Journal of Computational Mathematics
Vol.27, No.5, 2009, 604–623.

http://www.global-sci.org/jcm
doi:10.4208/jcm.2009.27.5.013

PARALLEL AUXILIARY SPACE AMG FOR $\boldsymbol{H}$(curl) PROBLEMS *

Tzanio V. Kolev Panayot S. Vassilevski
Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808,
L-560, Livermore, CA 94551, USA
Email: tzanio@llnl.gov; panayot@llnl.gov

Abstract

In this paper we review a number of auxiliary space based preconditioners for the second order definite and semi-definite Maxwell problems discretized with the lowest order Nédélec finite elements. We discuss the parallel implementation of the most promising of these methods, the ones derived from the recent Hiptmair-Xu (HX) auxiliary space decomposition [Hiptmair and Xu, SIAM J. Numer. Anal., 45 (2007), pp. 2483-2509]. An extensive set of numerical experiments demonstrate the scalability of our implementation on large-scale $\boldsymbol{H}$(curl) problems.

Mathematics subject classification: 65F10, 65N30, 65N55.
Key words: Parallel algebraic multigrid, H(curl) problems, Edge elements, Auxiliary space preconditioning.

1. Introduction

The numerical solution of electromagnetic problems based on Maxwell's equations is of critical importance in a number of engineering and physics applications. Recently, several implicit electromagnetic diffusion models have become popular in large-scale simulation codes [5, 21, 22]. These models require the solution of linear systems derived from discretizations of the weighted bilinear form

$$a(\boldsymbol{u},\boldsymbol{v}) = (\alpha\,\nabla\times\boldsymbol{u},\nabla\times\boldsymbol{v}) + (\beta\,\boldsymbol{u},\boldsymbol{v})\,, \tag{1.1}$$

where $\alpha > 0$ and $\beta \geq 0$ are piecewise-constant scalar coefficients describing the magnetic and electric properties of the medium. Problems involving (1.1) with $\beta > 0$ typically arise in time-domain electromagnetic simulations and are commonly referred to as second-order *definite* Maxwell equations. When β is zero in part of the domain, we call the problem *semi-definite*. Note that in this case the matrix corresponding to the bilinear form $a(\cdot,\cdot)$ is singular, and the right-hand side should satisfy appropriate compatibility conditions. One important semi-definite application is magnetostatics with a vector potential, where $\beta = 0$ in the whole domain.

Motivated by the needs of large multi-physics production codes, we are generally interested in efficient solvers for complicated systems of partial differential equations (PDEs), and in particular in the definite and semi-definite Maxwell problems. The target simulation codes run on parallel machines with tens of thousands of processors and typically lack discretization flexibility. Our focus is therefore on parallel algebraic methods since they can take advantage of needed discretization information about the problem only at the highest resolution.

If we restrict (1.1) to gradient fields, the familiar Poisson bilinear form is obtained, i.e.,

$$(\beta\,\nabla\phi,\nabla\psi) = a(\nabla\phi,\nabla\psi).$$

* Received December 19, 2007 / Revised version received June 23, 2008 / Accepted February 5, 2009 /

One very general algebraic approach, which has had a lot of success on Poisson and more general (scalar and vector) elliptic bilinear forms is the family of Algebraic Multigrid (AMG) methods. Classical AMG couples a simple relaxation scheme with a hierarchy of algebraically constructed coarse-grid problems. Parallel implementations of AMG have been under intensive research and development in the last decade, and several scalable software libraries are currently available [8,10]. Unlike Poisson problems, however, it is well known that the Maxwell system has a large near-nullspace of gradients which has to be addressed explicitly by the solver. Therefore, a straightforward application of AMG (developed for elliptic problems) fails. For a theoretical explanation of this phenomena, see, e.g., [27].

When a hierarchy of structured meshes is available and proper finite elements are used, geometric multigrid can be successfully applied to (1.1) based on additional discretization information [11]. One way to take advantage of this fact is to extend AMG by incorporating algebraic versions of the Hiptmair smoothers used in the geometric case. This was the approach taken in [4, 14, 20]. Alternatively, one can try to reduce the original unstructured problem to a problem for the same bilinear form on a structured mesh, which can then be handled by geometric multigrid. This idea, known as the *auxiliary mesh* approach, was explored in [12,15].

Recently, Hiptmair and Xu proposed in [13] an auxiliary space preconditioner for (1.1) and analyzed it in the case of constant coefficients. This approach is computationally more attractive since, in contrast to the auxiliary mesh idea, it uses a nodal conforming auxiliary space on the original mesh. This property significantly simplifies the computations and, more importantly, allows us to harness the power of AMG for Poisson problems. A related method, which arrives at the same nodal subspaces based on a compatible discretization principle, was introduced in [6].

In this paper, we examine the connections between the above algorithms, and their application to large definite and semi-definite Maxwell problems. We concentrate on the lowest order Nédélec finite element discretizations and consider topologically complicated domains, as well as variable coefficients (including some that lead to a singular matrix).

The remainder of the document is structured as follows: in Section 2, we introduce the notation and summarize some basic facts concerning Nédélec finite element discretizations. In Section 3, we show that the methods from [13], [15] and [6] can be derived from a single decomposition of the Nédélec finite element space by introducing one or several additional (auxiliary) spaces. The algebraic construction of auxiliary space preconditioners is reviewed in Section 4, and our parallel implementation of the Hiptmair-Xu (HX) auxiliary (nodal) space based preconditioner is discussed in Section 5. Next, we present in Section 6 a number of computational experiments that demonstrate the scalability of this approach. We finish with some conclusions in the last section.

2. Notation and Preliminaries

We consider definite and semi-definite Maxwell problems based on (1.1) and posed on a fixed three-dimensional polyhedral domain Ω. Let $\mathcal{T}_h$ be a tetrahedral or hexahedral meshing of the domain which is globally quasi-uniform of mesh size h. For generality, we allow internal holes and tunnels in the geometry, i.e., Ω may have a multiply-connected boundary and need not be simply connected. Meshes with such features arise naturally in many practical applications.

Let $\boldsymbol{L}_2(\Omega)$, $\boldsymbol{H}_0^1(\Omega)$, $\boldsymbol{H}_0(\Omega;\mathrm{curl})$ and $\boldsymbol{H}(\Omega;\mathrm{div})$ be the standard Hilbert spaces corresponding to our computational domain, with respective norms $\|\cdot\|_0$, $\|\cdot\|_1$, $\|\cdot\|_{\mathrm{curl}}$ and $\|\cdot\|_{\mathrm{div}}$. Here, and in the rest of the paper, we use boldface notation to denote vector functions and spaces of vector functions. In particular, $\boldsymbol{H}_0^1(\Omega) = H_0^1(\Omega)^3$. The following characterization of the kernel of the curl operator can be found in [2]:

$$\{\boldsymbol{u} \in \boldsymbol{H}_0(\Omega; \mathrm{curl}) \ : \ \nabla \times \boldsymbol{u} = 0\} = \nabla H_0^1(\Omega) \oplus \nabla K_N(\Omega)\,. \tag{2.1}$$

Here, $K_N(\Omega) \subset H^1(\Omega)$ is the space of harmonic functions which vanish on the exterior and are constant in the interior connected components of $\partial\Omega$.

We are interested in solving variational problems based on the bilinear form $a(\cdot,\cdot)$ posed in the (first kind) lowest order Nédélec space $\boldsymbol{V}_h \subset \boldsymbol{H}_0(\Omega; \mathrm{curl})$ associated with the triangulation $\mathcal{T}_h$. As it is well-known, the Nédélec edge elements are related to the space of continuous piecewise linear (or tri-linear) finite elements $S_h \subset H_0^1(\Omega)$. In particular,

$$\{\boldsymbol{u}_h \in \boldsymbol{V}_h \ : \ \nabla \times \boldsymbol{u}_h = 0\} = \nabla S_h \oplus \nabla K_{N,h}\,, \tag{2.2}$$

where $K_{N,h}$ is a discrete analog of $K_N(\Omega)$. These two spaces have the same finite dimension, which equals the number of internal holes in Ω.

Let $\boldsymbol{S}_h$ be the vector counterpart of S_h. Any $\boldsymbol{z} \in \boldsymbol{H}_0^1(\Omega)$ admits a stable component $\boldsymbol{z}_h \in \boldsymbol{S}_h$, that satisfies

$$h^{-1}\|\boldsymbol{z} - \boldsymbol{z}_h\|_0 + \|\boldsymbol{z}_h\|_1 \le C\,\|\boldsymbol{z}\|_1\,. \tag{2.3}$$

This component can be obtained by the application of an appropriate interpolation operator, see [7]. Here, and in what follows, C stands for a generic constant, independent of the functions and parameters involved in the given inequality.

Vector functions can also be approximated using the the Nédélec interpolation operator $\boldsymbol{\Pi}_h$. Let $\mathcal{E}_h$ be the set of edges in $\mathcal{T}_h$, while $\boldsymbol{t}_e$ is the unit tangent and $\boldsymbol{\Phi}_e \in \boldsymbol{V}_h$ is the basis function associated with the edge e. Then, $\boldsymbol{\Pi}_h$ is defined by

$$\boldsymbol{\Pi}_h \boldsymbol{v} = \sum_{e \in \mathcal{E}_h} \left(\int_e \boldsymbol{v} \cdot \boldsymbol{t}_e \, ds \right) \boldsymbol{\Phi}_e\,.$$

Note that we can apply $\boldsymbol{\Pi}_h$ only to sufficiently smooth functions. More specifically, the following result holds, see [3].

Theorem 2.1. *Let $\boldsymbol{z} \in \boldsymbol{H}_0^1(\Omega)$ be such that $\nabla \times \boldsymbol{z} \in \nabla \times \boldsymbol{V}_h$. Then*

(i) $\boldsymbol{\Pi}_h \boldsymbol{z}$ *is well-defined,*

(ii) $\nabla \times \boldsymbol{z} = \nabla \times \boldsymbol{\Pi}_h \boldsymbol{z}$,

(iii) $\|\boldsymbol{z} - \boldsymbol{\Pi}_h \boldsymbol{z}\|_0 \le C\,h\,\|\boldsymbol{z}\|_1$.

For example, the second item above follows by the commutativity between the curl operator, $\boldsymbol{\Pi}_h$, and its Raviart-Thomas counterpart $\boldsymbol{\Pi}_h^{RT}$. More details on the above topics, as well as further information regarding Nédélec finite elements, can be found in [18].

