Auxiliary Space Preconditioners for Mixed Finite Element Methods

Ray S. Tuminaro[1], Jinchao Xu[2], and Yunrong Zhu[3]

[1] MS 9214, Sandia National Laboratories, Livermore, CA 94551 USA
rstumin@sandia.gov
[2] Department of Mathematics, Pennsylvania State University, University Park, PA 16802 USA xu@math.psu.edu
[3] Department of Mathematics, University of California, San Diego, 9500 Gilman Driver, La Jolla, CA 92093 USA zhu@math.ucsd.edu

Summary. This paper is devoted to study of an auxiliary spaces preconditioner for $\mathbf{H}(\mathrm{div})$ systems and its application in the mixed formulation of second order elliptic equations. Extensive numerical results show the efficiency and robustness of the algorithms, even in the presence of large coefficient variations. For the mixed formulation of elliptic equations, we use the augmented Lagrange technique to convert the solution of the saddle point problem into the solution of a nearly singular $\mathbf{H}(\mathrm{div})$ system. Numerical experiments also justify the robustness and efficiency of this scheme.

1 Introduction

In this note, we discuss some implementation details of robust and efficient AMG preconditioners for the $\mathbf{H}(\mathrm{div})$ system:

$$(\lambda \mathrm{div}\,\mathbf{u}, \mathrm{div}\,\mathbf{v}) + (\mu \mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{H}(\mathrm{div}), \tag{1}$$

where $\mathbf{f} \in \mathbf{L}^2(\Omega)$ is a vector field and the coefficients $\lambda(x)$ and $\mu(x)$ are assumed to be uniformly positive but may have large variations in the whole domain Ω. Given a triangulation, the finite element problem reads:

$$\text{Find } \mathbf{u}_h \in \mathbf{V}_h(\mathrm{div}): \quad (\lambda \mathrm{div}\,\mathbf{u}_h, \mathrm{div}\,\mathbf{v}_h) + (\mu \mathbf{u}_h, \mathbf{v}_h) = (\mathbf{f}, \mathbf{v}_h), \quad \forall \mathbf{v}_h \in \mathbf{V}_h(\mathrm{div}), \tag{2}$$

where $\mathbf{V}_h(\mathrm{div}) \subset \mathbf{H}(\mathrm{div})$ is a conforming finite element space, e.g. Raviart-Thomas element, or BDM element (c.f. [6]). The finite element discretization (2) gives rise to the following linear system:

$$\mathbf{A}x = b, \tag{3}$$

where $\mathbf{A} = (a_{ij})$ is defined by $a_{ij} = \int_\Omega \lambda \mathrm{div}\varphi_j \mathrm{div}\varphi_i + \mu \varphi_j \cdot \varphi_i dx$ for any basis functions $\varphi_i, \varphi_j \in \mathbf{V}_h(\mathrm{div})$.

The importance of $\mathbf{H}(\mathrm{div})$-related problems has promoted vigorous research into efficient multilevel schemes for solving the linear system (3) (see [1, 10, 12, 18, 19] for example). The $\mathbf{H}(\mathrm{div})$ systems (1) arise naturally from numerous problems of practical importance, such as stabilized mixed formulations of the Stokes problem, least squares methods for H^1 systems, and mixed methods for second order elliptic equations, see [1, 19].

Recently, Hiptmair and Xu [12] proposed an innovative approach to solve $\mathbf{H}(\mathrm{curl})$ and $\mathbf{H}(\mathrm{div})$ systems, known as the HX-preconditioner. It relies on a discrete regular decomposition (see Section 2) and the framework of auxiliary space method ([20]). This decomposition links the vector fields in $\mathbf{H}(\mathrm{curl})$ and $\mathbf{H}(\mathrm{div})$ directly with functions in H^1. By using certain grid transfer operators, the evaluation of the preconditioners for $\mathbf{H}(\mathrm{curl})$ and $\mathbf{H}(\mathrm{div})$ systems is essentially reduced to several second-order elliptic operators. Hence, standard (algebraic) multigrid techniques for the H^1 equations can be applied.

In our implementation of the HX-preconditioner for $\mathbf{H}(\mathrm{div})$, we use a "grey-box" multilevel algorithm. More precisely, unlike the standard AMG technique, we rely on certain grid information for construction of the grid-transfer operators, namely the canonical interpolation operators Π_h^{curl}, Π_h^{div} and the discrete curl operator $\mathbf{C}$. The construction of these operators relies solely on coordinates and grid information on the finest level. In particular, we do not need a complete multilevel grid hierarchy which is crucial in standard geometric multilevel algorithms. A similar idea was used in [13] for the AMG implementation of the HX-preconditioner for $\mathbf{H}(\mathrm{curl})$ systems.

