NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. 2000; **00**:1–6 *Prepared using* nlaauth.cls *[Version: 2002/09/18 v1.02]*

Algebraic Multigrid for k-form Laplacians

Nathan Bell*and Luke N. Olson

Siebel Center for Computer Science, University of Illinois at Urbana-Champaign,
201 North Goodwin Avenue, Urbana, IL 61801, U.S.A.

SUMMARY

In this paper we describe an aggregation-based algebraic multigrid method for the solution of discrete k-form Laplacians. Our work generalizes Reitzinger and Schöberl's algorithm to higher dimensional discrete forms. We provide conditions on the tentative prolongators under which the commutativity of the coarse and fine de Rham complexes is maintained. Further, a practical algorithm that satisfies these conditions is outlined and smoothed prolongation operators and the associated finite element spaces are highlighted. Numerical evidence of the efficiency and generality of the proposed method is presented in the context of discrete Hodge decompositions. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: algebraic multigrid, Hodge decomposition, discrete forms, mimetic methods, Whitney forms

1. INTRODUCTION

Discrete differential k-forms arise in scientific disciplines ranging from computational electromagnetics to computer graphics. Examples include stable discretizations of the eddy-current problem [23, 6, 2], topological methods for sensor network coverage [8], visualization of complex flows [18, 20], and the design of vector fields on meshes [10].

In this paper we consider solving problems of the form,

$$\boldsymbol{\delta d}\alpha^k = \beta^k, \tag{1}$$

where $\boldsymbol{d}$ denotes the exterior derivative and $\boldsymbol{\delta}$ the codifferential relating k-forms α and β. For $k = 0, 1, 2$, $\boldsymbol{\delta d}$ is also expressed as $\boldsymbol{\nabla} \cdot \boldsymbol{\nabla}$, $\boldsymbol{\nabla} \times \boldsymbol{\nabla}\times$, and $\boldsymbol{\nabla\nabla}\cdot$ respectively. We refer to operator $\boldsymbol{\delta d}$ generically as a Laplacian, although it does not correspond to the Laplace-de Rham operator $\boldsymbol{\Delta} = \boldsymbol{d\delta}+\boldsymbol{\delta d}$ except for the case $k = 0$. We assume that (1) is discretized with mimetic first-order elements such as Whitney forms [22, 7] on simplicial meshes or the analog on hexahedral [5] or polyhedral elements [12]. In general we use $\mathcal{I}_k$ to denote the map from discrete k-forms (cochains) to their respective finite elements. Such discretizations give rise to a discrete

[*] Correspondence to: N. Bell, Siebel Center for Computer Science, University of Illinois at Urbana-Champaign, 201 North Goodwin Avenue, Urbana, IL 61801, U.S.A., email: wnbell@uiuc.edu

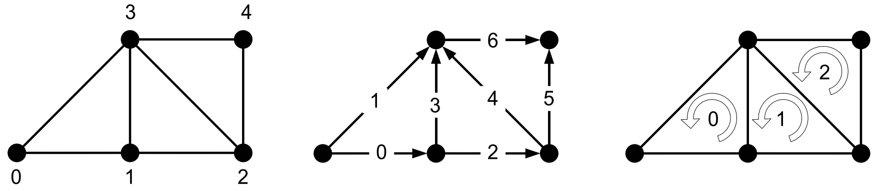

Figure 1: Enumeration of nodes (left), oriented edges (center), and oriented triangles(right) for a simple triangle mesh. We say that vertices 2 and 3 are *upper-adjacent* since they are joined by edge 4. Similarly, edges 5 and 6 are both faces of triangle 2 and therefore upper-adjacent.

exterior k-form derivative $\mathbb{D}_k$ and discrete k-form innerproduct $\mathbb{M}_k(i,j) = <\mathcal{I}_k e_i, \mathcal{I}_k e_j>$, which allows implementation of (1) in weak-form as

$$\mathbb{D}_k^T \mathbb{M}_{k+1} \mathbb{D}_k x = b, \tag{2}$$

under the additional assumption that $\boldsymbol{d}$ commutes with $\mathcal{I}$, i.e. $\mathcal{I}_{k+1}\mathbb{D}_k = \boldsymbol{d}_k \mathcal{I}_k$. This relationship is depicted as

$$\begin{tikzcd}
\Omega^k \arrow[r, "\boldsymbol{d}_k"] & \Omega^{k+1} \\
\Omega_d^k \arrow[u, "\mathcal{I}_k"] \arrow[r, "\mathbb{D}_k"] & \Omega_d^{k+1} \arrow[u, "\mathcal{I}_{k+1}"]
\end{tikzcd} \tag{3}$$

where Ω^k and Ω_d^k denote the spaces of differential k-forms and discrete k-forms respectively. For the remainder of the paper we restrict our attention to solving (2) on structured or unstructured meshes of arbitrary dimension and element type, provided that the elements satisfy the aforementioned commutativity property.

1.1. Example

Although our results hold more generally, it is instructive to examine a concrete example that satisfies the assumptions set out in Section 1. To this end, consider the three element simplicial mesh depicted in Figure 1, with the enumeration and orientation of vertices, edges, and triangles as shown. In this example, we choose Whitney forms [22] to define the interpolation operators $\mathcal{I}_0, \mathcal{I}_1, \mathcal{I}_2$ which in turn determine the discrete innerproducts $\mathbb{M}_0, \mathbb{M}_1 . \mathbb{M}_2$. Finally, sparse matrices

$$\mathbb{D}_{-1} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \qquad \mathbb{D}_0 = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}, \tag{4}$$

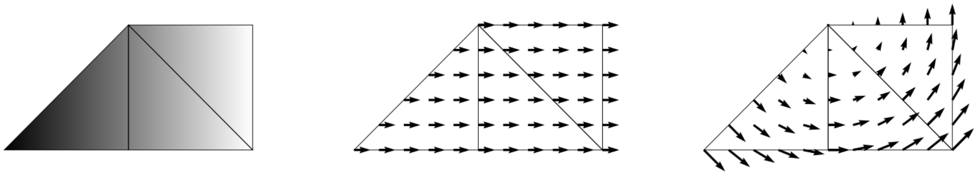

Figure 2: Forms $\mathcal{I}_0\alpha^0$, $\mathcal{I}_1\mathbb{D}_0\alpha^0$, and $\mathcal{I}_1\beta^1$ where $\mathcal{I}$ denotes Whitney interpolation. The left and center figures illustrate property (3). Whether the derivative is applied before or after interpolation, the result is the same.

$$\mathbb{D}_1 = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 \end{bmatrix}, \qquad \mathbb{D}_2 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}, \tag{5}$$

implement the discrete k-form derivative operators. A discrete k-form (cochain), denoted α^k, is represented by a column vector with entries corresponding to each of the k-simplices in the mesh. For example, the Whitney-interpolated fields corresponding to $\alpha^0 = [0,1,2,1,2]^T$, the gradient $\mathbb{D}_0\alpha^0 = [1,1,1,0,-1,0,1]^T$, and another 1-form $\beta^1 = [1,0,1,0,0,1,0]^T$ are shown in Figure 2. By convention $\mathbb{D}_{-1}$ and $\mathbb{D}_2$ are included to complete the exact sequence.

1.2. Related Work

There is significant interest in efficient solution methods for Maxwell's eddy-current problem

$$\nabla \times \nabla \times \vec{E} + \sigma\vec{E} = \vec{f}. \tag{6}$$

In particular, recent approaches focus on multilevel methods for both structured and unstructured meshes [13, 3, 19, 15]. Scalar multigrid performs poorly on edge element discretizations of (6) since error modes that lie in the kernel of $\nabla \times \nabla \times$ are not effectively damped by standard relaxation methods. Fortunately, the problematic modes are easily identified by the range of the discrete gradient operator $\mathbb{D}_0$, and an appropriate hybrid smoother [13, 3] can be constructed. An important property of these multigrid methods is commutativity between coarse and fine finite element spaces. The relationship is described as

$$\begin{tikzcd} \Omega_d^0 \arrow[r, "\mathbb{D}_0"] & \Omega_d^1 \\ \widehat{\Omega}_d^0 \arrow[u, "P_0"'] \arrow[r, "\widehat{\mathbb{D}}_0"] & \widehat{\Omega}_d^1 \arrow[u, "P_1"'] \end{tikzcd} \tag{7}$$

where $\widehat{\Omega}_d^k$ is the space of coarse discrete k-forms, $\widehat{\mathbb{D}}_0$ the coarse gradient operator, and P_0 and P_1 are the nodal and edge prolongation operators respectively. Combining (7) with (3) yields the same result for the corresponding fine and coarse finite element spaces.

