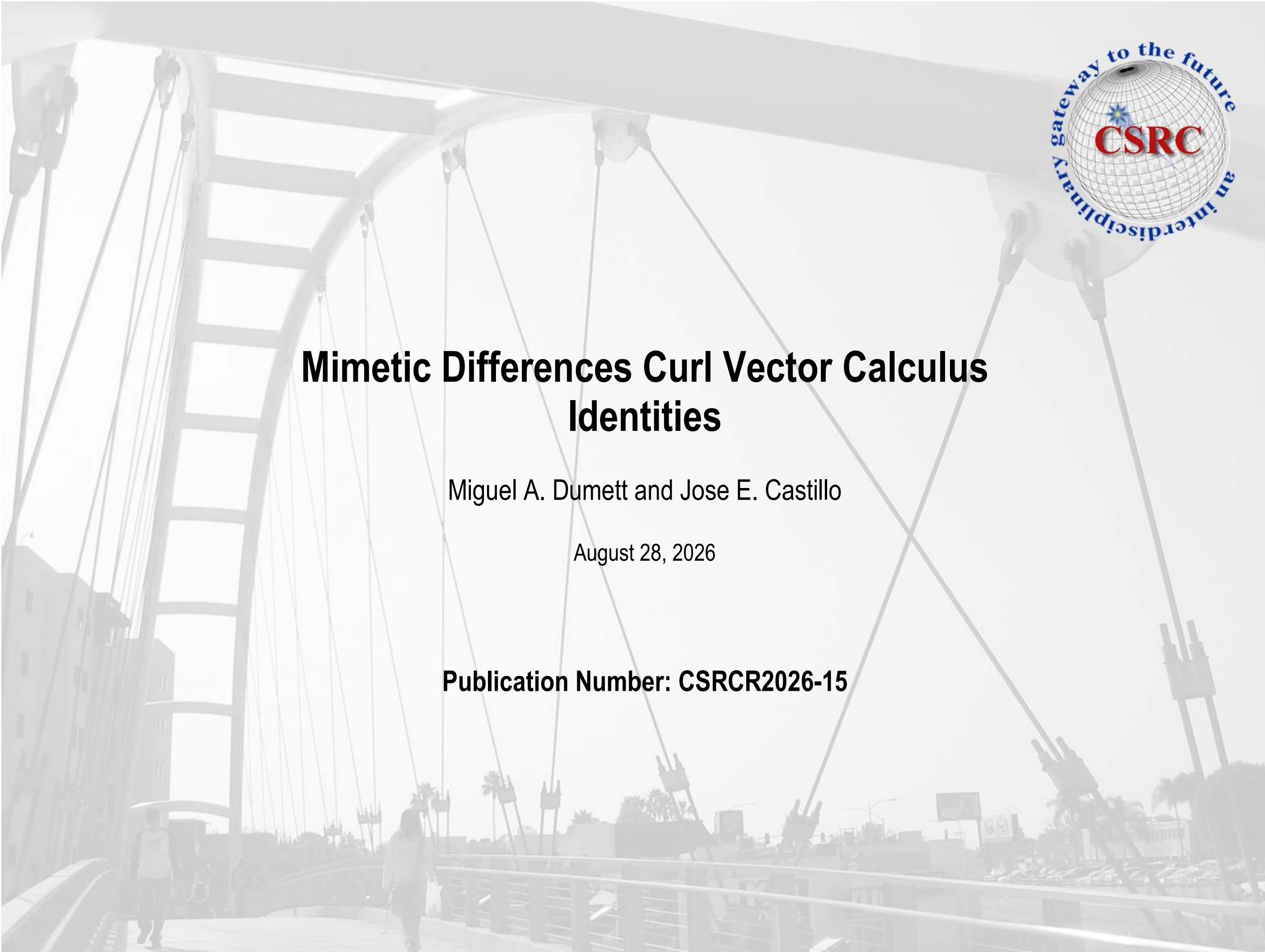

Mimetic Differences Curl Vector Calculus Identities

Miguel A. Dumett and Jose E. Castillo

August 28, 2026

Publication Number: CSRCR2026-15

Computational Science &
Engineering Faculty and Students
Research Articles

Database Powered by the
Computational Science Research Center
Computing Group & Visualization Lab

COMPUTATIONAL SCIENCE & ENGINEERING

SAN DIEGO STATE UNIVERSITY

Computational Science Research Center
College of Sciences
5500 Campanile Drive
San Diego, CA 92182-1245
(619) 594-3430

Mimetic Differences Curl Vector Calculus Identities *

Miguel A. Dumett † Jose E. Castillo ‡

August 28, 2026

Abstract

The following report registers the process followed to define convenient discrete analogs of the two- and three-dimensional curl operators. The problem is that the discrete analog curl operators not only have to approximate well the curl calculation, and reproduce in the discrete realm curl vector calculus identities, but also the location of the input data and the points at which curls are calculated should be compatible with the respective input and output spots of the discrete analog of gradient, divergence and Laplacian operators, when they are present in the partial differential operator of the equation to be numerically solved. This document gathers the evolution of that search through different approaches for defining appropriate curl operators.

1 Introduction

Discrete analogs of high-order Corbino-Castillo two- and three-dimensional mimetic difference curl operators are introduced. These operators are of three different classes.

1. Operators of the first type are constructed for vector fields that are defined in face centers, and its outcome is also defined in face centers. To prove analogs of the vector calculus identities, it is required to review some of the properties of the divergence and gradient operators defined on the mimetic differences staggered grid. Due to the different operator input and output domains involved in the curl construction and to demonstrate discrete analogs of curl vector calculus identities, several interpolation operators are required. This formulation has the advantage of using only mimetic operator identities and the Kronecker product properties to demonstrate discrete analogs of curl vector calculus identities. Besides the usage of interpolations, the proof of these identities imposes equal cell number constraints on each axis. More importantly, they may not preserve mass, momenta, and energy for systems of conservation laws.

*This work was supported by SDSU.

†Computational Science Research Center at San Diego State University (mdumett@sdsu.edu).
‡Computational Science Research Center at San Diego State University (jcastillo@sdsu.edu).

2. The curl operators of the second kind are derived from the curl coordinate free definition, and each of its components is expressed in terms of the two-dimensional divergence operator applied on a shifted staggered grid, following ideas from [17, 1, 19]. These curl operators consider each component of the curl as a scalar field and hence to be consistent with the mimetic difference approach for operator discrete analogs, their input data should be located at cell centers. However, following the same rationale as the definition of the curl coordinate free definition, the two- and three-dimensional curl calculations require the input data to be located at face centers and its output at cell centers. An interpolation from centers to faces can be applied to the output if it is needed to be located at face centers. The advantages of these second type of formulations is that they do not use interpolations (at least before the output is relocated from centers to faces) and do not have any restriction with respect to the number of cells used along each axis. As a drawback, the curl coordinate free definition allows also to define the curl at boundary faces, and hence the output should be at cell centers and boundary faces. Considering output in the primal and dual meshes of the staggered grid goes against the avenue followed in the construction of other operator mimetic discrete analogs as well as interpolations and quadratures. It is decided in this approach to set to zero the curl at boundary faces, which is achieved by utilizing two-dimensional divergence discrete analogs due to its property of being zero at the boundary. These versions of the curl discrete analog satisfy vector identities by verifying Schwarz formula.

3. The third approach followed in the construction of curl discrete analogs leverage on the advantages of both previous types and are well suited for differential operators that include also gradient, divergence and Laplacian operators, by defining the input data and the location of the output calculation to be cell centers. This method also allows to iterate in time since input data and location of the output coincide. A convenient centers to faces interpolation is required to move the input data to a location that can be operated by the divergence. This special interpolation is needed because each of the components of the curl computation requires data at faces located in two different axes.

1.1 Some history

In this document, we propose discrete analogs of the two-dimensional (2D) and three-dimensional (3D) curl operators. The advancement of our ideas is motivated by the earlier work of a 2D curl operators in [17, 19] and Chapter 4 of [1], as well as an abstract identity of the ∇ operator that connects to the exterior derivative in differential geometry [1, 13]. At this respect, curl vector calculus identities essentially follow from Schwarz' identity [1].

Mimetic differences [2, 4, 11] is a family of mimetic methods that construct high-order accurate discrete analogs of vector calculus spatial differential operators with the goal of preserving in the discrete realm continuum vector calculus identities and integral theorems. To achieve this in one-dimension (1D), divergence and gradient 1D discrete analogs that satisfy the Fundamental Theorem of Calculus were initially introduced on staggered grids, where discrete scalar fields are defined in cell centers (the dual grid X_C), and discrete vector fields are defined in face centers (the primal grid X_F). It is interesting to note that Yee's algorithm for discretizing Maxwell's equations follows the same approach [24]. High-order approximations of the Integration By Parts (IBP) formula required the introduction of generalized inner products Q and P, to approximate integrals where the derivatives are associated to divergence D or gradient G operators, respectively [1, 17]. Broadening of these operators to Cartesian product domains, such that the new operators approximate with high-order accuracy the Extended Gauss Divergence Theorem, led to mimetic interpolation operators [7] from cell centers from face centers and vise versa.

Even though, expressions for how to compute the 2D curl and 3D curl based on a 2D divergence were proposed in [1, 17], a matrix formulation of the 2D and 3D curl operators was never given[1]. This is complicated by the fact that the input of a curl operator is a vector field (defined in X_F) and its output is also a vector field (also in X_F), while the expressions proposed in [1, 17] are based on divergence operators that act on 2D projections that are defined at cell centers X_C of both primal and dual grids.

The purpose of this investigation is to explain how these 2D and 3D curl operators have to be defined, what their discrete analog matrices are in such a way that they do not impose any constraints in the number of cells that have to be chosen on each axis, and that they do not use interpolation operators. This will be in contrast to another more "natural" definition of the 3D curl operator, which is based on considering the divergence and gradient operators as ways of approximating first-order partial derivatives. The latter approach is restricted to having the same number of cells in each axis and uses interpolation operators intensively. Finally, a new method for constructing curl discrete analogs that leverages on the virtues of the previous two techniques is introduced.

2 Mimetic difference basic operators

These include discrete analogs of the divergence and gradient operators, as well as, inner products weights and diverse interpolation operators. Since discrete analogs of curl vector calculus identities will be mathematically demonstrated an introduction to these operators and the mimetic difference staggered grid is needed. This section summarizes these definitions, properties and identities.

In the following subsections we will present the staggered grids, the mimetic difference 1D operators (their domains and range, together with its exact identities and properties),

[1] Actual curl operators were never implemented in the Mimetic Tool Kit (MTK) [18], the Castillo-Grone mimetic differences repository. Furthermore, the Mimetic Operators Library Enhanced (MOLE) [5, 6], the Corbino-Castillo mimetic differences repository, only includes the implementation of the 2D curl operator

and the 2D and 3D extensions of these operators.

2.1 The staggered grids

In mimetic differences, staggered 1D grid X of an interval, which is divided into sub-intervals or cells, is composed of cells faces (or edges or nodes) of all sub-intervals X_F, and the cell centers of all sub-intervals X_C (see Figure 1). Assume a h-uniform mesh size.

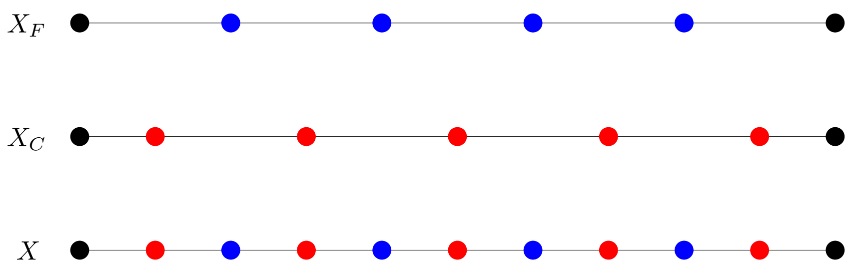

Figure 1: A 1D staggered grid X with 5 cells. It is composed of face centers (blue points and black boundary points) and cell centers (red points and black boundary points).