The finite element discretization of the mixed problem results in a saddle point problem. There is a significant amount of literature on designing robust preconditioners for the mixed problem, see [2, 7, 11]. Here we use the augmented Lagrangian method to reduce the saddle point problem into a nearly singular $\mathbf{H}(\mathrm{div})$ system, which can be efficiently solved by using the HX-preconditioner.

The remainder of this paper is organized as follows. In Section 2, we discuss some implementation details about the algorithm. In Section 3, we consider solving a mixed formulation of second order elliptic equations. We apply the augmented Lagrange method to reduce the mixed formulation into a nearly singular $\mathbf{H}(\mathrm{div})$ system. Then in Section 4, we present some numerical experiments to justify the robustness and efficiency of the algorithms.

2 HX-Preconditioner for $\mathbf{H}(\mathrm{div})$ Systems

In this section, we summarize the main ingredients used in [12] to derive and analyze the auxiliary space preconditioner. Here, we consider the lowest order Raviart-Thomas space $\mathbf{V}_h(\mathrm{div}) \subset \mathbf{H}(\mathrm{div})$, the lowest order Nédélec space $\mathbf{V}_h(\mathrm{curl}) \subset \mathbf{H}(\mathrm{curl})$ and the standard piecewise linear continuous nodal space $V_h(\mathrm{grad})$. We use Π_h^{grad}, Π_h^{curl} and Π_h^{div} to denote the canonical interpolation operators onto the finite element spaces $V_h(\mathrm{grad})$, $\mathbf{V}_h(\mathrm{curl})$ and $\mathbf{V}_h(\mathrm{div})$, respectively. The HX-preconditioners for $\mathbf{H}(\mathrm{curl})$ and $\mathbf{H}(\mathrm{div})$ systems exploit the following discrete regular decomposition.

Theorem 1. [12, Lemma 5.1] *Let D be the differential operator* curl *or* div, *and D^- be* grad *or* curl *respectively. Then for any $\mathbf{v}_h \in V_h(D)$, we have*

$$\mathbf{v}_h = \widetilde{\mathbf{v}}_h + \Pi_h^D \Phi_h + D^- p_h,$$

where $\widetilde{\mathbf{v}}_h \in V_h(D)$, $\Phi_h \in \mathbf{V}_h(\mathrm{grad})$ and $p_h \in V_h(D^-)$, such that

(1) $\|h^{-1}\widetilde{\mathbf{v}}_h\|_{0,\Omega}^2 + \|\Phi_h\|_{1,\Omega}^2 \lesssim \|D\mathbf{v}_h\|_{0,\Omega}^2$;
(2) $\|p_h\|_{H(D^-)} \lesssim \|\mathbf{v}_h\|_{H(D)}$.

In the above decomposition, when $D = \mathrm{div}$, the discrete potential $p_h \in \mathbf{V}_h(\mathrm{curl})$ is not entirely desirable. In order to avoid solving an $\mathbf{H}(\mathrm{curl})$-elliptic equation for p_h, we apply the decomposition Theorem 1 recursively and replace p_h by a $\Psi_h \in \mathbf{V}_h(\mathrm{grad})$ and some "high frequency" edge element function. More precisely, we obtain a decomposition

$$\begin{aligned}\mathbf{v}_h &= \sum_{\mathbf{b}\in\mathcal{B}(\mathrm{div})} \mathbf{v}_\mathbf{b} + \Pi_h^{\mathrm{div}} \Phi_h + \mathrm{curl}\, p_h \\ &= \sum_{\mathbf{b}\in\mathcal{B}(\mathrm{div})} \mathbf{v}_\mathbf{b} + \Pi_h^{\mathrm{div}} \Phi_h + \sum_{\mathbf{q}\in\mathcal{B}(\mathrm{curl})} \mathrm{curl}\, p_\mathbf{q} + \mathrm{curl}\, \Pi_h^{\mathrm{curl}} \Psi_h,\end{aligned}$$

where Φ_h, $\Psi_h \in \mathbf{V}_h(\mathrm{grad})$ and $\mathcal{B}(\mathrm{div})$, $\mathcal{B}(\mathrm{curl})$ are the sets of the basis functions in $\mathbf{V}_h(\mathrm{div})$ and $\mathbf{V}_h(\mathrm{curl})$ respectively. In this decomposition, we have used the fact that curl grad $= 0$. By Theorem 1, this decomposition is stable:

$$\sum_{\mathbf{b}\in\mathcal{B}(\mathrm{div})} \|\mathbf{v}_\mathbf{b}\|_A^2 + \|\Psi_h\|_{1,\Omega}^2 + \sum_{\mathbf{q}\in\mathcal{B}(\mathrm{curl})} \|\mathrm{curl}\, p_\mathbf{q}\|_{0,\Omega}^2 + \|\Phi_h\|_{1,\Omega}^2 \lesssim \|\mathbf{v}_h\|_A^2 . \qquad (4)$$