In [19] Reitzinger and Schöberl describe an algebraic multigrid method for solving (6) on unstructured meshes. In their method, property (7) is maintained by choosing nodal aggregates and using these aggregates to obtain compatible edge aggregates. The nodal and edge aggregates then give rise to piecewise-constant prolongators P_0 and P_1, which can be smoothed to achieve better multigrid convergence rates [15] while retaining property (7).

The method we present can be viewed as a natural extension of Reitzinger and Schöberl's work from 1-forms to general k-forms. Commutativity of the coarse and fine de Rham complexes is maintained for all k-forms, and their associated finite element spaces $\mathcal{I}_k\Omega_d^k \subset \Omega^k$. The relationship is described by

$$\begin{tikzcd}
\Omega_d^0 \arrow[r, "\mathbb{D}_0"] & \Omega_d^1 \arrow[r, "\mathbb{D}_1"] & \Omega_d^2 & \cdots & \Omega_d^k \arrow[r, "\mathbb{D}_k"] & \Omega_d^{k+1} \\
\widehat{\Omega}_d^0 \arrow[r, "\widehat{\mathbb{D}}_0"] \arrow[u, "\mathcal{P}_0"'] & \widehat{\Omega}_d^1 \arrow[r, "\widehat{\mathbb{D}}_1"] \arrow[u, "\mathcal{P}_1"'] & \widehat{\Omega}_d^2 \arrow[u, "\mathcal{P}_2"'] & \cdots & \widehat{\Omega}_d^k \arrow[r, "\widehat{\mathbb{D}}_k"] \arrow[u, "\mathcal{P}_k"'] & \widehat{\Omega}_d^{k+1} \arrow[u, "\mathcal{P}_{k+1}"']
\end{tikzcd} \tag{8}$$

where $\mathcal{P}_k$ denotes either the tentative prolongator P_k or smoothed prolongator $\mathcal{S}_kP_k$.

1.3. Focus and Applications

While our work is largely inspired by multigrid solvers for (6), our intended applications do not focus specifically on the eddy-current problem. Indeed, recent work suggests that the emphasis on multilevel commutativity, a property further developed in this paper, is at odds with developing efficient solvers for (6) in the presence of highly variable coefficients [16]. Although our method generalizes the work of [19] and [15], this additional generality does not specifically address the aforementioned eddy-current issues.

In Section 3 we discuss computing Hodge decompositions of discrete k-forms with the proposed method. The Hodge decomposition is a fundamental tool in both pure and applied mathematics that exposes topological information through differential forms. For example, the two harmonic 1-forms shown in Figure 3 exist because the manifold has genus 1. The efficient solution of discrete k-form Laplacians has substantial utility in computational topology. For instance, sufficient conditions on the coverage of sensor networks reduce to the discovery of harmonic forms on the simplicial Rips complex [8]. In such applications we do not encounter variable coefficients and often take the identity matrix for $\mathbb{M}_k$.

2. PROPOSED METHOD

2.1. Complex Coarsening

In this section we describe the construction of tentative prolongators P_k and coarse operators $\widehat{\mathbb{D}}_k$ which satisfy (8). In practice, the two-level commutativity depicted in (8) is extended recursively for use in a multilevel method. Also, it is important to note that when solving (2) for a specific k, it is not necessary to coarsen the entire complex.

As in [19], we presume the existence of a nodal aggregation algorithm which produces a piecewise-constant tentative prolongator P_0. This procedure, called `aggregate_nodes` in

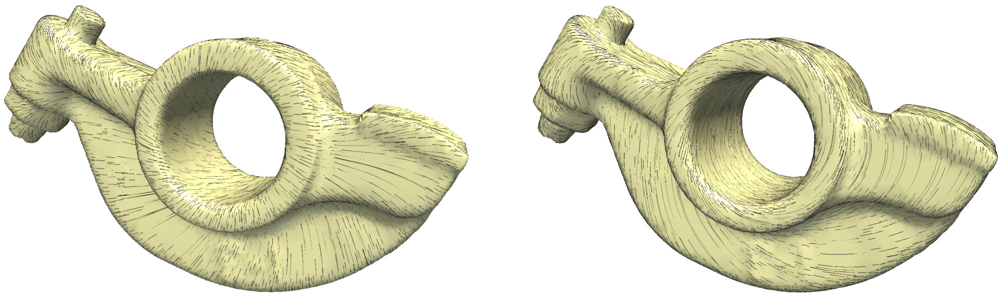

Figure 3: The two harmonic 1-forms of a rocker arm surface mesh.

Algorithm 1, is fulfilled by either Smoothed Aggregation [21] or a graph partitioner on matrices $\mathbb{D}_0^T \mathbb{M}_1 \mathbb{D}_0$ or $\mathbb{D}_0^T \mathbb{D}_0$. Ideally, the nodal aggregates are contiguous and have a small number of interfaces with other aggregates.

Algorithm 1: `coarsen_complex`$(\mathbb{D}_{-1}, \mathbb{D}_0, \ldots, \mathbb{D}_N)$

```
$P_0 \Leftarrow$ aggregate_nodes$(\mathbb{D}_0, \ldots)$
for $k = 0$ to $N-1$
    $P_{k+1} \Leftarrow$ induced_aggregates$(P_k, \mathbb{D}_k, \mathbb{D}_{k+1})$
    $\widehat{\mathbb{D}}_k \Leftarrow (P_{k+1}^T P_{k+1})^{-1} P_{k+1}^T \mathbb{D}_k P_k$
end
$\widehat{\mathbb{D}}_{-1} \Leftarrow P_0^T \mathbb{D}_{-1}$
$\widehat{\mathbb{D}}_N \Leftarrow \mathbb{D}_N P_N$
return $P_0, P_1, \ldots, P_N$ and $\widehat{\mathbb{D}}_{-1}, \widehat{\mathbb{D}}_0, \ldots, \widehat{\mathbb{D}}_N$
```

2.2. Induced Aggregates

The key concept in [19], which we apply and extend here, is that nodal aggregates *induce* edge aggregates; we denote P_1 as the resulting edge aggregation operator. As depicted in Figure 4, a coarse edge exists between two coarse nodal aggregates when any fine edge joins them. Multiple fine edges between the same two coarse nodal aggregates interpolate from a common coarse edge with weight 1 or -1 depending on their orientation relative to the coarse edge. The coarse nodes and coarse edges define a coarse derivative operator $\widehat{\mathbb{D}}_0$ which satisfies diagram (7).

We now restate the previous process in an algebraic manner that generalizes to arbitrary k-forms. Given P_0 as before, form the product $\overline{\mathbb{D}} = \mathbb{D}_0 P_0$ which relates coarse nodes to fine edges. Observe that each row of $\overline{\mathbb{D}}$ corresponds to a fine edge and each column to a coarse node. Notice that the i-th row of $\overline{\mathbb{D}}$ is zero when the endpoints of fine edge i lie within the same nodal aggregate. Conversely, the i-th row of $\overline{\mathbb{D}}$ is nonzero when the endpoints of fine edge i lie in different nodal aggregates. Furthermore, when two nonzero rows are equal up to a sign (i.e. linearly dependent), they interpolate from a common coarse edge.

Therefore, the procedure of aggregating edges reduces to computing sets of linearly

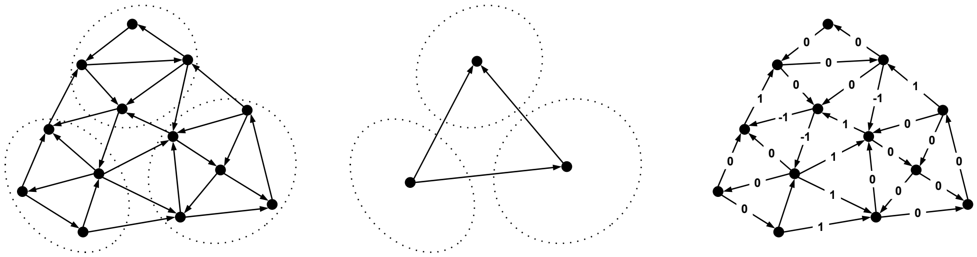

Figure 4: Nodal aggregates (left) determine coarse edges (center) through the algorithm `induced_aggregates`. Fine edges crossing between node aggregates interpolate from the corresponding coarse edge with weight 1 or -1 depending on their relative orientation. Edges contained within an aggregate do not correspond to any coarse edge and receive weight 0. These weights are determined by lines 10-13 of `induced_aggregates`.

dependent rows in $\overline{\mathbb{D}}$. Each set of dependent rows yields a coarse edge and thus a column of P_1. In the general case, sets of dependent rows in $\overline{\mathbb{D}} = \mathbb{D}_k P_k$ are identified and used to produce P_{k+1}. The process can be repeated to coarsen the entire de Rham complex. Alternatively, the coarsening can be stopped at a specific $k < N$. In Section 2.5 we discuss the coarse derivative operator $\widehat{\mathbb{D}}_k \Leftarrow (P_{k+1}^T P_{k+1})^{-1} P_{k+1}^T \mathbb{D}_k P_k$ and show that it satisfies diagram (8).