2.2 The 1D divergence and gradient operators

The number of cell centers (the cardinality of X_C) and cell faces (the cardinality of X_F) in a staggered grid is not the same. Discrete scalar fields domains are assumed to be in X_C, and discrete vector fields domains in X_F. One of the reasons for using a staggered grid is that in 2D and 3D the divergence and the gradient have different input and output.

The divergence acts on vector fields $\vec{v}$ and its output is a scalar field $\nabla\cdot\vec{v}$, while gradient acts on scalar fields f and its output is a vector field ∇f. Therefore, the discrete analog of the divergence operator $D^{(k)}$ that acts on the discrete projection $\vec{V}$ of $\vec{v}$, is supposed to approximate with order of accuracy k the derivative,

$$D^{(k)}\vec{V} = \nabla\cdot\vec{v} + \mathcal{O}(h^{k+1}),$$

must act on discrete vector fields (defined at X_F) and must return values at the discrete domain of scalar fields (X_C),

$$D : X_F \to X_C$$

its matrix representation is not a square matrix. Indeed, for m cells, $D \in \mathbb{R}^{(m+2)\times(m+1)}$. For specific discrete analogs D review [2, 1, 4, 8].

Similarly, the gradient $G^{(k)}$ when applied to the discrete projection F of the scalar field f, must approximate it with order of accuracy k the derivative, i.e.,

$$G^{(k)}F = \nabla f + \mathcal{O}(h^{k+1}),$$

and so

$$G : X_C \to X_F$$

and its matrix representation is not a square matrix. For m cells, $G \in \mathbb{R}^{(m+1)\times(m+2)}$. For specific discrete analogs G review [2, 1, 4, 8].

Both operators should approximate derivatives in 1D, and hence when applied to constant fields both should return a zero field. To enforce that, it is enough to have

$$D\,\mathbb{1} = 0, \qquad G\,\mathbb{1} = 0,$$

where both $\mathbb{1}$ and the zeros are of the appropriate sizes [2, 1, 4].

In addition, since the coordinate free definition of the divergence of a vector field at a point requires that the point should be in the interior region of smoothness of the vector field, one does not calculate the divergence at boundary points when this is not an open region. Therefore, the first and last rows of the 1D discrete analog of the divergence are zero [2, 1, 4], i.e.,

$$D_{1,:} = 0 = D_{m+2,:}.$$

One of the fundamental properties of the mimetic differences divergence D and gradient G operators is that they must satisfy the integration by part (IBP) formula

$$\int_a^b \vec{v} \cdot \nabla f \; dx + \int_a^b f\, \nabla \cdot \vec{v} \; dx = f(b)\, \vec{v}(b) - f(a)\, \vec{v}(a).$$

with high-order of accuracy. To attain that, it is needed to introduce inner product square diagonal matrix weights Q and P for the computation of both integrals by

$$h\, V^T D^T Q F = \langle DV, F \rangle_Q = \int_a^b f\, v' \; dx + \mathcal{O}(h^{k+1})$$

which implies for any vector field $\vec{v}$

$$h\, \mathbb{1} Q D V \approx \int_a^b v' \; dx,$$

and

$$h\, V^T P G F = \langle V, GF \rangle_P = \int_a^b v\, f' \; dx + \mathcal{O}(h^{k+1})$$

which implies for any scalar field f

$$h\, \mathbb{1} P G F \approx \int_a^b f' \; dx.$$

This way, in a 1D grid with m sub-intervals over $[a = x_0, b = x_{m+1}]$, the right hand side of the IBP formula is projected at a and b and the left hand side of the IBP formula is approximated by

$$h \langle GF, V \rangle_P + h \langle DV, F \rangle_Q = F_b V_b - F_a V_a,$$

where the identity is really an approximation.

If one substitutes $V = \mathbb{1}$ it can be shown that the Fundamental Theorem of Calculus is satisfied exactly for gradient derivatives. Similarly, for $F = \mathbb{1}$, the Fundamental Theorem of Calculus is satisfied for divergence derivatives. Furthermore, given D one can find that Q should satisfy

$$h\, D^T Q \mathbb{1} = (-1, 0, \cdots, 0, 1)^T_{m+1},$$

and analogously, given G one can find that P should verify

$$h\, G^T P \mathbb{1} = (-1, 0, \cdots, 0, 1)^T_{m+2}.$$

The last two identities are called the column sum property and this property is important for demonstrating preservation of physical quantities in systems of conservation laws as well as for other partial differential equations.

One interesting fact is that for $k = 2, 4, 6$ the entries of Q and P solutions of the previous linear systems are positive. Moreover, one can demonstrate that matrices Q and P are indeed quadratures [20], even though the positivity of their entries is not enforced. For $k = 8$ that is not the case. Nevertheless, it is possible to extend the stencil one neighbor and by solving a linear programming problem on the convex set defined by the linear systems and positivity conditions on the coefficients of Q and P, one can find and infinite number of those weights as solutions of different linear functionals.

2.3 The extensions of divergence and gradient operators to 2D and 3D

To facilitate the introduction of the divergence and gradient operators in several dimensions we need first to introduce the face centers and cell center points of the staggered grid in those cases. Observe that in several other mimetic methods, quantities are associated to points, edges, faces and volumes. However, this is not the case for mimetic differences, where we only consider cell centers and face centers.

Consider the Cartesian grid

$$\Pi_{i=1}^{d} [a_i, b_i] = [a_1, b_1] \times \cdots \times [a_d, b_d]$$

in d dimensions, and define

1. m_i the number of (uniform) cells along the x_i axes, $h_i = \frac{b_i - a_i}{m_i}$, $i = 1, \cdots, d$.

2. x_i-Grid: $X^i = \{x^i_j = a_i + \frac{j h_i}{2},\ j = 0, 1, \cdots, 2\, m_i\}$, $i = 1, \cdots, d$.

3. x_i-Faces: $X_F^i = \{x_j^i = a_i + jh_i,\ j = 0, 1, \cdots, m_i\}$, $i = 1, \cdots, d$.

4. x_i-Centers: $X_C^i = \{x_j^i = a_i + \frac{(2j-1)h_i}{2},\ j = 1, \cdots, m_i\}$, $i = 1, \cdots, d$.

5. x_i-Centers and Boundaries: $S^i = X_C^i \cup \{a_i, b_i\}$.

6. $X = X^1 \times \cdots \times X^d$.

7. $X_C = S^1 \times \cdots \times S^d$.

8. $X_F = \cup_{i=1}^d (X_F^1 \times \cdots \times X_F^{i-1} \times S^i \times X_F^{i+1} \times \cdots \times X_F^d)$.

In this way, the divergence D and gradient G operators are defined as linear transformations given by

$$G : X_C \to X_F, \qquad D : X_F \to X_C.$$

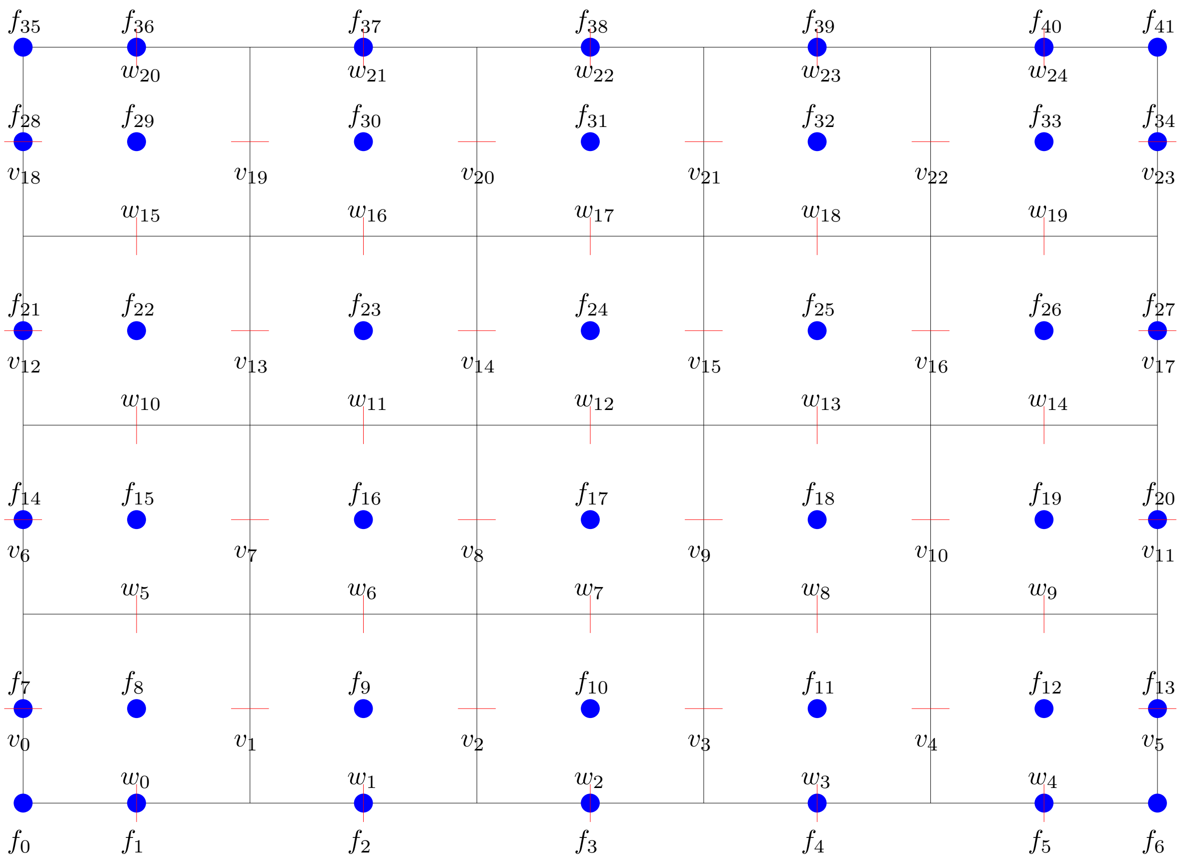

Figure 2: A staggered grid with 5 horizontal cells and 4 vertical cells.

As an example, consider in Figure 2, a (uniform) 2D staggered grid is displayed. It contains five cells along the x-axis and four cells along the y-axis.

On the one hand, scalar fields are defined at blue dots. On the other hand, the first component of vector fields is defined at horizontal red segments, while the second component of vector fields is defined at vertical red segments.

Observe that the input data of a mimetic divergence operator are located along the horizontal and vertical red segments, while its output is located at the blue dots that are not on the boundary, since the expression for the coordinate free divergence at a point P relies on a flux integral defined on shrinking neighborhoods of P that should be included in the domain, which is not possible to compute for P on the boundary of a domain.

In addition, the input data for a mimetic gradient operator is located at the blue dots, and its output is located at the horizontal red segments (for the first component of the gradient) and at the vertical red segments (for the second component of the gradient).

Extensions of these mimetic operators of order k for 2D and 3D in meshes with m cells along the x-axis, n cells along the y-axis and o cells along the z-axis, are constructed by utilizing Kronecker products. Examples are:

- The 2D divergence operators $D_{xy}^{(k)}$ of k-th order are given by

  $$D_{xy}^{(k)} = [\hat{I}_n \otimes D_x^{(k)}, \quad D_y^{(k)} \otimes \hat{I}_m],$$

  and the 3D divergence operators $D_{xyz}^{(k)}$ of k-th order are given by

  $$D_{xyz}^{(k)} = [\hat{I}_o \otimes \hat{I}_n \otimes D_x^{(k)}, \quad \hat{I}_o \otimes D_y^{(k)} \otimes \hat{I}_m, \quad D_z^{(k)} \otimes \hat{I}_n \otimes \hat{I}_m],$$

  where $D_p^{(k)}$ is the 1D divergence operator along the p-axis, and

  $$\hat{I}_q = \begin{bmatrix} 0_{1\times q} \\ I_{q\times q} \\ 0_{1\times q} \end{bmatrix},$$

  with $I_{q\times q}$ is the $q \times q$ identity matrix.