3. Edge Space Decompositions

The existence of certain stable decomposition of $\boldsymbol{V}_h$ is at the heart of the construction of all auxiliary space Maxwell solvers. In this section, we give an overview of several such decompositions and establish their stability. Even though the following theoretical estimates have been already essentially presented in [13, 15], we include them here for completeness, as well as to emphasize the connections between the different approaches.

Our starting point is the following result which was established by Pasciak and Zhao in [19], see also Lemma IV.2 in [26]. A careful examination of their argument reveals that the conclusion holds for the general type of domains that we are considering in this paper.

Theorem 3.1. *For any $\boldsymbol{u} \in \boldsymbol{H}_0(\Omega; \mathrm{curl})$ there exists a $\boldsymbol{z} \in \boldsymbol{H}_0^1(\Omega)$ such that $\nabla \times \boldsymbol{z} = \nabla \times \boldsymbol{u}$ which is bounded as follows*

$$\|\boldsymbol{z}\|_0 \leq C \, \|\boldsymbol{u}\|_0 \qquad \textit{and} \qquad \|\boldsymbol{z}\|_1 \leq C \, \|\nabla \times \boldsymbol{u}\|_0 \,.$$

Proof. Let B be a ball containing Ω and set $\Omega^c = B \setminus \Omega$. We can extend $\boldsymbol{u}$ to a function $\tilde{\boldsymbol{u}} \in \boldsymbol{H}_0(B; \mathrm{curl})$ by setting $\tilde{\boldsymbol{u}}|_{\Omega^c} = 0$. Consider the Helmholtz decomposition

$$\tilde{\boldsymbol{u}} = \tilde{\boldsymbol{z}} + \nabla \tilde{p} \,,$$

where $\tilde{p} \in H_0^1(B)$ can be defined as the unique solution of the variational problem

$$(\nabla \tilde{p}, \nabla \tilde{q})_{0,B} = (\tilde{\boldsymbol{u}}, \nabla \tilde{q})_{0,B} \qquad \forall q \in H_0^1(B) \,.$$

Note that $\tilde{\boldsymbol{z}}$ is divergence-free (by construction) and belongs to $\boldsymbol{H}_0(B; \mathrm{curl})$. Since B is convex, the space $\boldsymbol{H}_0(B; \mathrm{curl}) \cap \boldsymbol{H}(B; \mathrm{div})$ is continuously embedded in $\boldsymbol{H}^1(B)$, see [9]. Therefore, $\tilde{\boldsymbol{z}} \in \boldsymbol{H}^1(B)$ and the following estimates hold:

$$\|\tilde{\boldsymbol{z}}\|_{0,B} \leq \|\boldsymbol{u}\|_0 \,, \qquad \|\tilde{\boldsymbol{z}}\|_{1,B} \leq C \, \|\nabla \times \boldsymbol{u}\|_0 \,.$$

In Ω^c, we have $\nabla \tilde{p} = -\tilde{\boldsymbol{z}} \in \boldsymbol{H}^1(\Omega^c)$, so $\tilde{p} \in H^2(\Omega^c)$. If Ω^c has multiple connected components, we can subtract the mean value of $\tilde{p}$ on each of them without changing $\nabla \tilde{p}$. Thus, we can assume without a loss of generality that the following Poincaré inequality holds:

$$\|\tilde{p}\|_{1,\Omega^c} \leq C \, \|\nabla \tilde{p}\|_{0,\Omega^c} \,. \tag{3.1}$$

Since $\partial\Omega$ is Lipschitz, there exists an extension mapping from Ω^c to B that is bounded simultaneously in $\| \cdot \|_1$ and $\| \cdot \|_2$, see [24]. Let $\tilde{p}_B \in H^2(B)$ be the extension of $\tilde{p} \in H^2(\Omega^c)$. Then the function $\tilde{\boldsymbol{z}} + \nabla \tilde{p}_B$ is in $\boldsymbol{H}^1(B)$ and it vanishes in Ω^c. This means that $\boldsymbol{z} = (\tilde{\boldsymbol{z}} + \nabla \tilde{p}_B)|_\Omega$ is in $\boldsymbol{H}_0^1(\Omega)$, and clearly $\nabla \times \boldsymbol{z} = \nabla \times \boldsymbol{u}$. To complete the proof, use (3.1) and observe that

$$\|\boldsymbol{z}\|_{0,\Omega} \leq \|\tilde{\boldsymbol{z}}\|_{0,B} + C \, \|\tilde{p}\|_{1,\Omega^c} \leq \|\tilde{\boldsymbol{z}}\|_{0,B} + C \, \|\nabla \tilde{p}\|_{0,\Omega^c} \leq C \, \|\boldsymbol{u}\|_0 \,,$$

while

$$\|\boldsymbol{z}\|_{1,\Omega} \leq \|\tilde{\boldsymbol{z}}\|_{1,B} + C \, \|\tilde{p}\|_{2,\Omega^c} \leq \|\tilde{\boldsymbol{z}}\|_{1,B} + C \, \|\nabla \tilde{p}\|_{1,\Omega^c} \leq C \, \|\nabla \times \boldsymbol{u}\|_0 \,. \qquad \square$$

We apply Theorem 3.1 to a finite element function $\boldsymbol{u} = \boldsymbol{u}_h \in \boldsymbol{V}_h$. The corresponding function $\boldsymbol{z}$ satisfies the requirements of Theorem 2.1 and therefore

$$\nabla \times \boldsymbol{u}_h = \nabla \times \boldsymbol{\Pi}_h \boldsymbol{z} \,.$$

This means that $\boldsymbol{u}_h - \boldsymbol{\Pi}_h \boldsymbol{z} \in \boldsymbol{V}_h$ has a zero curl, and (2.2) tells us that it should equal ∇p_h for some $p_h \in S_h \oplus K_{N,h}$. Thus, we arrive at a decomposition of $\boldsymbol{V}_h$ with properties summarized in the next proposition.

Corollary 3.1. *Any $\boldsymbol{u}_h \in \boldsymbol{V}_h$ can be decomposed as follows*

$$\boldsymbol{u}_h = \boldsymbol{\Pi}_h \boldsymbol{z} + \nabla p_h \,,$$

where $\boldsymbol{z} \in \boldsymbol{H}_0^1(\Omega)$, $\nabla \times \boldsymbol{z} = \nabla \times \boldsymbol{u}_h$, and $p_h \in S_h \oplus K_{N,h}$. The following stability estimate holds:

$$\|\boldsymbol{\Pi}_h \boldsymbol{z}\|_0 + \|\nabla p_h\|_0 \leq C \, \|\boldsymbol{u}_h\|_0 \,.$$

Proof. First, observe that $\|\boldsymbol{z}\|_1 \leq C \|\nabla \times \boldsymbol{u}_h\|_0$ and the inverse inequality in $\boldsymbol{V}_h$ imply

$$h\|\boldsymbol{z}\|_1 \leq C\|\boldsymbol{u}_h\|_0 \,. \tag{3.2}$$

Therefore, Theorem 2.1 (iii) shows

$$\|\boldsymbol{\Pi}_h \boldsymbol{z}\|_0 \leq \|\boldsymbol{z}\|_0 + \|\boldsymbol{z} - \boldsymbol{\Pi}_h \boldsymbol{z}\|_0 \leq \|\boldsymbol{z}\|_0 + Ch\|\boldsymbol{z}\|_1 \leq C\|\boldsymbol{u}_h\|_0 \,.$$

The desired estimate now follows by the triangle inequality. □

The above result (Corollary 3.1) can be viewed as a semi-discrete version of the $\boldsymbol{H}_0(\Omega; \mathrm{curl})$ decomposition from [19]. There are several ways to further approximate the non-discrete component $\boldsymbol{z}$, thus ending up with various discrete stable decompositions. Two possible choices are:

1. Approximate $\boldsymbol{z}$ with a $\boldsymbol{z}_h \in \boldsymbol{S}_h$.
2. Approximate both $\boldsymbol{z}$ and p_h on a different mesh.

The first choice above leads to the HX decomposition from [13], which we consider in the next subsection. The second choice corresponds to the auxiliary mesh method and will be analyzed at the end of this section.

3.1. The HX Decomposition

Although the stable decomposition theory that we consider in the present paper cannot directly be applied to the definite Maxwell bilinear form (1.1) with variable coefficients, some insights can nevertheless be gained even if we assume constant coefficients. In what follows, we set $\alpha = 1$ and $\beta = \tau$, where τ is a nonnegative parameter.

Theorem 3.2. *Any $\boldsymbol{u}_h \in \boldsymbol{V}_h$ admits a decomposition*

$$\boldsymbol{u}_h = \boldsymbol{v}_h + \boldsymbol{\Pi}_h \boldsymbol{z}_h + \nabla p_h \,,$$

where $\boldsymbol{v}_h \in \boldsymbol{V}_h$, $\boldsymbol{z}_h \in \boldsymbol{S}_h$, $p_h \in S_h \oplus K_{N,h}$. The following stability estimate holds:

$$(h^{-2} + \tau)\|\boldsymbol{v}_h\|_0^2 + |||\boldsymbol{z}_h|||_\tau^2 + \tau \, \|\nabla p_h\|_0^2 \leq C \, \left(\|\nabla \times \boldsymbol{u}_h\|_0^2 + \tau \, \|\boldsymbol{u}_h\|_0^2\right) \,.$$

The above estimate is uniform with respect to the parameter $\tau \geq 0$. The norm $|||\cdot|||_\tau$ stands for one of the following expressions:

$$|||\boldsymbol{z}_h|||_\tau^2 = \begin{cases} \|\boldsymbol{z}_h\|_1^2 + \tau \, \|\boldsymbol{z}_h\|_0^2 & (a) \\ \|\nabla \times \boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0^2 + \tau \, \|\boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0^2 & (b) \end{cases} \tag{3.3}$$

Proof. Consider the decomposition of $\boldsymbol{u}_h$ given by Corollary 3.1, i.e., $\boldsymbol{u}_h = \boldsymbol{\Pi}_h \boldsymbol{z} + \nabla p_h$. Let $\boldsymbol{z}_h \in \boldsymbol{S}_h$ be a stable approximation of $\boldsymbol{z}$ in the sense of (2.3). Letting $\boldsymbol{v}_h = \boldsymbol{\Pi}_h(\boldsymbol{z} - \boldsymbol{z}_h)$, we have

$$\boldsymbol{u}_h = \boldsymbol{v}_h + \boldsymbol{\Pi}_h \boldsymbol{z}_h + \nabla p_h \,.$$

