Discrete Helmholtz–Hodge Decomposition on Polyhedral Meshes Using Compatible Discrete Operators

A. Lemoine, J.-P. Caltagirone, M. Azaïez, S. Vincent

Journal of Scientific Computing manuscript No.
(will be inserted by the editor)

Discrete Helmholtz-Hodge Decomposition on polyhedral meshes using Compatible Discrete Operators

A. Lemoine · J.-P. Caltagirone · M. Azaïez · S. Vincent

Received: date / Accepted: date

Abstract This article provides a methodology to perform discrete Helmholtz-Hodge decomposition on three-dimensional polyhedral meshes using structure-preserving schemes: the Compatible Discrete Operator schemes. We propose to extract the decomposition components independently with one equation to solve per component or potential. The key of the method is the choice of a discrete Hodge operator that makes a compromise between convergence rate and computational cost. Numerical experiments are performed to evaluate the convergence rate and the computational cost on various polyhedral meshes, in particular, on the FVCA benchmark meshes. We also investigate some linear solver capabilities to solve our equations. blueThe main contribution of this paper is the application of the CDO schemes to the Hodge decomposition and the required solvers.

A. Lemoine
Univ. Bordeaux, IPB, I2M, UMR 5295, F-33400 Talence, France.
Tel.: +3340006667
Fax: +3340006668
E-mail: antoine.lemoine@ipb.fr

J.-P. Caltagirone
Univ. Bordeaux, IPB, I2M, UMR 5295, F-33400 Talence, France.
Tel.: +3340006667
Fax: +3340006668
E-mail: calta@ipb.fr

M. Azaïez
Univ. Bordeaux, IPB, I2M, UMR 5295, F-33400 Talence, France.
Tel.: +3340006667
Fax: +3340006668
E-mail: azaiez@ipb.fr

S. Vincent
Univ. Bordeaux, IPB, I2M, UMR 5295, F-33400 Talence, France.
Tel.: +3340006667
Fax: +3340006688
E-mail: stephane.vincent@enscbp.fr

Keywords Helmholtz-Hodge · mimetic · polyhedral mesh · divergence-free · curl-free

1 Introduction

Computational physics often generates a lot of data that need to be processed to reveal features relevant to the phenomena studied. For instance, we can mention the detection of vortex centers in a turbulent flow, or the detection of the point of maximal curvature of a fingerprint. This can be achieved efficiently by the *Discrete Helmholtz-Hodge Decomposition* (DHHD). Helmholtz-Hodge decomposition consists in separating an initial vector field into an irrotational field, a solenoidal field and a harmonic field (1).

$$\begin{aligned} \boldsymbol{u} &= \boldsymbol{u}_\theta + \boldsymbol{u}_\psi + \boldsymbol{u_h} \\ &= \underbrace{\mathbf{grad}\,\theta}_{\text{compression}} + \underbrace{\mathbf{curl}\,\psi}_{\text{rotation}} + \underbrace{\boldsymbol{u_h}}_{\text{harmonic}} \end{aligned} \tag{1}$$

Moreover, irrotational and solenoidal parts are derived from potential fields. The extrema of the rotation potential $\boldsymbol{\psi}$ are the location of vortex centers and the extrema of the compression potential θ are the location of sources and sinks. These properties have been used in [1], [2] and [3] to perform the previously mentioned feature detections. The DHHD is not only a tool to analyze data. It can also be used in computational fluid dynamics for solving the Navier-Stokes equations, especially in the vector projection step [4]. The reader can refer to the survey of Bhatia *et al.* [5] for more applications. The versatility of this tool interests us to develop efficient methods to perform the decomposition.

Many authors such as Hyman & Shashkov [6] or Tong *et al.* [1] have emphasized the necessity to use structure preserving schemes (or mimetic, or compatible schemes) to ensure an exact discrete decomposition. That is, the irrotational term must be curl-free and the solenoidal term must be divergence-free to the unit roundoff. These schemes are designed to preserve some properties of the differential operators at the discrete level. For instance, these schemes guarantee that the identities $\mathbf{curl}\,\mathbf{grad} = 0$ and $\mathrm{div}\,\mathbf{curl} = 0$ are preserved at the discrete level. blueIn this article, we choose to use the *Compatible Discrete Operator* (CDO) schemes introduced by Bonelle & Ern [7] which belong to the family of structure-preserving schemes. These schemes come from the seminal ideas of Bossavit [8]. The reader can also refer to [9,10,11] for similar ideas.

While the DHHD has been widely explored by many authors, few have proposed a method to perform the DHHD on three-dimensional polyhedral meshes. For instance, Polthier & Preuß [3], Guo *et al.* [12] or Bluck & Walker [13] designed a method to perform the DHHD on triangular meshes. As shown in [14], the key to the problem is the choice of a suitable discrete Hodge operator, or equivalently, a discrete inner-product. Most of the discrete Hodge operators found in the literature are designed for tetrahedral or hexahedral meshes and those designed for polyhedral meshes require generally a minimization process

or an explicit matrix inversion (*e.g.* [15]). The computational cost of these operators is the limiting factor for applications on dense meshes. blueFollowing the idea of Bonelle & Ern [7], we chose the Hodge operator designed by Codecasa *et al.* [16] which can be computed explicitly without minimization or matrix inversion.

blueIn this paper, we explore many ways to perform the DHHD with a set of four equations, one per field of the decomposition ($\boldsymbol{u}_\theta$, $\boldsymbol{u}_\psi$, θ, or ψ). The novelty of this proposal consists in the exploration of various algorithms to extract these fields on polyhedral meshes with the CDO schemes and in the numerical study of the behaviors of these algorithms using various linear solvers.

The remainder of this paper is organized as follows. In section 2, we present the CDO framework. In section 3, we introduce the algorithms used to perform the DHHD. In section 4, numerical results on computational cost and convergence rate are presented. Finally, future works are envisaged and conclusions are drawn in section 5.

2 Compatible Discrete Operators

This section presents the CDO framework. The notations used in this article are widely based on [7]. The reader can refer to this article for more information about CDO schemes.

2.1 Discrete domain

The discrete domain is a polyhedral tessellation of a continuous domain Ω composed of the elements $M = \{V, E, F, C\}$. V is the set of vertices, E is the set of edges, F is the set of faces, and C is the set of cells. The numbers of elements of each kind are denoted $\{\#V, \#E, \#F, \#C\}$. We denote by the small characters v, e, f or c the elements of V, E, F or C. In the remainder, we will refer to this discrete domain by the words *cell-complex* or *primal mesh*.

In addition to this cell-complex, we consider a barycentric dual mesh $\tilde{M} = \{\tilde{V}, \tilde{E}, \tilde{F}, \tilde{C}\}$. Each element of the initial (or primal) cell-complex is associated with a unique element of the dual mesh. Namely, a primal vertex v is associated with a dual cell $\tilde{c}(v)$, a primal edge e is associated with a dual face $\tilde{f}(e)$, a primal face f is associated with a dual edge $\tilde{e}(f)$, and a primal cell c is associated with a dual vertex $\tilde{v}(c)$. Figure 1 presents a cell-complex composed of two polyhedral primal cells: a dodecahedron and a prism.

Note that we write "dual mesh" and not "dual cell-complex". To be a cell-complex, the dual mesh must have all its elements homeomorphic to a closed k-ball. By construction, the dual elements near the boundary of the domain are not closed. The closure of these elements provides a way to impose boundary conditions. This will be discussed in §2.3.

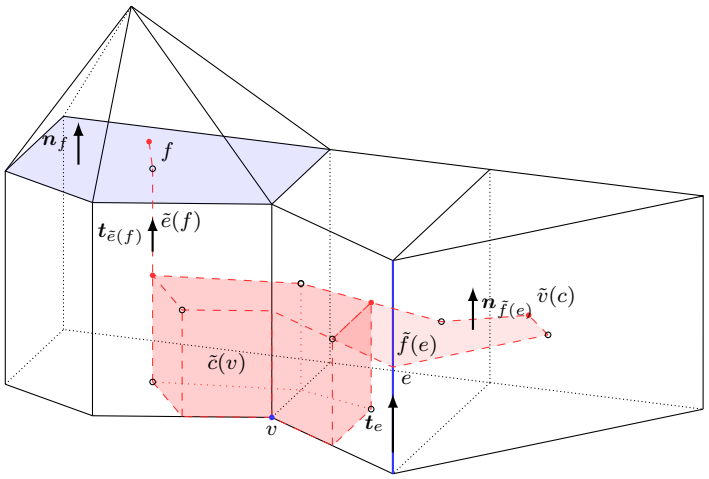


Fig. 1: blueExample of a cell-complex composed of prisms and a pyramid. Primal elements are emphasized in blue and their dual elements in red.

blueThe dual elements can be built using a collection of subsets of the primal mesh. Consider $\{\boldsymbol{x}_0, \cdots, \boldsymbol{x}_k\}$ a set of $k+1$ vertices with $k \in [\![0, 3]\!]$. We denote $\mathfrak{s}(\boldsymbol{x}_0, \cdots, \boldsymbol{x}_k)$ the interior of the simplex of dimension k generated by these vertices. Now consider $\boldsymbol{c}_a$ the barycenter of the primal element a. The dual elements are defined by the following formulas:

$$\forall c \in C \qquad \tilde{v}(c) \coloneqq \boldsymbol{c}_c \tag{2a}$$

$$\forall f \in F \qquad \tilde{e}(f) \coloneqq \bigcup_{c \in C_f} \mathfrak{s}(\boldsymbol{c}_c, \boldsymbol{c}_f) \tag{2b}$$

$$\forall e \in E \qquad \tilde{f}(e) \coloneqq \bigcup_{f \in F_e} \bigcup_{c \in C_f} \mathfrak{s}(\boldsymbol{c}_c, \boldsymbol{c}_f, \boldsymbol{c}_e) \tag{2c}$$

$$\forall v \in V \qquad \tilde{c}(v) \coloneqq \bigcup_{e \in E_v} \bigcup_{f \in F_e} \bigcup_{c \in C_f} \mathfrak{s}(\boldsymbol{c}_c, \boldsymbol{c}_f, \boldsymbol{c}_e, \boldsymbol{c}_v) \tag{2d}$$

blueWhere F_e denotes the set of faces that contain the edge e in their boundary.