- The 2D gradient operators $G_{xy}^{(k)}$ of k-th order are given by

  $$G_{xy}^{(k)} = \begin{bmatrix} \hat{I}_n^T \otimes G_x^{(k)} \\ G_y^{(k)} \otimes \hat{I}_m^T \end{bmatrix}$$

  and the 3D gradient operators $G_{xyz}^{(k)}$ of k-th order are given by

  $$G_{xyz}^{(k)} = \begin{bmatrix} \hat{I}_o^T \otimes \hat{I}_n^T \otimes G_x^{(k)} \\ \hat{I}_o^T \otimes G_y^{(k)} \otimes \hat{I}_m^T \\ G_z^{(k)} \otimes \hat{I}_n^T \otimes \hat{I}_m^T \end{bmatrix}$$

  where $G_p^{(k)}$ is the 1D gradient operator along the p-axis.

- The 2D quadrature weight operators $\mathcal{Q}_{xy}^{(k)}$ of k-h order associated to the divergence are given by

$$\mathcal{Q}_{xy}^{(k)} = \begin{bmatrix} I_{n+2} \otimes Q_{m+2}^{(k)} & \\ & Q_{n+2}^{(k)} \otimes I_{m+2} \end{bmatrix},$$

  and the 3D quadrature weight operators $\mathcal{Q}_{xyz}^{(k)}$ of k-h order associated to the divergence are given by

$$\mathcal{Q}_{xyz}^{(k)} = \begin{bmatrix} I_{o+2} \otimes I_{n+2} \otimes Q_{m+2}^{(k)} & & \\ & I_{o+2} \otimes Q_{n+2}^{(k)} \otimes I_{m+2} & \\ & & Q_{o+2}^{(k)} \otimes I_{n+2} \otimes I_{m+2} \end{bmatrix}$$

- The 2D quadrature weight operators $\mathcal{P}_{xy}^{(k)}$ of k-h order associated to the gradient are given by

$$\mathcal{P}_{xy}^{(k)} = \begin{bmatrix} I_{n+2} \otimes P_{m+2}^{(k)} & \\ & P_{n+2}^{(k)} \otimes I_{m+2} \end{bmatrix},$$

  and the 3D gradient quadrature weight operator $\mathcal{P}_{xyz}^{(k)}$ of k-h order associated to the gradient are given by

$$\mathcal{P}_{xyz}^{(k)} = \begin{bmatrix} I_{o+2} \otimes I_{n+2} \otimes P_{m+2}^{(k)} & & \\ & I_{o+2} \otimes P_{n+2}^{(k)} \otimes I_{m+2} & \\ & & P_{o+2}^{(k)} \otimes I_{n+2} \otimes I_{m+2} \end{bmatrix}.$$

Once the mimetic difference divergence and gradient operators are defined, one expects to extend the IBP formula to various dimensions, i.e., that the 2D and 3D versions of the divergence and gradient operators, together with the actual inner product weights, will accurately approximate the Extended Gauss Divergence Theorem given by

$$\int_U \vec{v} \cdot \nabla f \; dU + \int_U f \, \nabla \cdot \vec{v} \; dU = \int_{\partial U} f \, \vec{v} \cdot \vec{n} \; dS$$

At first glance, the discrete analog of this theorem is

$$h_x h_y h_z \, \langle P_{xyz} G_{xyz} F, \vec{V} \rangle + h_x h_y h_z \, \langle Q_{xyz} D_{xyz}^T F, \vec{V} \rangle = F^T \bar{B}_{xyz} \vec{V},$$

with

$$\bar{B}_{xyz} = \begin{pmatrix} I_{o+2} \otimes I_{n+2} \otimes \bar{B}_x & & \\ & I_{o+2} \otimes \bar{B}_y \otimes I_{m+2} & \\ & & \bar{B}_z \otimes I_{n+2} \otimes I_{m+2} \end{pmatrix},$$

where $\bar{B}_p$ is the one dimensional boundary $\bar{B}$ matrix along the p-axis given by

$$\bar{B} = \begin{pmatrix} -1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{pmatrix}.$$

However, when tracking where the input data for the different divergence and gradient operators one realized the need of utilizing interpolations to move data from/to cell centers to/from face centers.

Formally, i.e., without considering the compulsory interpolations one can establish relationships between the divergence and the gradient with Q and P, respectively. They are given by

$$h_x h_y h_z \, Q_{xyz} D_{xyz}^T \mathbb{1} = \mathbb{1} \otimes \mathbb{1} \otimes b_{m+1} + \mathbb{1} \otimes b_{n+1} \otimes \mathbb{1} + b_{o+1} \otimes \mathbb{1} \otimes \mathbb{1},$$

and

$$h_x h_y h_z \, G_{xyz}^T P_{xyz} \mathbb{1} = \mathbb{1} \otimes \mathbb{1} \otimes b_{m+2} + \mathbb{1} \otimes b_{n+2} \otimes \mathbb{1} + b_{o+2} \otimes \mathbb{1} \otimes \mathbb{1}.$$

The different 1D mimetic difference interpolation operators can be found in [8]. They can be extended to 2D and 3D in the following form:

- The 2D interpolation operators from cell centers to face centers $(I_D)_{xy}^{(k)}$ of k-th order are given by

$$(I_D)_{xy}^{(k)} = \begin{bmatrix} \hat{I}_n^T \otimes (I_D)_x^{(k)} & \\ & (I_D)_y^{(k)} \otimes \hat{I}_m^T \end{bmatrix},$$

  and the 3D interpolation operators from cell centers to face centers $(I_D)_{xyz}^{(k)}$ of k-th order are given by

$$(I_D)_{xyz}^{(k)} = \begin{bmatrix} \hat{I}_o^T \otimes \hat{I}_n^T \otimes (I_D)_x^{(k)} & & \\ & \hat{I}_o^T \otimes (I_D)_y^{(k)} \otimes \hat{I}_m^T & \\ & & (I_D)_z^{(k)} \otimes \hat{I}_n^T \otimes \hat{I}_m^T \end{bmatrix}$$

  where $(I_D)_p^{(k)}$ is the corresponding 1D interpolation operator along the p-axis.

- The 2D interpolation operators from face centers to cell centers $(I_G)_{xy}^{(k)}$ of k-th order are given by

$$(I_G)_{xy}^{(k)} = \begin{bmatrix} \hat{I}_n \otimes (I_G)_x^{(k)} & \\ & (I_G)_y^{(k)} \otimes \hat{I}_m \end{bmatrix},$$

and the 3D interpolation operators from face centers to cell centers $(I_G)^{(k)}_{xyz}$ of k-th order are

$$(I_G)^{(k)}_{xyz} = \begin{bmatrix} \hat{I}_o \otimes \hat{I}_n \otimes (I_G)^{(k)}_x & & \\ & \hat{I}_o \otimes (I_G)^{(k)}_y \otimes \hat{I}_m & \\ & & (I_G)^{(k)}_z \otimes \hat{I}_n \otimes \hat{I}_m \end{bmatrix},$$

where $(I_G)^{(k)}_p$ is the corresponding 1D interpolation operator along the p-axis.

The mentioned mimetic difference operators have been successfully used to demonstrate preservation of mass, momenta, and energy for general systems of conservation laws [10].

3 The curl operator calculated at cell faces

These 3D curl discrete analogs are represented by square matrices where the number of rows and columns is given by the number of cell face centers. They take data from face centers and return values at face centers. We will show that this way of constructing curl operators, following the ideas presented in [8], uses many of the properties established for the different operators in the previous section for mimetic differences operators[2], but requires many interpolations, and it is restricted to the case where the number of cells is the same in each direction. On the other hand, exclusively from these properties without specifying the actual coefficients of D or G, discrete analogs of curl vector calculus identities are easily shown.

Although these curl operators are only defined in 3D, there have been some previous attempts to construct discrete analogs of mimetic differences of 2D and 3D curl operators, in terms of 2D divergence operators [1, 19], in the context of the Castillo-Grone approach [2, 19]. However, these ideas can be applied to the Corbino-Castillo method. We discuss that in the next section.

3.1 Construction of the 3D curl based on the gradient operator

For a Cartesian grid with m, n, o cells in the x, y, z-axes, respectively, the 3D curl mimetic operators are constructed by utilizing Kronecker products $\otimes$.

Given a 3D vector field $\vec{v} = (v_1, v_2, v_3)^T$, the curl of $\vec{v}$ is defined by

$$\nabla \times \vec{v} = \left[\frac{\partial v_3}{\partial y} - \frac{\partial v_2}{\partial z}, \frac{\partial v_1}{\partial z} - \frac{\partial v_3}{\partial x}, \frac{\partial v_2}{\partial x} - \frac{\partial v_1}{\partial y}\right]^T.$$

To demonstrate some of the vector calculus properties, we will rely on the following characterization. If $\vec{V}$ is a discrete version of $\vec{v}$ with V_1, V_2, V_3 being the projections of the

[2]No matter if one is using Corbino-Castillo, Castillo-Grone, or any other mimetic difference representation of the operators.

v_1, v_2, v_3 functions onto the staggered grid, where

$$V_1 : X_F^1 \times X_C^2 \times X_C^3 \to \mathbb{R}, \qquad V_2 : X_C^1 \times X_F^2 \times X_C^3 \to \mathbb{R}, \qquad V_3 : X_C^1 \times X_C^2 \times X_F^3 \to \mathbb{R},$$

then the discrete analog of the curl operator $C_{xyz}^{(k)}$ is given by

$$C_{xyz}^{(k)} = (I_D)_{xyz}^{(k)} \begin{bmatrix} & -(I_G)_{xyz,3}^{(k)} G_{xyz,3} & (I_G)_{xyz,2}^{(k)} G_{xyz,2} \\ (I_G)_{xyz,3}^{(k)} G_{xyz,3} & & -(I_G)_{xyz,1}^{(k)} G_{xyz,1} \\ -(I_G)_{xyz,2}^{(k)} G_{xyz,2} & (I_G)_{xyz,1}^{(k)} G_{xyz,1} & \end{bmatrix} (I_G)_{xyz}^{(k)},$$

where $G_p^{(k)}$ is the 1D gradient operator along the p-axis, $(I_D)_{xyz}^{(k)}$ is the interpolation from cell centers to face centers, $(I_G)_{xyz}^{(k)}$ the interpolation from face centers to cell centers of k-th order [7].

3.2 Alternative versions of the 3D Curl based on gradient operators

By utilizing the definition of the interpolator operators and properties of the Kronecker products, one gets that the curl operator defined utilizing gradient operators for computing partial derivatives, $C_{xyz}^{(k)}$ holds the following identities

$$= (I_D)_{xyz}^{(k)} \begin{bmatrix} & -(I_G)_{xyz,3}^{(k)} G_{xyz,3} & (I_G)_{xyz,2}^{(k)} G_{xyz,2} \\ (I_G)_{xyz,3}^{(k)} G_{xyz,3} & & -(I_G)_{xyz,1}^{(k)} G_{xyz,1} \\ -(I_G)_{xyz,2}^{(k)} G_{xyz,2} & (I_G)_{xyz,1}^{(k)} G_{xyz,1} & \end{bmatrix} (I_G)_{xyz}^{(k)} \tag{1}$$

$$= (I_D)_{xyz}^{(k)} \begin{bmatrix} & -(I_G)_{xyz,3}^{(k)} G_{xyz,3} (I_G)_{xyz,2}^{(k)} & (I_G)_{xyz,2}^{(k)} G_{xyz,2} (I_G)_{xyz,3}^{(k)} \\ (I_G)_{xyz,3}^{(k)} G_{xyz,3} (I_G)_{xyz,1}^{(k)} & & -(I_G)_{xyz,1}^{(k)} G_{xyz,1} (I_G)_{xyz,3}^{(k)} \\ -(I_G)_{xyz,2}^{(k)} G_{xyz,2} (I_G)_{xyz,1}^{(k)} & (I_G)_{xyz,1}^{(k)} G_{xyz,1} (I_G)_{xyz,2}^{(k)} & \end{bmatrix} \tag{2}$$