Algorithm 2: induced_aggregates($P_k, \mathbb{D}_k, \mathbb{D}_{k+1}$)

```
1  $\overline{\mathbb{D}} \Leftarrow \mathbb{D}_k P_k$
2  $\mathbb{G} \Leftarrow \mathbb{D}_{k+1}^T \mathbb{D}_{k+1}$
3  $V \Leftarrow \{\}$
4  $n \Leftarrow 0$
5
6  for $i$ in $rows(\overline{\mathbb{D}})$ such that $\overline{\mathbb{D}}(i,:) \neq 0$
7     if $i \notin V$
8        $A_n \Leftarrow$ dependent_rows($\mathbb{G}, \overline{\mathbb{D}}, i$)
9        for $j \in A_n$
10          if $\overline{\mathbb{D}}(i,:) = \overline{\mathbb{D}}(j,:)$
11             $P_{k+1}(j,n) \Leftarrow 1$
12          else
13             $P_{k+1}(j,n) \Leftarrow -1$
14          end
15       end
16       $n \Leftarrow n+1$
17       $V \Leftarrow V \cup A_n$
18    end
19 end
20 return $P_{k+1}$
```

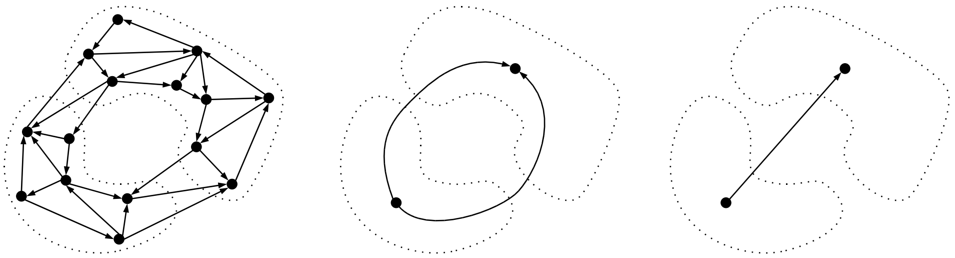

Figure 5: Example where contiguous (center) and non-contiguous (right) aggregation differs. Contiguous aggregates are reflected through our choice of $\mathbb{G}$ defined in `induced_aggregates` and later used in `dependent_rows`.

Intuitively, linear dependence between rows in $\overline{\mathbb{D}} = \mathbb{D}_k P_k$ indicates redundancy created by operator P_k. Aggregating dependent rows together removes redundancy from the output of $\overline{\mathbb{D}}$ and compresses the remaining degrees of freedom into a smaller set of variables. By construction, the tentative prolongators have full column rank and satisfy

$$\mathcal{R}(\mathbb{D}_k P_k) \subset \mathcal{R}(P_{k+1}) \tag{9}$$

where $\mathcal{R}(A)$ denotes the range of matrix A. Note that property (9) is clearly necessary to satisfy diagram (8).

Using disjoint sets of dependent rows $A_0, A_1, \ldots$, the function `induced_aggregates` constructs the aggregation operator P_{k+1} described above. Non-zero entry $P_{k+1}(i,j)$ indicates membership of the i-th row of $\overline{\mathbb{D}}$—i.e. the i-th $k+1$-dimensional element—to the j-th aggregate A_j.

2.3. Computing Aggregates

For a given row index i, the function `dependent_rows` constructs a set of rows that are linearly dependent to $\overline{\mathbb{D}}(i,:)$. In the matrix graph of $\mathbb{G}$, a nonzero entry $\mathbb{G}(i,j)$ indicates that the $k+1$-dimensional elements with indices i and j are *upper-adjacent* [17]. In other words, i and j are both faces of some $k+2$ dimensional element. For example, two edges in a simplicial mesh are upper-adjacent if they belong to the same triangle. All linearly dependent rows that are adjacent in the matrix graph of $\mathbb{G}$ are aggregated together. This construction ensures that the aggregates produced by `dependent_rows` are contiguous. As shown in Figure 5, such aggregates are more natural than those that result from aggregating all dependent rows together (i.e. using $\mathbb{G} = \overline{\mathbb{D}}\,\overline{\mathbb{D}}^T$).

Algorithm 3: `dependent_rows`$(\mathbb{G}, \overline{\mathbb{D}}, i)$

```
$Q \Leftarrow \{i\}$
$A \Leftarrow \{i\}$
while $Q \neq \{\}$
    $j \Leftarrow$ pop$(Q)$
    $Q \Leftarrow Q \setminus \{j\}$
    for $k$ such that $\mathbb{G}(j,k) \neq 0$
        if $k \notin A$ and $\overline{\mathbb{D}}(i,:) = \pm\overline{\mathbb{D}}(k,:)$
            $A \Leftarrow A \cup \{k\}$
            $Q \Leftarrow Q \cup \{k\}$
        end
    end
end
return $A$
```

2.4. Example

In this section we describe the steps of our algorithm applied to the three element simplicial mesh depicted in Figure 1. Matrices $\mathbb{D}_{-1}, \mathbb{D}_0, \mathbb{D}_1$, and $\mathbb{D}_2$, shown in Section 1.1, are first computed and then passed to `coarsen_complex`. The externally-defined procedure `aggregate_nodes` is then called to produce the piecewise-constant nodal aggregation operator

$$P_0 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

whose corresponding aggregates are shown in Figure 6. At this stage of the procedure, a more general nodal problem $\mathbb{D}_0^T \mathbb{M}_1 \mathbb{D}_0$ may be utilized in determining the coarse aggregates. Next, `induced_aggregates` is invoked with arguments $P_0, \mathbb{D}_0, \mathbb{D}_1$ and the sparse matrix

$$\overline{\mathbb{D}} = \mathbb{D}_0 P_0 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \tag{11}$$

is constructed. Recall from Section 2.2 that the rows of $\overline{\mathbb{D}}$ are used to determine the induced edge aggregates. The zero rows of $\overline{\mathbb{D}}$, namely rows 0, 1, and 3, correspond to interior edges, which is confirmed by Figure 6. Linear dependence between rows 2 and 4 indicates that edges 2 and 4 have common coarse endpoints, with the difference in sign indicating opposite orientations.

For each non-zero and un-aggregated row of $\overline{\mathbb{D}}$, `dependent_rows` traverses

$$\mathbb{G} = \mathbb{D}_1^T \mathbb{D}_1 = \begin{bmatrix} 1 & -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 \\ 1 & -1 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 2 & -1 & 1 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 1 \end{bmatrix}, \tag{12}$$

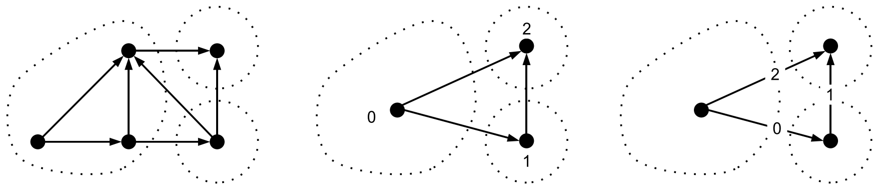

Figure 6: Original mesh with nodal aggregates (left), coarse nodes (center), and coarse edges (right).

to find dependent rows among upper-adjacent edges. In this case, edges 3 and 4 are upper-adjacent to 2, however only row 4 in $\overline{\mathbb{D}}$ is linearly dependent to row 2 in $\overline{\mathbb{D}}$. Rows 5 and 6 of $\overline{\mathbb{D}}$ are not linearly dependent to any other rows, thus forming single aggregates for edges 5 and 6. The resulting aggregation operator

$$P_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{13}$$

is then used to produce the coarse discrete derivative operator

$$\widehat{\mathbb{D}}_0 = (P_1^T P_1)^{-1} P_1^T \mathbb{D}_0 P_0 = \begin{bmatrix} -1 & 1 & 0 \\ 0 & -1 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \tag{14}$$

for the mesh in Figure 6. Subsequent iterations of the algorithm produce operators

$$P_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \qquad \widehat{\mathbb{D}}_1 = (P_2^T P_2)^{-1} P_2^T \mathbb{D}_1 P_1 = \begin{bmatrix} 1 & 1 & -1 \end{bmatrix}, \qquad \widehat{\mathbb{D}}_2 = \mathbb{D}_2 P_2 = \begin{bmatrix} 0 \end{bmatrix}, \tag{15}$$

which complete the coarse de Rham complex.