2.2 Degrees of freedom

The degrees of freedom (DoF) are defined by the de Rham map $\mathsf{R}_{\mathcal{A}} \colon \mathcal{S} \to \mathcal{A}$ where $\mathcal{S}$ denotes a suitable space and $\mathcal{A}$ denotes the resultant DoF space or *cochain* space. We denote by $\{\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{C}\}$ the set of DoF associated with the elements $\{V, E, F, C\}$. Consider p as a scalar field and $\boldsymbol{u}$ as a vector field. The

DoF are defined by the following formulas:

$$\begin{aligned} &\forall v \in V \quad (\mathsf{R}_{\mathcal{V}}(p))_v \coloneqq p(v), && \forall c \in C \quad (\mathsf{R}_{\mathcal{C}}(p))_c \coloneqq \int_c p, \\ &\forall e \in E \quad (\mathsf{R}_{\mathcal{E}}(\boldsymbol{u}))_e \coloneqq \int_e \boldsymbol{u} \cdot \boldsymbol{t}_e, && \forall f \in F \quad (\mathsf{R}_{\mathcal{F}}(\boldsymbol{u}))_f \coloneqq \int_f \boldsymbol{u} \cdot \boldsymbol{n}_f \end{aligned} \tag{3}$$

Note that these formulas involve the tangent vectors $\boldsymbol{t}_e$ of edges and the normal vectors $\boldsymbol{n}_f$ of faces. That means that the geometric elements need to be (arbitrary) *oriented*. This orientation is represented in figure 1 by little arrows.

To understand what these DoF represent, we can use the metaphor of Tonti [17]: the mesh elements can be seen as a set of sensors that are sensitive to specific physical quantities. For instance, the DoF on the edges can be seen as sensors of circulations along lines and those of the faces can be seen as sensors of fluxes across surfaces. In general, scalar fields can be discretized (or measured) on points or cells and vector fields can be discretized on edges or faces. The de Rham maps (3) measure the physical quantities with the right sensor, that is, the right geometric element. This right geometric element is given by the nature of the physical quantity. The reader can refer to the article of Tonti [17] for a classification of the physical variables and their relation to the geometric elements.

We also emphasize that the DoF are not defined on some points of the geometric elements or approached by a polynomial approximation like in the finite volumes method or the finite elements method. To distinguish the nature of the DoF of the CDO schemes from those of the other methods, we choose to use the word *cochain*, borrowed from the vocabulary of algebraic topology, instead of DoF.

These definitions can be naturally extended to dual elements to build the dual cochain spaces $\{\tilde{\mathcal{V}}, \tilde{\mathcal{E}}, \tilde{\mathcal{F}}, \tilde{\mathcal{C}}\}$.

2.3 Discrete operators

The CDO schemes define discrete differential operators as applications between cochain spaces, such as:

$$\mathsf{GRAD}\colon \mathcal{V} \to \mathcal{E}, \qquad \mathsf{CURL}\colon \mathcal{E} \to \mathcal{F}, \qquad \mathsf{DIV}\colon \mathcal{F} \to \mathcal{C} \tag{4}$$

The Stokes formulas allow these operators to be expressed in terms of a matrix composed only of the elements $\{-1, 0, 1\}$. For instance, the Stokes formula applied to the divergence on a cell c for any vector field $\boldsymbol{\phi}$ gives

$$\int_c \operatorname{div} \boldsymbol{\phi} = \sum_{f \in F} \imath_{f,c} \int_f \boldsymbol{\phi} \cdot \boldsymbol{n}_f \tag{5}$$

where $\imath_{f,c}$ is the *incidence matrix* between faces and cells. It is given by

$$\imath_{f,c} = \begin{cases} 0 & \text{if } f \text{ does not belong to } \partial c \\ 1 & \text{if } \boldsymbol{n}_f \text{ points towards } \boldsymbol{n} \\ -1 & \text{otherwise} \end{cases} \tag{6}$$

where $\boldsymbol{n}$ denotes the outward normal of the cell c, $\boldsymbol{n}_f$ the arbitrary oriented normal of face f and ∂c the boundary of the cell c. Rewrite (5) using the de Rham map, we define the discrete divergence operator as

$$(\mathsf{DIV}\,\mathsf{R}_{\mathcal{F}}(\boldsymbol{\phi}))_c \coloneqq (\mathsf{R}_{\mathcal{C}}(\operatorname{div}\boldsymbol{\phi}))_c = \sum_{f\in F} \imath_{f,c}(\mathsf{R}_{\mathcal{F}}(\boldsymbol{\phi}))_f \tag{7}$$

Similarly, the three discrete differential operators of the equation (4) can be expressed with incidence matrices. Consider $\mathbf{p} \in \mathcal{V}$, $\mathbf{u} \in \mathcal{E}$ and $\boldsymbol{\psi} \in \mathcal{F}$, the discrete differentials operators are defined by:

$$\begin{gathered}\forall e \in E \quad (\mathsf{GRAD}\,\mathbf{p})_e = \sum_{v\in V} \imath_{v,e}\mathbf{p}_v, \qquad \forall f \in F \quad (\mathsf{CURL}\,\mathbf{u})_f = \sum_{e\in E} \imath_{e,f}\mathbf{u}_e, \\ \forall c \in C \quad (\mathsf{DIV}\,\boldsymbol{\psi})_c = \sum_{f\in F} \imath_{f,c}\boldsymbol{\psi}_f\end{gathered} \tag{8}$$

The reader can check that, by construction, we have the following identities:

$$\forall \mathbf{p} \in \mathcal{V} \quad \mathsf{CURL}\cdot\mathsf{GRAD}\,\mathbf{p} = \mathbf{0} \qquad \forall \mathbf{u} \in \mathcal{E} \quad \mathsf{DIV}\cdot\mathsf{CURL}\,\mathbf{u} = \mathbf{0} \tag{9}$$

One possibility to define discrete differential operators on the dual mesh is to introduce the discrete duality product. Consider two cochain spaces $\mathcal{A}$ and $\tilde{\mathcal{B}}$, the discrete duality product between $\mathbf{u} \in \mathcal{A}$ and $\boldsymbol{\phi} \in \tilde{\mathcal{B}}$ is given by

$$[\![\mathbf{u}, \boldsymbol{\phi}]\!]_{\mathcal{A}\tilde{\mathcal{B}}} = \sum_{a\in A} \mathbf{u}_a \boldsymbol{\phi}_{\tilde{b}(a)} \tag{10}$$

blueWhere $\mathbf{u}_a$ denotes the value of cochain $\mathbf{u}$ on the primal element $a \in A$ and $\boldsymbol{\phi}_{\tilde{b}(a)}$ denotes the value of cochain $\boldsymbol{\phi}$ on the dual element $\tilde{b}(a) \in \tilde{B}$.

Thus, the discrete differential operators are defined by adjunction

$$\begin{aligned}\forall(\mathbf{p},\boldsymbol{\phi}) \in \mathcal{V}\times\tilde{\mathcal{F}} \quad & \left[\!\!\left[\mathbf{p}, \widetilde{\mathsf{DIV}}\,\boldsymbol{\phi}\right]\!\!\right]_{\mathcal{V}\tilde{\mathcal{C}}} \coloneqq -\,[\![\mathsf{GRAD}\,\mathbf{p}, \boldsymbol{\phi}]\!]_{\mathcal{E}\tilde{\mathcal{F}}}\,, \\ \forall(\boldsymbol{\psi},\mathbf{u}) \in \mathcal{E}\times\tilde{\mathcal{E}} \quad & \left[\!\!\left[\boldsymbol{\psi}, \widetilde{\mathsf{CURL}}\,\mathbf{u}\right]\!\!\right]_{\mathcal{E}\tilde{\mathcal{F}}} \coloneqq [\![\mathsf{CURL}\,\boldsymbol{\psi}, \mathbf{u}]\!]_{\mathcal{F}\tilde{\mathcal{E}}}\,, \\ \forall(\boldsymbol{\phi},\boldsymbol{\rho}) \in \mathcal{F}\times\tilde{\mathcal{C}} \quad & \left[\!\!\left[\boldsymbol{\phi}, \widetilde{\mathsf{GRAD}}\,\boldsymbol{\rho}\right]\!\!\right]_{\mathcal{F}\tilde{\mathcal{E}}} \coloneqq -\,[\![\mathsf{DIV}\,\boldsymbol{\phi}, \boldsymbol{\rho}]\!]_{\mathcal{V}\tilde{\mathcal{C}}}\end{aligned} \tag{11}$$

To build second order operators such as $\operatorname{div}\mathbf{grad}$ or $\mathbf{curl}\,\mathbf{curl}$, we need operators that link primal cochains to dual cochains. This is the purpose of the discrete Hodge operator. In a three-dimensional cell-complex, we have four discrete Hodge operators:

$$\begin{aligned}\mathsf{H}^{\mathcal{V}\tilde{\mathcal{C}}} &: \mathcal{V} \to \tilde{\mathcal{C}} \qquad \mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}} : \mathcal{E} \to \tilde{\mathcal{F}} \\ \mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}} &: \mathcal{F} \to \tilde{\mathcal{E}} \qquad \mathsf{H}^{\mathcal{C}\tilde{\mathcal{V}}} : \mathcal{C} \to \tilde{\mathcal{V}}\end{aligned} \tag{12}$$

The design of these operators is the main issue of CDO schemes. Namely, these operators govern the convergence rate and the computation cost of the method. blueIn short, we require these operators to be symmetric definite positive. The discrete Hodge operators we use in our simulations are presented in section 4.