$$= (I_D)_{xyz}^{(k)} \begin{bmatrix} & -((I_G)_z^{(k)} G_z^{(k)} \otimes \hat{I}_n^T \otimes \hat{I}_m^T) & (\hat{I}_o^T \otimes (I_G)_y^{(k)} G_y^{(k)} \otimes \hat{I}_m^T) \\ ((I_G)_z^{(k)} G_z^{(k)} \otimes \hat{I}_n^T \otimes \hat{I}_m^T) & & -(\hat{I}_o^T \otimes \hat{I}_n^T \otimes (I_G)_x^{(k)} G_x^{(k)}) \\ -(\hat{I}_o^T \otimes (I_G)_y^{(k)} G_y^{(k)} \otimes \hat{I}_m^T) & (\hat{I}_o^T \otimes \hat{I}_n^T \otimes (I_G)_x^{(k)} G_x^{(k)}) & \end{bmatrix} (I_G)_{xyz}^{(k)} \tag{3}$$

$$= (I_D)_{xyz}^{(k)} \begin{bmatrix} & -((I_G)_z^{(k)} G_z^{(k)} \otimes \hat{I}_n^T (I_G)_y^{(k)} \otimes \hat{I}_m^T) & (\hat{I}_o^T (I_G)_z^{(k)} \otimes (I_G)_y^{(k)} G_y^{(k)} \otimes \hat{I}_m^T) \\ ((I_G)_z^{(k)} G_z^{(k)} \otimes \hat{I}_n^T \otimes \hat{I}_m^T (I_G)_x^{(k)}) & & -(\hat{I}_o^T (I_G)_z^{(k)} \otimes \hat{I}_n^T \otimes (I_G)_x^{(k)} G_x^{(k)}) \\ -(\hat{I}_o^T \otimes (I_G)_y^{(k)} G_y^{(k)} \otimes \hat{I}_m^T (I_G)_x^{(k)}) & (\hat{I}_o^T \otimes \hat{I}_n^T (I_G)_y^{(k)} \otimes (I_G)_x^{(k)} G_x^{(k)}) & \end{bmatrix} \tag{4}$$

$$= \begin{bmatrix} & -(I_G)_z^{(k)} G_z^{(k)} \otimes \hat{I}_n^T (I_G)_y^{(k)} \otimes (I_D)_x^{(k)} \hat{I}_m^T & \hat{I}_o^T (I_G)_z^{(k)} \otimes (I_G)_y^{(k)} G_y^{(k)} \otimes (I_D)_x^{(k)} \hat{I}_m^T \\ (I_G)_z^{(k)} G_z^{(k)} \otimes (I_D)_y^{(k)} \hat{I}_n^T \otimes \hat{I}_m^T (I_G)_x^{(k)} & & -\hat{I}_o^T (I_G)_z^{(k)} \otimes (I_D)_y^{(k)} \hat{I}_n^T \otimes (I_G)_x^{(k)} G_x^{(k)} \\ -(I_D)_z^{(k)} \hat{I}_o^T \otimes (I_G)_y^{(k)} G_y^{(k)} \otimes \hat{I}_m^T (I_G)_x^{(k)} & (I_D)_z^{(k)} \hat{I}_o^T \otimes \hat{I}_n^T (I_G)_y^{(k)} \otimes (I_G)_x^{(k)} G_x^{(k)} & \end{bmatrix} \tag{5}$$

3.3 Construction of the 3D curl based on the divergence operator

Observe that one could equivalently define the curl operator in terms of divergence operators to approach the first-order partial derivatives in the curl operators as

$$C_{xyz} = (I_D)^{(k)}_{xyz} \begin{bmatrix} & -\bar{D}_{xyz,3}(I_D)^{(k)}_{xyz,3} & \bar{D}_{xyz,2}(I_D)^{(k)}_{xyz,2} \\ \bar{D}_{xyz,3}(I_D)^{(k)}_{xyz,3} & & -\bar{D}_{xyz,1}(I_D)^{(k)}_{xyz,1} \\ -\bar{D}_{xyz,2}(I_D)^{(k)}_{xyz,2} & \bar{D}_{xyz,1}(I_G)^{(k)}_{xyz,1} & \end{bmatrix} (I_G)^{(k)}_{xyz}. \quad (6)$$

Similar identities to (2)-(5) can be obtained for the 3D curl operator that uses the 3D divergence. For example, the following is similar to (2),

$$C_{xyz} = \begin{bmatrix} & -(I_D)^{(k)}_{xyz,1}\bar{D}_{xyz,3}(I_D)^{(k)}_{xyz,3} & (I_D)^{(k)}_{xyz,1}\bar{D}_{xyz,2}(I_D)^{(k)}_{xyz,2} \\ (I_D)^{(k)}_{xyz,2}\bar{D}_{xyz,3}(I_D)^{(k)}_{xyz,3} & & -(I_D)^{(k)}_{xyz,2}\bar{D}_{xyz,1}(I_D)^{(k)}_{xyz,1} \\ -(I_D)^{(k)}_{xyz,3}\bar{D}_{xyz,2}(I_D)^{(k)}_{xyz,2} & (I_D)^{(k)}_{xyz,3}\bar{D}_{xyz,1}(I_G)^{(k)}_{xyz,1} & \end{bmatrix} (I_G)^{(k)}_{xyz}. \quad (7)$$

3.4 Some vector calculus identities

Here, mimetic discrete analogs of some of the Vector Calculus identities are derived. In some cases, algebraic relationships obtained from the operator formulas found in the previous sections are leveraged to simplify the work.

Given the 3D scalar fields f, g and the 3D vector fields $\vec{v}, \vec{w}$, the discrete analogs of the scalar fields are $F, G : S^1 \times S^2 \times S^3 \to \mathbb{R}$, and the discrete analogs of the vector fields are $\vec{V}, \vec{W}$ with $\vec{V} = (V_1, V_2, V_3), \vec{W} = (W_1, W_2, W_3)$, where $V_1, W_1 : X^1_F \times S^2 \times S^3 \to \mathbb{R}, V_2, W_2 : S^1 \times X^2_F \times S^3 \to \mathbb{R}, V_3, W_3 : S^1 \times S^2 \times X^3_F \to \mathbb{R}$.

The list of curl vector calculus identities is the following.

1. Curl of the gradient is zero, or
$$\nabla \times \nabla f = \vec{0}.$$
For any discrete version F of the scalar field f, the discrete analog of this property is $C_{xyz}G_{xyz}F = \vec{0}$. One needs to verify that $C_{xyz}G_{xyz}$ is the zero matrix. For, from (2), $C_{xyz}G_{xyz}$ is

$$\begin{aligned}
&= (I_D)_{xyz} \begin{bmatrix} & -((I_G)_{xyz,3}G_{xyz,3})(I_G)_{xyz,2} & ((I_G)_{xyz,2}G_{xyz,2})(I_G)_{xyz,3} \\ ((I_G)_{xyz,3}G_{xyz,3})(I_G)_{xyz,1} & & -((I_G)_{xyz,1}G_{xyz,1})(I_G)_{xyz,3} \\ -((I_G)_{xyz,2}G_{xyz,2})(I_G)_{xyz,1} & ((I_G)_{xyz,1}G_{xyz,1})(I_G)_{xyz,2} & \end{bmatrix} \begin{bmatrix} G_{xyz,1} \\ G_{xyz,2} \\ G_{xyz,3} \end{bmatrix} \\
&= (I_D)_{xyz} \begin{bmatrix} -((I_G)_{xyz,3}G_{xyz,3})((I_G)_{xyz,2}G_{xyz,2}) + ((I_G)_{xyz,2}G_{xyz,2})((I_G)_{xyz,3}G_{xyz,3}) \\ ((I_G)_{xyz,3}G_{xyz,3})((I_G)_{xyz,1}G_{xyz,1}) - ((I_G)_{xyz,1}G_{xyz,1})((I_G)_{xyz,3}G_{xyz,3}) \\ -((I_G)_{xyz,2}G_{xyz,2})((I_G)_{xyz,1}G_{xyz,1}) + ((I_G)_{xyz,1}G_{xyz,1})((I_G)_{xyz,2}G_{xyz,2}) \end{bmatrix} = (I_D)_{xyz} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.
\end{aligned}$$

The last identity follows because $(I_G)_pG_p, p = x, y, z$ are the same when $m = n = o$.

2. Divergence of curl is zero, or
$$\nabla \cdot \nabla \times \vec{v} = 0.$$
For any discrete version $\vec{V}$ of the vector field $\vec{v}$, the discrete analog of this property is $\bar{D}_{xyz} C_{xyz} \vec{V} = 0$, since the curl is not zero on the boundary and the expanded divergence is required. One needs to verify that $\bar{D}_{xyz} C_{xyz} = 0$. From (7), $\bar{D}_{xyz} C_{xyz}$ is equal to
$$\begin{bmatrix} \bar{D}_{xyz,2}(I_D)_{xyz,2}\bar{D}_{xyz,3}(I_D)_{xyz,3} - \bar{D}_{xyz,3}(I_D)_{xyz,3}\bar{D}_{xyz,2}(I_D)_{xyz,2} \\ \bar{D}_{xyz,3}(I_D)_{xyz,3}\bar{D}_{xyz,1}(I_G)_{xyz,1} - \bar{D}_{xyz,1}(I_D)_{xyz,1}\bar{D}_{xyz,3}(I_D)_{xyz,3} \\ \bar{D}_{xyz,1}(I_D)_{xyz,1}\bar{D}_{xyz,2}(I_D)_{xyz,2} - \bar{D}_{xyz,2}(I_D)_{xyz,2}\bar{D}_{xyz,1}(I_D)_{xyz,1} \end{bmatrix}^T (I_G)_{xyz} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T (I_G)_{xyz} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}^T.$$
The last identity follows because of $\bar{D}_p(I_D)_p$ are the same when $m = n = o$.

3. The next three properties require the definition of a discretization of the cross product for vector fields, which is an operation that requires two arguments. A possible definition for discrete vector fields $\vec{V} = (V_1, V_2, V_3)^T, \vec{W} = (W_1, W_2, W_3)^T)$, that assumes $m = n = o$ is
$$\vec{V} \times \vec{W} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \times \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix} = I_D \begin{bmatrix} & -I_{G,3}V_3 & I_{G,2}V_2 \\ I_{G,3}V_3 & & -I_{G,1}V_1 \\ -I_{G,2}V_2 & I_{G,1}V_1 & \end{bmatrix} I_G \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix}.$$

   (a) Product rule for a multiplication by a scalar of the curl, or
$$\nabla \times (f\,\vec{v}) = \nabla f \times \vec{v} + f\,\nabla \times \vec{v}.$$
   The discrete analog of this property for this discrete cross product is
$$\begin{aligned} C_{xyz}(I_D)_{xyz} \begin{bmatrix} F \circ V_1 \\ F \circ V_2 \\ F \circ V_3 \end{bmatrix} &= (I_D)_{xyz} \begin{bmatrix} & -I_{G,3}G_{xyz,3}F & I_{G,2}G_{xyz,2}F \\ I_{G,3}G_{xyz,3}F & & -I_{G,1}G_{xyz,1}F \\ -I_{G,2}G_{xyz,2}F & I_{G,1}G_{xyz,1}F & \end{bmatrix} (I_G)_{xyz} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \\ &+ ((I_D)_{xyz}F) \circ \left[C_{xyz} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} \right]. \end{aligned}$$

   (b) The product rule
$$\nabla \cdot (\vec{v} \times \vec{w}) = \vec{w} \cdot (\nabla \times \vec{v}) - \vec{v} \cdot (\nabla \times \vec{w}).$$
   The analog of this property for this discrete cross product is
$$D_{xyz}(I_D)_{xyz} \begin{bmatrix} & -(I_G)_{xyz,3}V_3 & (I_G)_{xyz,2}V_2 \\ (I_G)_{xyz,3}V_3 & & -(I_G)_{xyz,1}V_1 \\ -(I_G)_{xyz,2}V_2 & (I_G)_{xyz,1}V_1 & \end{bmatrix} (I_G)_{xyz} \begin{bmatrix} W_1 \\ W_2 \\ W_3 \end{bmatrix} = \vec{W}^T \cdot C_{xyz}\vec{V} - \vec{V}^T \cdot C_{xyz}\vec{W}.$$

(c) The product rule

$$\nabla \cdot (\nabla f \times \nabla g) = 0.$$

The discrete analog of this property for this discrete cross product is

$$D_{xyz}(I_D)_{xyz} \begin{bmatrix} & -(I_G)_{xyz,3}G_{xyz,3}F & (I_G)_{xyz,2}G_{xyz,2}F \\ (I_G)_{xyz,3}G_{xyz,3}F & & -(I_G)_{xyz,1}G_{xyz,1}F \\ -(I_G)_{xyz,2}G_{xyz,2}F & (I_G)_{xyz,1}G_{xyz,1}F & \end{bmatrix} (I_G)_{xyz}G_{xyz}G = 0.$$

If the discrete analog of the product rule $\nabla \cdot (\vec{v} \times \vec{w}) = \vec{w} \cdot (\nabla \times \vec{v}) - \vec{v} \cdot (\nabla \times \vec{w})$ holds, then by substituting $\vec{v}, \vec{w}$ by $\nabla f, \nabla g$, respectively, and by the discrete analog of $\nabla \times \nabla f = \vec{0}$, then the discrete analog of $\nabla \cdot (\nabla f \times \nabla g) = 0$ will hold.