2.5. Commutativity

We now prove tentative prolongators $P_0, P_1, \ldots, P_K$ and coarse derivative operators $\widehat{\mathbb{D}}_0, \widehat{\mathbb{D}}_1, \ldots, \widehat{\mathbb{D}}_K$ produced by Algorithm 1 satisfy commutative diagram (8). The result is summarized by the following theorem.

Theorem 1. *Let $P_k : \widehat{\Omega}_d^k \to \Omega_d^k$ denote the discrete k-form prolongation operators with the following properties*

$$P_{k+1} \quad \text{has full column rank} \tag{16i}$$

$$\mathcal{R}(\mathbb{D}_k P_k) \subset \mathcal{R}(P_{k+1}) \tag{16ii}$$

$$\widehat{\mathbb{D}}_k \Leftarrow (P_{k+1}^T P_{k+1})^{-1} P_{k+1}^T \mathbb{D}_k P_k \tag{16iii}$$

Then, diagram (8) holds. That is,

$$\mathbb{D}_k P_k = P_{k+1}\widehat{\mathbb{D}}_k \tag{17}$$

Proof. Since P_{k+1} has full column rank, the pseudoinverse is given by

$$P_{k+1}^{+} = (P_{k+1}^T P_{k+1})^{-1} P_{k+1}^T. \tag{18}$$

Recall that for an arbitrary matrix A the pseudoinverse satisfies $AA^{+}A = A$. Furthermore, $\mathcal{R}(\mathbb{D}_k P_k) \subset \mathcal{R}(P_{k+1})$ implies that $\mathbb{D}_k P_k = P_{k+1}\mathbb{X}$ for some matrix $\mathbb{X}$. Combining these observations,

$$\begin{aligned} P_{k+1}\widehat{\mathbb{D}}_k &= P_{k+1}P_{k+1}^{+}\mathbb{D}_k P_k, \\ &= P_{k+1}P_{k+1}^{+}P_{k+1}\mathbb{X}, \\ &= P_{k+1}\mathbb{X}, \\ &= \mathbb{D}_k P_k \end{aligned}$$

□

Since Algorithm 1 meets assumptions (16i),(16ii), and (16iii) it follows that diagram (8) is satisfied. Also, assuming disjoint aggregates, the matrix $(P_{k+1}^T P_{k+1})$ appearing in (18) is a diagonal matrix, so its inverse is easily computed.

2.6. Exact Sequences

The de Rham complex formed by the fine-level discrete derivative operators,

$$\begin{tikzcd} 0 \arrow[r, "\mathbb{D}_{-1}"] & \Omega_d^0 \arrow[r, "\mathbb{D}_0"] & \Omega_d^1 \arrow[r, "\mathbb{D}_1"] & \dots \arrow[r, "\mathbb{D}_{N-1}"] & \Omega_d^N \arrow[r, "\mathbb{D}_N"] & 0 \end{tikzcd} \tag{19}$$

is an exact sequence, i.e. $\mathrm{img}(\mathbb{D}_k) \subset \ker(\mathbb{D}_{k+1})$ or equivalently $\mathbb{D}_{k+1}\mathbb{D}_k = 0$. A natural question to ask is whether the coarse complex retains this property. As argued in Section 2.5, $\mathbb{D}_k P_k = P_{k+1}\mathbb{X}$ for some matrix $\mathbb{X}$, therefore it follows

$$\begin{aligned} \widehat{\mathbb{D}}_{k+1}\widehat{\mathbb{D}}_k &= P_{k+2}^{+}\mathbb{D}_{k+1}P_{k+1}P_{k+1}^{+}\mathbb{D}_k P_k, \\ &= P_{k+2}^{+}\mathbb{D}_{k+1}P_{k+1}P_{k+1}^{+}P_{k+1}\mathbb{X}, \\ &= P_{k+2}^{+}\mathbb{D}_{k+1}P_{k+1}\mathbb{X}, \\ &= P_{k+2}^{+}\mathbb{D}_{k+1}\mathbb{D}_k P_k, \\ &= 0, \end{aligned}$$

since $\mathbb{D}_{k+1}\mathbb{D}_k = 0$ by assumption. From diagram (3) we infer the same result for the associated finite element spaces.

2.7. Smoothed Prolongators

While the tentative prolongators $P_0, P_1, \dots$ produced by `coarsen_complex` commute with $\mathbb{D}_k$ and give rise to an coarse exact sequence, their piecewise-constant nature leads to suboptimal multigrid scaling [19, 15]. In Smoothed Aggregation [21], the tentative prolongator P is smoothed to produce another prolongator $\mathcal{P} = \mathcal{S}P$ with superior approximation characteristics. We consider prolongation smoothers of the form $\mathcal{S} = (I - \mathbb{S}A)$. Possible implementations include Richardson $\mathbb{S} = \omega I$, Jacobi $\mathbb{S} = \omega\,\mathrm{diag}(A)^{-1}$, and polynomial $\mathbb{S} = p(A)$ [1].

Smoothed prolongation operators are desirable, but straightforward application of smoothers to each of $P_0, P_1, \ldots$ violates commutativity. The solution proposed in [15] smooths P_0 and P_1 with *compatible* smoothers $\mathcal{S}_0, \mathcal{S}_1$ such that commutativity of the smoothed prolongators $\mathcal{P}_0, \mathcal{P}_1$ is maintained, i.e. $\mathbb{D}_0\mathcal{P}_0 = \mathcal{P}_1\widehat{\mathbb{D}}_0$. In the following theorem we generalize this result to arbitrary k.

Theorem 2. *Given discrete k-form prolongation operators P_k satisfying (16i),(16ii), and (16iii), let $\mathcal{P}_k : \widehat{\Omega}_d^k \to \Omega_d^k$ denote the smoothed discrete k-form prolongation operators with the following properties*

$$\mathcal{P}_k = \mathcal{S}_k P_k \tag{20i}$$

$$\mathcal{S}_0 = (I - \mathbb{S}_0\mathbb{D}_0^T\mathbb{M}_1\mathbb{D}_0) \tag{20ii}$$

$$\mathcal{S}_k = (I - \mathbb{S}_k\mathbb{D}_k^T\mathbb{M}_{k+1}\mathbb{D}_k - \mathbb{D}_{k-1}\mathbb{S}_{k-1}\mathbb{D}_{k-1}^T\mathbb{M}_k) \quad \text{for k} > 0 \tag{20iii}$$

where $\mathbb{S}_k$ defines the type of prolongation smoother. Then, diagram (8) holds. That is,

$$\mathbb{D}_k\mathcal{P}_k = \mathcal{P}_{k+1}\widehat{\mathbb{D}}_k \tag{21}$$

Proof. First, if

$$\mathbb{D}_k\mathcal{S}_k = \mathcal{S}_{k+1}\mathbb{D}_k \tag{22}$$

then

$$\begin{aligned}
\mathcal{P}_{k+1}\widehat{\mathbb{D}}_k &= \mathcal{S}_{k+1}P_{k+1}\widehat{\mathbb{D}}_k \\
&= \mathcal{S}_{k+1}P_{k+1}(P_{k+1}^TP_{k+1})^{-1}P_{k+1}^T\mathbb{D}_kP_k \\
&= \mathcal{S}_{k+1}\mathbb{D}_kP_k \\
&= \mathbb{D}_k\mathcal{S}_kP_k \\
&= \mathbb{D}_k\mathcal{P}_k
\end{aligned}$$

Therefore, it suffices to show that (22) holds for all k. For $k = 0$ we have

$$\begin{aligned}
\mathcal{S}_1\mathbb{D}_0 &= (I - \mathbb{S}_1\mathbb{D}_1^T\mathbb{M}_2\mathbb{D}_1 - \mathbb{D}_0\mathbb{S}_0\mathbb{D}_0^T\mathbb{M}_1)\mathbb{D}_0 \\
&= (\mathbb{D}_0 - \mathbb{S}_1\mathbb{D}_1^T\mathbb{M}_2\mathbb{D}_1\mathbb{D}_0 - \mathbb{D}_0\mathbb{S}_0\mathbb{D}_0^T\mathbb{M}_1\mathbb{D}_0) \\
&= (\mathbb{D}_0 - \mathbb{D}_0\mathbb{S}_0\mathbb{D}_0^T\mathbb{M}_1\mathbb{D}_0) \\
&= \mathbb{D}_0(I - \mathbb{S}_0\mathbb{D}_0^T\mathbb{M}_1\mathbb{D}_0) \\
&= \mathbb{D}_0\mathcal{S}_0
\end{aligned}$$