Based on this decomposition, the matrix representation of the (additive) auxiliary space preconditioner for the equation (2) is given by

$$\mathbf{B}_h^{\mathrm{div}} := \mathbf{S}_h^{\mathrm{div}} + \mathbf{C}\mathbf{S}_h^{\mathrm{curl}}\mathbf{C}^T + \mathbf{P}_h^{\mathrm{div}}\left(\mathbf{A}_h^{\mathrm{grad}}\right)^{-1}\mathbf{P}_h^{\mathrm{div}\,T} + \mathbf{C}\mathbf{P}_h^{\mathrm{curl}}\left(\mathbf{A}_h^{\mathrm{grad}}\right)^{-1}\mathbf{P}_h^{\mathrm{curl}\,T}\mathbf{C}^T, \qquad (5)$$

where

- $\mathbf{S}_h^{\mathrm{div}}$ and $\mathbf{S}_h^{\mathrm{curl}}$ are certain smoothers in the Raviart-Thomas and the Nédélec finite element spaces, for example Jacobi or symmetric Gauss-Seidel iterations, which we denote by StandardRelaxation() in the algorithms;
- $\mathbf{C}$ is the discrete curl operator;
- $\mathbf{P}_h^{\mathrm{div}}$ and $\mathbf{P}_h^{\mathrm{curl}}$ are the matrix representation of the canonical interpolation operators Π_h^{div} and Π_h^{curl} respectively;
- $\mathbf{A}_h^{\mathrm{grad}}$ is the related (vectorial) elliptic operator on the finite element spaces $\mathbf{V}_h(\mathrm{grad})$.

Remark 1. We remark that the first two terms in (5) together form an additive *Hiptmair smoother* (see [10]). In a multiplicative version of the preconditioner, we will denote the functions Pre(/Post)FineRelaxation() as multiplicative Hiptmair

Algorithm 1: $u = \mathsf{FineRelaxation}(\mathbf{A}, \mathbf{C}, u, b)$

1 $u \leftarrow \mathsf{StandardRelaxation}(\mathbf{A}, u, b)$;
2 $e \leftarrow \mathsf{StandardRelaxation}(\mathbf{C}^T A\mathbf{C}, 0, \mathbf{C}^T(b - \mathbf{A}u))$;
3 $u \leftarrow u + \mathbf{C}e$;
4 $u \leftarrow \mathsf{StandardRelaxation}(\mathbf{A}, u, b)$;

smoothers, see Algorithm 1. The function PreFineRelaxation() is identical to Algorithm 1 except step 1 is omitted, and the function PostFineRelaxation() is identical to Algorithm 1 except step four is omitted to keep the preconditioner symmetric.

It is important to realize that this special smoother is only needed on the finest level in our implementation, instead of using this smoother on each level as in [10].

When a hierarchy of structured grids is available, standard geometric multigrid can be applied to $\mathbf{A}_h^{\text{grad}}$ in the preconditioner (5). However, in general, the hierarchical information of the grids is not available, for example when the mesh is unstructured. In this case, one may consider using algebraic multigrid (AMG) algorithms. Moreover, instead of assembling the stiffness matrix $\mathbf{A}_h^{\text{grad}}$ explicitly by using the mesh data, we replace it with the following two matrices:

$$\mathbf{A}_1 := \mathbf{P}_h^{\text{div}\,T}\mathbf{A}\mathbf{P}_h^{\text{div}},$$
$$\mathbf{A}_2 := \mathbf{P}_h^{\text{curl}\,T}\mathbf{C}^T\mathbf{A}\mathbf{C}\mathbf{P}_h^{\text{curl}} = \mathbf{P}_h^{\text{curl}\,T}\mathbf{C}^T\mathbf{M}(\mu)\mathbf{C}\mathbf{P}_h^{\text{curl}},$$

where $\mathbf{A}$ is the stiffness matrix defined in (3), and $\mathbf{M}(\mu)$ is the mass matrix defined by $M = (m_{ij})$ with $m_{ij} = \int_\Omega \mu\varphi_i \cdot \varphi_j$. In the formulation of $\mathbf{A}_2$, we used the fact that $\text{div}\,\text{curl} = 0$.