We first note that for any $\boldsymbol{z}_h \in \boldsymbol{S}_h$, we have $\nabla \times \boldsymbol{z}_h \in \nabla \times \boldsymbol{V}_h$. Since $\nabla \times \boldsymbol{z} = \nabla \times \boldsymbol{u}_h \in \nabla \times \boldsymbol{V}_h$, applying Theorem 2.1 to $\boldsymbol{z} - \boldsymbol{z}_h$ gives

$$\begin{aligned} \|\boldsymbol{v}_h\|_0 &\le \|(\boldsymbol{z} - \boldsymbol{z}_h) - \boldsymbol{v}_h\|_0 + \|(\boldsymbol{z} - \boldsymbol{z}_h)\|_0 \\ &= \|(\boldsymbol{z} - \boldsymbol{z}_h) - \boldsymbol{\Pi}_h(\boldsymbol{z} - \boldsymbol{z}_h)\|_0 + \|\boldsymbol{z} - \boldsymbol{z}_h\|_0 \\ &\le C\, h\, \|(\boldsymbol{z} - \boldsymbol{z}_h)\|_1 + C\, h\, \|\boldsymbol{z}\|_1 \le C\, h\, \|\boldsymbol{z}\|_1 \,. \end{aligned}$$

Therefore, by Theorem 3.1 and (3.2),

$$(h^{-2} + \tau)\|\boldsymbol{v}_h\|_0^2 \le C\,(1 + \tau h^2)\, \|\boldsymbol{z}\|_1^2 \le C\, \left(\|\nabla \times \boldsymbol{u}_h\|_0^2 + \tau\, \|\boldsymbol{u}_h\|_0^2\right) \,.$$

The estimate of $\boldsymbol{z}_h$ in case (a) follows from (2.3) and (3.2). We have

$$\|\boldsymbol{z}_h\|_1 \le C\, \|\boldsymbol{z}\|_1 \le C\, \|\nabla \times \boldsymbol{u}_h\|_0 \,,$$

and

$$\|\boldsymbol{z}_h\|_0 \le \|\boldsymbol{z}\|_0 + \|\boldsymbol{z} - \boldsymbol{z}_h\|_0 \le C\, \|\boldsymbol{u}_h\|_0 \,.$$

On the other hand, for any $\boldsymbol{z}_h \in \boldsymbol{S}_h$, recalling that $\nabla \times \boldsymbol{z}_h \in \nabla \times \boldsymbol{V}_h$, Theorem 2.1 implies

$$\|\nabla \times \boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0 = \|\nabla \times \boldsymbol{z}_h\|_0 \le C\, \|\boldsymbol{z}_h\|_1 \,,$$

while the inverse inequality in $\boldsymbol{S}_h$ yields

$$\|\boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0 \le \|\boldsymbol{z}_h - \boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0 + \|\boldsymbol{z}_h\|_0 \le C\, \|\boldsymbol{z}_h\|_0 \,.$$

The bound involving p_h was already established in Corollary 3.1. □

Remark 3.1. The result of Theorem 3.2 holds for other stable approximations of $\boldsymbol{z}$. For example, we can choose $\boldsymbol{z}_h = \boldsymbol{z}_H$ from a coarse subspace $\boldsymbol{S}_H$ of $\boldsymbol{S}_h$ as long as $H \simeq h$ (i.e. $\boldsymbol{S}_H$ and $\boldsymbol{S}_h$ have comparable mesh sizes).

In magnetostatic applications, it is useful to introduce the factor-space $\boldsymbol{V}_h/\nabla S_h$ which contains the equivalence classes of finite element functions that differ by a discrete gradient. Let $[\boldsymbol{u}_h] \in \boldsymbol{V}_h/\nabla S_h$ be the equivalence class containing $\boldsymbol{u}_h \in \boldsymbol{V}_h$. By definition

$$\|\nabla \times [\boldsymbol{u}_h]\|_0 = \|\nabla \times \boldsymbol{u}_h\|_0 \quad \text{and} \quad \|[\boldsymbol{u}_h]\|_0 = \inf_{p_h \in S_h} \|\boldsymbol{u}_h - \nabla p_h\|_0 \,.$$

With this notation, we have the following gradient-free analog of the HX decomposition.

Corollary 3.2. *For any $[\boldsymbol{u}_h] \in \boldsymbol{V}_h/\nabla S_h$ there are $\boldsymbol{v}_h \in \boldsymbol{V}_h$ and $\boldsymbol{z}_h \in \boldsymbol{S}_h$, such that*

$$[\boldsymbol{u}_h] = [\boldsymbol{v}_h] + [\boldsymbol{\Pi}_h \boldsymbol{z}_h] \,,$$

and the following stability estimate holds:

$$h^{-2}\|[\boldsymbol{v}_h]\|_0^2 + \|\nabla \times [\boldsymbol{\Pi}_h \boldsymbol{z}_h]\|_0^2 \le C\, \|\nabla \times [\boldsymbol{u}_h]\|_0^2 \,.$$

This result follows from Theorem 3.2 applied to $\boldsymbol{u}_h$ and the fact that $\|[\boldsymbol{v}_h]\|_0 \le \|\boldsymbol{v}_h\|_0$. It implies that the space $[\boldsymbol{\Pi}_h \boldsymbol{S}_h]$ can be used to approximate the discretely divergence-free elements of $\boldsymbol{V}_h$, i.e. if $\boldsymbol{u}_h$ is orthogonal to ∇S_h, then

$$\inf_{\boldsymbol{z}_h \in \boldsymbol{S}_h} \|\boldsymbol{u}_h - [\boldsymbol{\Pi}_h \boldsymbol{z}_h]\|_0 \le C\, h\, \|\nabla \times \boldsymbol{u}_h\|_0 \,.$$

3.2. A Component-wise (Scalar) HX Decomposition

Let $\boldsymbol{z}_h = (z_h^1, z_h^2, z_h^3)$ and introduce

$$\varPi_h^1 z_h^1 = \boldsymbol{\varPi}_h(z_h^1, 0, 0)\,, \quad \varPi_h^2 z_h^2 = \boldsymbol{\varPi}_h(0, z_h^2, 0)\,, \quad \varPi_h^3 z_h^3 = \boldsymbol{\varPi}_h(0, 0, z_h^3)\,.$$

By construction then, we have

$$\boldsymbol{\varPi}_h \boldsymbol{z}_h = \sum_{k=1}^{3} \varPi_h^k z_h^k\,.$$

Using this representation in the HX decomposition, we end up with a different version that utilizes only scalar subspaces. We refer to this decomposition as the "scalar" HX decomposition. Similarly to 3.3(a), it gives rise to a block-diagonal matrix associated with the subspace $\boldsymbol{S}_h$. A magnetostatics result analogous to that of Corollary 3.2 is valid as well.

Theorem 3.3. *Any $\boldsymbol{u}_h \in \boldsymbol{V}_h$ admits the decomposition*

$$\boldsymbol{u}_h = \boldsymbol{v}_h + \sum_{k=1}^{3} \varPi_h^k z_h^k + \nabla p_h\,,$$

where $\boldsymbol{v}_h \in \boldsymbol{V}_h$, $z_h^k \in S_h$, $p_h \in S_h \oplus K_{N,h}$, which satisfies the following stability estimate

$$(h^{-2} + \tau)\|\boldsymbol{v}_h\|_0^2 + \sum_{k=1}^{3} |||z_h^k|||_{k,\tau}^2 + \tau\,\|\nabla p_h\|_0^2 \le C\,\left(\|\nabla \times \boldsymbol{u}_h\|_0^2 + \tau\,\|\boldsymbol{u}_h\|_0^2\right)\,.$$

The constant C is independent of the parameter $\tau \ge 0$. The norm $|||.|||_{k,\tau}$ is defined by

$$|||z_h^k|||_{k,\tau}^2 = \|\nabla \times \varPi_h^k z_h^k\|_0^2 + \tau\,\|\varPi_h^k z_h^k\|_0^2.$$

Proof. In the proof of Theorem 3.2 we showed that for any $\boldsymbol{z}_h \in \boldsymbol{S}_h$

$$\|\nabla \times \boldsymbol{\varPi}_h \boldsymbol{z}_h\|_0^2 + \tau\,\|\boldsymbol{\varPi}_h \boldsymbol{z}_h\|_0^2 \le C\,\left(\|\boldsymbol{z}_h\|_1^2 + \tau\,\|\boldsymbol{z}_h\|_0^2\right)\,. \tag{3.4}$$

Therefore,

$$|||z_h^k|||_{k,\tau}^2 \le C\,\left(\|z_h^k\|_1^2 + \tau\,\|z_h^k\|_0^2\right)\,.$$

The desired stability follows from

$$\sum_{k=1}^{3} |||z_h^k|||_{k,\tau}^2 \le C\,\sum_{k=1}^{3}\left(\|z_h^k\|_1^2 + \tau\,\|z_h^k\|_0^2\right) \le C\,\left(\|\boldsymbol{z}_h\|_1^2 + \tau\,\|\boldsymbol{z}_h\|_0^2\right)$$

and the estimates established for 3.3(a). □

3.3. The HX Decomposition Involving Discrete Divergence

We first define the discrete divergence operator $\mathrm{div}_h : \boldsymbol{V}_h \mapsto S_h$ as the unique solution to the problem

$$(\mathrm{div}_h \boldsymbol{\varPi}_h \boldsymbol{z}_h, q_h) = -(\boldsymbol{\varPi}_h \boldsymbol{z}_h, \nabla q_h) \tag{3.5}$$

for all $q_h \in S_h$. The following result is readily seen.

Lemma 3.1. *We have*

$$\|\mathrm{div}_h \boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0 \le C\, \|\boldsymbol{z}_h\|_1 \,.$$

Proof. The result follows from the definition of div_h, the inverse inequality on S_h and the approximation properties of $\boldsymbol{\Pi}_h$. We have

$$\begin{aligned}\|\mathrm{div}_h \boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0^2 &= -(\boldsymbol{\Pi}_h \boldsymbol{z}_h, \nabla\, \mathrm{div}_h \boldsymbol{\Pi}_h \boldsymbol{z}_h)\\ &= (\boldsymbol{z}_h - \boldsymbol{\Pi}_h \boldsymbol{z}_h, \nabla\, \mathrm{div}_h \boldsymbol{\Pi}_h \boldsymbol{z}_h) + (\mathrm{div}\, \boldsymbol{z}_h, \mathrm{div}_h \boldsymbol{\Pi}_h \boldsymbol{z}_h)\\ &\le C\, \|\boldsymbol{z}_h\|_1 \|\mathrm{div}_h \boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0 \,.\end{aligned}$$

This completes the proof of the lemma. □

As proposed in [6], the above stability estimate allows us to modify the norm used in the stability properties of the HX decomposition.