and for all $k > 1$ we have

$$\begin{aligned}
\mathcal{S}_{k+1}\mathbb{D}_k &= (I - \mathbb{S}_{k+1}\mathbb{D}_{k+1}^T\mathbb{M}_{k+2}\mathbb{D}_{k+1} - \mathbb{D}_k\mathbb{S}_k\mathbb{D}_k^T\mathbb{M}_{k+1})\mathbb{D}_k \\
&= (\mathbb{D}_k - \mathbb{S}_{k+1}\mathbb{D}_{k+1}^T\mathbb{M}_{k+2}\mathbb{D}_{k+1}\mathbb{D}_k - \mathbb{D}_k\mathbb{S}_k\mathbb{D}_k^T\mathbb{M}_{k+1}\mathbb{D}_k) \\
&= (\mathbb{D}_k - \mathbb{D}_k\mathbb{S}_k\mathbb{D}_k^T\mathbb{M}_{k+1}\mathbb{D}_k) \\
&= (\mathbb{D}_k - \mathbb{D}_k\mathbb{S}_k\mathbb{D}_k^T\mathbb{M}_{k+1}\mathbb{D}_k - \mathbb{D}_k\mathbb{D}_{k-1}\mathbb{S}_{k-1}\mathbb{D}_{k-1}^T\mathbb{M}_k) \\
&= \mathbb{D}_k(I - \mathbb{S}_k\mathbb{D}_k^T\mathbb{M}_{k+1}\mathbb{D}_k - \mathbb{D}_{k-1}\mathbb{S}_{k-1}\mathbb{D}_{k-1}^T\mathbb{M}_k) \\
&= \mathbb{D}_k\mathcal{S}_k
\end{aligned}$$

which completes the proof of (21). □

On subsequent levels, the coarse innerproducts $\widehat{\mathbb{M}}_k = \mathcal{P}_k^T \mathbb{M}_k \mathcal{P}_k$ and derivatives $\widehat{\mathbb{D}}_k$ replace $\mathbb{M}_k$ and $\mathbb{D}_k$ in the definition of $\mathcal{S}_k$. As shown below, the Galerkin product $\widehat{A}_k = \mathcal{P}_k^T A_k \mathcal{P}_k$ can also be written in terms of the coarse operators.

$$\begin{aligned} \widehat{A}_k &= \mathcal{P}_k^T A_k \mathcal{P}_k \\ &= \mathcal{P}_k^T \mathbb{D}_k^T \mathbb{M}_{k+1} \mathbb{D}_k \mathcal{P}_k \\ &= \widehat{\mathbb{D}}_k^T \mathcal{P}_{k+1}^T \mathbb{M}_{k+1} \mathcal{P}_{k+1} \widehat{\mathbb{D}}_k \\ &= \widehat{\mathbb{D}}_k^T \widehat{\mathbb{M}}_{k+1} \widehat{\mathbb{D}}_k \end{aligned}$$

2.8. Extensions and Applications

Note that condition (9) permits some freedom in our choice of aggregates. For instance, in restricting ourselves to contiguous aggregates we have slightly enriched the range of P_{k+1} beyond what is necessary. Provided that P_{k+1} already satisfies (9), additional coarse basis functions, can be introduced to better approximate low-energy modes. As in Smoothed Aggregation, these additional columns of P_{k+1} can be chosen to exactly interpolate given near-nullspace vectors [21].

So far we have only discussed coarsening the cochain complex (8). It is worth noting that `coarsen_complex` works equally well on the chain complex formed by the mesh boundary operators $\partial_k = \mathbb{D}_{k-1}^T$,

$$\begin{tikzcd} 0 & \Omega_d^0 \arrow[l, "\mathbb{D}_{-1}^T"'] & \cdots \arrow[l, "\mathbb{D}_0^T"'] & \Omega_d^{N-1} \arrow[l, "\mathbb{D}_{N-2}^T"'] & \Omega_d^N \arrow[l, "\mathbb{D}_{N-1}^T"'] & 0, \arrow[l, "\mathbb{D}_N^T"'] \end{tikzcd} \tag{23}$$

by simply reversing the order of the complex, i.e. $(\mathbb{D}_{-1}, \mathbb{D}_{,0}, \ldots, \mathbb{D}_N) \Rightarrow (\mathbb{D}_N^T, \mathbb{D}_{N-1}^T, \ldots, \mathbb{D}_{-1})$. In this case `aggregate_nodes` will aggregate the top-level elements, for instance the triangles in Figure 1. Intuitively, ∂_k acts like a derivative operator that maps k-cochains to $(k+1)$-cochains, however one typically refers to these as k-chains rather than cochains [14]. In Section 3 we coarsen both complexes when computing Hodge decompositions.

3. HODGE DECOMPOSITION

The *Hodge Decomposition* [11] states that the space of k-forms on a closed manifold can be decomposed into three orthogonal subspaces,

$$\Omega^k = \boldsymbol{d}_{k-1}\Omega^{k-1} \oplus \boldsymbol{\delta}_{k+1}\Omega^{k+1} \oplus \mathcal{H}^k, \tag{24}$$

where $\mathcal{H}^k$ is the space of *harmonic* k-forms, $\mathcal{H}^k = \{h \in \Omega^k | \boldsymbol{\Delta}^k h = 0\}$. The analogous result holds for the space of discrete k-forms Ω_d^k, where the *derived* codifferential [4]

$$\boldsymbol{\delta}_k = \mathbb{M}_{k-1}^{-1} \mathbb{D}_{k-1}^T \mathbb{M}_k, \tag{25}$$

is defined to be the adjoint of $\mathbb{D}_{k-1}$ in the discrete innerproduct $\mathbb{M}_k$. Convergence of the discrete approximations to the Hodge decomposition is examined in [9].

In practice, for a discrete k-form ω^k we seek a decomposition

$$\omega^k = \mathbb{D}_{k-1}\alpha^{k-1} + \mathbb{M}_k^{-1} \mathbb{D}_k^T \mathbb{M}_{k+1} \beta^{k+1} + h^k, \tag{26}$$

for some $\alpha^{k-1} \in \Omega_d^{k-1}$, $\beta^{k+1} \in \Omega_d^{k+1}$, and $h^k \in \Omega_d^k$ where $\mathbf{\Delta}^k h^k = 0$. Note that α^{k-1} and β^{k+1} are generally not unique, since the kernels of $\mathbb{D}_{k-1}$ and $\mathbb{M}_k^{-1}\mathbb{D}_k^T\mathbb{M}_{k+1}$ are non-empty. However, the discrete k-forms $(\mathbb{D}_{k-1}\alpha^{k-1})$ and $(\mathbb{M}_k^{-1}\mathbb{D}_k^T\mathbb{M}_{k+1}\beta^{k+1})$ *are* uniquely determined. We decompose ω^k into (26) by solving

$$\begin{aligned} \left(\mathbb{D}_{k-1}^T\mathbb{M}_k\mathbb{D}_{k-1}\right)\alpha^{k-1} &= \mathbb{D}_{k-1}^T\mathbb{M}_k\omega^k, & (27)\\ \left(\mathbb{D}_k\mathbb{M}_k^{-1}\mathbb{D}_k^T\right)\mathbb{M}_{k+1}\beta^{k+1} &= \mathbb{D}_k\omega^k, & (28)\\ h^k &= \omega^k - \mathbb{D}_{k-1}\alpha^{k-1} - \mathbb{M}_k^{-1}\mathbb{D}_k^T\mathbb{M}_{k+1}\beta^{k+1}. & (29) \end{aligned}$$

Note that (28) involves the explicit inverse $\mathbb{M}_k^{-1}$ which is typically dense[†]. In the following sections, we first consider the special case $\mathbb{M}_k = I$ and then show how (28) can be circumvented in the general case. Equation (27) is obtained by left multiplying $\mathbb{M}_{k-1}\mathbb{D}_{k-1}^T\mathbb{M}_k$ on both sides of (26). Likewise, applying $\mathbb{D}_k$ to both sides of (26) yields (28). Equivalently, one may seek minima of the following functionals.

$$\left|\left|\mathbb{D}_{k-1}\alpha^{k-1} - \omega^k\right|\right|_{\mathbb{M}_k} \qquad \left|\left|\mathbb{M}_k^{-1}\mathbb{D}_k^T\mathbb{M}_{k+1}\beta^{k+1} - \omega^k\right|\right|_{\mathbb{M}_k} \tag{30}$$

3.1. *Special Case*

Taking the appropriate identity matrix for all discrete innerproducts $\mathbb{M}_k$ in (27) - (29) yields

$$\begin{aligned} \mathbb{D}_{k-1}^T\mathbb{D}_{k-1}\alpha^{k-1} &= \mathbb{D}_{k-1}^T\omega^k, & (31)\\ \mathbb{D}_k\mathbb{D}_k^T\beta^{k+1} &= \mathbb{D}_k\omega^k, & (32)\\ h^k &= \omega^k - \mathbb{D}_{k-1}\alpha^{k-1} - \mathbb{D}_k^T\beta^{k+1}. & (33) \end{aligned}$$

Although (31) - (33) are devoid of metric information, some fundamental topological properties of the mesh are retained. For instance, the *number* of harmonic k-forms, which together form a *cohomology basis*, is independent of the choice of innerproduct[‡]. In applications where metric information is either irrelevant or simply unavailable[8] these "nonphysical" equations are sufficient.