We note that $\mathbf{A}_1$ and $\mathbf{A}_2$ are vector Laplacian-like operators defined on the nodal space. Therefore, $\mathbf{A}_1$ and $\mathbf{A}_2$ are amenable to standard AMG algorithms. Here, we make use of the interpolation $\mathbf{P}_h^{\text{div}}$ and $\mathbf{P}_h^{\text{curl}}$, as well as the discrete curl $\mathbf{C}$. All of these three matrices can be constructed using grid information on the fine level. In fact, to compute the matrix $\mathbf{C}$, one needs to expand $\text{curl}\varphi_E$ in terms of the basis of $\mathbf{V}_h(\text{div})$ for any basis function $\varphi_E \in \mathbf{V}_h(\text{curl})$. If $\mathbf{V}_h(\text{div})$ and $\mathbf{V}_h(\text{curl})$ are the lowest order Raviart-Thomas and Nédélec spaces respectively, $\mathbf{C}$ is simply a signed "edge-to-face" incidence matrix. The sign of each entry is determined by the signs of basis functions, i.e. the preset edge and face orientations in the grid. The matrix $\mathbf{P}_h^{\text{div}} = (\mathbf{P}_h^x, \mathbf{P}_h^y, \mathbf{P}_h^z)$ can be computed component-wise, where each of the blocks has the same sparsity pattern as the "face-to-node" incidence matrix. The entries are computed by the surface integral of the nodal basis functions on the normal direction on the face. The computation of $\mathbf{P}_h^{\text{curl}}$ is similar, which can be found in [3, 13].

The operator (5) and the discussion above suggest an additive version of the preconditioner: Algorithm 2.

In Algorithm 2, we may update the solution u after computing each u_1-u_4. By updating the solution and the residual at each step, we define a multiplicative version of the preconditioner. In the multiplicative preconditioner, we replace

Algorithm 2: $u = \mathsf{HX\_Additive\_Preconditioner}(\mathbf{A}, b)$

1 %Setup Phase
2 Form $\mathbf{A}_1 \leftarrow {\mathbf{P}_h^{\mathrm{div}}}^T \mathbf{A} \mathbf{P}_h^{\mathrm{div}}$ efficiently;
3 $\mathsf{Standard\_AMG\_Setup}(\mathbf{A}_1)$;
4 Form $\mathbf{A}_2 \leftarrow {\mathbf{P}_h^{\mathrm{curl}}}^T \mathbf{C}^T \mathbf{A} \mathbf{C} \mathbf{P}_h^{\mathrm{curl}}$ efficiently;
5 $\mathsf{Standard\_AMG\_Setup}(\mathbf{A}_2)$;
6 —— ;
7 %Solve Phase;
8 $u_1 \leftarrow \mathsf{StandardRelaxation}(\mathbf{A}, 0, b)$;
9 $x \leftarrow \mathsf{StandardRelaxation}(\mathbf{C}^T \mathbf{A} \mathbf{C}, 0, \mathbf{C}^T b)$;
10 $u_2 \leftarrow \mathbf{C}x$;
11 %Perform V-cycles on $\mathbf{A}_1$ and $\mathbf{A}_2$ $a \leftarrow \mathsf{Standard\_AMG\_Vcycle}\big(\mathbf{A}_1, 0, {\mathbf{P}_h^{\mathrm{div}}}^T b\big)$;
12 $u_3 \leftarrow \mathbf{P}_h^{\mathrm{div}} a$;
13 $p \leftarrow \mathsf{Standard\_AMG\_Vcycle}\big(\mathbf{A}_2, 0, {\mathbf{P}_h^{\mathrm{curl}}}^T \mathbf{C}^T b\big)$;
14 $u_4 \leftarrow \mathbf{C}\mathbf{P}_h^{\mathrm{curl}} p$;
15 —— ;
16 $u \leftarrow u_1 + u_2 + u_3 + u_4$;

the additive Hiptmair smoother (line 8-10 in Algorithm 2) by a multiplicative one (PreFineRelaxation). The rest of the algorithm is similar to Algorithm 2, except that after each step, we update the solution u, compute the residual r and replace the b in Line 11 and 13 by the residual r. Finally, in order to guarantee the symmetry of the overall preconditioner, we need to preform a post-smoothing step (PostFineRelaxation) in the end of the algorithm.