All the discrete operators can be summed using the discrete de Rham complex represented in figure 2.

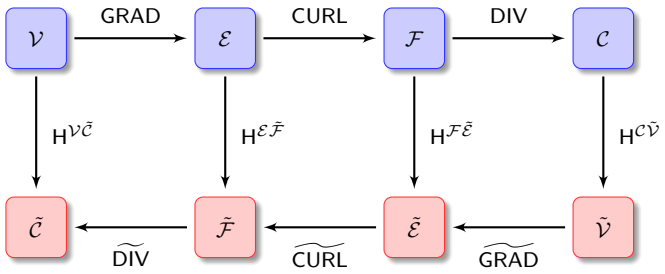
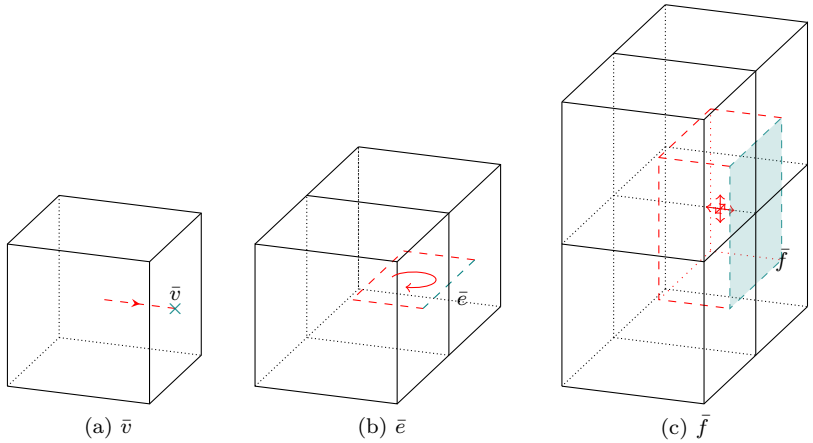


Fig. 2: Discrete de Rham complex

By construction, the dual mesh is not closed on the boundary. Thus, the dual operators require boundary conditions to be computed. For instance, consider a dual cell $\tilde{c}(v)$ near the boundary. The dual divergence operator $\widetilde{\mathsf{DIV}}$ can be closed by addition of an extra flux $\boldsymbol{\phi}_{\bar{f}}$ on the closing face $\bar{f}$ represented in figure 3c:

$$(\widetilde{\mathsf{DIV}}\,\boldsymbol{\phi})_{\tilde{c}} = \sum_{\tilde{f}\in\tilde{F}} \iota_{\tilde{f},\tilde{v}}\boldsymbol{\phi}_{\tilde{f}} + \boldsymbol{\phi}_{\bar{f}} \tag{13}$$

The same way, an extra vertex $\bar{v}$ and an extra edge $\bar{e}$ (figures 3a and 3b) are used to impose boundary conditions for the operators $\widetilde{\mathsf{GRAD}}$ and $\widetilde{\mathsf{CURL}}$.



(a) $\bar{v}$ (b) $\bar{e}$ (c) $\bar{f}$

Fig. 3: Closure of dual discrete operators

3 Discrete decomposition strategies

In this section, we give a description of the way the DHHD can be carried out. The starting point comes from the fact that the de Rham operator can discretize any vector field in two ways. The vector fields can be discretized on edges and considered as circulations, or they can be discretized on faces and considered as fluxes. These two discretizations lead to two families of DHHD: *edge-based* and *face-based* that we will present in the following sections. For the sake of clarity, we add, as an exponent, the space to which the cochains belong.

3.1 Edge-based DHHD

The edge-based decomposition considers the initial discrete vector field $\mathbf{u}^{\mathcal{E}}$ as a set of circulations. The degrees of freedom are located on the edges E. The following equation denotes the DHHD for circulations:

$$\begin{array}{rcccccc} \mathbf{u}^{\mathcal{E}} & = & \mathsf{GRAD}\,\boldsymbol{\theta}^{\mathcal{V}} & + & (\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\cdot\widetilde{\mathsf{CURL}}\cdot\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\boldsymbol{\psi}^{\mathcal{F}} & + & \mathbf{u_h}^{\mathcal{E}} \\ & = & \mathbf{u_{\boldsymbol{\theta}}}^{\mathcal{E}} & + & \mathbf{u_{\boldsymbol{\psi}}}^{\mathcal{E}} & + & \mathbf{u_h}^{\mathcal{E}} \end{array} \tag{14}$$

Note that we can also consider the vector potential as a cochain on the dual mesh, that is $\boldsymbol{\psi}^{\tilde{\mathcal{E}}} = \mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\boldsymbol{\psi}^{\mathcal{F}}$.

The two potentials and their derivatives can be extracted independently using the following equations:

$$(\mathsf{H}^{\mathcal{V}\tilde{\mathcal{C}}})^{\text{-}1}\,\widetilde{\mathsf{DIV}}\,\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}}\,\mathsf{GRAD}\,\boldsymbol{\theta}^{\mathcal{V}} = (\mathsf{H}^{\mathcal{V}\tilde{\mathcal{C}}})^{\text{-}1}\,\widetilde{\mathsf{DIV}}\,\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}}\,\mathbf{u}^{\mathcal{E}} \tag{15}$$

$$\mathsf{CURL}\,(\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\boldsymbol{\psi}^{\mathcal{F}} = \mathsf{CURL}\,\mathbf{u}^{\mathcal{E}} \tag{16}$$

$$\mathsf{GRAD}\,(\mathsf{H}^{\mathcal{V}\tilde{\mathcal{C}}})^{\text{-}1}\,\widetilde{\mathsf{DIV}}\,\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}}\,\mathbf{u_{\boldsymbol{\theta}}}^{\mathcal{E}} = \mathsf{GRAD}\,(\mathsf{H}^{\mathcal{V}\tilde{\mathcal{C}}})^{\text{-}1}\,\widetilde{\mathsf{DIV}}\,\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}}\,\mathbf{u}^{\mathcal{E}} \tag{17}$$

$$(\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\mathsf{CURL}\,\mathbf{u_{\boldsymbol{\psi}}}^{\mathcal{E}} = (\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\mathsf{CURL}\,\mathbf{u}^{\mathcal{E}} \tag{18}$$

The harmonic field $\mathbf{u_h}^{\mathcal{E}}$ is deduced by subtraction:

$$\mathbf{u_h}^{\mathcal{E}} = \mathbf{u}^{\mathcal{E}} - \mathbf{u_{\boldsymbol{\theta}}}^{\mathcal{E}} - \mathbf{u_{\boldsymbol{\psi}}}^{\mathcal{E}} \tag{19}$$

Equations (17) and (18) were proposed by Angot *et al.* [4] to perform the DHHD. They introduced these equations with an extra penalization term $\varepsilon\mathrm{Id}$, where ε is a small real number. In a following section, §4.2, we will discuss how to get rid of it by using suitable linear solvers.

Concerning the boundary conditions, we distinguish two cases. If the first operator acting on the unknown discrete field is a primal operator, both primal and dual boundary conditions are available. For instance, in equation (15), we can impose the value of $\boldsymbol{\theta}^{\mathcal{V}}$ on the primal boundary or the flux generated by the gradient $\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}}\cdot\mathsf{GRAD}\,\boldsymbol{\theta}^{\mathcal{V}}$ on the closure of the dual mesh, as in figure 3c. While, if the first operator acting on the unknown discrete field is a dual operator, only dual boundary conditions are available. Table 1 summarizes the boundary conditions available for equations (15) to (18).

Equation	Primal	Location	Dual	Location
$\boldsymbol{\theta}^{\mathcal{V}}$	θ	primal vertices	$\mathbf{grad}\,\theta \cdot \boldsymbol{n}$	dual faces
$\boldsymbol{\psi}^{\mathcal{F}}$	n/a		$\boldsymbol{\psi} \cdot \boldsymbol{t}$	dual edges
${\mathbf{u}_{\boldsymbol{\theta}}}^{\mathcal{E}}$	n/a		$\boldsymbol{u}_\theta \cdot \boldsymbol{n}$	dual faces
${\mathbf{u}_{\boldsymbol{\psi}}}^{\mathcal{E}}$	$\boldsymbol{u}_{\boldsymbol{\psi}} \cdot \boldsymbol{t}$	primal edges	$\mathbf{curl}\,\boldsymbol{u}_{\boldsymbol{\psi}} \cdot \boldsymbol{t}$	dual edges

Table 1: Boundary conditions available for the edge-based equations written in terms of continuous variables and their locations.

