of piecewise polynomials of degree r, while the flux F is approximated in the Raviart–Thomas space RT_r. For stability reasons, the time stepping of e_k and E is implicit, which results in a nonlinear problem in each time step, as $B(T_k(\rho_k, e_k))$ is nonlinear. Newton's method is used to solve the resulting system. As E and e_k are discontinuous, they can be eliminated locally during each Newton iteration, which is inexpensive and inherently parallel. Upon eliminating E and e_k, an $H(\mathrm{div})$ linear system for F is obtained, which we use to test the performance of the two proposed approaches, as well as AMS (2D) and ADS (3D).

Because each method ultimately solves a different linear system, this example is used to obtain better perception of the execution time needed for each method to obtain similar solution accuracy with respect to the original system. Starting with identical initial conditions, each computation performs exactly one linear solve, and then the F residual is calculated from the definition of the nonlinear system. Execution times are presented for computations reaching similar residuals. For this particular example, this is achieved by setting the relative tolerances to 1e-6, 1e-5, and 1e-12 for AMS/ADS, static condensation+AMS/ADS, and hybridization+AMG, respectively.

As observed in Tables 6.6 and 6.7, even though the hybridized system must be solved to a much lower relative tolerance, the hybridization approach still has the shortest solving time, followed by static condensation and then ADS. Except for RT_0 in the 2D case, the hybridization approach is at least 10 times faster than AMS/ADS, which is a substantial improvement.

6.4. Coefficient from SPE10 benchmark. Our last numerical example is the SPE10 benchmark coefficient [15] from the reservoir simulation community. This coefficient is well-known for its multiscale heterogeneity and anisotropy. The initial mesh is a $60 \times 220 \times 85$ hexahedral grid, with the size of each cell being $20 \times 10 \times 2$. β is taken to be the SPE10 coefficient (see Figure 6.5), and $\alpha \equiv 1$. The time spent on different components of the solution process for the hybridization approach and ADS are shown in Tables 6.8 and 6.9, respectively. This example also demonstrates the superiority, in terms of performance, of the hybridization solver.

7. Conclusion. In this paper we studied and compared two popular techniques in finite elements, hybridization and static condensation. When applied to $H(\mathrm{div})$ problems, both techniques lead to a reduced system of the same size. The formation of the reduced system involves inversion of independent local matrices, which can be done in parallel.

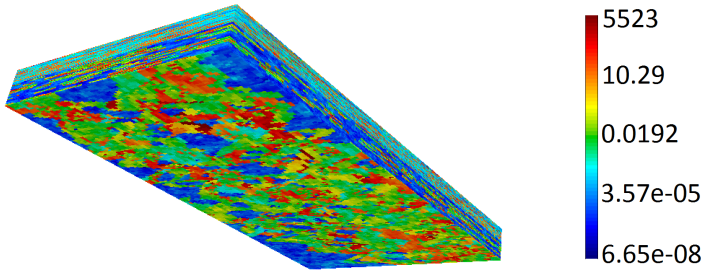

FIG. 6.5. *The SPE*10 *coefficient.*

TABLE 6.8
*The SPE*10 *example solved by hybridization and AMG.*

#proc	Size	Hybridize	AMG setup	PCG Solve (#iter)	Back sub.	Total time
3	3,403,000	21.3	2.21	18.6 (48)	0.51	42.62
24	27,076,000	68.75	3.76	23.59 (45)	0.62	96.72
192	216,016,000	111.82	6.93	27.85 (44)	0.63	147.22
1,536	1,725,760,000	131.85	9.6	29.58 (40)	0.65	171.68

TABLE 6.9
*The SPE*10 *example solved by ADS.*

#proc	Size	ADS setup	PCG solve (#iter)	Total time
3	3,403,000	85.77	405.76 (123)	491.53
24	27,076,000	290.17	1416.49 (300)	1706.66
192	216,016,000	492.11	3529.19 (635)	4021.3
1,536	1,725,760,000	585.7	5977.23 (959)	6562.93

We discussed suitable preconditioning strategies using various algebraic multigrid methods (AMG/AMS/ADS from *hypre* [29]) for the respective reduced systems and demonstrated that if these methods work well for the original problem, they also work well for the reduced systems obtained from static condensation. In the case of hybridization, we showed that its application to $H(\mathrm{div})$ problems leads to H^1-equivalent problems for which AMG is a well-suited preconditioner.

We presented several numerical examples comparing the performance of the proposed approaches, as well as solving the original problem directly by ADS/AMS. Our results show good weak scaling of all the proposed approaches. In general, for $H(\mathrm{div})$ problems, the approach using hybridization and AMG is more efficient than the approach using static condensation and ADS/AMS, while both of these approaches are faster than a direct application of ADS/AMS to the original problem. We observed substantial savings in the solve time when the hybridization approach is used, especially for higher-order discretizations.

Highly performant implementations of the two approaches are freely available in the MFEM finite element library [37].

REFERENCES

[1] T. ARBOGAST AND Z. CHEN, *On the implementation of mixed methods as nonconforming methods for second-order elliptic problems*, Math. Comp., 64 (1995), pp. 943–972, http://www.jstor.org/stable/2153478.