3 Application to Mixed Method

As an application, we present the augmented Lagrangian method for solving systems arising from a mixed finite element discretization of the elliptic boundary value problem (see e.g., [6]):

$$\Delta p = f \ \text{ in } \Omega, \quad p|_{\partial\Omega} = 0. \tag{6}$$

The aim is to show that implementing an efficient iterative method for the resulting indefinite linear system reduces to designing an efficient method for the solution of an auxiliary *nearly singular* $\mathbf{H}(\mathrm{div})$ problem. The augmented Lagrangian method has been applied to the mixed formulation of equation (6) in [11].

Given a conforming triangulation $\mathcal{T}_h$, let $\mathbf{V}_h(\mathrm{div}) \subset \mathbf{H}(\mathrm{div})$ and $V_h(0) \subset L^2(\Omega)$ be the corresponding finite element spaces. Then the mixed finite element method for the model problem (6) reads: find $(\mathbf{u}_h, p_h) \in \mathbf{V}_h(\mathrm{div}) \times V_h(0)$ such that

$$\begin{cases} (\mathbf{u}_h, \mathbf{v}_h) + (p_h, \mathrm{div}\mathbf{v}_h) = 0 & \forall \mathbf{v}_h \in V_h(\mathrm{div}) \\ (\mathrm{div}\mathbf{u}_h, q_h) = (f, q_h) & \forall q_h \in V_h(0). \end{cases} \tag{7}$$

A sufficient condition for the well-posedness of the mixed method (7) is the discrete inf-sup condition. Several finite element spaces satisfying the inf-sup condition have been introduced, such as those of Raviart-Thomas [15] and Brezzi-Douglas-Marini [5]. Here we restrict ourselves to the Raviart-Thomas spaces.

The mixed finite element method (7) results in the following linear system:

$$\begin{bmatrix} A & B^* \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix}. \tag{8}$$

It is not difficult to see that A is the mass matrix of the Raviart-Thomas element and B is a matrix representation of div^*.

The augmented Lagrangian algorithm solves the following equivalent problem to (8) by the Uzawa method:

$$\begin{bmatrix} A+\varepsilon^{-1}B^*B & B^* \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \varepsilon^{-1}B^*f \\ f \end{bmatrix}. \tag{9}$$

Given $(\mathbf{u}^{(k)}, p^{(k)})$, the new iterate $(\mathbf{u}^{(k+1)}, p^{(k+1)})$ is obtained by solving the following system:

$$\begin{cases} (A+\varepsilon^{-1}B^*B)\mathbf{u}^{(k+1)} = \varepsilon^{-1}B^*f - B^*p^{(k)}, \\ p^{(k+1)} = p^{(k)} - \varepsilon^{-1}(f - B\mathbf{u}^{(k+1)}). \end{cases} \tag{10}$$

Convergence of this algorithm has been discussed in many works, see for example [7, 8, 14, 17].

Theorem 2. [14, Lemma 2.1] *Let $(\mathbf{u}^{(0)}, p^{(0)})$ be a given initial guess and for $k \geq 1$, let $(\mathbf{u}^{(k)}, p^{(k)})$ be the iterates obtained via the augmented Lagrangian algorithm* (10)*. Then the following estimates hold:*

$$\|p - p^{(k)}\|_{0,\Omega} \leq \left(\frac{\varepsilon}{\varepsilon+\lambda_0}\right)^k \|p - p^{(0)}\|_{0,\Omega},$$

$$\|\mathbf{u} - \mathbf{u}^{(k)}\|_A \leq \sqrt{\varepsilon}\|p - p^{(k)}\|_{0,\Omega} \leq \sqrt{\varepsilon}\left(\frac{\varepsilon}{\varepsilon+\lambda_0}\right)^k \|p - p^{(0)}\|_{0,\Omega},$$

where λ_0 is the minimum eigenvalue of $S = BA^{-1}B^*$.

According to this theorem, the iteration procedure (10) converges rapidly to the solution of (7) for small ε. However, at each iteration one needs to solve a nearly singular $\mathbf{H}(\mathrm{div})$ system

$$(\varepsilon A + B^*B)\mathbf{u}^{(k+1)} = B^*f - \varepsilon B^*p^{(k)}. \tag{11}$$

Thus, an efficient and robust $\mathbf{H}(\mathrm{div})$ solver will result in an optimal iterative method for the saddle point problem (7). We refer to Section 4.3 for the numerical justification.