One can observe that the two discrete analogs of the 3D curl operator introduced in this section, approximate the partial derivatives of the curl utilizing the 3D divergence or the 3D gradient operators, have a common problem. They use too many interpolations, and are restricted to have the same number of cells along each axis, and most important, because of the former, they almost surely will not preserve physical quantities for systems of conservation laws, as is the case of Maxwell's equations.

4 The 2D and 3D curl operators using 2D divergence

In this section, we propose a 2D and a 3D curl operators whose discrete analog matrix representation is independent of interpolations, and are not restricted to have the same number of cells. This 3D construction repeats for each of the curl components the approximation used for the 2D curl, the latter based on the 2D divergence [1, 17, 19], and for that reason the 2D curl is considered first. However, there are several possibilities for the 2D curl discrete analogs depending on whether or not all three components of the 2D curl are computed. Furthermore, this approach is based on the curl coordinate free formulation or alternatively on a analogous of the ∇ operator in the context of the exterior derivative of exterior calculus. A significant difference exists between the proposed 2D curl discrete analog (which extends to the 3D curl) and that of Runyan's approach [17], and also with respect to the abstract ∇ approach proposed by Castillo and Miranda [1].

It will turn out that the 3D curl discrete analog matrix representation we propose will be composed of a three by six block of matrices, and this matrix will operate on two copies of the components of the vector field calculated at different cell face centers.

In what follows, we have some subsections explaining Runyan's approach, and Castillo and Miranda approach to better distinguish the one we propose from those two.

4.1 Runyan's approach for the 2D curl operator

The 3D curl can be defined in several ways. One possibility is to define the curl of a vector field $\vec{E}$ at a point P implicitly through its components along various axes represented

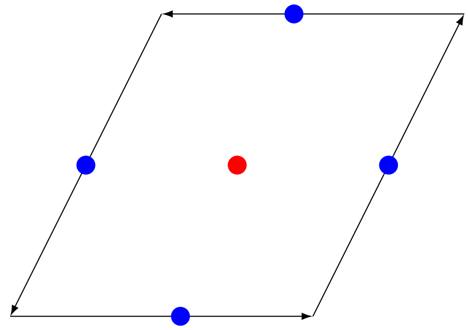

Figure 3: The coordinate free definition of the 2D curl in terms of a line integral around a closed loop surrounding a plane cell. The curl calculation at the red cell center uses data from blue 2D face centers. This plot is for h fixed. Keep in mind the $h \to 0$ in the 2D coordinate free curl definition.

by their unit vectors $\vec{e}$ passing through the point, that is, the limiting value of a closed line integral in a plane perpendicular to $\vec{e}$ divided by the area it encloses, as the path of integration γ shrinks indefinitely around P, i.e.,

$$(\nabla \times \vec{E})(P) \cdot \vec{e} = \lim_{A \to 0} \frac{1}{A} \oint_{\gamma} \vec{E} \cdot d\vec{l}.$$

This coordinate free component-wise definition is convenient for the 2D curl. Notice that in 2D, the $(\nabla \times \vec{E})(P)$ counter-clockwise loop γ can be restricted to a square with P as its center (see Figure 3). Using Figure 3 to approximate this integral around square loops γ, which surrounds a cell of area h^2, we get for the discrete analog C_z of the third component of the curl of the vector field $\vec{E} = (E_1, E_2)$ at the center cell with coordinates $P = (i + 1/2, j + 1/2)$ the following

$$\begin{aligned} C_z(\vec{E})(P) &\approx \frac{1}{h^2}(hE_1(i+1/2,j) + hE_2(i+1,j+1/2) - hE_1(i+1/2,j+1) - hE_2(i,j+1/2)) \\ &= \frac{1}{h}([E_2(i+1,j+1/2) - E_2(i,j+1/2))] - [E_1(i+1/2,j+1) - E_1(i+1/2,j)], \end{aligned}$$

which turns out to be a second-order approximation.

This 2D curl approximation for points at the red cell center uses data from blue 2D face centers. There is an apparent inconsistency in this approach, since the curl is supposed to take data from face centers and return values at cell face centers (blue dots in the plot). But this is not the case, because the formula is for each scalar field component of the curl (which are located at cell centers) and not for the whole curl. Yet, this needs to be investigated further.

This apparent inconsistency can also be perceived algebraically. Consider Cartesian coordinates. Let $P = (x, y)$. Suppose $\vec{E}(x, y) = (E_1(x, y), E_2(x, y))$ is a 2D vector

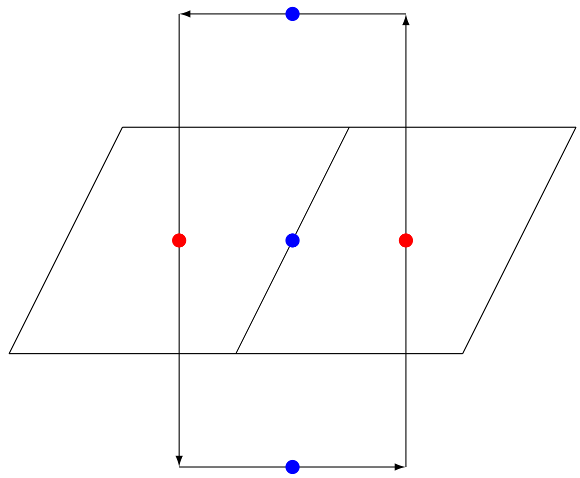

Figure 4: The coordinate free definition of the 2D curl in terms of a line integral around a closed loop surrounding a plane cell. The curl calculation at the blue face center uses data from red 2D red cell centers and the black face centers. The plot is for a fixed h. Keep in mind that $h \to 0$ in the 2D coordinate free curl definition.

field, where E_1, E_2 are scalar fields. To compute its curl, embed $\vec{E}$ in 3D by $\vec{E}(x,y) = (E_1(x,y), E_2(x,y), 0)$. The 3D curl is given by

$$\nabla \vec{E}(x,y) = \left(-\frac{\partial E_2}{\partial z}(x,y), \frac{\partial E_1}{\partial z}(x,y), \frac{\partial E_2}{\partial x}(x,y) - \frac{\partial E_1}{\partial y}(x,y)\right) = \left(0, 0, \frac{\partial E_2}{\partial x}(x,y) - \frac{\partial E_1}{\partial y}(x,y)\right).$$

Therefore, the only non-zero 2D curl component is the last one. If one wants to add the 2D curl vector in Figure 3, one needs to add an orthogonal vector to the plane of the figure at the red point P [17]. But if the computation is restricted to the curl components (which are scalar fields and as such their data are located at cell centers), then it is not necessary to draw the 2D curl vector at the red point P in Figure 3.

With this understanding, we explore the second-order calculation of the 2D curl at the blue 2D face center points in Figure 3. For, first consider the right vertical face center in Figure 3. This is better depicted in Figure 4. In this case, the input data, for the 2D curl computation at the blue dot on the plane, are partially composed of blue (out of the plane) face centers and red (on the plane) red cell centers. Again, there is an apparent inconsistency with the fact that the location of the input and output data is not exclusively at face centers, which is harder to explain.

4.2 The generalized ∇ operator formulation of Castillo and Miranda

In [1], Castillo and Miranda introduced the generalized coordinate-free formulation of the ∇ operator as

$$\nabla()\phi = \lim_{\text{diam}(\Omega)\to 0} \frac{1}{\text{vol}(\Omega)} \int_\sigma \vec{n}()\phi \, d\sigma,$$

where Ω taken as a 3D cell with center P, σ the six-face boundary surface of Ω, vol(Ω) the volume of Ω, and diam(Ω) its diameter. In this formula, one must replace () by $\cdot$, $\times$, or nothing, depending on the type of field ϕ being operated upon. This formula synthesizes the coordinate-free definition of the divergence, curl, and gradient operators, as they can be seen as concrete versions of the differential geometry exterior derivative acting on skew-symmetric d-forms ($d = 1, 2, 3$), respectively.

Taking the inner product of $(\nabla \times \vec{E})(P)$ with the unit axis vector $\vec{e}$, one can compute each of the three scalar components of the curl by

$$(\nabla \times \vec{E})(P) \cdot \vec{e} = \lim_{\text{diam}(\Omega)\to 0} \frac{1}{\text{vol}(\Omega)} \int_\sigma (\vec{n} \times \vec{E}) \cdot \vec{e} \, d\sigma = \lim_{\text{diam}(\Omega)\to 0} \frac{1}{\text{vol}(\Omega)} \int_\sigma (\vec{E} \times \vec{e}) \cdot (\vec{n} \, d\sigma),$$

where the transitivity of the triple product has been used in the last identity.

Using the definition of the generalized ∇ operator for the inner product, one gets

$$(\nabla \cdot \phi)(P) = \lim_{\text{diam}(\Omega)\to 0} \frac{1}{\text{vol}(\Omega)} \int_\sigma \phi \cdot (\vec{n} \, d\sigma),$$

where the commutativity of the inner product has been used.

Looking at the expression found for $(\nabla \times \vec{E})(P) \cdot \vec{e}$ and the relationship just found for the inner product, notice that

$$(\nabla \times \vec{E})(P) \cdot \vec{e} = \lim_{\text{diam}(\Omega)\to 0} \frac{1}{\text{vol}(\Omega)} \int_\sigma (\vec{E} \times \vec{e}) \cdot (\vec{n} \, d\sigma) = (\nabla \cdot (\vec{E} \times \vec{e}))(P),$$

and as said in [1], "it is sufficient to have 2D mimetic divergence discrete operators with the desired even order of accuracy, coupled with adequate 2D or 3D staggering grids, in order to obtain the single scalar component of the 2D curl, or the three scalar components of the 3D curl". Therefore, the curl operators reduce, in computational terms, to a 2D divergence[3].

[3] A similar idea can be used for the gradient of a scalar field ϕ. In this case,

$$(\nabla\phi)(P) \cdot \vec{e} = \lim_{\text{diam}(\Omega)\to 0} \frac{1}{\text{vol}(\Omega)} \int_\sigma (\phi \, \vec{n}) \cdot \vec{e} \, d\sigma = \lim_{\text{diam}(\Omega)\to 0} \frac{1}{\text{vol}(\Omega)} \int_\sigma \vec{n} \cdot (\phi \, \vec{e}) \, d\sigma = (\nabla \cdot (\phi\vec{e}))(P),$$

in computational terms, the gradient reduces to a 1D divergence. Obviously, the divergence of a vector field reduces to a 3D divergence calculation. Again, the connection between the general ∇ operator and exterior derivative of differential geometry produce these formulas using the 1D, 2D, 3D divergence operators.