Note that the decomposition is unique up to the given boundary conditions (*e.g.* see [5]). Furthermore, some boundary conditions are not sufficient to give the uniqueness of the potentials. For instance, the discrete scalar potential $\boldsymbol{\theta}^{\mathcal{V}}$, obtained with equation (15), is unique up to a constant when using dual or periodic boundary conditions while it is unique when primal boundary conditions are applied. The other operators (16), (17), and (18) have a null space of large dimension, we will see in §4.2 how to find a solution which does not belong to the kernel.

3.2 Face-based DHHD

The face-based decomposition considers the initial discrete vector field $\mathbf{u}^{\mathcal{F}}$ as a set of fluxes. The degrees of freedom are located on the faces F. The following equation denotes the DHHD for fluxes:

$$\begin{array}{rccccc} \mathbf{u}^{\mathcal{F}} & = & (\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}})^{-1} \cdot \widetilde{\mathsf{GRAD}} \cdot \mathsf{H}^{\mathcal{C}\tilde{\mathcal{V}}}\,\boldsymbol{\theta}^{\mathcal{C}} & + & \mathsf{CURL}\,\boldsymbol{\psi}^{\mathcal{E}} & + \mathbf{u_h}^{\mathcal{F}} \\ & = & {\mathbf{u}_{\boldsymbol{\theta}}}^{\mathcal{F}} & + & {\mathbf{u}_{\boldsymbol{\psi}}}^{\mathcal{F}} & + \mathbf{u_h}^{\mathcal{F}} \end{array} \tag{20}$$

The two potentials and their derivatives can be extracted independently using the following equations:

$$\mathsf{DIV}\,(\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}})^{-1}\,\widetilde{\mathsf{GRAD}}\,\mathsf{H}^{\mathcal{C}\tilde{\mathcal{V}}}\,\boldsymbol{\theta}^{\mathcal{C}} = \mathsf{DIV}\,\mathbf{u}^{\mathcal{F}} \tag{21}$$

$$(\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{-1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\mathsf{CURL}\,\boldsymbol{\psi}^{\mathcal{E}} = (\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{-1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\mathbf{u}^{\mathcal{F}} \tag{22}$$

$$(\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}})^{-1}\,\widetilde{\mathsf{GRAD}}\,\mathsf{H}^{\mathcal{C}\tilde{\mathcal{V}}}\,\mathsf{DIV}\,{\mathbf{u}_{\boldsymbol{\theta}}}^{\mathcal{F}} = (\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}})^{-1}\,\widetilde{\mathsf{GRAD}}\,\mathsf{H}^{\mathcal{C}\tilde{\mathcal{V}}}\,\mathsf{DIV}\,\mathbf{u}^{\mathcal{F}} \tag{23}$$

$$\mathsf{CURL}\,(\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{-1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,{\mathbf{u}_{\boldsymbol{\psi}}}^{\mathcal{F}} = \mathsf{CURL}\,(\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{-1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\mathbf{u}^{\mathcal{F}} \tag{24}$$

The harmonic field $\mathbf{u_h}^{\mathcal{F}}$ is deduced by subtraction:

$$\mathbf{u_h}^{\mathcal{F}} = \mathbf{u}^{\mathcal{F}} - {\mathbf{u}_{\boldsymbol{\theta}}}^{\mathcal{F}} - {\mathbf{u}_{\boldsymbol{\psi}}}^{\mathcal{F}} \tag{25}$$

Table 2 summarizes the boundary conditions available for equations (21) to (24).

Equation	Primal	Location	Dual	Location
$\boldsymbol{\theta}^{\mathcal{C}}$	n/a		θ	dual vertices
$\boldsymbol{\psi}^{\mathcal{E}}$	$\boldsymbol{\psi} \cdot \boldsymbol{t}$	primal edges	$\mathbf{curl}\, \boldsymbol{\psi} \cdot \boldsymbol{t}$	dual edges
$\mathbf{u}_{\boldsymbol{\theta}}{}^{\mathcal{F}}$	$\boldsymbol{u}_\theta \cdot \boldsymbol{n}$	primal faces	$\operatorname{div} \boldsymbol{u}_\theta$	dual vertices
$\mathbf{u}_{\boldsymbol{\psi}}{}^{\mathcal{F}}$	n/a		$\boldsymbol{u}_{\boldsymbol{\psi}} \cdot \boldsymbol{t}$	dual edges

Table 2: Boundary conditions available for the face-based equations written in terms of continuous variables and their location.

4 Numerical Results

We start this section by defining the various discrete norms we will use to evaluate the errors and we explain the choice of the iterative methods to solve the algebraic systems issued from the DHHD. Finally, we present some numerical results to validate our approach.

4.1 Discrete norms

To measure errors at the discrete level, a discrete norm must be designed. We could use the discrete duality product defined on equation (10) with a discrete Hodge operator to create an inner product and use this inner product to build a discrete norm. For example:

$$\operatorname{norm}(\mathbf{u}) = \sqrt{\left[\!\left[\mathbf{u}, \mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}}\, \mathbf{u}\right]\!\right]_{\mathcal{E}\tilde{\mathcal{F}}}} \tag{26}$$

However, this norm depends on a discrete Hodge operator. Instead, we chose a Hodge-independent discrete norm based on a generic partition of the whole domain. The following equations define the discrete p-norms for all the cochain spaces:

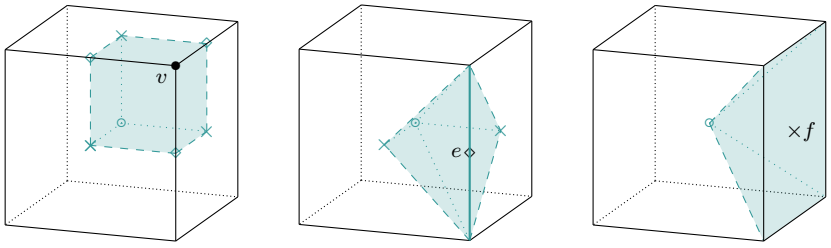
$$\forall \boldsymbol{\theta}^{\mathcal{V}} \in \mathcal{V} \quad \|\boldsymbol{\theta}^{\mathcal{V}}\|_{p,\mathcal{V}} \coloneqq \left(\sum_{c \in C} \sum_{v \in V_c} |\mathfrak{p}_{v,c}| |\boldsymbol{\theta}_c^{\mathcal{V}}|^p \right)^{1/p} \tag{27}$$

$$\forall \mathbf{u}^{\mathcal{E}} \in \mathcal{E} \quad \|\mathbf{u}^{\mathcal{E}}\|_{p,\mathcal{E}} \coloneqq \left(\sum_{c \in C} \sum_{e \in E_c} |\mathfrak{p}_{e,c}| \left(\frac{|\mathbf{u}_e^{\mathcal{E}}|}{|e|} \right)^p \right)^{1/p} \tag{28}$$

$$\forall \boldsymbol{\psi}^{\mathcal{F}} \in \mathcal{F} \quad \|\boldsymbol{\psi}^{\mathcal{F}}\|_{p,\mathcal{F}} \coloneqq \left(\sum_{c \in C} \sum_{f \in F_c} |\mathfrak{p}_{f,c}| \left(\frac{|\boldsymbol{\psi}_f^{\mathcal{F}}|}{|f|} \right)^p \right)^{1/p} \tag{29}$$

$$\forall \boldsymbol{\rho}^{\mathcal{C}} \in \mathcal{C} \quad \|\boldsymbol{\rho}^{\mathcal{C}}\|_{p,\mathcal{C}} \coloneqq \left(\sum_{c \in C} |c| \left(\frac{|\boldsymbol{\rho}^{\mathcal{C}}|}{|c|} \right)^p \right)^{1/p} \tag{30}$$

In these equations, the errors are computed and summed cell by cell. We use the notation V_c to denote the set of all the vertices which belong to a cell c. This notation is also used for the set of edges and faces which belong to a cell c: E_c and F_c. Each geometric element of the cell is associated to a subvolume of the cell to which it belongs. These subvolumes are built to ensure a partition of unity of the cell. They are denoted $\mathfrak{p}$ with the associated geometric element and the cell where it belongs to in subscript. For instance, the subvolume associated to an edge e of a cell c is denoted $\mathfrak{p}_{e,c}$. Some examples of subvolumes are represented in figure 4.



(a) $\mathfrak{p}_{v,c}$ (b) $\mathfrak{p}_{e,c}$ (c) $\mathfrak{p}_{f,c}$

Fig. 4: Subvolumes associated with mesh elements

Consider a field ϕ and a cochain $\boldsymbol{\phi}^{\mathcal{A}} \in \mathcal{A}$ which approaches ϕ at the discrete level. The cochain space $\mathcal{A}$ can be any of the spaces $\{\mathcal{V}, \mathcal{E}, \mathcal{F}, \mathcal{C}\}$. The discrete L^2 error on ϕ is defined by:

$$\mathrm{Er}_{\mathcal{A}}(\phi) \coloneqq \|\boldsymbol{\phi}^{\mathcal{A}} - \mathsf{R}_{\mathcal{A}}(\phi)\|_{2,\mathcal{A}} \tag{31}$$

Consider a sequence of meshes of increasing density indexed with integers. The convergence rate R between two meshes $i-1$ and i is given by

$$\mathrm{R} \coloneqq -3 \log\left(\frac{\mathrm{Er}^{i}_{\mathcal{A}}}{\mathrm{Er}^{i-1}_{\mathcal{A}}}\right) / \log\left(\frac{\#A^{i}}{\#A^{i-1}}\right) \tag{32}$$

bluewhere $\mathrm{Er}^{i}_{\mathcal{A}}$ is the discrete error relative to mesh i and $\#A^i$ is the number of geometric elements of type A in mesh i. Note that coefficient 3 depends on the dimension of the cell-complex. For instance, in a planar case, the coefficient will be 2. The reader can refer to [18] for this definition.