Theorem 3.4. *Any $\boldsymbol{u}_h \in \boldsymbol{V}_h$ admits the decomposition*

$$\boldsymbol{u}_h = \boldsymbol{v}_h + \boldsymbol{\Pi}_h \boldsymbol{z}_h + \nabla p_h \,,$$

where $\boldsymbol{v}_h \in \boldsymbol{V}_h$, $\boldsymbol{z}_h \in \boldsymbol{S}_h$, $p_h \in S_h \oplus K_{N,h}$ satisfy the following stability estimates

$$(h^{-2}+\tau)\|\boldsymbol{v}_h\|_0^2 + |||\boldsymbol{z}_h|||_{div,\,\tau}^2 + \tau\, \|\nabla p_h\|_0^2 \le C\, \left(\|\nabla \times \boldsymbol{u}_h\|_0^2 + \tau\, \|\boldsymbol{u}_h\|_0^2\right) \,.$$

The constant C is independent of the parameter $\tau \ge 0$. The norm $|||.||_{div,\,\tau}$ is defined by

$$|||\boldsymbol{z}_h|||_{div,\,\tau}^2 = \|\nabla \times \boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0^2 + \|\mathrm{div}_h \boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0^2 + \tau\, \|\boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0^2 \,.$$

We note that, for domains with no internal holes, the leading term $\|\nabla\times\boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0^2 + \|\mathrm{div}_h \boldsymbol{\Pi}_h \boldsymbol{z}_h\|_0^2$ of $|||\boldsymbol{z}_h|||_{div,\,\tau}^2$ gives rise to an invertible operator on Range $(\boldsymbol{\Pi}_h \boldsymbol{S}_h)$. However, $\boldsymbol{\Pi}_h$ may have a nontrivial kernel on $\boldsymbol{S}_h$.

3.4. Auxiliary Mesh Decompositions

Consider a different mesh, $\mathcal{T}_H$, contained in Ω such that H is of comparable size as h, i.e., $H \simeq h$. Typically, such an auxiliary mesh is the restriction of a uniform grid to the interior of the domain Ω. Let $\boldsymbol{V}_H$ and S_H be the Nédélec and piecewise linear finite element spaces associated with $\mathcal{T}_H$. The following result was proved in [15]. It assumes that the domain has no internal holes or tunnels.

Theorem 3.5. *Any $\boldsymbol{u}_h \in \boldsymbol{V}_h$ admits the decomposition*

$$\boldsymbol{u}_h = \boldsymbol{v}_h + \nabla q_h + \boldsymbol{\Pi}_h \boldsymbol{u}_H \,,$$

where $\boldsymbol{v}_h \in \boldsymbol{V}_h$, $q_h \in S_h$, $\boldsymbol{u}_H \in \boldsymbol{V}_H$ satisfy the following stability estimate uniformly in $\tau \ge 0$,

$$(h^{-2}+\tau)\|\boldsymbol{v}_h\|_0^2 + \tau\, h^{-2}\|q_h\|_0^2 + \|\nabla \times \boldsymbol{\Pi}_h \boldsymbol{u}_H\|_0^2 + \tau\, \|\boldsymbol{\Pi}_h \boldsymbol{u}_H\|_0^2 \le C\, \left(\|\nabla \times \boldsymbol{u}_h\|_0^2 + \tau\, \|\boldsymbol{u}_h\|_0^2\right) \,.$$

Proof. Due to the assumptions on the auxiliary mesh $\mathcal{T}_H$, there is a $p_H \in S_H$ such that

$$h^{-1}\|p_h - p_H\|_0 + \|\nabla p_H\|_0 \le C\, \|\nabla p_h\|_0 \,.$$

Similarly, let $\boldsymbol{z}_H \in \boldsymbol{S}_H$ be an approximation to $\boldsymbol{z}$ such that

$$\|\boldsymbol{z} - \boldsymbol{z}_H\|_0 \le C\, h\, \|\boldsymbol{z}\|_1 \,.$$

Note that $\boldsymbol{\Pi}_h$ is well defined on $\boldsymbol{V}_H$ and that $\nabla \Pi_h^S p_H = \boldsymbol{\Pi}_h \nabla p_H$ where Π_h^S is the nodal interpolation operator in S_h. Therefore, letting

$$\boldsymbol{v}_h = \boldsymbol{\Pi}_h(\boldsymbol{z} - \boldsymbol{\Pi}_H \boldsymbol{z}_H), \quad q_h = p_h - \Pi_h^S p_H, \text{ and } \boldsymbol{u}_H = \boldsymbol{\Pi}_H \boldsymbol{z}_H + \nabla p_H \in \boldsymbol{V}_H,$$

we arrive at the decomposition

$$\boldsymbol{u}_h = \boldsymbol{v}_h + \nabla q_h + \boldsymbol{\Pi}_h \boldsymbol{u}_H \,.$$

This time, the estimate of $\boldsymbol{v}_h$ is more involved and requires manipulations on element level described in detail in [15]. Applying this technique results in

$$\|\boldsymbol{\Pi}_h(\boldsymbol{z} - \boldsymbol{\Pi}_H \boldsymbol{z}_H)\|_0 \le C\, h\, \|\boldsymbol{z}\|_1 \,,$$

which establishes the estimates

$$\|\boldsymbol{v}_h\|_0 \le C\, h\, \|\nabla \times \boldsymbol{u}_h\|_0 \text{ and } \|\boldsymbol{v}_h\|_0 \le C\, \|\boldsymbol{u}_h\|_0.$$

For q_h, we have

$$\|q_h\|_0 \le \|p_h - p_H\|_0 + \|p_H - \Pi_h^S p_H\|_0 \le C\, h\, \|\nabla p_h\|_0 \le C\, h\, \|\boldsymbol{u}_h\|_0 \,.$$

Finally, the triangle inequality and inverse estimates imply

$$\|\boldsymbol{\Pi}_h \boldsymbol{u}_H\|_0 = \|\boldsymbol{u}_h - \boldsymbol{v}_h - \nabla q_h\|_0 \le C\, \|\boldsymbol{u}_h\|_0 \quad \text{and} \quad \|\nabla \times \boldsymbol{\Pi}_h \boldsymbol{u}_H\|_0 \le C\, \|\nabla \times \boldsymbol{u}_h\|_0 \,. \qquad \Box$$

Remark 3.2. The convergence of a two–level geometric multigrid (with Hiptmair smoothing) follows as a special case of the above Theorem 3.5. This is the case since in geometric multigrid the meshes are obtained by refinement of an initial coarse mesh, thus we can set (in Theorem 3.5) the auxiliary mesh $\mathcal{T}_H$ to be a coarse sub-mesh of $\mathcal{T}_h$.

4. The Construction of Auxiliary Space Preconditioners

In this section we review the construction of auxiliary space preconditioners based on given stable decompositions. We transition from operator to matrix notation, in order to clarify the practical implementation of the algorithms, as well as to facilitate the implementation discussion in the following section. Matrices will be typeset using a roman style font.

Let A_h be the stiffness matrix corresponding to $a(\cdot,\cdot)$ on $\boldsymbol{V}_h$. The matrix representation of the mapping

$$\varphi \in S_h \mapsto \nabla\varphi \in \boldsymbol{V}_h$$

is commonly called the discrete gradient matrix further denoted by G_h. This matrix is usually readily available since it simply describes the edges of $\mathcal{T}_h$ in terms of its vertices: if e is an edge with vertices v_1 and v_2, then the only two non-zero entries of G_h in the row corresponding to e are

$$(\mathrm{G}_h)_{e,v_1} = -(\mathrm{G}_h)_{e,v_2} = 1\,.$$

The interpolation operator $\boldsymbol{\Pi}_h$ induces different matrices depending on which auxiliary discrete space it is acting from. If $\{\boldsymbol{\Psi}_j\}$ spans a specific auxiliary space, the entries of the corresponding matrix are computed by evaluating the Nédélec space degrees of freedom:

$$(\mathbf{\Pi}_h)_{ij} = \int_{e_i} \boldsymbol{\Psi}_j \cdot \boldsymbol{t}_{e_i}\, ds\,.$$

For example, G_h is the matrix representation of the mapping $\boldsymbol{\Pi}_h : \nabla S_h \mapsto \boldsymbol{V}_h$, with spanning set $\{\nabla\varphi_j\}$, where $\{\varphi_j\}$ is the nodal basis of S_h.

Suppose that we have a small number of auxiliary spaces $\{\boldsymbol{V}_{k,h}\}$ with certain norms $\|\cdot\|_{\boldsymbol{V}_{k,h}}$ such that $\{\boldsymbol{\Pi}_{k,h}\boldsymbol{V}_{k,h}\}$ provide a stable decomposition of $\boldsymbol{V}_h$ for some operators $\boldsymbol{\Pi}_{k,h} : \boldsymbol{V}_{k,h} \mapsto \boldsymbol{V}_h$. That is, for any $\boldsymbol{u}_h \in \boldsymbol{V}_h$ there are $\boldsymbol{u}_{k,h} \in \boldsymbol{V}_{k,h}$ such that

$$\boldsymbol{u}_h = \sum_k \boldsymbol{\Pi}_{k,h}\boldsymbol{u}_{k,h} \quad \text{and} \quad \sum_k \|\boldsymbol{u}_{k,h}\|^2_{\boldsymbol{V}_{k,h}} \leq C\,\|\boldsymbol{u}_h\|^2_{\boldsymbol{V}_h}\,. \tag{4.1}$$

Let $\mathbf{\Pi}_{k,h}$ and $\mathrm{A}_{k,h}$ be the matrices corresponding to the mapping $\boldsymbol{\Pi}_h : \boldsymbol{\Pi}_{k,h}\boldsymbol{V}_{k,h} \mapsto \boldsymbol{V}_h$ and the bilinear form that defines the norm $\|\cdot\|_{\boldsymbol{V}_{k,h}}$, respectively. The auxiliary space preconditioner based on (4.1) can be viewed as a Schwarz method with non-traditional subspaces involving the above matrices, see [25]. For ease of presentation, in what follows, we concentrate on its additive form. It is defined simply as

$$\mathrm{B}_h = \sum_k \mathbf{\Pi}_{k,h}\mathrm{B}_{k,h}\mathbf{\Pi}^T_{k,h}\,,$$

where $\{\mathrm{B}_{k,h}\}$ are efficient preconditioners for $\{\mathrm{A}_{k,h}\}$.