[2] D. N. ARNOLD AND F. BREZZI, *Mixed and nonconforming finite element methods: Implementation, postprocessing and error estimates*, ESAIM Math. Model. Numer. Anal., 19 (1985), pp. 7–32, http://eudml.org/doc/193443.

[3] A. Baker, R. Falgout, T. Kolev, and U. Yang, *Scaling hypre's multigrid solvers to* 100,000 *cores*, in High Performance Scientific Computing: Algorithms and Applications, Springer, New York, 2012, pp. 261–279.

[4] A. Barker, V. Dobrev, J. Gopalakrishnan, and V. Kolev, *A scalable preconditioner for a primal discontinuous Petrov-Galerkin method*, SIAM J. Sci. Comput., 40 (2018), pp. A1187–A1203.

[5] D. Boffi, F. Brezzi, and M. Fortin, *Mixed Finite Element Methods and Applications*, Springer Ser. Comput. Math. 44, Springer, New York, 2013, https://books.google.com/books?id=mRhAAAAAQBAJ.

[6] S. Brenner, *A multigrid algorithm for the lowest-order Raviart-Thomas mixed triangular finite element method*, SIAM J. Numer. Anal., 29 (1992), pp. 647–678, https://doi.org/10.1137/0729042.

[7] F. Brezzi, J. Douglas, and L. D. Marini, *Two families of mixed finite elements for second order elliptic problems*, Numer. Math., 47 (1985), pp. 217–235, https://doi.org/10.1007/BF01389710.

[8] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer, New York, 1991.

[9] F. Brezzi and D. Marini, *A survey on mixed finite element approximations*, IEEE Trans. Magnetics, 30 (1994), pp. 3547–3551, https://doi.org/10.1109/20.312705.

[10] T. A. Brunner, *Forms of Approximate Radiation Transport*, Technical Report SAND2002-1778, Sandia National Laboratory, 2002.

[11] T. A. Brunner and T. V. Kolev, *Algebraic multigrid for linear systems obtained by explicit element reduction*, SIAM J. Sci. Comput., 33 (2011), pp. 2706–2731.

[12] Z. Cai, R. Lazarov, T. A. Manteuffel, and S. F. McCormick, *First-order system least squares for second-order partial differential equations: Part* I, SIAM J. Numer. Anal., 31 (1994), pp. 1785–1799, http://www.jstor.org/stable/2158377.

[13] J. Cerveny, V. Dobrev, and T. Kolev, *Non-conforming mesh refinement for high-order finite elements*, submitted.

[14] Z. Chen, *Equivalence between and multigrid algorithms for mixed and nonconforming methods for second order elliptic problems*, East-West J. Numer. Math, 4 (1996), pp. 1–33.

[15] M. A. Christie and M. J. Blunt, *Tenth SPE Comparative Solution Project: A Comparison of Upscaling Techniques*, Society of Petroleum Engineers, Richardson, TX, 2001.

[16] B. Cockburn and J. Gopalakrishnan, *A characterization of hybridized mixed methods for second order elliptic problems*, SIAM J. Numer. Anal., 42 (2004), pp. 283–301, https://doi.org/10.1137/S0036142902417893.

[17] B. Cockburn and J. Gopalakrishnan, *Error analysis of variable degree mixed methods for elliptic problems via hybridization*, Math. Comp., 74 (2005), pp. 1653–1677.

[18] B. Cockburn and J. Gopalakrishnan, *New hybridization techniques*, GAMM-Mitt., 28 (2005), pp. 154–182, https://doi.org/10.1002/gamm.201490017.

[19] B. Cockburn, J. Gopalakrishnan, and R. Lazarov, *Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems*, SIAM J. Numer. Anal., 47 (2009), pp. 1319–1365, https://doi.org/10.1137/070706616.

[20] L. C. Cowsar, J. Mandel, and M. F. Wheeler, *Balancing domain decomposition for mixed finite elements*, Math. Comp., 64 (1995), pp. 989–1015, http://www.jstor.org/stable/2153480.

[21] C. R. Dohrmann, *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput., 25 (2003), pp. 246–258, https://doi.org/10.1137/S1064827502412887.

[22] H. Egger and J. Schöberl, *A hybrid mixed discontinuous Galerkin finite-element method for convection-diffusion problems*, IMA J. Numer. Anal., 30 (2010), pp. 1206–1234, https://doi.org/10.1093/imanum/drn083.

[23] B. X. Fraeijis de Veubeke, *Displacement and equilibrium models in the finite element method*, in Stress Analysis, O. C. Zienkiewicz and G. S. Holister, eds., John Wiley & Sons, New York, 1965, pp. 145–197.

[24] R. Glowinski and M. F. Wheeler, *Domain decomposition and mixed finite element methods for elliptic problems*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Paris, 1987), SIAM, Philadelphia, 1988, pp. 144–172.

[25] J. Gopalakrishnan, *A Schwarz preconditioner for a hybridized mixed method*, Comput. Methods Appl. Math., 3 (2003), pp. 116–134.