In [1], the 2D curl is presented as a preparation for the 3D curl. For, given the 2D vector field $\vec{E} = (E_1, E_2)$, an auxiliary vector field $\vec{E}^* = \vec{E} \times \vec{k}$, $\vec{k} = (0,0,1)$, and hence $\vec{E}^* = (E_1^*, E_2^*) = (E_2, -E_1)$. Since

$$\text{curl}(\vec{E}) = \left(\frac{\partial E_2}{\partial x} - \frac{\partial E_1}{\partial y}\right)\vec{k},$$

then

$$\text{curl}(\vec{E}) \cdot \vec{k} = \frac{\partial E_1^*}{\partial x} + \frac{\partial E_2^*}{\partial y} = \text{div}(\vec{E}^*),$$

so only a 2D divergence calculation at a point is required for the single scalar component of the 2D curl. Since $\text{curl}(\vec{E})(x,y) = \text{div}(\vec{E}^*)\vec{k}$ then, according to the next three formulas which correspond to formulas (4.61), (4.62), and (4.63) of [1], respectively

$$\text{curl}(\vec{E})(x,y) \approx \text{CURL}(\vec{E})(i, j+1/2) = \text{DIV}(\vec{E}^*)(i, j+1/2)$$

where

$$\begin{aligned} \text{CURL}(\vec{E})(i,j+1/2)\cdot\vec{k} &= \frac{1}{h}\big(\,(E_2(x_{i+1/2}, y_{j+1/2}) - E_2(x_{i-1/2}, y_{j+1/2}) \\ &- \;(E_1(x_i, y_{j+1/2}) - E_1(x_i, y_{j-1/2})))\,\big) \end{aligned}$$

and

$$\text{DIV}(\vec{E}^*) = \frac{1}{h}\left(E_1^*(i+1/2, j+1/2) - E_1^*(i-1/2, j+1/2) + E_2^*(i, j+1) - E_2^*(i,j)\right).$$

The approximation in (4.61) shifts the second coordinate for the approximation when changing from curl to CURL, and more importantly, both formulas (4.62) and (4.63) use cell center information.

In the next subsection, following the ideas of [17] and [1], we propose an approach for computing the 2D and 3D curls that explicitly use face center information.

Notice, that in mimetic differences, by construction the 2D divergence operators are set to zero at the domain boundary. Since the 3D curl operators are intended to be calculated at cell face centers, and since the 2D divergence operators are calculated at cell centers, interpolations from cell centers to cell face centers will be necessary. One can foresee at least potential two problems.

1. There are cell face centers on the boundary. Interpolation from cell centers to face centers use values at cell centers. Cell centers data also includes data on the boundary, in particular at the corners of the domain, and at those points at the boundary the divergence is zero. For a face center near a corner, the zero value of any cell data on the boundary (set by the 2D divergence), in particular the zero at the corner, will affect the non-zero divergence values coming from the interior near cell centers of the domain that are used for interpolation of cell centers to cell face centers. This will create a definite bias of the interpolated values at the cell face centers.

2. In addition, the 3D curl discrete analog calculates the curl at cell centers and also at boundary cell face centers. Indeed, the output of the curl operator will be in both meshes of the staggered grid. All other mimetic difference operators have their input data in one of the meshes and their output on the other of the meshes. This allows to use interpolations to move data from one of the meshes to the other when complex differential operators are composed of gradient, divergence, Laplacian, etc. But, having output data in both meshes will impede to locate input data in one of the meshes, which will be required, for example, when iterating discrete time derivatives of evolution operators.

If one would like to compute the curl at boundary cell face centers, one may think about extending the divergence used in this curl formulation to be non-zero at the boundary. But that is not needed[4]. These potential problems can be fixed by the fact that utilizing the divergence discrete analogs that have zero boundary data will imply necessarily zero values for the curl at boundary cell face centers. Assuming that one constructs a curl discrete analog that necessarily is set to zero at boundary cell face centers will be enough to avoid computing output in two different meshes. This fact will be useful when trying to prove mass, momenta, and energy preservation, for differential operators that contain curls[5].

4.3 Our 2D curl approach based on the divergence operator

In Figure 5, the red $2h$ side square loop γ is used to approximate $\frac{1}{A}\oint_\gamma \vec{E}\cdot\, d\vec{l}$. The surrounded area is $4h^2$. The discrete analog C_z of the third component of the curl of the vector field $\vec{E}=(E_1,E_2)$ at the face center that is located in the middle of the blue points, with coordinates $P=(i,j+1/2)$ is the following

$$\begin{aligned}
C_z(\vec{E})(P) \quad &\approx \quad \frac{1}{4h^2}(2hE_1(i,j-1/2)+2hE_2(i+1,j+1/2)-2hE_1(i,j+3/2)-2hE_2(i-1,j+1/2)) \\
&= \quad \frac{1}{2h}([E_2(i+1,j+1/2)-E_2(i-1,j+1/2))]-[E_1(i,j+3/2)-E_1(i,j-1/2)] \\
&\to_{h\to 0} \quad \frac{\partial E_2}{\partial x}(i,j+1/2)-\frac{\partial E_1}{\partial y}(i,j+1/2)=\nabla\cdot(E_2,-E_1)(i,j+1/2),
\end{aligned}$$

which turns out to be a second-order approximation of the divergence at the face point. However, $P=(i,j+1/2)$ is a face center and mimetic divergences are always computed

[4] There is no guarantee that those extended divergence operators will do the work. For, one could extend first the 1D D (which has zero first and last rows, and satisfy the column sum property derived from the enforcement of a high-order approximation of the IBP formula), with the introduction of non-zero values in the first and last rows, but that will break the column sum property, due to the reflectivity with respect to the boundaries of the coefficients that will be added. Consistently, one can recompute Q accordingly to satisfy the IBP formula using the modified D but there is no guaranteed that the new Q coefficients obtained will lose the preservation of important quantities for systems of conservation laws

[5] As examples of mathematically demonstrating preservation of quantities for system of conservation laws see [10].

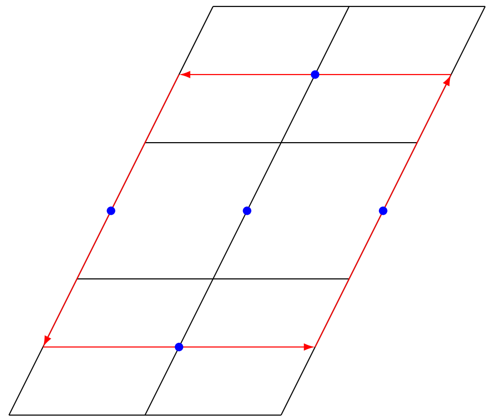

Figure 5: The coordinate free definition of the 2D curl in terms of a line integral around a red closed loop surrounding a face center. The curl calculation at the blue face center uses data exclusively from face centers. Cells are in black. The plot is for a fixed h. Keep in mind that $h \to 0$ in the 2D coordinate free curl definition.

at cell centers using data at face centers. To calculate the divergence at P, it should be a cell center. Hence, for vertical face centers coordinates of the form $(i, j + 1/2)$, we need to work with an additional staggered grid with axes coordinates $\bar{i}$ and $\bar{j}$, where P has coordinates $(\bar{i} + 1/2, \bar{j} + 1/2)$. Therefore, the coordinates of P in both systems are related by $(i, j + 1/2) = (\bar{i} + 1/2, \bar{j} + 1/2)$, i.e., P is a center in the new coordinate system. The additional staggered grid can be seen as the original coordinate system shifted to the left, half a unit. This way, all cell centers of the original system with coordinates $(i \pm 1/2, j \pm 1/2)$ are located in the new system at vertical face centers with coordinates $(\bar{i}, \bar{j} \pm 1/2)$, and all vertical face centers of the form $(i, j \pm 1/2))$ will be located at centers with coordinates $(\bar{i} + 1/2, \bar{j} \pm 1/2)$. Unfortunately, horizontal face centers of the form $(i \pm 1/2, j)$, will become corners $(\bar{i}, \bar{j})$ in the new system. Therefore, for horizontal face centers we need to work with another coordinate system with axes $\tilde{i}$ and $\tilde{j}$, which is the original coordinate system shifted down, half a unit, i.e., $\tilde{i} = i, \tilde{j} = j + 1/2$.

4.3.1 The matrix representation of the 2D curl

The 2D curl of the 2D vector field $\vec{E} = (E_1, E_2)$ with input data at face centers is the normal vector to the plane $\nabla \times \vec{E} = (0, 0, \nabla \cdot (E_2, -E_1))$ at face centers. Since its third component can be calculated by a divergence, and a mimetic difference divergence is calculated at cell centers, we can use this divergence characterization to obtain the next matrix $C_z^{(k)} \in \mathbb{R}^{mn \times (m(n+1)+(m+1)n)}$ representation of the discrete analog of order k of the

k	m=n=5	m=n=9	m=n=13	m=n=17	m=n=25
2	0.11731	0.078185	0.056351	0.043752	0.030093
4		0.012697	0.0043952	0.0019488	0.00057731
6			4.7339e-14	3.9728E-14	1.1368e-13
8				2.5123E-14	1.3131e-13

Table 1: L^2 error of the discrete analog of the 2D curl third component for the zero curl vector field $\vec{E} = (6x^5y^6, 6x^6y^5, 0)$. The accuracy order k requires at least $2k+1$ cells in each direction.

scalar field third component of the curl

$$C_z^{(k)} = \begin{pmatrix} -\bar{D}_n^{(k)} \otimes I_m & \quad I_n \otimes \bar{D}_m^{(k)} \end{pmatrix}$$

where $\bar{D}_p^{(k)} \in \mathbb{R}^{p\times(p+1)}$ is the 1D divergence operator $D_p^{(k)}$ of order k along the p-axis stripped off its first and last rows, and I_p is the $p \times p$ identity matrix. It is understood that $C_z^{(k)}$ is calculated at cell centers.

Moreover, the 2D curl third component applied to the vector field $\vec{E} = (E_1, E_2)$ is

$$(\nabla \times \vec{E}) \cdot \vec{k} = C_z^{(k)} \begin{pmatrix} E_1 \\ E_2 \end{pmatrix},$$

where E_1 acts on the horizontal 2D face centers and E_2 acts on the vertical 2D face centers. An Octave implementation of computational error of $C_z^{(k)}, k = 2, 4, 6, 8$, for different numbers of horizontal cells m and vertical cells $n = m$ is displayed in Table 1 for the 2D vector field $\vec{E} = (6x^5y^6, 6x^6y^5, 0)$ with zero 2D curl. One can see that for a fixed order of accuracy, the L^2 error decreases when the number of cells increases keeping the number of cells in all directions the same. On the other hand, for a fixed number of cells, the L^2 error decreases, when the degree of accuracy increases, until maximum computational is achieved.

k	m=n=5	m=n=9	m=n=13	m=n=17	m=n=25
2	0.16957	0.079077	0.050387	0.036734	0.023698
4		0.003975	0.0010699	0.00041376	0.00010679
6			7.2666e-15	5.168e-15	1.9431e-14
8				4.494e-15	2.1836e-14

Table 2: First component L^2 error of the vector 2D curl discrete analog for $\vec{E} = (6x^5y^6, 6x^6y^5, x^6y^6)$. Accuracy order k requires at least $2k+1$ cells in each direction.

k	m=n=5	m=n=9	m=n=13	m=n=17	m=n=25
2	0.16957	0.079077	0.050387	0.036734	0.023698
4		0.003975	0.0010699	0.00041376	0.00010679
6			7.0582e-15	5.191e-15	1.8853e-14
8				4.5095e-15	2.1167e-14

Table 3: Second component L^2 error of the vector 2D curl discrete analog for $\vec{E} = (6x^5y^6, 6x^6y^5, x^6y^6)$. Accuracy order k requires at least $2k+1$ cells in each direction.

k	m=n=5	m=n=9	m=n=13	m=n=17	m=n=25
2	0.2916	0.15726	0.1069	0.080791	0.05417
4		0.018816	0.0063972	0.0028174	0.00083029
6			6.163e-14	5.6691e-14	1.4413e-13
8				3.3509e-14	1.7776e-13

Table 4: L^2 error of the vector 2D curl discrete analog for $\vec{E} = (6x^5y^6, 6x^6y^5, x^6y^6)$. Accuracy order k requires at least $2k+1$ cells in each direction.