The following well-known result, see e.g. [13], states that if the auxiliary space operators $\mathrm{A}_{k,h}$ bound the variationally defined ones $(\mathbf{\Pi}^T_{k,h}\mathrm{A}_h\mathbf{\Pi}_{k,h})$, then (4.1) ensures that B_h is an optimal preconditioner for the original matrix A_h.

Theorem 4.1. *Assume* (4.1) *and let the following estimates hold*

$$\|\boldsymbol{\Pi}_{k,h}\boldsymbol{u}_{k,h}\|_{\boldsymbol{V}_h} \leq C\,\|\boldsymbol{u}_{k,h}\|_{\boldsymbol{V}_{k,h}}\,. \tag{4.2}$$

Also, let $\mathrm{B}_{k,h}$ *be spectrally equivalent to* $\mathrm{A}^{-1}_{k,h}$*. Then,* B_h *is spectrally equivalent to* A^{-1}_h*.*

To illustrate the construction, consider the HX decomposition (3.3)a

$$(h^{-2}+\tau)\|\boldsymbol{v}_h\|^2_0 \;+\; \|\boldsymbol{z}_h\|^2_1 + \tau\,\|\boldsymbol{z}_h\|^2_0 \;+\; \tau\,\|\nabla p_h\|^2_0 \;\leq\; C\,\|\boldsymbol{u}_h\|^2_{\boldsymbol{V}_h}\,,$$

where $\|\boldsymbol{u}_h\|^2_{\boldsymbol{V}_h} = a(\boldsymbol{u}_h,\boldsymbol{u}_h)$ with $\alpha = 1$, $\beta = \tau$. In this case the auxiliary spaces are $\boldsymbol{V}_{1,h} = \boldsymbol{V}_h$, $\boldsymbol{V}_{2,h} = \boldsymbol{S}_h$ and $\boldsymbol{V}_{3,h} = S_h$ equipped with the following norms:

$$\|\boldsymbol{v}_h\|^2_{\boldsymbol{V}_{1,h}} = (h^{-2}+\tau)\|\boldsymbol{v}_h\|^2_0\,, \quad \|\boldsymbol{z}_h\|^2_{\boldsymbol{V}_{2,h}} = \|\boldsymbol{z}_h\|^2_1 + \tau\,\|\boldsymbol{z}_h\|^2_0 \quad \text{and} \quad \|p_h\|^2_{\boldsymbol{V}_{3,h}} = \tau\|\nabla p_h\|^2_0\,.$$

The mappings $\{\boldsymbol{\Pi}_{k,h}\}$ are defined accordingly as $\boldsymbol{\Pi}_{1,h} = I$, $\boldsymbol{\Pi}_{2,h} = \boldsymbol{\Pi}_h$ and $\boldsymbol{\Pi}_{3,h} = \nabla$ (i.e., the gradient operator). These choices are justified, not only because of the above stability estimate, but also because they satisfy (4.2). Indeed,

$$\|\boldsymbol{\Pi}_{1,h}\boldsymbol{v}_h\|_{\boldsymbol{V}_h} = \|\boldsymbol{v}_h\|_{\boldsymbol{V}_h} \leq C\,\|\boldsymbol{v}_h\|_{\boldsymbol{V}_{1,h}}$$

follows by inverse inequality, and $\|\boldsymbol{\Pi}_h\boldsymbol{z}_h\|_{\boldsymbol{V}_h} \leq C\,\|\boldsymbol{z}_h\|_{\boldsymbol{V}_{2,h}}$ is the same as (3.4). Finally,

$$\|\boldsymbol{\Pi}_{3,h}p_h\|^2_{\boldsymbol{V}_h} = \tau\,\|\nabla p_h\|^2_{\boldsymbol{V}_h} = \|p_h\|^2_{\boldsymbol{V}_{3,h}}.$$

The last prerequisite of Theorem 4.1 is that the auxiliary space solvers $\mathrm{B}_{k,h}$ are spectrally equivalent to $\mathrm{A}^{-1}_{k,h}$. This is satisfied if $\mathrm{B}_{1,h}$ corresponds to a smoothing based on A_h (since $\mathrm{A}_{1,h}$ is a scaled mass matrix spectrally equivalent to the diagonal of A_h), while $\mathrm{B}_{2,h}$ and $\mathrm{B}_{3,h}$ are efficient (AMG) preconditioners for the parameter dependent vector Poisson–like matrix $\mathrm{A}_{2,h}$ and the scalar Poisson matrix $\mathrm{A}_{3,h}$, respectively.

Similar considerations can be applied to all decompositions presented in the previous section, and below we list several of the resulting additive auxiliary space preconditioners. In all cases R_h denotes a *point smoother* for A_h (such as Gauss-Seidel). Note that we define the methods for general coefficients, even though the stability of the decompositions, and therefore Theorem 4.1, was established only for constant α and β.

(A) HX decomposition (3.3)a with Poisson auxiliary matrices:

$$\mathrm{B}_h = \mathrm{R}_h + \mathrm{G}_h \mathrm{B}_{v,h} \mathrm{G}_h^T + \boldsymbol{\Pi}_h \mathbf{B}_{v,h} \boldsymbol{\Pi}_h^T \,,$$

where $\mathrm{B}_{v,h}$ and $\mathbf{B}_{v,h}$ correspond to *AMG V-cycles* for the Poisson problems $(\beta\,\nabla u, \nabla v)$ and $(\alpha\nabla \boldsymbol{u}, \nabla \boldsymbol{v}) + (\beta \boldsymbol{u}, \boldsymbol{v})$ discretized on S_h and $\boldsymbol{S}_h$, respectively.

(B) HX decomposition (3.3)b with variational auxiliary matrices:

$$\mathrm{B}_h = \mathrm{R}_h + \mathrm{G}_h \mathrm{B}_{v,h} \mathrm{G}_h^T + \boldsymbol{\Pi}_h \mathbf{B}_{v,h} \boldsymbol{\Pi}_h^T \,,$$

where $\mathrm{B}_{v,h}$, $\mathbf{B}_{v,h}$ correspond to *AMG V-cycles* for $\mathrm{G}_h^{\mathrm{T}} \mathrm{A}_h \mathrm{G}_h$ and $\boldsymbol{\Pi}_h^{\mathrm{T}} \mathrm{A}_h \boldsymbol{\Pi}_h$, respectively. The magnetostatics form reads (cf. Corollary 3.2)

$$\mathrm{B}_h = \mathrm{R}_h + \boldsymbol{\Pi}_h \mathbf{B}_{v,h} \boldsymbol{\Pi}_h^T \,.$$

(C) Scalar HX decomposition from Theorem 3.3:

$$\mathrm{B}_h = \mathrm{R}_h + \mathrm{G}_h \mathrm{B}_{v,h} \mathrm{G}_h^T + \sum_{k=1}^{3} \Pi_h^k \, \mathrm{B}_{v,h}^k \, (\Pi_h^k)^T \,,$$

where $\mathrm{B}_{v,h}^k$ correspond to *AMG V-cycles* for $(\Pi_h^k)^T \mathrm{A}_h \, \Pi_h^k$.

(D) HX decomposition with discrete divergence from Theorem 3.4:

$$\mathrm{B}_h = \mathrm{R}_h + \mathrm{G}_h \mathrm{B}_{v,h} \mathrm{G}_h^T + \boldsymbol{\Pi}_h \mathbf{B}_{v,h} \boldsymbol{\Pi}_h^T \,,$$

where $\mathrm{B}_{v,h}$, $\mathbf{B}_{v,h}$ correspond to *AMG V-cycles* for $\mathrm{G}_h^{\mathrm{T}} \mathrm{A}_h \mathrm{G}_h$ and $\boldsymbol{\Pi}_h^{\mathrm{T}} \widehat{\mathrm{A}}_h \boldsymbol{\Pi}_h$ respectively. Here, the matrix $\widehat{\mathrm{A}}_h$ is obtained by discretizing $a(\boldsymbol{u}_h, \boldsymbol{v}_h) + (\mathrm{div}_h \boldsymbol{u}_h, \mathrm{div}_h \boldsymbol{v}_h)$ in $\boldsymbol{V}_h$ with possible lumping of the mass matrix inverse (see (3.5)).

(E) Auxiliary mesh decomposition:

$$\mathrm{B}_h = \mathrm{R}_h + \mathrm{G}_h \mathrm{R}_{v,h} \mathrm{G}_h^T + \boldsymbol{\Pi}_h \mathbf{B}_H \boldsymbol{\Pi}_h^T$$

where $\mathrm{R}_{v,h}$ is a point smoother for $\mathrm{G}_h^{\mathrm{T}} \mathrm{A}_h \mathrm{G}_h$, and B_H corresponds to a structured mesh *geometric multigrid* on $\boldsymbol{V}_H$.

The above list offers a variety of preconditioners for the definite and semi-definite Maxwell problems which are optimal, at least in the case of constant coefficients. In our experience, the nodal auxiliary space approaches (B) and (C) offer the best combination of robust performance and low requirements for additional discretization information.

5. Parallel Implementation

We recently developed a new parallel auxiliary space Maxwell solver (AMS) in the *hypre* library [1]. AMS is based on the HX decomposition methods (A), (B) and (C) from the previous section and employs *hypre*'s scalable algebraic multigrid solver BoomerAMG [10].

To solve a definite Maxwell problem, the user provides the linear system, the discrete gradient matrix and the vertex coordinates, as described in the following code segment:

```
HYPRE_Solver solver;
HYPRE_AMSCreate(&solver);

/* Set discrete gradient matrix */
HYPRE_AMSSetDiscreteGradient(solver, G);
/* Set vertex coordinates */
HYPRE_AMSSetCoordinateVectors(solver, x, y, z);

HYPRE_AMSSetup(solver, A, b, x);
HYPRE_AMSSolve(solver, A, b, x);
```

The interpolation matrices and coarse grid operators are then constructed automatically. The Poisson matrices on S_h and $\boldsymbol{S}_h$ can be utilized when available. A simplified magnetostatic mode corresponding to Corollary 3.2 is also supported.