[26] R. J. Guyan, *Reduction of stiffness and mass matrices*, AIAA J., 3 (1965), p. 380, https://doi.org/10.2514/3.2874.

[27] V. E. Henson and U. M. Yang, *BoomerAMG: A parallel algebraic multigrid solver and preconditioner*, Appl. Numer. Math., 41 (2002), pp. 155–177, http://dx.doi.org/10.1016/S0168-9274(01)00115-5.

[28] R. Hiptmair and J. Xu, *Nodal auxiliary space preconditioning in H(curl) and H(div) spaces*, SIAM J. Numer. Anal., 45 (2007), pp. 2483–2509, http://www.jstor.org/stable/40233189.

[29] *hypre: High Performance Preconditioners.*, http://www.llnl.gov/casc/hypre (2018).

[30] B. M. Irons, *Structural eigenvalue problems—elimination of unwanted variables*, AIAA J., 3 (1965), pp. 961–962, https://doi.org/10.2514/3.3027.

[31] E. Kikinzon, Y. Kuznetsov, K. Lipnikov, and M. J. Shashkov, *Approximate Static Condensation Algorithm for Solving Multi-Material Diffusion Problems on Mesh Non-Aligned with Material Interfaces*, Technical Report LA-UR-17-22479, Los Alamos National Laboratory, Los Alamos, NM, 2017, https://doi.org/10.13140/RG.2.2.22290.66243.

[32] T. V. Kolev and P. S. Vassilevski, *Parallel auxiliary space AMG solver for H(curl) problems*, J. Comput. Math., 27 (2009), pp. 604–623.

[33] T. V. Kolev and P. S. Vassilevski, *Parallel auxiliary space AMG solver for H(div) problems*, SIAM J. Sci. Comput., 34 (2012), pp. A3079–A3098, https://doi.org/10.1137/110859361.

[34] C. S. Lee and P. S. Vassilevski, *Parallel solver for H(div) problems using hybridization and AMG*, in Domain Decomposition Methods in Science and Engineering 23, C.-O. Lee, X.-C. Cai, D. E. Keyes, H. H. Kim, A. Klawonn, E.-J. Park, and O. B. Widlund, eds., Springer, New York, 2017, pp. 69–80.

[35] L. Marini, *An inexpensive method for the evaluation of the solution of the lowest order Raviart-Thomas mixed method*, SIAM J. Numer. Anal., 22 (1985), pp. 493–496, https://doi.org/10.1137/0722029.

[36] J. Maryška, M. Rozložník, and M. Tůma, *Schur complement systems in the mixed-hybrid finite element approximation of the potential fluid flow problem*, SIAM J. Sci. Comput., 22 (2000), pp. 704–723, https://doi.org/10.1137/S1064827598339608.

[37] *MFEM: Modular Finite Element Methods.*, http://mfem.org (2018).

[38] D.-S. Oh, O. B. Widlund, S. Zampini, and C. R. Dohrmann, *BDDC algorithms with deluxe scaling and adaptive selection of primal constraints for Raviart-Thomas vector fields*, Math. Comp., 87 (2018), pp. 659–692, https://doi.org/10.1090/mcom/3254.

[39] P. A. Raviart and J. M. Thomas, *A mixed finite element method for second order elliptic problems*, in Mathematical Aspects of the Finite Element Method, Lecture Notes in Math. 606, Springer, New York, 1977, pp. 292–315.

[40] J. Roberts and J.-M. Thomas, *Mixed and hybrid methods*, in Finite Element Methods (Part 1), Handbook of Numerical Analysis, Vol. 2, Elsevier, Amsterdam, 1991, pp. 523–639, https://doi.org/10.1016/S1570-8659(05)80041-9.

[41] P. S. Vassilevski and U. Villa, *A block-diagonal algebraic multigrid preconditioner for the Brinkman problem*, SIAM J. Sci. Comput., 35 (2013), pp. S3–S17, https://doi.org/10.1137/120882846.

[42] M. Vohralík, *Unified primal formulation-based a priori and a posteriori error analysis of mixed finite element methods*, Math. Comp., 79 (2010), pp. 2001–2032, http://www.jstor.org/stable/20779133.

[43] M. F. Wheeler and I. Yotov, *Multigrid on the interface for mortar mixed finite element methods for elliptic problems*, Comput. Methods Appl. Mech. Engrg., 184 (2000), pp. 287–302, https://doi.org/10.1016/S0045-7825(99)00232-7.

[44] E. L. Wilson, *Structural analysis of axisymmetric solids*, AIAA J., 3 (1965), pp. 2269–2274, https://doi.org/10.2514/3.3356.

[45] S. Zampini, *PCBDDC: A class of robust dual-primal methods in PETSc*, SIAM J. Sci. Comput., 38 (2016), pp. S282–S306, https://doi.org/10.1137/15M1025785.

[46] S. Zampini and D. E. Keyes, *On the robustness and prospects of adaptive BDDC methods for finite element discretizations of elliptic PDEs with high-contrast coefficients*, in Proceedings of the Platform for Advanced Scientific Computing Conference, ACM, New York, 2016, pp. 6:1–6:13, http://doi.acm.org/10.1145/2929908.2929919.