4.2 Linear solvers

blueWe have seen in section 3 above that most of the components of the DHHD can be extracted using singular linear systems. Without loss of generality, consider the extraction of the solenoidal field. This problem consists in finding

a cochain $\mathbf{u}_{\boldsymbol{\psi}}$ in $\mathrm{Im}((\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}) \subset \mathcal{E}$ with respect to the following equation:

$$(\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\mathsf{CURL}\,\mathbf{u}_{\boldsymbol{\psi}}{}^{\varepsilon} = (\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\mathsf{CURL}\,\mathbf{u}^{\varepsilon} \tag{33}$$

blueWe distinguish at least three ways to resolve this problem. The first consists in solving a constrained problem involving a Lagrange multiplier. The second consists in approaching the solution using regularization, such as the VPP an RPP methods proposed by Angot *et al.* [19]. The third way consists in using well designed discrete operators coupled with an iterative linear solver. This is the way we choose to perform the DHHD.

blueEquation (33) can be rewritten $\mathsf{A}x = b$, where:

$$\begin{aligned} \mathsf{A} &= (\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\mathsf{CURL} \\ b &= (\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\mathsf{CURL}\,\mathbf{u}^{\varepsilon} \\ x &= \mathbf{u}_{\boldsymbol{\psi}}{}^{\varepsilon} \end{aligned} \tag{34}$$

blueNotice that $\mathrm{Im}(A) \subset \mathrm{Im}((\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}})$ and $b \in \mathrm{Im}((\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}})$. If we make an initial guess x that belongs to $\mathrm{Im}((\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}})$ and choose an iterative method that performs only linear combinations or matrix-vector products of these elements, the solution will remain in $\mathrm{Im}((\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}})$. For instance, we can consider Krylov-based linear solvers. blueThe matrix involved in our linear systems are non-symmetric. Thus, we need to use suitable Krylov-based linear solvers. In this paper, we focus on three of them: BiCGStab2 [20], BiCGStab(l) [21] and (Full GMRES) FGMRES [22].

blueHowever, $\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}}$ and $\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}$ are not diagonal on polyhedral meshes. Thus, their inverses are not sparse. We have found to ways to manage this problem depending on the position of the inverse of the Hodge operator in the operator A. If the inverse is at the extremity of the operator — such as in equations (18), (22) and (23) — we use the preconditioning trick. Otherwise, if the inverse is in the middle of the operator — such as in equations (16), (21) and (24) — we use the saddle-point trick.

blueThe preconditioning trick consists in considering the Hodge operator as a left preconditioner M of the linear system $\mathsf{A}'x = b'$. Where $\mathsf{A}' = \widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\mathsf{CURL}$ and $b' = \widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}\,\mathsf{CURL}\,\mathbf{u}^{\varepsilon}$. That is $\mathsf{M}^{-1}\mathsf{A}'x = \mathsf{M}^{-1}b'$, with $\mathsf{M} = \mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}}$. Note that we cannot simply remove $(\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}$ by multiplying by $\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}}$. If we do so, the approached solution x will not remain in $\mathrm{Im}((\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}})$ since $\mathrm{Im}(\mathsf{A}') \not\subset \mathrm{Im}((\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{\text{-}1}\,\widetilde{\mathsf{CURL}}\,\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}})$. However, for evident performance reasons, we refuse to explicitly compute an inverse matrix and we refuse to solve a linear system for each iteration of the Krylov-based linear system. Fortunately, the Hodge operator is a sparse symmetric definite positive matrix. Thus, we can compute a permuted Cholesky decomposition $\mathsf{M} = \mathsf{P}\mathsf{L}\mathsf{L}^{\top}\mathsf{P}^{\top}$ that remains sparse. This decomposition can be performed at the start of the iterative method. Then, at every iteration, we have just to perform a fast bottom-up operation to apply preconditioning where it is required.

blueThe saddle-point trick involves less computation than the preconditioning trick but involves a bigger linear system. For instance, the extraction of the scalar potential can be equivalently rewritten in the following saddle-point problem: find $(\boldsymbol{\theta}^{\tilde{\mathcal{V}}}, \mathbf{u}_{\boldsymbol{\theta}}{}^{\mathcal{F}}) \in \tilde{\mathcal{V}} \times \mathcal{F}$ such that

$$\begin{cases} \mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}} \cdot \mathbf{u}_{\boldsymbol{\theta}}{}^{\mathcal{F}} - \widetilde{\mathsf{GRAD}}\, \boldsymbol{\theta}^{\tilde{\mathcal{V}}} & = \mathbf{0}^{\tilde{\mathcal{E}}} \\ \mathsf{DIV}\, \mathbf{u}_{\boldsymbol{\theta}}{}^{\mathcal{F}} & = \mathsf{DIV}\, \mathbf{u}^{\mathcal{F}} \end{cases} \tag{35}$$

blueNote that this linear system allows the extraction of the irrotational component too.

blueAnother way to explore this is the implementation of the algebraic multigrid proposed by Bell [23].

4.3 Numerical experiments

To illustrate the efficiency of our approaches, we performed several numerical experiments that we split into two families. First, we attempted to find the best linear solver to compute the DHHD. This step was carried out on Cartesian meshes with periodic boundary conditions. Then, we used this solver to compute the DHHD on polyhedral meshes with boundary conditions different than periodic. Each of the numerical results will be computed on some mesh sequences of increasing density.

4.3.1 Cartesian meshes

The table 3 depicts the mesh sequence used for our tests. These meshes comes from tessellation of the hexahedron $[-1, 1] \times [-2, 2] \times [-3, 3]$.

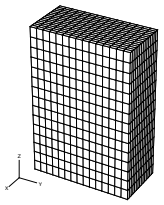


Fig. 5: Cartesian 16

Mesh	$\#V$	$\#E$	$\#F$	$\#C$
cart 8	512	1,344	1,176	343
cart 16	4,096	11,520	10,800	3,375
cart 32	32,768	95,232	92,256	29,791
cart 64	262,144	774,144	762,048	250,047
cart 128	2,097,152	6,242,304	6,193,536	2,048,383

Table 3: Cartesian mesh sequence

The following periodic potentials are used for our tests:

$$\boldsymbol{\psi} = -\frac{1}{\pi} \begin{pmatrix} \sin^3(\pi y)\cos^3(\pi z) \\ \sin^3(\pi z)\cos^3(\pi x) \\ \sin^3(\pi x)\cos^3(\pi y) \end{pmatrix} \quad \theta = -\frac{1}{\pi}\cos^3(\pi x)\cos^3(\pi y)\cos^3(\pi z) \tag{36}$$

We also introduce the following harmonic field:

$$\boldsymbol{u}_h = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^{\mathsf{T}} \tag{37}$$

Thus, the initial vector field writes:

$$\boldsymbol{u} = \mathbf{curl}\,\boldsymbol{\psi} + \mathbf{grad}\,\theta + \boldsymbol{u}_h \tag{38}$$

This field is discretized onto the edges and faces using the de Rham maps:

$$\mathbf{u}^{\mathcal{E}} = \mathsf{R}_{\mathcal{E}}(\boldsymbol{u}) \qquad \mathbf{u}^{\mathcal{F}} = \mathsf{R}_{\mathcal{F}}(\boldsymbol{u}) \tag{39}$$

In practice, we use high order quadrature formulas to approximate the de Rham map near the unit roundoff.

For the Cartesian meshes we use the following diagonal Hodge operators:

$$\forall (a, a') \in A^2 \quad (\mathsf{H}^{\mathcal{A}\tilde{\mathcal{B}}})_{a,a'} = \begin{cases} \dfrac{|\tilde{b}(a)|}{|a|} & \text{if } a = a' \\ 0 & \text{otherwise} \end{cases} \tag{40}$$

where $\mathcal{A}$ denotes any cochain space and $\tilde{\mathcal{B}}$ its dual. The measure of a vertex is considered as equal to 1.

For the tests, we set the maximum number of iterations to 128 for both FGMRES and BiCGStab2. The time taken to extract the components of the DHHD are reported in the figure 6. Note that BiCGStab2 is about 5 times faster than FGMRES. Some issues have been experienced with BiCGStab2. The solver becomes unstable once the minimal residual is reached. We had to modify the solver to select the solution with the minimum residual. This behavior is not encountered for the FGMRES. However, the dimension of the Krylov basis for the FGMRES solver has to be set large enough to not produce a restart, since any restart causes the divergence of the solver. We also recall that FGMRES has a memory cost proportional to the number of iterations, which means that it has a memory cost much larger than BiCGStab2. Concerning the BiCGStab(l) solver, it provides results similar to those of BiCGStab2 for $l = 2$. With $l = 4$, we observe that one iteration of BiCGStab(4) is equivalent to two iterations of BiCGStab(2) with no performance gain.