The construction of $\boldsymbol{\Pi}_h$ in AMS uses only G_h and the coordinate vectors x, y and z. Specifically, if $\boldsymbol{z}_h = (z_1, z_2, z_3)$ is a vector of linear functions and e is the edge with vertices v_1 and v_2, then

$$\int_e \boldsymbol{z}_h \cdot \boldsymbol{t}_e \, ds = \frac{\boldsymbol{z}_h(v_1) + \boldsymbol{z}_h(v_2)}{2} \cdot (v_2 - v_1) \, .$$

If $\boldsymbol{z}_h$ is the x-basis function at v_1 or v_2, we get $\int_e \boldsymbol{z}_h \cdot \boldsymbol{t}_e \, ds = \frac{1}{2}(\mathrm{G}_h \mathrm{x})_e$. Therefore,

$$\boldsymbol{\Pi}_h = \left[\Pi_h^1, \, \Pi_h^2, \, \Pi_h^3\right]$$

where each block has the same sparsity structure as G_h and entries

$$(\Pi_h^1)_{e,v_1} = (\Pi_h^1)_{e,v_2} = \frac{(\mathrm{G}_h \mathrm{x})_e}{2} \, , \quad (\Pi_h^2)_{e,v_1} = (\Pi_h^2)_{e,v_2} = \frac{(\mathrm{G}_h \mathrm{y})_e}{2} \, , \quad (\Pi_h^3)_{e,v_1} = (\Pi_h^3)_{e,v_2} = \frac{(\mathrm{G}_h \mathrm{z})_e}{2} \, .$$

The behavior of the algorithm can be influenced by a number of tuning parameters. In their default settings, AMS is a multiplicative solver based on decomposition (B), which uses a convergent parallel hybrid smoother and single BoomerAMG V-cycles (with aggressive HMIS coarsening) for the algebraic problems in S_h and $\boldsymbol{S}_h$. We note that these parameters were optimized with respect to the total time to solution. Different choices are available, which give methods with better convergence properties (but slower in practice).

6. Numerical Experiments

6.1. Constant Coefficients

This section is devoted to assessing the numerical performance and parallel scalability of our implementation on a variety of definite and semi-definite Maxwell problems. Sequential results

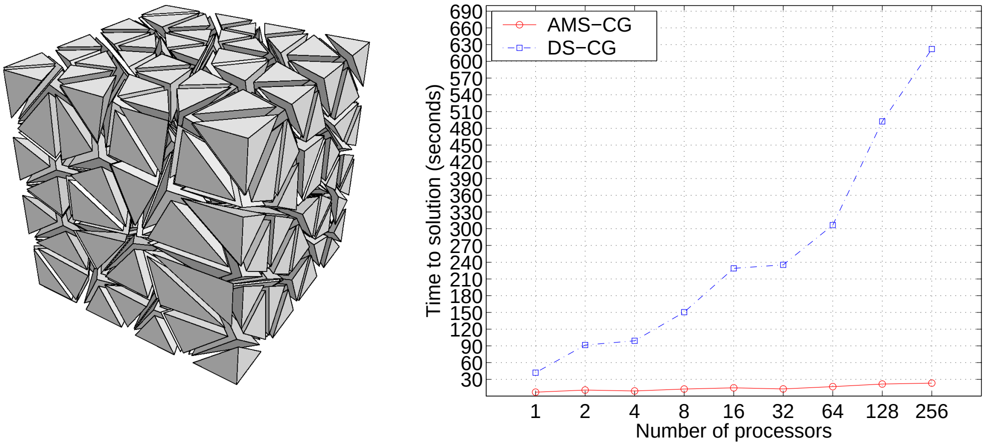


Fig. 6.1. Initial mesh and time to solution for the constant coefficients problem on the unit cube: AMS versus a Jacobi preconditioner (DS-CG).

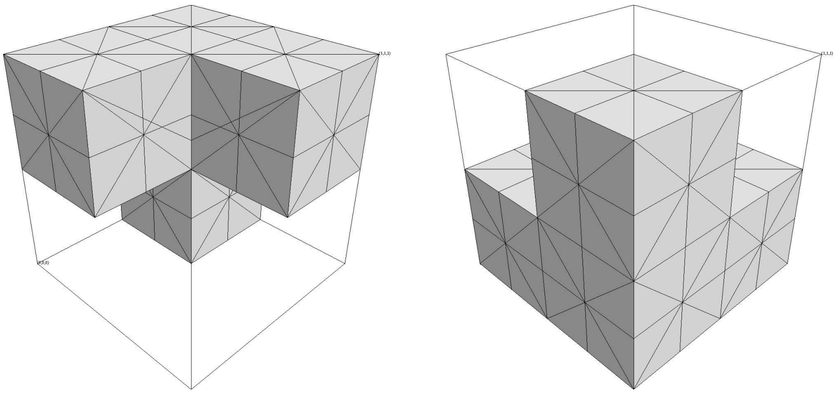


Fig. 6.2. The unit cube split into two symmetrical regions. Shown are the initial tetrahedral meshes.

with an earlier version of the method can be found in [17]. In all experiments we use AMS as a preconditioner in a conjugate-gradient (CG) iteration. The convergence tolerance corresponds to reduction in the preconditioned residual norm, which is typically set to 10^{-6}.

We tried to keep the problem size per processor in our scaling tests approximately the same, while increasing the number of processors. The unstructured nature of the problems, however, led to somewhat varying load balance. The following notation was used to record the results: np denotes the number of processors in the run, N is the total problem size, n_{it} is the number of AMS-CG iterations, while t_{setup}, t_{solve} and t denote the average times (in seconds) needed for setup, solve, and time to solution (setup plus solve) respectively. The code was executed on a machine with 2.4GHz Xeon processors. In these settings, an algorithm with perfect (weak) scalability will require the same time to run, regardless of how many processors are used.

We first consider a simple problem with $\alpha = \beta = 1$. The domain is the unit cube meshed with an unstructured tetrahedral mesh. The initial coarse mesh, before serial or parallel refinement is shown in Figure 6.1. The input matrices and vectors for this, and following similar tests, were constructed in parallel using the scalable finite element package aFEM.

In Table 6.1 we report scalability results and compare the AMS preconditioner with the

Table 6.1: Numerical results for the problem with constant coefficients ($\alpha = \beta = 1$) on a cube.

	AMS				AMG			
np	N	n_{it}	t_{setup}	t_{solve}	N	n_{it}	t_{setup}	t_{solve}
`default solver parameters`								
1	105,877	11	2.5s	4.9s	17,478	13	0.1s	0.3s
2	184,820	12	3.5s	8.4s	29,059	14	0.2s	0.5s
4	293,224	13	2.9s	6.9s	43,881	15	0.1s	0.4s
8	697,618	14	4.2s	9.7s	110,745	18	0.2s	0.7s
16	1,414,371	16	4.6s	11.0s	225,102	18	0.3s	0.8s
32	2,305,232	16	3.9s	9.7s	337,105	20	0.4s	0.9s
64	5,040,829	18	5.2s	12.8s	779,539	22	0.5s	1.4s
128	10,383,148	19	6.5s	15.9s	1,682,661	23	0.8s	1.8s
256	18,280,864	21	7.3s	17.0s	2,642,337	25	1.1s	2.1s
512	38,367,625	23	9.0s	22.0s	5,845,443	28	1.7s	2.8s
1024	78,909,936	25	17.9s	33.2s	12,923,121	30	4.0s	4.8s
`alternative solver parameters`								
1	105,877	4	5.5s	8.8s	17,478	6	0.4s	0.4s
2	184,820	4	9.8s	13.5s	29,059	7	0.7s	0.7s
4	293,224	4	10.5s	10.8s	43,881	7	1.2s	0.8s
8	697,618	5	21.2s	18.3s	110,745	7	2.1s	1.1s
16	1,414,371	5	38.0s	20.5s	225,102	7	3.9s	1.3s
32	2,305,232	5	53.8s	19.4s	337,105	8	7.3s	1.9s
64	5,040,829	6	79.3s	25.3s	779,539	8	10.7s	2.5s

BoomerAMG preconditioner in *hypre* applied to a Laplace problem discretized with linear finite elements on the same mesh. The data shows that the behavior of AMS is qualitatively similar to that of BoomerAMG. This trend is often observed in our experiments, and implies that any future improvements in (standard, i.e., designed for elliptic problems) AMG technology will lead to further improvements in AMS. In our view, this is a main advantage of the HX decomposition approach compared with the alternatives discussed earlier.

Since the problem has constant coefficients, we can apply Theorem 4.1 and conclude that the preconditioner should be optimal. With the default parameters, the number of AMS iterations increases slightly, but the total run time grows slowly and remains less than a minute. On the other hand, we can use an alternative set of parameters, see [16], which does indeed give constant number of iterations, but also leads to (almost linearly) increasing setup time.

While not perfect, the scalability of the AMS preconditioner is enough to significantly outperform traditional solvers. This is clearly demonstrated in Figure 6.1, where we compare AMS with the commonly used diagonally-scaled CG (or DS-CG) method. We only include results up to 256 processors, since on 512 processors (around 38M unknowns) DS-CG did not converge in $100,000$ iterations.

Table 6.2: Number of iterations for the problem on a cube with α and β having different values in the regions shown in Figure 6.2.

np	N	p									t
		-8	-4	-2	-1	0	1	2	4	8	
$\alpha = 1, \beta \in \{1, 10^p\}$											
1	83,278	9	9	9	9	9	9	10	11	11	5s
2	161,056	10	10	10	10	10	10	10	11	11	9s
4	296,032	11	12	12	12	11	11	12	13	13	9s
8	622,030	13	13	13	12	12	12	13	15	14	12s
16	1,249,272	13	13	13	13	13	13	13	15	14	13s
32	2,330,816	15	15	15	15	15	15	15	16	15	14s
64	4,810,140	16	16	16	16	16	15	16	18	17	17s
128	9,710,856	16	16	16	16	16	16	16	17	17	23s
256	18,497,920	19	19	19	19	19	19	19	21	20	27s
512	37,864,880	21	20	20	20	20	20	20	23	22	32s
1024	76,343,920	20	20	20	20	20	20	20	21	21	56s
$\beta = 1, \alpha \in \{1, 10^p\}$											
1	83,278	10	10	11	10	9	11	12	13	13	6s
2	161,056	10	10	11	10	10	11	12	12	12	10s
4	296,032	11	11	13	12	11	13	14	15	15	11s
8	622,030	13	13	15	14	12	14	16	16	16	14s
16	1,249,272	13	13	14	14	13	14	15	16	16	15s
32	2,330,816	14	15	16	16	15	17	17	18	18	16s
64	4,810,140	16	17	18	17	16	18	19	19	20	20s
128	9,710,856	14	17	17	17	16	18	18	18	19	26s
256	18,497,920	17	19	20	20	19	21	21	22	22	29s
512	37,864,880	19	20	22	22	20	23	24	24	25	36s
1024	76,343,920	17	20	21	21	20	22	23	22	23	76s

6.2. Definite Problems with Discontinuous Coefficients

In this subsection we explore several definite Maxwell problems, where the coefficients are discontinuous and have large jumps. We consider both model problems and real applications.