Algorithm 4: `construct_solver`$(k, \mathbb{M}_k, \mathbb{D}_{-1}, \mathbb{D}_0, \ldots, \mathbb{D}_N)$

```
$A_0 \Leftarrow \mathbb{D}_{k-1}^T\mathbb{M}_k\mathbb{D}_{k-1}$
$\mathbb{D}_{-1}^0, \ldots, \mathbb{D}_N^0 \Leftarrow \mathbb{D}_{-1}, \ldots, \mathbb{D}_N$
for $l = 0$ to NUM_LEVELS $- 1$
    $P_0^l, \ldots, P_N^l, \mathbb{D}_{-1}^{l+1}, \ldots, \mathbb{D}_N^{l+1} \Leftarrow$ coarsen_complex($\mathbb{D}_{-1}^l, \ldots, \mathbb{D}_N^l$)
end
for $l = 0$ to NUM_LEVELS $- 1$
    $\mathcal{P}_l \Leftarrow$ smooth_prolongator($A_l, P_{k-1}^l$)
    $A_{l+1} \Leftarrow \mathcal{P}_l^T A_l \mathcal{P}_l$
end
return MG_solver($A_0, A_1, \ldots, A_{\texttt{NUM\_LEVELS}}, \mathcal{P}_0, \mathcal{P}_1, \ldots, \mathcal{P}_{\texttt{NUM\_LEVELS}-1}$)
```

[†] The covolume Hodge star is a notable exception.

[‡] In the case $\mathbb{M} = I$, the cohomology basis is actually a *homology basis* also.

Algorithm 5: `decompose_special`$(\omega^k, \mathbb{D}_{-1}, \mathbb{D}_0, \ldots, \mathbb{D}_N)$

```
solver1 ⇐ construct_solver($k, I, \mathbb{D}_{-1}, \mathbb{D}_{,0}, \ldots, \mathbb{D}_N$)
solver2 ⇐ construct_solver($N - k - 1, I, \mathbb{D}_N^T, \mathbb{D}_{N-1}^T, \ldots, \mathbb{D}_{-1}^T$)

$\alpha^{k-1} \Leftarrow$ solver1($\mathbb{D}_{k-1}^T \omega^k$)
$\beta^{k+1} \Leftarrow$ solver2($\mathbb{D}_k \omega^k$)
$h \Leftarrow \omega^k - \mathbb{D}_{k-1}\alpha^{k-1} - \mathbb{D}_k^T \beta^{k+1}$

return $\alpha^{k-1}, \beta^{k+1}, h^k$
```

Algorithm 5 demonstrates how the proposed method is used to compute Hodge decompositions in the special case. Multigrid solvers `solver1` and `solver2` are constructed for the solution of linear systems (31) and (32) respectively. In the latter case, the direction of the chain complex is reversed when being passed as an argument to `construct_solver`. As mentioned in Section 2.8, `coarsen_complex` coarsens the reversed complex with this simple change of arguments.

Using the identity innerproduct, `construct_solver` applies the proposed method recursively to produce a progressively-coarser hierarchy of tentative prolongators P_k^l and discrete derivatives $\mathbb{D}_k^l$. The tentative prolongators are then smoothed by a user-defined function `smoothprolongator` to produce the final prolongators $\mathcal{P}_l$ and Galerkin products $A_{l+1} \Leftarrow \mathcal{P}_l^T A_l \mathcal{P}_l$. Finally, the matrices $A_0, \ldots, A_{\texttt{NUM\_LEVELS}}$ and $\mathcal{P}_0, \ldots, \mathcal{P}_{\texttt{NUM\_LEVELS}-1}$ determine the multigrid cycle in a user-defined class `MGsolver`. Choices for `smoothprolongator` and `MGsolver` are discussed in Section 4.

3.2. General Case

The multilevel solver outlined in Section 3.1 can be directly applied to linear system (27) by passing the innerproduct $\mathbb{M}_k$, instead of the identity, in the arguments to `construct_solver`. However, a different strategy is needed to solve (28) since $\mathbb{M}_k^{-1}$ is generally dense and cannot be formed explicitly. In the following we outline a method for computing Hodge Decompositions in the general case.

We first remark that if a basis for the space of Harmonic k-forms, $\mathcal{H}^k = \text{span}\{h_0^k, h_1^k, \ldots h_H^k\}$, is known, then the harmonic component of the Hodge Decomposition is easily computed by projecting ω^k onto the basis elements. Furthermore, since α^{k-1} in (27) can also be obtained, we can compute the value of the remaining component $(\omega^k - \mathbb{D}_{k-1}\alpha^{k-1} - h^k)$ which must lie in the range of $\mathbb{M}_k^{-1}\mathbb{D}_k^T\mathbb{M}_{k+1}$ due to orthogonality of the three spaces.

Therefore, the task of computing general Hodge Decompositions can be reduced to computing a basis for $\mathcal{H}^k$. Sometimes a basis is known a priori. For instance, $\mathcal{H}^0$ which corresponds to the nullspace of the pure-Neumann problem, is spanned by constant vectors on each connected component of the domain. Furthermore, if the domain is *contractible* then $\mathcal{H}^k = \{\}$ for $k > 0$. However, in many cases of interest we cannot assume that a basis for $\mathcal{H}^k$ is known, and therefore it must be computed.

Note that `decompose_special` can be used to determine a Harmonic k-form basis for the identity innerproduct by decomposing randomly generated k-forms until their respective harmonic components become linearly dependent. We denote this basis $\{\overline{h_0^k}, \overline{h_1^k}, \ldots \overline{h_m^k}\}$ and their span $\overline{\mathcal{H}^k}$. Using these k-forms, a basis for the harmonic k-forms with innerproduct $\mathbb{M}_k$

can be produced by solving

$$\begin{aligned}\mathbb{D}_{k-1}^T\mathbb{M}_k\mathbb{D}_{k-1}\alpha_i^{k-1} &= \mathbb{D}_{k-1}^T\mathbb{M}_k\overline{h_i^k}, &(34)\\ h_i^k &= \overline{h_i^k} - \mathbb{D}_{k-1}\alpha_i^{k-1}. &(35)\end{aligned}$$

It is readily verified that $h_0^k, \dots, h_m^k$ are harmonic,

$$\begin{aligned}\mathbb{D}_k h_i^k &= \mathbb{D}_k\overline{h_i^k} - \mathbb{D}_k\mathbb{D}_{k-1}\alpha_i^{k-1} = 0, &(36)\\ \mathbb{M}_{k-1}^{-1}\mathbb{D}_{k-1}^T\mathbb{M}_k h_i^k &= \mathbb{M}_{k-1}^{-1}(\mathbb{D}_{k-1}^T\mathbb{M}_k\overline{h_i^k} - \mathbb{D}_{k-1}^T\mathbb{M}_k h_i^k\mathbb{D}_{k-1}\alpha_i^{k-1}) = 0, &(37)\end{aligned}$$

since $\mathbb{D}_k\mathbb{D}_{k-1} = 0$ and $\mathbb{D}_k\overline{h_i^k} = 0$ by assumption. It remains to be shown that $h_0^k, \dots, h_m^k$ are linearly independent. Supposing $h_0^k, \dots, h_m^k$ to be linearly dependent, there exist scalars $c_0, \dots, c_H$ not all zero such that

$$\begin{aligned}0 &= \sum_{i=0}^{m} c_i h_i^k,\\ &= \sum_{i=0}^{m} c_i(\overline{h_i^k} - \mathbb{D}_{k-1}\alpha_i^{k-1}),\\ &= \sum_{i=0}^{m} c_i\overline{h_i^k} - \sum_{i=0}^{m} c_i\mathbb{D}_{k-1}\alpha_i^{k-1},\end{aligned}$$

which is a contradiction, since $\left(\sum_{i=0}^{N-1} c_i\overline{h_i^k}\right) \in \overline{\mathcal{H}^k}$ is nonzero and $\overline{\mathcal{H}^k} \perp \mathcal{R}(\mathbb{D}_{k-1})$. Note that the harmonic forms $h_0^k, \dots, h_m^k$ are not generally the same as the harmonic components of the random k-forms used to produce $\overline{h_0^k}, \dots \overline{h_m^k}$.