4 Numerical Results

The proposed solvers are implemented as preconditioners for the conjugate gradient method (CG) in MATLAB. ML's smoothed aggregation solver (c.f. [16]) is used for $\mathbf{A}_1$ and $\mathbf{A}_2$ through the mlmex MATLAB interface [9]. Part of the numerical experiments was done and reported in [4]. Unless otherwise stated, we use two steps of symmetric Gauss-Seidel sub-smoothing on both faces and edges. For all experiments, the convergence is attained when the ℓ^2-norm of the residual is reduced by 1×10^{-10}.

4.1 Constant Coefficients

As the first experiment, we consider the constant coefficient case. We triangulate the domain $\Omega = [0,2]^3$ with an *unstructured* grid. We assume that $\mu > 0$ is a constant in Ω. The following table shows CG-accelerated auxiliary AMG solvers for the $\mathbf{H}(\mathrm{div})$ system:

$$(\mathrm{div}\,\mathbf{u}, \mathrm{div}\,\mathbf{v}) + \mu(\mathbf{u},\mathbf{v}) = (\mathbf{f},\mathbf{v}), \quad \forall \mathbf{v} \in \mathbf{H}(\mathrm{div})$$

with respect to different constant values of μ.

Grid		μ							
		10^{-9}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10^{1}	10^{2}
9^3	Additive	15	16	16	17	18	19	21	25
	Multiplicative	5	5	5	6	6	6	6	7
18^3	Additive	15	18	18	19	19	21	23	26
	Multiplicative	5	6	7	7	7	8	8	9
27^3	Additive	15	18	18	19	19	21	24	26
	Multiplicative	5	7	7	7	8	8	9	9

Table 1. Number of CG iterations for AMG $\mathbf{H}(\mathrm{div})$ preconditioners on the unstructured 3D tetrahedral mesh. $\lambda = 1$ and μ is a different constant for each experiment.

From Table 1, we observe that for different mesh sizes, both additive and multiplicative preconditioners result in a uniform and small number of CG iterations. Therefore, the preconditioners in both algorithms are robust with respect to the mesh size, which agrees with the theoretical results in [12]. Also, the iteration numbers are fairly robust with respective to the variation of the coefficient μ. From the table, one may also observe that the multiplicative preconditioner behaves better than the additive ones.

4.2 Variable Coefficients

In this subsection, we consider cases with variable coefficients. We conduct the experiments on the 3D unit cube $[0,1]^3$, triangulated by a uniform tetrahedron mesh (c.f. Fig. 1, each small cube is partitioned into six tetrahedrons).

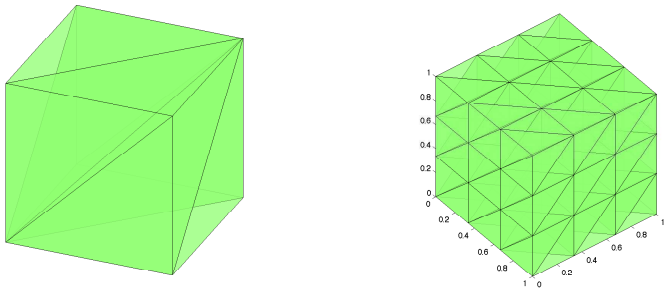

Fig. 1. Uniform Tetrahedra Meshes

First, we experiment with jumps in μ by considering two regions with constant values of μ. Specifically, define

$$\Omega_0 = \left\{ (x,y,z) : \frac{1}{3} \le x,y,z \le \frac{2}{3} \right\}, \ \ \Omega_1 = \Omega \setminus \Omega_0;$$

let $\mu \equiv 1$ in Ω_1 and choose $\mu = \mu_0$ to be a constant inside Ω_0. λ is fixed to be 1 throughout the whole domain Ω. Table 2 reports the number of iterations on different mesh sizes. Note that the number of iterations are again robust with respect to the

Grid		μ_0								
		10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10^{1}	10^{2}	10^{3}	10^{4}
9^3	Additive	19	19	19	19	18	19	21	23	23
	Multiplicative	6	5	5	5	5	5	6	6	5
18^3	Additive	19	19	20	18	17	18	20	23	24
	Multiplicative	6	6	6	6	5	5	6	6	6
27^3	Additive	18	19	19	17	17	17	19	22	24
	Multiplicative	6	6	6	5	5	5	6	6	6

Table 2. Number of iterations for CG-accelerated AMG on the 3D tetrahedral mesh problem with jump coefficients. μ_0 is defined inside $[1/3,2/3]^3$ and is a different constant for each experiment, elsewhere μ is 1, and $\lambda \equiv 1$.

variation of the coefficient μ.