A different version of the 2D curl operator $C^{(k)}$ for the vector field $\vec{E}(x, y)$ with components $(E_x(x, y), E_y(x, y), E_z(x, y))$ is given by

$$\nabla\vec{E}(x,y) = \left(\frac{\partial E_3}{\partial y}(x,y), -\frac{\partial E_3}{\partial x}(x,y), \frac{\partial E_2}{\partial x}(x,y) - \frac{\partial E_1}{\partial y}(x,y)\right).$$

Its discrete analog matrix representation is

$$C^{(k)} = \begin{pmatrix} & & \bar{D}_n^{(k)} \otimes I_{m+1} \\ & & -I_{n+1} \otimes \bar{D}_m^{(k)} \\ -\bar{D}_n^{(k)} \otimes I_m & I_n \otimes \bar{D}_m^{(k)} & \end{pmatrix}.$$

Its action on the vector field $\vec{E} = (E_1, E_2, E_3)$ is computed by

$$\nabla \times \vec{E} = C^{(k)} \begin{pmatrix} E_1 \\ E_2 \\ E_3 \end{pmatrix}.$$

An Octave implementation of computational error of the first and second components of the vector 2D curl $C^{(k)}, k = 2, 4, 6, 8$, for different numbers of horizontal cells m and vertical cells $n = m$, are displayed in Tables 2 and 3, respectively, for the 2D vector field $\vec{E} = (6x^5y^6, 6x^6y^5, x^6y^6)$ with zero curl. The third component errors matches those given in Table 1, since the first two components of this example and the example for $C_z^{(k)}$ are the same. In addition, in Table 4 the errors of the vector 2D curl (all components together) are shown. In all these tables, for a fixed order of accuracy, the L^2 error decreases when the number of cells increases keeping the number of cells in all directions the same. On the other hand, for a fixed number of cells, the L^2 error decreases, when the degree of accuracy increases, until maximum computational is achieved.

4.3.2 Spectral radius of the 2D curl operator

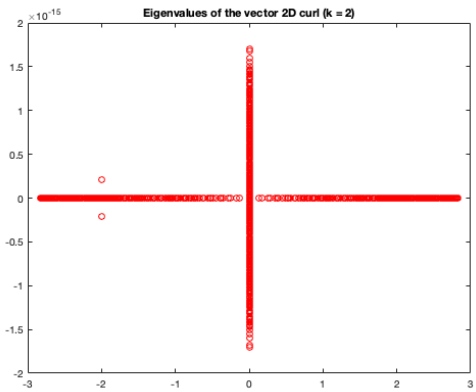

Figure 6: Vector 2D curl eigenvalues, for order of accuracy $k = 2$ and $m = n = 25$.

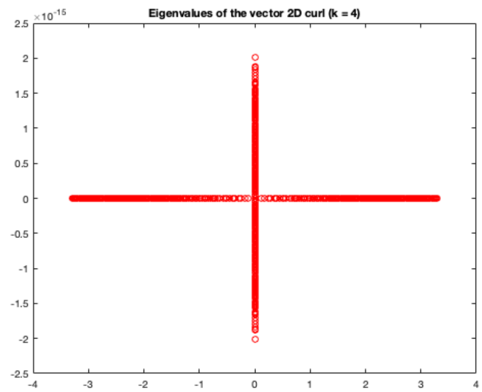

Figure 7: Vector 2D curl eigenvalues, for order of accuracy $k = 4$ and $m = n = 25$.

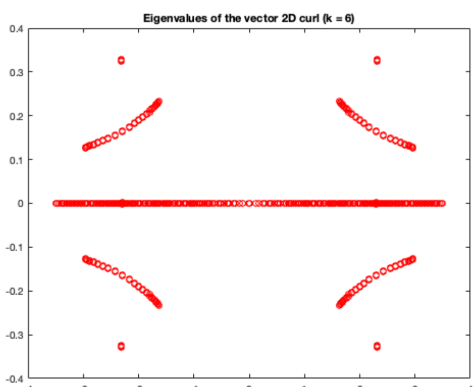

Figure 8: Vector 2D curl eigenvalues, for order of accuracy $k = 6$ and $m = n = 25$.

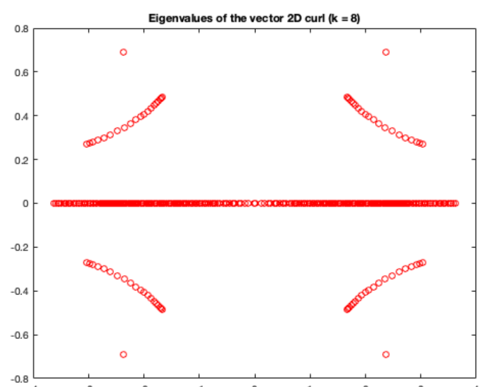

Figure 9: Vector 2D curl eigenvalues, for order of accuracy $k = 8$ and $m = n = 25$.

Since our 2D curl discrete analog based on the divergence operator has a square matrix representation, one can compute its spectral radius.

Table 5 displays the spectral radius of the vector 2D curl discrete analog. One can see that for a fixed order of accuracy, the spectral radius converges asymptotically when the number of cell increases. Figures 6, 7, 8, and 9, display the eigenvalues of the vector 2D

k	m=n=6	m=n=10	m=n=14	m=n=18	m=n=25
2	2.7321	2.7936	2.8106	2.8177	2.8228
4		3.1967	3.2572	3.2768	3.2891
6			3.4268	3.4717	3.4952
8				3.5717	3.6138

Table 5: Spectral radius of the vector 2D curl discrete analog.

curl operator for orders of accuracy $k = 2, 4, 6, , 8$, respectively. The number of horizontal and vertical cells is set to $m = n = 25$.

4.4 The 3D curl operator based on the divergence approach

A similar argument to the one presented in the 2D curl section, can be applied to each of the components of the 3D curl. The next discrete analog of the 3D curl does not use interpolation operators and it does not have any constraint in terms of the number of cells in each direction.

4.4.1 The matrix representation of the 3D curl

The 3D curl of the 3D vector field $\vec{E} = (E_1, E_2, E_3)$ with input data at face centers is the normal vector to the plane

$$\nabla \times \vec{E} = (\nabla \cdot (E_3, -E_2), \nabla \cdot (-E_3, E_1), \nabla \cdot (E_2, -E_1))$$

at face centers. Since the 3D curl can be calculated by three 2D divergences, and a mimetic difference divergence is calculated at cell centers, we can use this divergence characterization to obtain the next matrix $C^{(k)} \in \mathbb{R}^{3\,mno \times 2(m(n+1)o+(m+1)no+mn(o+1))}$ representation of the discrete analog of order k of the 3D curl $C^{(k)}$ of order k of the curl, computed at cell centers, is given by

$$C^{(k)} = \begin{bmatrix} & & & & -\bar{D}_o^{(k)} \otimes I_n \otimes I_m & I_o \otimes \bar{D}_n^{(k)} \otimes I_m \\ & & -I_o \otimes I_n \otimes \bar{D}_m^{(k)} & \bar{D}_o^{(k)} \otimes I_n \otimes I_m & & \\ -I_o \otimes \bar{D}_n^{(k)} \otimes I_m & I_o \otimes I_n \otimes \bar{D}_m^{(k)} & & & & \end{bmatrix},$$

where $\bar{D}_p^{(k)} \in \mathbb{R}^{p \times (p+1)}$ is the 1D divergence operator $D_p^{(k)}$ of order k along the p-axis stripped off its first and last rows, and I_p is the $p \times p$ identity matrix. It is understood that $C^{(k)}$ is calculated at cell centers.

Moreover, the 3D curl applied to the vector field $\vec{E} = (E_1, E_2, E_3)$ is

$$\nabla \times \vec{E} = C^{(k)} \begin{pmatrix} E_1 & E_2 & E_3 & E_1 & E_2 & E_3 \end{pmatrix}^T,$$

where

- the first component E_1 acts on the $x-z$ cell face centers,
- the second component E_2 acts on the $y-z$ cell face centers,
- the third component E_3 acts on the $x-z$ cell face centers,
- the fourth component E_1 acts on the $x-y$ cell face centers,
- the fifth component E_2 acts on the $x-z$ cell face centers, and
- the sixth component E_3 acts on the $x-z$ cell face centers.

k	m=n=o=5	m=n=o=9	m=n=o=13	m=n=o=17	m=n=o=25
2	0.026079	0.025513	0.022567	0.020197	0.016949
4		0.0041434	0.0017601	0.0008996	0.00032514
6			1.3574e-14	1.4886e-14	6.3206e-14
8				1.5874e-14	8.3301e-14

Table 6: L^2 error of the discrete analog of the 3D curl for the zero curl vector field $\vec{E} = (x^5y^6z^6, x^6y^5z^6, x^6y^6z^5)$. The accuracy order k requires at least $2k+1$ cells in each direction.

An Octave implementation of computational error of $C^{(k)}, k = 2, 4, 6, 8$, for different numbers of horizontal cells m, vertical cells $n = m$, and depth cell $o = m$ is displayed in Table 6 for the 3D vector field $\vec{E} = (x^5y^6z^6, x^6y^5z^6, x^6y^6z^5)$ with zero curl. One can see that for a fixed order of accuracy, the L^2 error decreases when the number of cells increases keeping the number of cells in all directions the same. On the other hand, for a fixed number of cells, the L^2 error decreases, when the degree of accuracy increases, until maximum computational is achieved.

4.5 Matrix representation for general 3D operators

One of the difficulties with the discrete analog of the 3D curl is that, for a three-dimensional vector field $\vec{V} = (P, Q, R)$, it requires each of the scalar field components P, Q, R receives data from two different sets of faces (P from y- and z-faces, Q from x- and z-faces, and R from x- and y-faces), and this is very unique among mimetic difference discrete analogs whose input data is usually at the centers or all three sets of faces. In general, when vector fields components are considered (instead of the whole vector field), one works with cell centers. When considering this, one would like to have input data and output computation at cell centers. Therefore, the corresponding 3D curl discrete analog C_g will be a composition of a particular interpolation that constructs two-faced data from center cells $\tilde{I}_c^f$ and the actual discrete curl computation, or, in other words,

$$C_g^{(k)} = C^{(k)}(\tilde{I}_c^f)^{(k)},$$

where

$$(\tilde{I}_c^f)^{(k)} = \begin{pmatrix} (I_c^f)_2^{(k)} & & \\ & (I_c^f)_1^{(k)} & \\ & & (I_c^f)_1^{(k)} \\ (I_c^f)_3^{(k)} & & \\ & (I_c^f)_3^{(k)} & \\ & & (I_c^f)_2^{(k)} \end{pmatrix},$$

where

$$(I_c^f)^{(k)} = \begin{pmatrix} (I_c^f)_1^{(k)} & & \\ & (I_c^f)_2^{(k)} & \\ & & (I_c^f)_3^{(k)} \end{pmatrix},$$

is the interpolation operator from cell centers to cell faces of order k. Hence,

$$\begin{aligned} C_g^{(k)} &= C^{(k)} \begin{pmatrix} \hat{I}_o^T \otimes (I_D)_y^{(k)} \otimes \hat{I}_m^T & & \\ & \hat{I}_o^T \otimes \hat{I}_n^T \otimes (I_D)_x^{(k)} & \\ & & \hat{I}_o^T \otimes \hat{I}_n^T \otimes (I_D)_x^{(k)} \\ (I_D)_z^{(k)} \otimes \hat{I}_n^T \otimes \hat{I}_m^T & & \\ & (I_D)_z^{(k)} \otimes \hat{I}_n^T \otimes \hat{I}_m^T & \\ & & \hat{I}_o^T \otimes (I_D)_y^{(k)} \otimes \hat{I}_m^T \end{pmatrix} \\ &= \begin{pmatrix} & -D_z^{(k)}(I_D)_z^{(k)} \otimes \tilde{I}_n \otimes \tilde{I}_m & \tilde{I}_o \otimes D_y^{(k)}(I_D)_y^{(k)} \otimes \tilde{I}_m \\ D_z^{(k)}(I_D)_z^{(k)} \otimes \tilde{I}_n \otimes \tilde{I}_m & & -\tilde{I}_o \otimes \tilde{I}_n \otimes D_x^{(k)}(I_D)_x^{(k)} \\ -\tilde{I}_o \otimes D_y^{(k)}(I_D)_y^{(k)} \otimes \tilde{I}_m & \tilde{I}_o \otimes \tilde{I}_n \otimes D_x^{(k)}(I_D)_x^{(k)} & \end{pmatrix}, \end{aligned}$$

where $(I_D)_p^{(k)}$ is the 1D interpolation operator from cell centers to cell faces of order k along the p-axis, and $\hat{I}_q$ is the $(q+2) \times q$ matrix, with first and last zero rows, and the $q \times q$ identity matrix in the middle, or

$$\hat{I}_q = \begin{pmatrix} 0 & \cdots & 0 \\ & I_q & \\ 0 & \cdots & 0 \end{pmatrix},$$

and

$$\tilde{I}_q = \hat{I}_q \hat{I}_q^T = \begin{pmatrix} 0 & \cdots & 0 \\ \vdots & I_q & \vdots \\ 0 & \cdots & 0 \end{pmatrix},$$

is the identity matrix of order $q \times q$ surrounded by one layer of zeros.