4. NUMERICAL RESULTS

We have applied the proposed method to a number of structured and unstructured problems. In all cases, a multigrid V(1,1)-cycle is used as a preconditioner to conjugate gradient iteration. Unless stated otherwise, a symmetric Gauss-Seidel sweep is used during pre- and post-smoothing stages. Iteration on the positive-semidefinite systems,

$$\mathbb{D}_k^T\mathbb{D}_k, \qquad \mathbb{D}_k\mathbb{D}_k^T, \qquad \mathbb{D}_k^T\mathbb{M}_{k+1}\mathbb{D}_k, \tag{38}$$

proceeds until the relative residual is reduced by 10^{-10}. The matrix $\mathbb{D}_0^T\mathbb{M}_1\mathbb{D}_0$ corresponds to a Poisson problem with pure-Neumann boundary conditions. Similarly, $\mathbb{D}_1^T\mathbb{M}_2\mathbb{D}_1$ is an eddy-current problem (6) with $\sigma = 0$. As explained in Section 3, matrices (38) arise in discrete Hodge decompositions.

The multigrid hierarchy extends until the number of unknowns falls below 500, at which point a pseudoinverse is used to perform the coarse level solve. The tentative prolongators are smoothed *twice* with a Jacobi smoother

$$\begin{aligned}\mathcal{S} &= I - \frac{4}{3\lambda_{max}}\operatorname{diag}(A)^{-1}A &(39)\\ \mathcal{P} &= \mathcal{S}\mathcal{S}P &(40)\end{aligned}$$

System	Grid	Unknowns	Convergence	Work/Digit	Complexity	Levels
$\mathbb{D}_0^T\mathbb{D}_0$	250^2	63,001	0.075	8.172	1.636	4
	500^2	251,001	0.100	9.321	1.661	4
	1000^2	1002,001	0.063	7.866	1.686	5
$\mathbb{D}_1^T\mathbb{D}_1$	250^2	125,500	0.096	8.370	1.506	4
	500^2	501,000	0.103	8.741	1.527	5
	1000^2	2,002,000	0.085	8.142	1.545	5
$\mathbb{D}_0\mathbb{D}_0^T$	250^2	125,500	0.124	9.529	1.530	4
	500^2	501,000	0.133	9.932	1.542	5
	1000^2	2,002,000	0.094	8.550	1.553	5
$\mathbb{D}_1\mathbb{D}_1^T$	250^2	62,500	0.063	7.664	1.641	4
	500^2	250,000	0.063	7.758	1.664	4
	1000^2	1,000,000	0.063	7.868	1.687	5
$\mathbb{D}_0^T\mathbb{M}_1\mathbb{D}_0$	250^2	63,001	0.043	5.894	1.415	4
	500^2	251,001	0.055	6.480	1.432	4
	1000^2	1,002,001	0.041	5.963	1.448	5
$\mathbb{D}_1^T\mathbb{M}_2\mathbb{D}_1$	250^2	125,500	0.095	8.362	1.506	4
	500^2	501,000	0.103	8.738	1.527	5
	1000^2	2,002,000	0.085	8.140	1.545	5

Table I: Two-dimensional scaling results.

where λ_{max} is an upper bound on the spectral radius of $\text{diag}(A)^{-1}A$. When zero or near zero values appear on the diagonal of the Galerkin product $\mathcal{P}^T A\mathcal{P}$, the corresponding rows and columns are zeroed and ignored during smoothing. We discuss this choice of prolongation smoother in Section 4.1.

Tables I and II show the result of applying the proposed method to regular quadrilateral and hexahedral meshes of increasing size. In both cases, the finite element spaces described in [5] are used to produce the innerproducts $\mathbb{M}_k$. The systems are solved with a random initial value for x. Since the matrices are singular, the solution x is an arbitrary null vector. Column labels are explained as follows:

- 'Grid' - dimensions of the quadrilateral/hexahedral grid
- 'Convergence'- geometric mean of residual convergence factors $\sqrt[N]{||r_N||/||r_0||}$
- 'Work/Digit' - averaged operation cost of 1/10 residual reduction in units of $nnz(A)$ [§]
- 'Complexity' - total memory cost of multigrid hierarchy relative to 'System'
- 'Levels' - number of levels in the multigrid hierarchy

For each k, the algorithm exhibits competitive convergence factors while maintaining low operator complexity. Together, the work per digit-of-accuracy remains bounded as the problem size increases.

In Table III, numerical results are presented for the unstructured tetrahedral mesh depicted

[§] including the cost of conjugate gradient iteration

System	Grid	Unknowns	Convergence	Work/Digit	Complexity	Levels
$\mathbb{D}_0^T\mathbb{D}_0$	25^3	17,576	0.120	7.976	1.268	3
	50^3	132,651	0.151	9.118	1.300	3
	100^3	1,030,301	0.105	7.960	1.358	4
$\mathbb{D}_1^T\mathbb{D}_1$	25^3	50,700	0.192	10.432	1.296	3
	50^3	390,150	0.216	11.587	1.342	4
	100^3	3,060,300	0.208	11.849	1.415	4
$\mathbb{D}_2^T\mathbb{D}_2$	25^3	48,750	0.188	9.342	1.156	3
	50^3	382,500	0.218	10.447	1.180	3
	100^3	3,030,000	0.267	12.350	1.217	4
$\mathbb{D}_0\mathbb{D}_0^T$	25^3	50,700	0.287	13.323	1.246	3
	50^3	390,150	0.391	17.594	1.235	4
	100^3	3,060,300	0.323	14.811	1.252	4
$\mathbb{D}_1\mathbb{D}_1^T$	25^3	48,750	0.187	10.928	1.389	3
	50^3	382,500	0.264	13.855	1.403	4
	100^3	3,030,000	0.194	11.630	1.455	4
$\mathbb{D}_2\mathbb{D}_2^T$	25^3	15,625	0.089	7.152	1.302	3
	50^3	125,000	0.102	7.649	1.318	3
	100^3	1,000,000	0.103	7.949	1.368	4
$\mathbb{D}_0^T\mathbb{M}_1\mathbb{D}_0$	25^3	17,576	0.037	4.804	1.178	3
	50^3	132,651	0.053	5.495	1.200	3
	100^3	1,030,301	0.038	5.054	1.241	4
$\mathbb{D}_1^T\mathbb{M}_2\mathbb{D}_1$	25^3	50,700	0.097	6.838	1.184	3
	50^3	390,150	0.113	7.461	1.214	4
	100^3	3,060,300	0.088	6.932	1.264	4
$\mathbb{D}_2^T\mathbb{M}_3\mathbb{D}_2$	25^3	48,750	0.188	9.334	1.156	3
	50^3	382,500	0.223	10.585	1.180	3
	100^3	3,030,000	0.265	12.294	1.217	4

Table II: Three-dimensional scaling results.

in Figure 7. As with classical algebraic multigrid methods, performance degrades in moving from a structured to an unstructured tessellation. However the decrease in performance for the scalar problems $\mathbb{D}_0^T\mathbb{D}_0$ and $\mathbb{D}_0^T\mathbb{M}_1\mathbb{D}_0$ is less significant than that of the other problems.

4.1. Prolongation Smoother

On the nonscalar problems considered, we found second degree prolongation smoothers (39) noticeably more efficient than first degree prolongation smoothers. While additional smoothing operations generally improve the convergence rate of Smoothed Aggregation methods, this improvement is typically offset by an increase in operator complexity and therefore the resultant work per digit of accuracy is not improved. However, there is an important difference between the tentative prolongators in the scalar and nonscalar problems. In the scalar case, all degrees of freedom are associated with a coarse aggregate and therefore the tentative

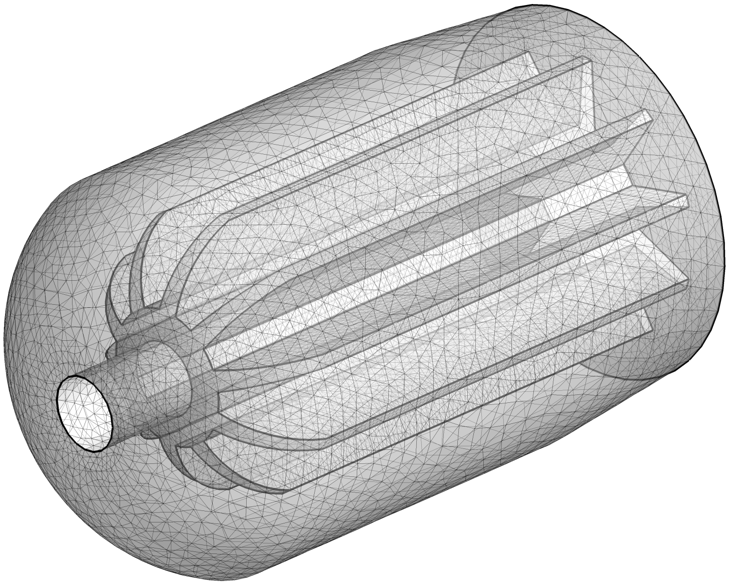

Figure 7: Titan IV rocket mesh.