We now consider a jump in λ. As before, we choose $\lambda = \lambda_0$ to be a constant, which varies for different experiments inside the domain Ω_0, and $\lambda = 1$ elsewhere. This time, we fix μ to be 1 in the whole domain Ω. Table 3 reports the number of iterations on different mesh sizes. In this case, the number of iterations varies a little bit. This may be due to some mild deficiencies in the underlying standard AMG solver.

Grid		λ_0								
		10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10^{1}	10^{2}	10^{3}	10^{4}
9^3	Additive	31	28	22	19	18	18	18	17	16
	Multiplicative	12	11	8	6	5	5	5	5	5
18^3	Additive	33	29	22	18	17	17	17	16	16
	Multiplicative	11	10	8	6	5	5	5	5	5
27^3	Additive	32	28	21	17	17	16	16	16	16
	Multiplicative	10	9	7	6	5	5	5	5	5

Table 3. Number of iterations for CG-accelerated AMG on the 3D tetrahedral mesh problem with jump coefficients. λ_0 is defined inside $[1/3,2/3]^3$ and is a different constant for each experiment, elsewhere λ is 1, and $\mu \equiv 1$.

4.3 Augmented Lagrangian Iterations

The augmented Lagrangian algorithm presented in Section 3 requires the solution of a nearly singular $\mathbf{H}(\mathrm{div})$ system (11) at each iteration. This implies that the $\mathbf{H}(\mathrm{div})$ solver should be robust with respect to the (penalty) parameter ε. Table 4 shows the CG-accelerated auxiliary AMG solver for the $\mathbf{H}(\mathrm{div})$ system:

$$(\mathrm{div}\,\mathbf{u}, \mathrm{div}\,\mathbf{v}) + \varepsilon(\mathbf{u},\mathbf{v}) = (\mathbf{f},\mathbf{v}), \;\; \forall \mathbf{v} \in \mathbf{H}(\mathrm{div})$$

with respect to different ε on structured meshes with different mesh sizes. That is, take $\mu = \varepsilon$ in the example in Subsection 4.1. Here we use ε for consistency with Section 3. As we can see from this table, both additive and multiplicative preconditioners are robust with respect to ε.

Grid		ε							
		10^{-9}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10^{1}	10^{2}
9^3	Additive	14	15	15	15	17	18	20	23
	Multiplicative	5	5	5	5	5	5	5	6
18^3	Additive	14	15	15	15	16	17	19	20
	Multiplicative	5	5	5	5	5	5	5	5
27^3	Additive	15	15	15	15	15	17	18	20
	Multiplicative	5	5	5	5	5	5	5	5

Table 4. Number of iterations for CG-accelerated AMG on the 3D tetrahedral mesh $\mathbf{H}(\mathrm{div})$ problem. ε is a different constant for each experiment.

Table 5 shows the number of outer iterations for the augmented Lagrangian method for the mixed formulation of the elliptic equation with respect to different ε, where we used the auxiliary AMG $\mathbf{H}(\mathrm{div})$ solver above to solve the nearly singular system. The tolerance for the augmented Lagrangian iteration is 10^{-8}. In particular, according to the theory, the augmented Lagrangian method converges faster

	ε							
Grid	10^{-9}	10^{-4}	10^{-3}	10^{-2}	10^{-1}	1	10^{1}	10^{2}
9^3	1	2	3	3	4	7	16	83
18^3	1	2	3	3	4	7	17	87
27^3	1	2	3	3	5	7	17	88

Table 5. Number of iterations for the augmented Lagrangian method for mixed method for elliptic equations on a 3D tetrahedral mesh using the $\mathbf{H}(\mathrm{div})$ solver. ε is a different constant for each experiment.

for smaller ε. We observe this phenomenon in Table 5. Most notably, if we choose $\varepsilon \leq 10^{-9}$ then only one iteration is needed.

5 Conclusions

In this paper, we discuss the implementation of an AMG based HX-preconditioners for the $\mathbf{H}(\mathrm{div})$ systems on unstructured grids. The numerical experiments show the robustness and efficiency of the algorithms even in the presence of large jump coefficients. As an application, we applied these preconditioners to solve the mixed finite element problem by augmented Lagrangian technique. The numerical experiments also show the efficiency of this approach.