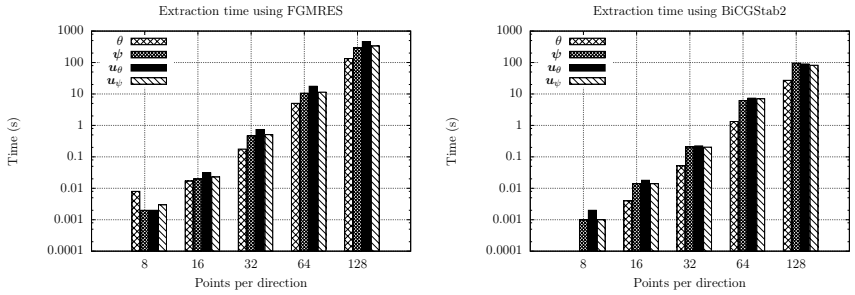


Fig. 6: Comparison of the times elapsed between FGMRES and BiCGStab2 for the extraction of the various components for the edge-based decomposition

The number of iterations for both solvers are reported in figure 7. Note that the extraction of the scalar potential $\boldsymbol{\theta}^{\mathcal{V}}$ requires as many iterations as the other fields. This result is counter-intuitive because of the system size. The other systems seem better-conditioned than the extraction of $\boldsymbol{\theta}^{\mathcal{V}}$, but the memory cost is about 3 times greater (see table 3 for the number of degrees of freedom). This behavior for the system conditioning has already been observed in [24] and [19]. The maximum number of iterations is reached for $\mathbf{u}_{\boldsymbol{\theta}}{}^{\mathcal{E}}$ with 112 iterators using FGMRES and for $\boldsymbol{\theta}^{\mathcal{V}}$ with 56 iterations using BiCGStab2.

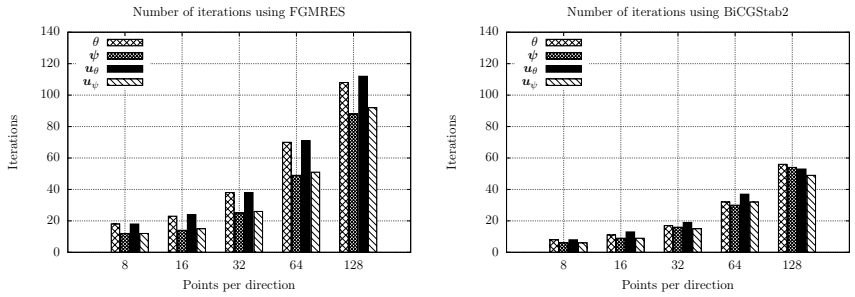


Fig. 7: Comparison between the number of iterations used by FGMRES and BiCGStab2 for the extraction of the various components for the edge-based decomposition

As expected, table 4 shows a convergence rate of 2. Note that the errors on $\boldsymbol{u}_{\psi}$ are not represented since they are the same as the errors on $\boldsymbol{u}_{\theta}$ up to the unit roundoff.

Mesh	$\mathrm{Er}_{\mathcal{V}}(\theta)$	R	$\mathrm{Er}_{\mathcal{F}}(\boldsymbol{\psi})$	R	$\mathrm{Er}_{\mathcal{E}}(\boldsymbol{u}_{\theta})$	R
cart 8	3.5×10^{-1}		2.3×10^{-1}		1.1×10^{0}	
cart 16	7.5×10^{-2}	2.3	1.2×10^{-1}	0.9	4.4×10^{-1}	1.4
cart 32	1.1×10^{-2}	2.7	2.8×10^{-2}	2.1	9.9×10^{-2}	2.2
cart 64	2.5×10^{-3}	2.2	6.9×10^{-3}	2.0	2.4×10^{-2}	2.1
cart 128	5.9×10^{-4}	2.1	1.7×10^{-3}	2.0	5.8×10^{-3}	2.0

Table 4: Discrete errors and convergence rates for the Cartesian mesh sequence for the edge-based decomposition

The same parameters as the edge-based decomposition are used for the face-based decomposition. We only have to adjust the maximum number of iterations to 150. We found similar results as those obtained with edge-based decomposition for time (figure 8) andi for the number of iterations (figure 9). The maximum number of iterations is reached for $\mathbf{u}_{\boldsymbol{\psi}}{}^{\mathcal{F}}$ with 135 iterations using the FGMRES solver and also for $\mathbf{u}_{\boldsymbol{\psi}}{}^{\mathcal{F}}$a with 61 iterations using BiCGStab2.

As well as edge-based schemes, a convergence rate of 2 is observed for these schemes (table 9).

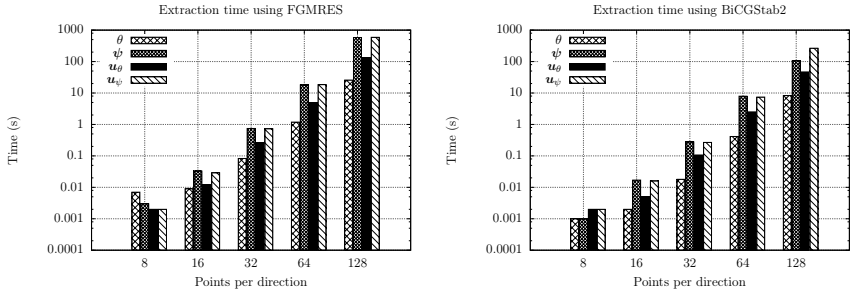


Fig. 8: Comparison of the elapsed times between FGMRES and BiCGStab2 for the extraction of the various components for the face-based decomposition

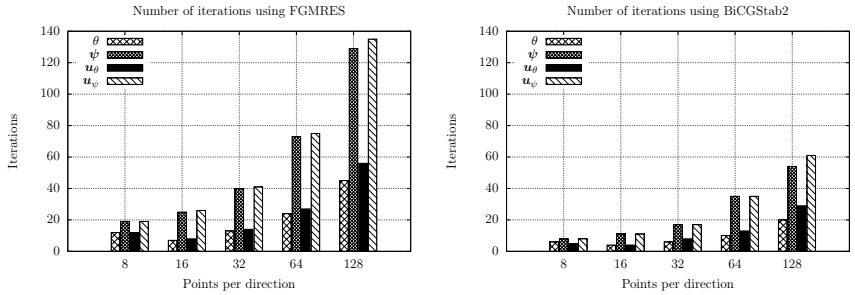


Fig. 9: Comparison between the number of iterations used by FGMRES and BiCGStab2 for the extraction of the various components for the face-based decomposition

Mesh	$\mathrm{Er}_{\mathcal{E}}(\boldsymbol{\psi})$	R	$\mathrm{Er}_{\mathcal{F}}(\boldsymbol{u}_\theta)$	R	$\mathrm{Er}_{\mathcal{C}}(\theta)$	R
cart 8	4.7×10^{-2}		2.1×10^{-1}		4.5×10^{-2}	
cart 16	1.6×10^{-2}	1.5	1.1×10^{-1}	0.8	3.4×10^{-2}	0.4
cart 32	2.9×10^{-3}	2.4	2.9×10^{-2}	2.0	9.5×10^{-3}	1.8
cart 64	6.7×10^{-4}	2.1	7.1×10^{-3}	2.0	2.4×10^{-3}	2.0
cart 128	1.6×10^{-4}	2.0	1.8×10^{-3}	2.0	5.9×10^{-4}	2.0

Table 5: Discrete errors and convergence rates for the Cartesian mesh sequence for the face-based decomposition

For both edge-based and face-based decompositions, the errors on the harmonic terms ${\mathbf{u_h}}^{\mathcal{E}}$ and ${\mathbf{u_h}}^{\mathcal{F}}$ are observed being equal to the unit roundoff. The properties of extracted fields (zero curl and divergence) are also verified to the unit roundoff.

Figure 10 summarizes the errors for both edge-based and face-based methods.

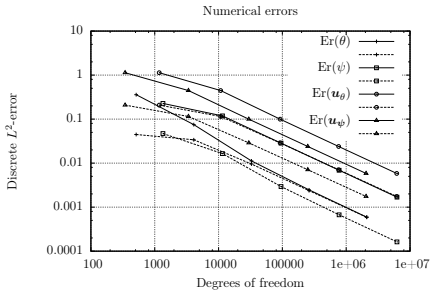


Fig. 10: Numerical errors on extracted components. Plain lines denote errors on edge-based method and dashed lines denote errors on face-based method

For the sake of completeness, we provide the gain offered by preconditioners for the extraction of the discrete scalar potentials in figure 11. We can see that ILU0 decreases at least by a factor of 3 the number of iterations of the BiCGStab2 solver in comparison with BiCGStab2 without preconditioner.

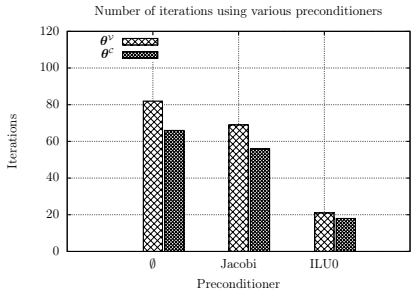


Fig. 11: Comparison between the number of iterations for various preconditioners using BiCGStab2.

This first series of numerical experiments confirms that the numerical schemes provide correct results. We have seen that BiCGStab2 is efficient to solve our equations. We will use this numerical solver in the remainder to present the DHHD on polyhedral meshes.

4.3.2 Polyhedral meshes

For the polyhedral meshes, we chose the discrete Hodge operators designed by [16] for the Discrete Geometry Approach (DGA) schemes. These operators allow an explicit reconstruction of gradients and fluxes without minimization or explicit matrix inversion. According to [7], these operators require some regularities of the primal mesh. For instance, the primal faces must be planar. Since these operators are only designed for gradients and fluxes ($\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}}$ and $\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}}$), we keep the diagonal Hodge operators for $\mathsf{H}^{\mathcal{V}\tilde{\mathcal{C}}}$ and $\mathsf{H}^{\mathcal{C}\tilde{\mathcal{V}}}$.