System	Unknowns	Convergence	Work/Digit	Complexity	Levels
$\mathbb{D}_0^T\mathbb{D}_0$	84,280	0.073	6.601	1.304	3
$\mathbb{D}_1^T\mathbb{D}_1$	554,213	0.378	18.816	1.391	4
$\mathbb{D}_2^T\mathbb{D}_2$	920,168	0.366	15.856	1.186	4
$\mathbb{D}_0\mathbb{D}_0^T$	554,213	0.236	19.848	2.289	4
$\mathbb{D}_1\mathbb{D}_1^T$	920,168	0.390	17.068	1.197	4
$\mathbb{D}_2\mathbb{D}_2^T$	450,235	0.370	14.400	1.043	3
$\mathbb{D}_0^T\mathbb{M}_1\mathbb{D}_0$	84,280	0.144	8.949	1.304	3
$\mathbb{D}_1^T\mathbb{M}_2\mathbb{D}_1$	554,213	0.518	29.428	1.483	4
$\mathbb{D}_2^T\mathbb{M}_3\mathbb{D}_2$	920,168	0.348	15.111	1.187	4

Table III: Solver performance on the unstructured tetrahedral mesh in Figure 7.

prolongator has no zero rows. As described in Section 2.4, the tentative prolongator for nonscalar problems has zero rows for elements contained in the interior of a nodal aggregate. In the nonscalar case, additional smoothing operations incorporate a greater proportion of these degrees of freedom into the range of the final prolongator.

The influence of higher degree prolongation smoothers on solver performance is reported in Table IV. Column 'Degree' records the degree d of the prolongation smoother $\mathcal{P} = \mathcal{S}^d P$ while 'Percent Zero' reflects the percentage of zero rows in the first level prolongator. As expected the operator complexity increases with smoother degree. However, up to a point, this increase is less significant than the corresponding reduction in solver convergence. Second

System	Grid	Degree	Percent Zero	Convergence	Work/Digit	Complexity
$\mathbb{D}_1^T\mathbb{M}_2\mathbb{D}_1$	250^2	0	66.8	0.697	42.255	1.123
		1	66.8	0.357	14.774	1.123
		2	22.9	0.096	8.379	1.506
		3	0.4	0.063	9.515	2.084
		4	0.0	0.063	10.188	2.250
$\mathbb{D}_1^T\mathbb{M}_2\mathbb{D}_1$	50^3	0	67.6	0.567	25.043	1.034
		1	66.5	0.290	11.497	1.035
		2	8.8	0.096	7.460	1.214
		3	0.3	0.063	9.011	1.577
		4	0.0	0.063	9.074	1.632
$\mathbb{D}_2^T\mathbb{M}_3\mathbb{D}_2$	50^3	0	89.63	0.549	23.670	1.034
		1	89.63	0.382	14.753	1.034
		2	63.93	0.214	10.304	1.180
		3	23.77	0.122	9.203	1.481
		4	6.48	0.098	8.348	1.487
		5	2.07	0.089	10.267	1.953

Table IV: Comparison of prolongation smoothers.

degree smoothers exhibit the best efficiency in both instances of the problem $\mathbb{D}_1^T\mathbb{M}_2\mathbb{D}_1$ and remain competitive with higher degree smoothers in the last test. Since work per digit figures exclude the cost of constructing multigrid transfer operators, these higher degree smoothers may be less efficient in practice.

5. CONCLUSION

We have described an extension of Reitzinger and Schöberl's methodology [19] to higher dimensional k-forms with the addition of smoothed prolongation operators. Furthermore, we have detailed properties of the prolongation operator that arise from this generalized setting. Specifically we have identified necessary and sufficient conditions under which commutativity is maintained. The prolongation operators give rise to a hierarchy of exact finite element sequences. The generality of the method is appealing since the components are constructed independently of a particular mimetic discretization. Lastly, we have initiated a study of algebraic multigrid for the Hodge decomposition of general k-forms.

REFERENCES

1. M. Adams, M. Brezina, J. Hu, and R. Tuminaro. Parallel multigrid smoothing: polynomial versus Gauss-Seidel. *J. Comput. Phys.*, 188(2):593–610, 2003.
2. D. N. Arnold. Differential complexes and numerical stability. In *Proceedings of the International Congress of Mathematicians, Beijing 2002, Volume 1 : Plenary Lectures*, 2002.
3. D. N. Arnold, R. S. Falk, and R. Winther. Multigrid in $H(\text{div})$ and $H(\text{curl})$. *Numer. Math.*, 85(2):197–217, 2000.

4. P. B. Bochev and J. M. Hyman. Principles of mimetic discretizations of differential operators. In D. N. Arnold, P. B. Bochev, R. B. Lehoucq, R. A. Nicolaides, and M. Shashkov, editors, *Compatible Spatial Discretizations*, volume 142 of *The IMA Volumes in Mathematics and its Applications*, pages 89–119. Springer, Berlin, 2006.
5. P. B. Bochev and A. C. Robinson. Matching algorithms with physics: exact sequences of finite element spaces. In D. Estep and S. Tavener, editors, *Collected Lectures on Preservation of Stability Under Discretization*, chapter 8, pages 145–166. Society for Industrial and Applied Mathematics (SIAM), 2002.
6. A. Bossavit. On the numerical analysis of eddy-current problems. *Computer Methods in Applied Mechanics and Engineering*, 27(3):303–318, 1981.
7. A. Bossavit. Whitney forms : a class of finite elements for three-dimensional computations in electromagnetism. *IEE Proceedings*, 135, Part A(8):493–500, November 1988.
8. V. de Silva and R. Ghrist. Homological Sensor Networks. *Notices of the American Mathematical Society*, 54:10–17, 2007.
9. J. Dodziuk. Finite-difference approach to the Hodge theory of harmonic forms. *Amer. J. Math.*, 98(1):79–104, 1976.
10. M. Fisher, P. Schröder, M. Desbrun, and H. Hoppe. Design of tangent vector fields. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 56, New York, NY, USA, 2007. ACM.
11. T. Frankel. *The Geometry of Physics*. Cambridge University Press, Cambridge, second edition, 2004. An introduction.
12. V. Gradinaru and R. Hiptmair. Whitney Elements on Pyramids. *Electronic Transactions on Numerical Analysis*, 8:154–168, 1999.
13. R. Hiptmair. Multigrid method for maxwell's equations. *SIAM J. Numer. Anal.*, 36(1):204–225, 1999.
14. A. N. Hirani. *Discrete Exterior Calculus*. PhD thesis, California Institute of Technology, May 2003.
15. J. J. Hu, R. S. Tuminaro, P. B. Bochev, C. J. Garasi, and A. C. Robinson. Toward an h-independent algebraic multigrid method for Maxwell's equations. *SIAM Journal on Scientific Computing*, 27:1669–1688, 2006.
16. J. Jones and B. Lee. A multigrid method for variable coefficient maxwell's equations. *SIAM J. Sci. Comput.*, 27(5):1689–1708, 2006.
17. A. Muhammad and M. Egerstedt. Control Using Higher Order Laplacians in Network Topologies. *Proc. of 17th International Symposium on Mathematical Theory of Networks and Systems, Kyoto, Japan*, pages 1024–1038, 2006.
18. K. Polthier and E. Preuss. Identifying vector field singularities using a discrete hodge decomposition. In H. C. Hege and K. Polthier, editors, *Visualization and Mathematics, VisMath*. Springer–Verlag, 2002.
19. S. Reitzinger and J. Schöberl. An algebraic multigrid method for finite element discretizations with edge elements. *Numer. Linear Algebra Appl.*, 9:223–238, 2002.
20. Y. Tong, S. Lombeyda, A. N. Hirani, and M. Desbrun. Discrete multiscale vector field decomposition. *ACM Transactions on Graphics (Special issue of SIGGRAPH 2003 Proceedings)*, 22(3):445–452, July 2003.
21. P. Vaněk, J. Mandel, and M. Brezina. Algebraic Multigrid by Smoothed Aggregation for Second and Fourth Order Elliptic Problems. *Computing*, 56(3):179–196, 1996.
22. H. Whitney. *Geometric Integration Theory*. Princeton University Press, Princeton, N. J., 1957.
23. K. S. Yee. Numerical solution of initial boundary value problems involving Maxwells equations in isotropic media. *IEEE Transactions on Antennas and Propagation*, AP-14(3):302–307, May 1966.