In the first experiment, we choose α and β to have different values in two regions of the unit cube. The geometry and the numerical results are presented in Figure 6.2 and Table 6.2. We chose this particular test problem, because it was reported to be problematic for geometric multigrid [11].

The iteration counts in Table 6.2 are comparable to the constant coefficient case in Table 6.1. Furthermore, the sensitivity to the magnitude of the jumps in the coefficients appears to be very mild. We can conclude that even though the existing theory does not cover variable coefficients, the actual performance of AMS does not degrade in this particular case.

The solution times in Table 6.2 also compare favorably to the previous experiment, and hint that AMS may enable the fast solution of very challenging problems. For example, we solved a problem with more than 76 million unknowns and 8 orders of magnitude jumps in the coefficients in less than a minute.

Next, we report results from a simulation of electromagnetic diffusion of a copper wire in air using the MHD package described in [22]. In this problem, β corresponds to the electric conductivity, which has a jump: $\beta_{air} = 10^{-6}\beta_{copper}$, while $\alpha \equiv 1$. The geometry, and the

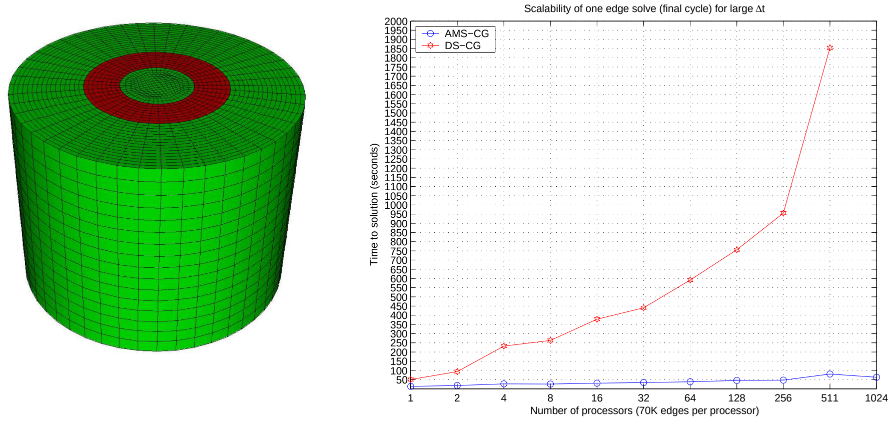


Fig. 6.3. Initial mesh and time to solution for the copper wire model.

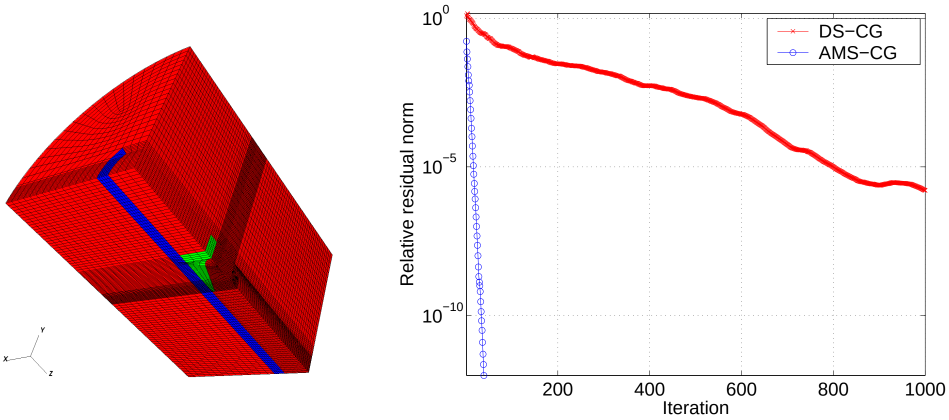


Fig. 6.4. Initial mesh and (logarithmic) converge history plot for a MHD simulation.

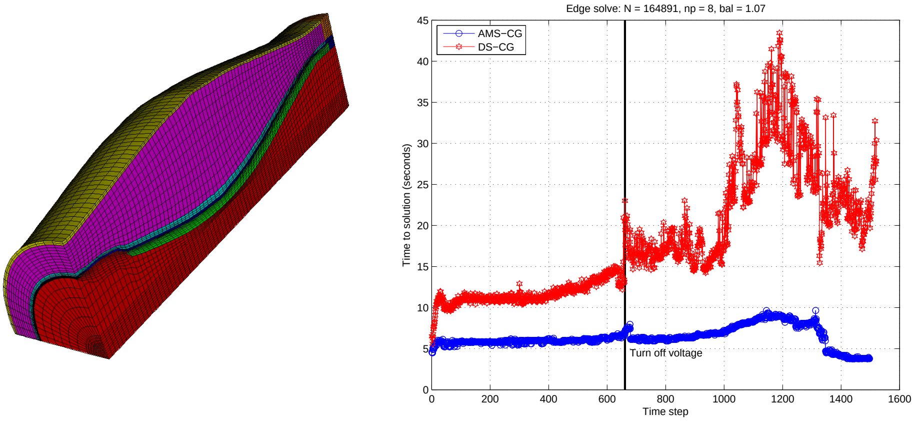
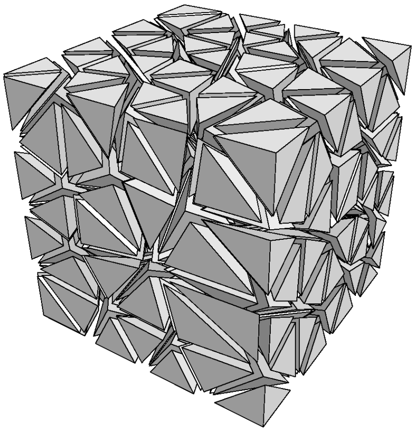


Fig. 6.5. The magnetic flux compression generator mesh in the middle of the calculation (left) and the time needed for the Maxwell solves at each time step during the simulation (right).

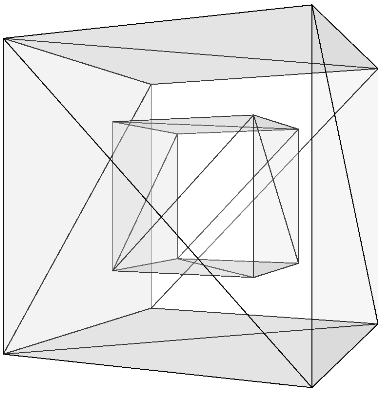
initial mesh are presented in Figure 6.3, where the copper region is colored in red. Unlike the previous problems, here we deal with hexahedral elements as well as mixed natural and essential

Table 6.3: Numerical results for the semi-definite problem on a cube with $\alpha = 1$ and $\beta = 0$.



np	N	n_{it}	t_{setup}	t_{solve}
1	105,877	11	2.0s	3.7s
2	184,820	12	2.8s	6.1s
4	293,224	13	2.3s	5.1s
8	697,618	15	3.3s	7.3s
16	1,414,371	16	3.8s	7.9s
32	2,305,232	17	3.3s	7.0s
64	5,040,829	19	4.5s	9.2s
128	10,383,148	20	5.4s	11.1s
256	18,280,864	23	5.9s	13.1s
512	38,367,625	24	10.4s	14.9s
1024	78,909,936	26	10.8s	21.6s

Table 6.4: Initial mesh and numerical results for the problem on a cube with $\alpha = 1$ and β equal to 1 inside and 0 outside the interior cube.



np	N	n_{it}	t_{setup}	t_{solve}
1	90,496	11	2.0s	4.7s
2	146,055	12	2.7s	7.3s
4	226,984	13	2.3s	6.3s
8	569,578	15	3.7s	9.3s
16	1,100,033	16	3.9s	10.8s
32	1,806,160	17	3.4s	9.1s
64	4,049,344	19	5.1s	12.8s
128	8,123,586	20	6.7s	15.8s
256	14,411,424	22	6.5s	17.2s
512	30,685,829	25	9.5s	22.6s
1024	61,933,284	24	19.5s	44.8s

boundary conditions. The convergence tolerance for this test was lowered to 10^{-9}.

To investigate the scalability of AMS, we discretized the problem in parallel, keeping approximately 70K edges per processor. The plot in Figure 6.3 shows that the auxiliary space preconditioner scales well, and can be more than 25 times faster than DS-CG when a large number of processors are used. Note that this problem arises in the context of time discretization, and the performance of DS-CG will degrade further for larger time-steps. In contrast, AMS is not sensitive to the size of the time step and allows larger jumps in the conductivity. In fact, it allows setting $\beta_{air} = 0$, as shown later in Section 6.3.

We remark that switching from the default options to the version of AMS corresponding to decomposition (C) from Section 4 in a large run of this problem ($N = 73$M, $np = 1024$) resulted in 38% improvement in the run time.

Another application in the same simulation settings is shown in Figure 6.4. Here, the mesh is locally refined and several materials are present. The coefficient β varies in a scale spanning seven orders of magnitude. The convergence tolerance is set to 10^{-8}, and the problem size is $372,644$ on 12 processors. The traditionally used DS-CG solver needed *409 seconds* and performed 16,062 iterations to converge. AMS solved the problem *18.6 seconds* (27 iterations).