Acknowledgement. The first author was supported in part by the DOE Office of Science ASCR Applied Math Research program and by the ASC program at Sandia Laboratories. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. The second author was supported in part by NSF DMS-0609727, NSFC-10528102 and Alexander von Humboldt Research Award for Senior US Scientists. The third author would like to thank Sandia National Lab for the support in summer 2007, and especially thank Bochev Pavel and Chris Siefert for inspiring discussions. He would also like to thank his postdoctoral advisor Professor Michael Holst for his encouragement and support through NSF Awards 0715146 and 0411723.

References

[1] Arnold, D.N., Falk, R.S., Winther, R.: Preconditioning in $H(\mathrm{div})$ and applications. *Math. Comp.*, 66:957–984, 1997.

[2] Bank, R.E., Welfert, B.D., Yserentant, H.: A class of iterative methods for solving mixed finite element equations. *Numer. Math.*, 56:645–666, 1990.

[3] Beck, R.: Graph-based algebraic multigrid for Lagrange-type finite elements on simplicial. Technical Report Preprint SC 99-22, Konrad-Zuse-Zentrum für Informationstechnik Berlin, 1999.

[4] Bochev, P.B., Seifert, C., Tuminaro, R., Xu, J., Zhu, Y.: Compatible gauge approaches for $H(\mathrm{div})$ equations. In *CSRI Summer Proceedings*, 2007.
[5] Brezzi, F., Douglas, J., Marini, L.D.: Two families of mixed finite elements for second order elliptic problems. *Numer. Math.*, 47(2):217–235, 1985.
[6] Brezzi, F., Fortin, M.: *Mixed and hybrid finite element methods*. Springer, 1991.
[7] Chen, Z., Ewing, R.E., Lazarov, R.D., Maliassov, S., Kuznetsov, Y.A.: Multilevel preconditioners for mixed methods for second order elliptic problems. *Numer. Linear Algebra Appl.*, 3(5):427–453, 1996.
[8] Fortin, M., Glowinski, R. *Augmented Lagrangian Methods: Application to the numerical solution of boundary value problems*. North-Holland, Amsterdam, 1983.
[9] Gee, M., Siefert, C., Hu, J., Tuminaro, R., Sala, M.: ML 5.0 smoothed aggregation user's guide. Technical Report SAND2006-2649, Sandia National Laboratories, 2006.
[10] Hiptmair, R.: Multigrid method for $H(\mathrm{div})$ in three dimensions. *Electron. Trans. Numer. Anal.*, 6:133–152, 1997.
[11] Hiptmair, R., Schiekofer, T., Wohlmuth, B. Multilevel preconditioned augmented Lagrangian techniques for 2nd order mixed problems. *Computing*, 57(1):25–48, 1996.
[12] Hiptmair, R., Xu, J.: Nodal Auxiliary Space Preconditioning in $H(\mathrm{curl})$ and $H(\mathrm{div})$ Spaces. *SIAM J. Numer. Anal.*, 45:2483, 2007.
[13] Kolev, T., Vassilevski, P.: Some experience with a H^1-based auxiliary space AMG for $H(\mathrm{curl})$ problems. Technical Report UCRL-TR-221841, Lawrence Livermore Nat. Lab., 2006.
[14] Lee, Y., Wu, J., Xu, J., Zikatanov, L. Robust subspace correction methods for nearly singular systems. *Math. Models Methods Appl. Sci.*, 17(11):1937, 2007.
[15] Raviart, P.A., Thomas, J.: A mixed finite element method fo 2nd order elliptic problems. In I. Galligani and E. Magenes, editors, *Mathematical aspects of the Finite Elements Method*, Lectures Notes in Math. 606, pages 292–315. Springer, Berlin, 1977.
[16] Vaněk, P., Brezina, M., Mandel, J. Convergence of algebraic multigrid based on smoothed aggregation. *Numer. Math.*, 88(3):559–579, 2001.
[17] Vassilevski, P., Lazarov, R.: Preconditioning Mixed Finite Element Saddle-point Elliptic Problems. *Numer. Linear Algebra Appl.*, 3(1):1–20, 1996.
[18] Vassilevski, P.S., Wang, J.: Multilevel iterative methods for mixed finite element discretizations of elliptic problems. *Numer. Math.*, 63(1):503–520, 1992.
[19] Wohlmuth, B.I., Toselli, A., Widlund, O.B.: An iterative substructuring method for Raviart–Thomas vector fields in three dimensions. *SIAM J. Numer. Anal.*, 37(5):1657–1676, 2000.
[20] Xu, J.: The auxiliary space method and optimal multigrid preconditioning techniques for unstructured meshes. *Computing*, 56:215–235, 1996.