To evaluate our method on polyhedral meshes, we use the meshes provided by the FVCA benchmark [25]. For the sake of clarity, we present only two of the mesh sequences (figures 12 and 13). These mesh sequences summarizes the two convergence behaviors using DGA Hodge operators. The characteristics of the mesh sequences are provided in tables 6 and 7.

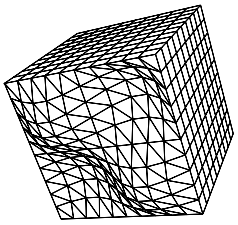


Fig. 12: BLS 10

Mesh	$\#V$	$\#E$	$\#F$	$\#C$
bls 10	1,331	4,730	5,400	2,000
bls 20	9,261	34,860	41,600	16,000
bls 30	29,791	114,390	138,600	54,000
bls 40	68,921	267,320	326,400	128,000

Table 6: Prism mesh sequence

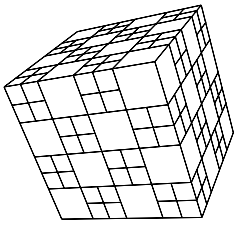


Fig. 13: Checkerboard 4

Mesh	$\#V$	$\#E$	$\#F$	$\#C$
chkb 4	625	1,536	1,200	288
chkb 8	4,417	11,520	9,408	2,304
chkb 16	33,025	89,088	74,496	18,432
chkb 32	254,977	700,416	592,896	147,456

Table 7: Checkerboard mesh sequence

Since these polyhedral meshes are not periodic, we use the discrete fields (36) introduced for the Cartesian mesh sequence without the constant field (37). We also need boundary conditions. We recall that the components of the DHHD depend on these boundary conditions. Unsuitable boundary conditions lead to a different decomposition (*e.g.* see [5]). To retrieve our analytical components, we need to impose the exact solution on the boundary using primal or dual boundary conditions.

In equations (16), (21) and (24), we observe that the inverse of a DGA Hodge operator is required. To avoid explicit matrix inversion, we propose to solve equation (16) using a saddle-point problem. Considering only the dual vector potential $\boldsymbol{\psi}^{\tilde{\varepsilon}}$ and introducing $\mathbf{u}_{\boldsymbol{\psi}}{}^{\varepsilon} = (\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}})^{-1} \cdot \boldsymbol{\psi}^{\tilde{\varepsilon}}$, we rewrite equa-

tion (16) to obtain the following saddle-point problem: find $(\mathbf{u}_{\boldsymbol{\psi}}{}^{\mathcal{E}}, \boldsymbol{\psi}^{\tilde{\mathcal{E}}}) \in \mathcal{E} \times \tilde{\mathcal{E}}$ such that

$$\begin{cases} -\mathsf{H}^{\mathcal{E}\tilde{\mathcal{F}}} \cdot \mathbf{u}_{\boldsymbol{\psi}}{}^{\mathcal{E}} + \widetilde{\mathsf{CURL}}\, \boldsymbol{\psi}^{\tilde{\mathcal{E}}} & = \mathbf{0}^{\tilde{\mathcal{F}}} \\ \quad \mathsf{CURL}\, \mathbf{u}_{\boldsymbol{\psi}}{}^{\mathcal{E}} & = \mathsf{CURL}\, \mathbf{u}^{\mathcal{E}} \end{cases} \tag{41}$$

This problem allows the vector potential and the solenoidal component to be computed at the same time. We impose dual boundary conditions on $\boldsymbol{\psi}^{\tilde{\mathcal{E}}}$ to obtain the results presented in figure 14. Table 8 shows that we obtain a first order of convergence for the vector potential $\boldsymbol{\psi}^{\tilde{\mathcal{E}}}$. However, the solenoidal component $\mathbf{u}_{\boldsymbol{\psi}}{}^{\mathcal{E}}$ presents two convergence behaviors depending on the mesh regularity. We also find two different convergence behaviors for the irrotational component $\mathbf{u}_{\boldsymbol{\theta}}{}^{\mathcal{E}}$ while the scalar potential $\boldsymbol{\theta}^{\mathcal{V}}$ reaches the second order of convergence. This behavior is in line with the results found by [7].

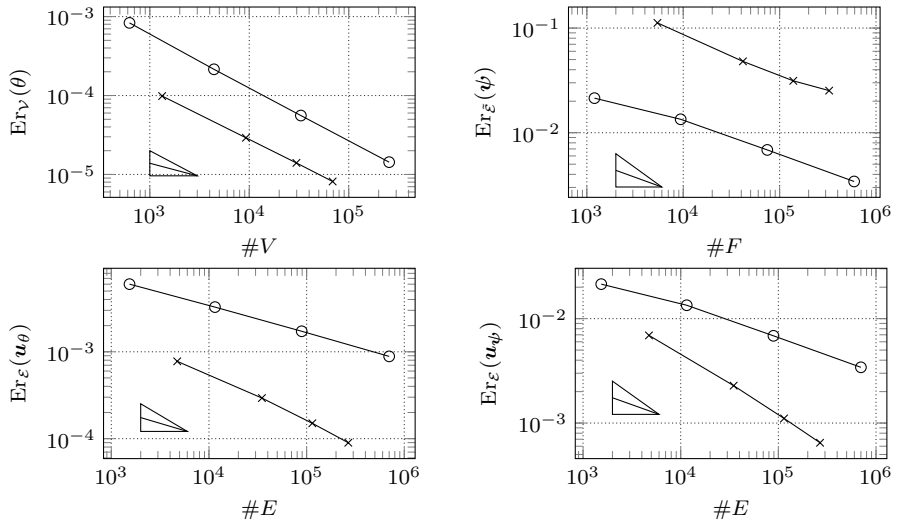


Fig. 14: Error measurements for the polyhedral mesh sequences for the edge-based DHHD. ∘ denotes the checkerboard and × denotes the prism mesh.

Mesh	$\mathrm{Er}_{\mathcal{V}}(\theta)$	R	$\mathrm{Er}_{\tilde{\mathcal{E}}}(\psi)$	R	$\mathrm{Er}_{\mathcal{E}}(\boldsymbol{u}_{\boldsymbol{\theta}})$	R	$\mathrm{Er}_{\mathcal{E}}(\boldsymbol{u}_{\psi})$	R
bls 10	9.8×10^{-5}		1.1×10^{-1}		7.8×10^{-4}		6.9×10^{-3}	
bls 20	2.9×10^{-5}	1.9	4.8×10^{-2}	1.2	2.9×10^{-4}	1.5	2.3×10^{-3}	1.7
bls 30	1.4×10^{-5}	1.9	3.1×10^{-2}	1.1	1.5×10^{-4}	1.7	1.1×10^{-3}	1.8
bls 40	8.1×10^{-6}	1.9	2.5×10^{-2}	0.8	9.0×10^{-5}	1.8	6.5×10^{-4}	1.9
chkb 4	8.3×10^{-4}		2.1×10^{-2}		6.0×10^{-3}		2.1×10^{-2}	
chkb 8	2.1×10^{-4}	2.1	1.3×10^{-2}	0.7	3.3×10^{-3}	0.9	1.3×10^{-2}	0.7
chkb 16	5.6×10^{-5}	2.0	6.8×10^{-3}	1.0	1.7×10^{-3}	0.9	6.8×10^{-3}	0.9
chkb 32	1.4×10^{-5}	2.0	3.4×10^{-3}	1.0	8.9×10^{-4}	1.0	3.4×10^{-3}	1.0

Table 8: Discrete errors and convergence rates for the polyhedral mesh sequences for the edge-based decomposition

For the face-based schemes, we also need to solve a saddle-point problem to extract the scalar potential $\boldsymbol{\theta}^{\tilde{\mathcal{V}}}$. By introducing the irrotational component $\mathbf{u}_{\boldsymbol{\theta}}{}^{\mathcal{F}} = (\mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}})^{-1} \cdot \boldsymbol{\theta}^{\tilde{\mathcal{V}}}$, equation (21) can be transformed in the following saddle-point problem: find $(\boldsymbol{\theta}^{\tilde{\mathcal{V}}}, \mathbf{u}_{\boldsymbol{\theta}}{}^{\mathcal{F}}) \in \tilde{\mathcal{V}} \times \mathcal{F}$ such that

$$\begin{cases} \mathsf{H}^{\mathcal{F}\tilde{\mathcal{E}}} \cdot \mathbf{u}_{\boldsymbol{\theta}}{}^{\mathcal{F}} - \widetilde{\mathsf{GRAD}}\, \boldsymbol{\theta}^{\tilde{\mathcal{V}}} & = \mathbf{0}^{\tilde{\mathcal{E}}} \\ \mathsf{DIV}\, \mathbf{u}_{\boldsymbol{\theta}}{}^{\mathcal{F}} & = \mathsf{DIV}\, \mathbf{u}^{\mathcal{F}} \end{cases} \tag{42}$$

We endow this system with Dirichlet boundary conditions. The discrete errors and convergence rates for the face-based DHHD are presented in figure 15 and table 9.