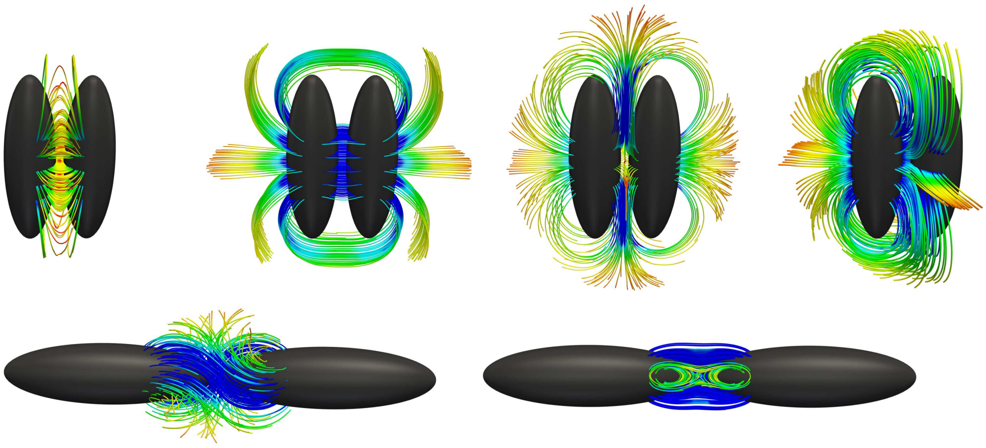


Fig. 6.6. Streamlines of the magnetic fields generated by interactions between two magnetic orbs.

In addition, the steady convergence rate shown on the plot in Figure 6.4 allowed setting the convergence tolerance even lower, which was important to the users in this case.

Our final example involves the simulation of a magnetically-driven compression generator, where the definite Maxwell problem is coupled with hydrodynamics on each time step. For more details we refer to [22, 23]. In this calculation, the mesh deforms in time, which produces hexahedral elements with bad aspect ratios as shown in Figure 6.5. A number of different materials occupy portions of the domain, which results in large jumps in β and small variations in α. The magnitude of these jumps changes throughout the course of the calculation.

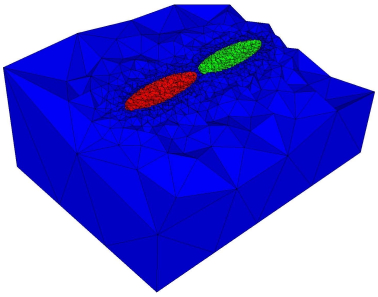
The full simulation run of this model can take hours of execution time. Therefore, we considered a coarse discretization having approximately 165K edges on 8 processors. This problem required around 1500 time steps, and took *more than 9 hours* using DS-CG. As plotted in Figure 6.5, the performance of AMS was much more stable, which reduced the total simulation time to *less than 4 hours*.

6.3. Semi-definite Problems

In this subsection we report results for problems where β is identically zero in the domain (Table 6.3) or in part of it (Table 6.4). In both cases we set $\alpha = 1$.

The iteration counts in Table 6.3 are very similar to those in Table 6.1. Since $\beta = 0$ we employ the magnetostatic version of decomposition (B) from Section 4.

Table 6.5: Initial mesh and numerical results corresponding to vertical attraction between two orbs.



np	N	n_{it}	t_{setup}	t_{solve}
2	397,726	10	4.71s	22.00s
16	2,007,616	12	5.34s	17.41s
128	13,143,442	13	5.57s	16.88s
1024	95,727,110	14	9.39s	19.68s

This results in significant reduction in the setup and solution times compared to Table 6.1, while the overall scalability remains very good.

When β is zero only in part of the domain, we cannot use the magnetostatic version of decomposition (B). The results in Table 6.4 are nevertheless very good and comparable to the experiments with constant coefficients.

Finally, as a more realistic magnetostatic test case, we consider the modeling of the interaction between two magnetic orbs, see Figure 6.6. These are just magnets with poles on the sides which, for simplicity, are modeled by current loops. This means that we are solving a semi-definite problem with $\alpha = 1$, $\beta = 0$ and right-hand side which traces the ellipsoidal geometry of each orb. To insure that this right-hand side is compatible, we first remove its gradient components using a Poisson solve.

We illustrate the behavior of the solver on these types of problems by simulating the vertical attraction between the orbs corresponding to the last configuration in Figure 6.6. Here we used the version of AMS based on the "scalar" HX decomposition (C) from Section 4. Table 6.5 shows the initial locally refined mesh and reports the convergence results. They are another confirmation of the scalable behavior of AMS on large-scale magnetostatic problems.

7. Conclusions

The auxiliary space AMG based on the recent Hiptmair-Xu decomposition from [13] provides a number of scalable algebraic preconditioners for a variety of unstructured definite and semi-definite Maxwell problems.

Our AMS implementation in the *hypre* library requires some additional user input besides the problem matrix and r.h.s; namely, the discrete gradient matrix and the coordinates of the vertices of the mesh. It can handle both variable coefficients and problems with zero conductivity. In model simulations the performance on such problems is very similar to that on problems with $\alpha = \beta = 1$. In MHD applications, AMS-CG can be orders of magnitude faster than DS-CG.

The behavior of AMS on Maxwell problems is qualitatively similar to that of AMG on (scalar and vector) elliptic problems discretized on the same mesh. Thus, any further improvements in AMG, will likely lead to additional improvements in AMS.

Acknowledgments. This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. UCRL-JRNL-237306.

References

[1] *hypre* : High performance preconditioners. http://www.llnl.gov/CASC/hypre/.

[2] C. Amrouche, C. Bernardi, M. Dauge, and V. Girault, Vector potentials in three-dimensional nonsmooth domains, *Math. Method. Appl. Sci.*, **21**:9 (1998), 823-864.

[3] D. N. Arnold, R. S. Falk, and R. Winther, Multigrid in H(div) and H(curl), *Numer. Math.*, **85** (2000), 197-217.

[4] P. Bochev, C. Garasi, J. Hu, A. Robinson, and R. Tuminaro, An improved algebraic multigrid method for solving Maxwell's equations, *SIAM J. Sci. Comput.*, **25**:2 (2003), 623-642.

[5] P. Bochev, J. Hu, A. Robinson, and R. Tuminaro, Towards robust 3D Z-pinch simulations: discretization and fast solvers for magnetic diffusion in heterogeneous conductors, *Electron. T. Numer. Ana.*, **15** (2003), 186-210.

[6] P. Bochev, J. Hu, C. Siefert, and R. Tuminaro, An algebraic multigrid approach based on a compatible gauge reformulation of Maxwell's equations, Technical Report SAND2007-1633, Sandia, Albuquerque, New Mexico, USA, 2007. To appear in *SIAM J. Sci. Comput.*

[7] P. Clément, Approximation by finite element functions using local regularization, *Rev. Française Automat. Informat. Recherche Opérationnelle Sér. Rouge Anal. Numér.*, **9**:R-2 (1975), 77-84.

[8] M. Gee, C. Siefert, J. Hu, R. Tuminaro, and M. Sala, ML 5.0 smoothed aggregation user's guide, Technical Report SAND2006-2649, Sandia National Laboratories, 2006.

[9] V. Girault and P. Raviart, Finite Element Approximation of the Navier-Stokes Equations, volume 749 of Lecture Notes in Mathematics. Springer-Verlag, New York, 1981.

[10] V. E. Henson and U. M. Yang, BoomerAMG: a parallel algebraic multigrid solver and preconditioner, *Appl. Numer. Math.*, **41** (2002), 155-177.

[11] R. Hiptmair, Multigrid method for Maxwell's equations, *SIAM J. Numer. Anal.*, **36** (1999), 204-225.

[12] R. Hiptmair, G. Widmer, and J. Zou, Auxiliary space preconditioning in H_0(curl; Ω), *Numer. Math.*, **103**:3 (2006), 435-459.

[13] R. Hiptmair and J. Xu, Nodal auxiliary space preconditioning in H(curl) and H(div) spaces, *SIAM J. Numer. Anal.*, **45**:6 (2007), 2483-2509.

[14] J. Jones and B. Lee, A multigrid method for variable coefficient Maxwell's equations, *SIAM J. Sci. Comput.*, **27**:5 (2006), 1689-1708.

[15] T. V. Kolev, J. E. Pasciak, and P. S. Vassilevski, H(curl) auxiliary mesh preconditioning, *Numer. Linear Algebr.*, **15**:5 (2008), 455-471.

[16] T. V. Kolev and P. S. Vassilevski, Parallel H1-based auxiliary space AMG solver for H(curl) problems, Technical Report UCRL-TR-222763, LLNL, Livermore, California, USA, 2006.

[17] T. V. Kolev and P. S. Vassilevski, Auxiliary space AMG for H(curl) problems, In Domain Decomposition Methods in Science and Engineering XVII, volume 60 of Lecture Notes in Computational Science and Engineering, pages 147-154. Springer, 2008.

[18] P. Monk, Finite Element Methods for Maxwell's Equations, Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, UK, 2003.

[19] J. E. Pasciak and J. Zhao, Overlapping Schwarz Methods in $H(curl)$ on Nonconvex Domains, *East West J. Num. Anal.*, **10** (2002), 221-234.

[20] S. Reitzinger and J. Schöberl, An algebraic multigrid method for finite element discretizations with edge elements, *Numer. Linear Algebr.*, **9**:3 (2002), 223-238.

[21] R. Rieben and D. White, Verification of high-order mixed finite-element solution of transient magnetic diffusion problems, *IEEE T. Magn.*, **42**:1 (2006), 25-39.

[22] R. Rieben, D. White, B. Wallin, and J. Solberg, An arbitrary Lagrangian-Eulerian discretization of MHD on 3D unstructured grids, *J. Comput. Phys.*, **226**:1 (2007), 534-570.

[23] J. W. Shearer, F. F. Abraham, C. M. Aplin, B. P. Benham, J. E. Faulkner, F. C. Ford, M. M. Hill, C. A. McDonald, W. H. Stephens, D. J. Steinberg, and J. R. Wilson, Explosive-driven magnetic-field compression generators, *J. Appl. Phys.*, **39**:4 (1968), 2102-2116.

[24] E. M. Stein, Singular Integrals and Differentiability Properties of Functions, Princeton Mathematical Series. Princeton University Press, Princeton, NJ, 1970.

[25] J. Xu, The auxiliary space method and optimal preconditioning techniques for unstructured grids, *Computing*, **56**:3 (1996), 215-235.

[26] J. Zhao, Analysis of Finite Element Approximation and Iterative Methods for Time-Dependent Maxwell Problems, Ph.D. dissertation, Department of Mathematics, Texas A&M University, College Station, 2002.

[27] L. T. Zikatanov, Two-sided bounds on the convergence rate of two-level methods, *Numer. Linear Algebr.*, **15**:5 (2008), 439-454.