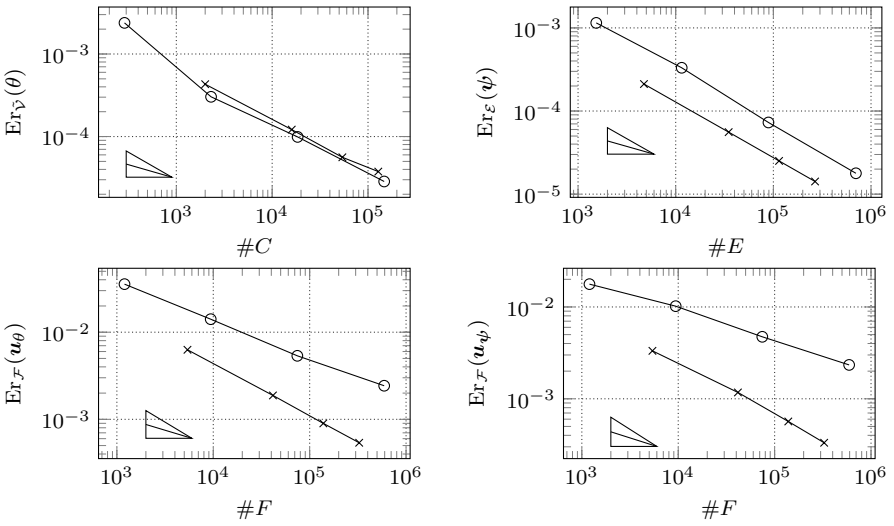


Fig. 15: Error measurements for the polyhedral mesh sequences for face-based DHHD. ∘ denotes the checkerboard and × denotes the prism mesh.

Mesh	$\mathrm{Er}_{\tilde{\mathcal{V}}}(\theta)$	R	$\mathrm{Er}_{\mathcal{E}}(\psi)$	R	$\mathrm{Er}_{\mathcal{F}}(\boldsymbol{u_\theta})$	R	$\mathrm{Er}_{\mathcal{F}}(\boldsymbol{u_\psi})$	R
bls 10	4.3×10^{-4}		2.1×10^{-4}		6.3×10^{-3}		3.3×10^{-3}	
bls 20	1.2×10^{-4}	1.8	5.6×10^{-5}	2.0	1.9×10^{-3}	1.8	1.2×10^{-3}	1.5
bls 30	5.6×10^{-5}	1.9	2.5×10^{-5}	2.0	9.0×10^{-4}	1.8	5.7×10^{-4}	1.8
bls 40	3.8×10^{-5}	1.4	1.4×10^{-5}	2.0	5.4×10^{-4}	1.8	3.3×10^{-4}	1.9
chkb 4	2.4×10^{-3}		6.3×10^{-3}		3.6×10^{-2}		1.8×10^{-2}	
chkb 8	3.0×10^{-4}	2.9	1.9×10^{-3}	1.9	1.4×10^{-2}	1.3	1.0×10^{-2}	0.8
chkb 16	9.9×10^{-5}	1.6	9.0×10^{-4}	2.2	5.4×10^{-3}	1.4	4.7×10^{-3}	1.1
chkb 32	2.9×10^{-5}	1.8	5.4×10^{-4}	2.0	2.4×10^{-3}	1.1	2.3×10^{-3}	1.1

Table 9: Discrete errors and convergence rates for the polyhedral mesh sequences for the face-based decomposition

We measure a second order of convergence for all the vector components and potentials on the prism mesh sequence. A second order of convergence is also found for the potentials on the checkerboard mesh sequence but only a first order is found on the vector components.

During our numerical experiments, we observe that the vector potential is harder to extract than the other components. The equations and methods to extract $\boldsymbol{\psi}$ are extremely sensitive to boundary conditions and numerical perturbations. Great caution must be taken in the implementation of the CDO schemes and solvers to avoid disturbing this extraction.

We use BiCGStab2 to solve our equations. The number of iterations required for this solver are close to those of the Cartesian mesh sequence (<100) to reach the same residual. However, the FGMRES solver is unusable on polyhedral meshes. Our numerical experiments have shown that we cannot converge in a reasonable number of iterations with this solver.

5 Conclusion

We propose a methodology to perform the discrete Helmholtz-Hodge decomposition on polyhedral meshes using numerical schemes that preserve the properties of the decomposition at the discrete level, namely $\mathbf{curl}\,\boldsymbol{u_\theta} = \mathbf{0}$ and $\operatorname{div} \boldsymbol{u_\psi} = 0$. We have chosen the Compatible Discrete Operator schemes designed by Bonelle & Ern [7] to ensure these properties. These schemes lead to two kinds of DHHD, edge-based and face-based, depending on the discrete representation of the discrete vector field. We can identify four components (two vector fields and two potentials) plus a harmonic field in each kind of DHHD. We propose four algorithms to extract these components independently. Most of these algorithms lead to singular linear systems, but we have shown that we can treat them as regular linear systems using Krylov-based linear solvers to get the expected solution.

We performed a two-step numerical validation. First, we performed validations on Cartesian meshes with periodic boundary conditions to check that our algorithms can effectively extract the various components of the DHHD and

to find the most suitable linear solver. Our numerical experiments show that the BiCGStab2 is a good linear solver, better than the FGMRES in terms of time and memory cost. Then, we use this linear solver to perform the DHHD on polyhedral meshes with boundary conditions other than periodic. This second step involved a choice of discrete Hodge operators required by the CDO schemes. Since we focus on performance, we chose the discrete Hodge operators designed by Codecasa *et al.* [16], given by an explicit formula without minimization or explicit matrix inversion. For many reasons intrinsic to these operators, we only reached a first order of convergence for some components of the decomposition. However, for most of the components, we were able to reach the second order of convergence whatever the mesh, as long as the primal faces were planar. In both validation steps, we checked that the properties of the DHHD are verified to the unit roundoff.

Future investigations to improve the methods may focus on finding other Hodge operators to reach an optimal convergence rate for all the components. To improve the performance in terms of computational time, the implementation of the algebraic multigrid of Bell [23] seems to be unavoidable.

Acknowledgements We would like to thank Jérôme Bonelle and Bruno Audebert from EDF and Marc Gerritsma from Delft University of Technology for meaningful discussions on the DHHD and discrete operators.

References

1. Y. Tong, S. Lombeyda, A.N. Hirani, M. Desbrun, in *ACM SIGGRAPH 2003 Papers on - SIGGRAPH '03* (ACM Press, New York, New York, USA, 2003), 1, p. 445
2. A. Wiebel, G. Scheuermann, C. Garth, Master's thesis. University of Kaiserslautern (2004)
3. K. Polthier, E. Preuß, Work **5**, 1 (2002)
4. P. Angot, J.P. Caltagirone, P. Fabrie, Appl. Math. Lett. **25**(11), 1681 (2012)
5. H. Bhatia, G. Norgard, V. Pascucci, P.T. Bremer, IEEE Transactions on Visualization and Computer Graphics **99**(PrePrints), 1 (2012)
6. J.M. Hyman, M. Shashkov, SIAM J. Numer. Anal **36**, 788 (1999)
7. J. Bonelle, A. Ern, ESAIM: Mathematical Modelling and Numerical Analysis (2013)
8. A. Bossavit, The Japan Society of Applied Electromagnetics and Mechanics **7**(2), 150 (1999)
9. J.M. Hyman, M. Shashkov, Applied Numerical Mathematics **25**(4), 413 (1997)
10. J.M. Hyman, M.J. Shashkov, Comput. Math. Appl. **33**(4), 81 (1997)
11. K. Lipnikov, G. Manzini, M. Shashkov, Journal of Computational Physics **257**, 1163 (2014)
12. Q. Guo, M.K. Mandal, M. Li, Pattern Recognition Letters **26**(4), 493 (2005)
13. M. Bluck, S. Walker, Antennas and Propagation, IEEE Transactions on **55**(5), 1338 (2007)
14. P.B. Bochev, J.M. Hyman, in *Compatible Spatial Discretization*, *The IMA Volumes in mathematics and its applications*, vol. 142, ed. by D.D. Arnold, P.B. Bochev, R.B. Lehoucq, R.A. Nicolaides, M. Shashkov (Springer, 2005), *The IMA Volumes in mathematics and its applications*, vol. 142, pp. 89–120
15. F. Brezzi, K. Lipnikov, M. Shashkov, V. Simoncini, Comput. Methods Appl. Mech. Engrg. **196**, 3682–3692 (2007)
16. L. Codecasa, R. Specogna, F. Trevisan, J. Comput. Physics **229**(19), 7401 (2010)
17. E. Tonti, Journal of Computational Physics **257, Part B**(0), 1260 (2014). DOI http://dx.doi.org/10.1016/j.jcp.2013.08.016. Physics-compatible numerical methods

18. J. Bonelle, A. Ern, arXiv preprint arXiv:1401.7842 (2014)
19. P. Angot, J.P. Caltagirone, P. Fabrie, Applied Mathematics Letters **26**, to appear (2013). 8 pages
20. M.H. Gutknecht, SIAM journal on scientific computing **14**(5), 1020 (1993)
21. G.L. Sleijpen, D.R. Fokkema, Electronic Transactions on Numerical Analysis **1**(11), 2000 (1993)
22. Intel. Math kernel library. http://developer.intel.com/software/products/mkl/
23. W.N. Bell, Algebraic multigrid for discrete differential forms. Ph.D. thesis, Champaign, IL, USA (2008). AAI3337696
24. Z. Ren, Magnetics, IEEE Transactions on **32**(3), 655 (1996)
25. R. Eymard, G. Henry, R. Herbin, F. Hubert, R. Klofkorn, G. Manzini, in *Proceedings of Finite Volumes for Complex Applications VI*. Springer (Springer, Praha, 2011), pp. 895–930