An Octave implementation of computational error of $C^{(k)}, k = 2, 4, 6, 8$, for different numbers of horizontal cells m, vertical cells $n = m$, and depth cell $o = m$ is displayed in

k	m=n=o=5	m=n=o=9	m=n=o=13	m=n=o=17	m=n=o=25
2	0.2746	0.45652	0.54219	0.59119	0.64487
4		0.011086	0.0066443	0.0043903	0.0023086
6			4.5379e-05	1.579e-05	3.4148e-06
8				1.9102e-14	8.9297e-14

Table 7: L^2 error of the discrete analog of the 3D curl for the zero curl vector field $\vec{E} = (x^5y^6z^6, x^6y^5z^6, x^6y^6z^5)$. The accuracy order k requires at least $2k+1$ cells in each direction.

k	m=n=o=7	m=n=o=9	m=n=o=13	m=n=o=17	m=n=o=25
2	0.034677	0.025513	0.022567	0.020197	0.016949
4		0.0041434	0.0017601	0.0008996	0.00032514
6			7.252e-15	2.7771e-14	6.7671e-14
8				1.9102e-14	8.9297e-14

Table 8: L^2 error of the discrete analog of the 3D curl for the zero curl vector field $\vec{E} = (x^5y^6z^6, x^6y^5z^6, x^6y^6z^5)$. The accuracy order k that uses a specific version of the interpolation of 8-th order requires at least $\min\{2k+1, 7\}$ cells in each direction.

Tables 8 (with interpolations of 8-th order, since the main source of errors is the initial interpolation from centers to faces) and 7 (using interpolations from centers to faces of order k) for the 3D vector field $\vec{E} = (x^5y^6z^6, x^6y^5z^6, x^6y^6z^5)$ with zero curl. One can see that for a fixed order of accuracy, the L^2 error decreases when the number of cells increases keeping the number of cells in all directions the same. On the other hand, for a fixed number of cells, the L^2 error decreases when the degree of accuracy increases, until maximum computational efficiency is achieved.

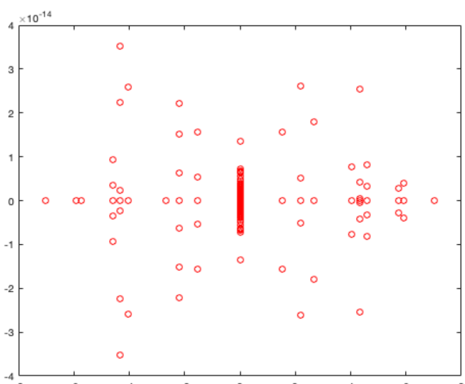

Figure 10: 3D curl eigenvalues, for order of accuracy $k = 2$ and $m = n = o = 5$.

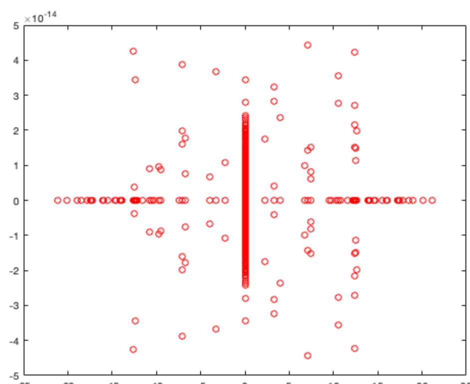

Figure 11: 3D curl eigenvalues, for order of accuracy $k = 4$ and $m = n = o = 9$.

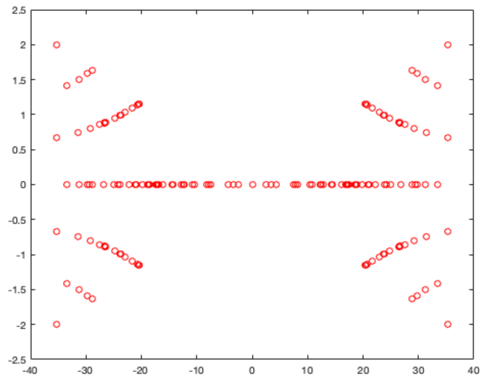

Figure 12: 3D curl eigenvalues, for order of accuracy $k = 6$ and $m = n = o = 13$.

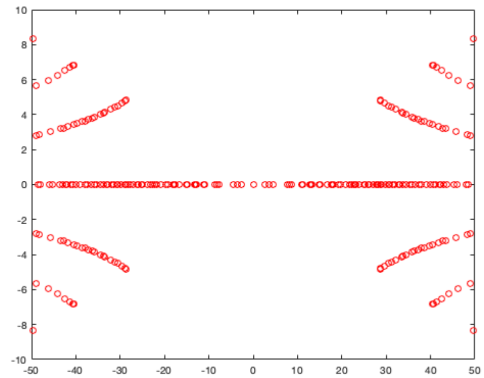

Figure 13: 3D curl eigenvalues, for order of accuracy $k = 8$ and $m = n = o = 17$.

4.5.1 Spectral radius of the 3D curl operator

The general purpose discrete analog of the 3D curl operator has input data and output computation at cell centers. Therefore, its matrix representation is square.

One can see that for a fixed order of accuracy, the spectral radius converges asymptotically when the number of cell increases. Figures 10, 11, 12, and 13, display the eigenvalues of the 3D curl operator for orders of accuracy $k = 2, 4, 6, 8$, respectively. The number of cells in each direction is set to $m = n = o = 25$. Keep in mind that it has been shown in the previous subsection that the initial special interpolation that the general curl 3D requires is responsible for most of the approximation error of this version of the 3D curl. For that reason, the figures are intended for reporting the collection of eigenvalues for each case under specific values of the number of cells.

4.6 Matrix representation for general 2D operators

The purpose of this section is to derive discrete analogs of the general 2D curl operator which can be used for complex differential operators that may include gradient, divergence, Laplacian operators. The input data and output computation locations will be as for the generalized 3D curl, at cell centers, and their values at boundary cell face centers will be zero. It turns out that the general 2D curl discrete analog is just a projection onto 2D of the general 3D curl discrete analog, i.e.,

$$C_g^{(k)} = C_z^{(k)} (\tilde{I}_c^f)^{(k)},$$

where

$$(\tilde{I}_c^f)^{(k)} = \begin{pmatrix} (I_c^f)_2^{(k)} & \\ & (I_c^f)_1^{(k)} \end{pmatrix},$$

where

$$(I_c^f)^{(k)} = \begin{pmatrix} (I_c^f)_1^{(k)} & \\ & (I_c^f)_2^{(k)} \end{pmatrix},$$

is the interpolation operator from cell centers to cell faces of order k. Hence,

$$\begin{aligned} C_g^{(k)} &= C_z^{(k)} \begin{pmatrix} (I_D)_y^{(k)} \otimes \hat{I}_m^T & \\ & \hat{I}_n^T \otimes (I_D)_x^{(k)} \end{pmatrix} \\ &= \begin{pmatrix} -D_y^{(k)} (I_D)_y^{(k)} \otimes \tilde{I}_m & \tilde{I}_n \otimes D_x^{(k)} (I_D)_x^{(k)} \end{pmatrix}, \end{aligned}$$

where $(I_D)_p^{(k)}$ is the 1D interpolation operator from cell centers to cell faces of order k along the p-axis, and $\tilde{I}_q = \hat{I}_q \hat{I}_q^T$.

5 Future work

Currently, we are working on several examples where the different curl 2D and 3D operators are tested and compared whenever appropriate. Among them the numerical solution of Maxwell equations, the magneto-hydrodynamic equations, and the Navier-Stokes equations (derived from the magneto-hydrodynamic equations, where the enstrophy is an integral whose integrand contains one factor given by the square norm of the curl of the velocity vector field).

References

[1] Castillo, J.E., and Miranda, G.F., Mimetic Discretization Methods, Chapman and Hall CRC, 2013.

[2] Castillo, J.E., and Grone, R.D., A matrix analysis approach to higher-order approximations for divergence and gradients satisfying a global conservative law, SIAM J. Matrix Anal. Appl., Vol. 25, No. 1, pp. 128-142. CRC Press, Boca Raton, Florida, USA, 2013.

[3] Castillo, J.E, and Grone, R.D., Using Kronecker products to construct mimetic gradients, Linear and Multilinear Algebra, DOI: 10.1080/03081087.2017.1298564, 2017.

[4] Corbino, J., and Castillo, J.E., High-order mimetic difference operators satisfying the extended Gauss divergence theorem, Journal of Computational and Applied Mathematics, 364 (2020) 112326.

[5] Corbino, J., Dumett M.A., and Castillo, J.E., MOLE: Mimetic Operators Library Enhanced, Journal of Open Source Software, 9(99), 6288, 2024.

[6] Corbino, J., and Castillo, J.E., MOLE: Mimetic Operators Library Enhanced: The Open-Source Library for Solving Partial Differential Equations using Mimetic Methods, 2017. https://github.com/csrc-sdsu/mole

[7] Dumett M.A., and Castillo, J.E., Interpolation Operators for Staggered Grids, CSRC Report, Publication Number: CSRCR2022-02, 6-Dec-2022.

[8] Dumett M.A., and Castillo, J.E., Mimetic Analogs of Vector Calculus Identities, CSRC Report, Publication Number: CSRCR2023-01, 6-Jul-2023.

[9] Dumett M.A., and Castillo, J.E., Energy Conservation and Convergence of High-Order Mimetic Schemes for the 3D Advection Equation, CSRC Report, Publication Number: CSRCR2023-05, 21-Jul-2023.

[10] Dumett M.A., Mimetic Difference Schemes Preserve Mass, Momenta and Energy for Systems of Conservation Laws in Curvilinear Structured Grids, CSRC Report, Publication Number: CSRCR2025-04, 27-August-2025.

[11] Dumett M.A., and Castillo, J.E., General Framework for Mimetic Differences, CSRC Report, Publication Number: CSRCR2024-07, 25-June-2024.

[12] Fornberg B., Some Numerical Techniques for Maxwell's Equations in Different Types of Geometries, University of Colorado, Department of Applied Mathematics, 526 UCB, Boulder, CO 80309, USA (fornberg@colorado.edu).

[13] Kreeft J., Palha A., and Gerritsma M., Mimetic Framework On Curvilinar Quadrilaterals Of Arbitrary Order, arXiv:1111.4304v1 [math.NA], 2011. https://doi.org/10.48550/arXiv.1111.4304.

[14] Hairer E., Lubich C., and Wanner G., Geometrical Numerical Integrators, Second edition, Springer, 2006.

[15] Marsden J., and Weinstein, A., The Hamiltonian Structure of Maxwell-Vlasov equations, Physica 4D, pp 394-406, 1982, North-Holland Publishing Company.

[16] Rawashdeh E.A., A simple method for finding the inverse of a Vandermonde matrix, Matematički Vesnik, September 2019.

[17] Runyan J.B., A Novel Higher Order Finite Difference Time Domain Method Based on the Castillo-Grone Mimetic Curl Operator with Applications Concerning the Time-Dependent Maxwell Equations. Master's thesis, San Diego State University, 2011.

[18] Sanchez E.J., Paolini C.P., and Castillo J.E., The Mimetic Methods Toolkit: An object-oriented API for Mimetic Finite Differences, Journal of Computational and Applied Mathematics, 270(0):308–322, 2014. Fourth International Conference on Finite Element Methods in Engineering and Sciences (FEMTEC 2013).

[19] Sanchez E., Miranda G., and Castillo, J.E., Constructing a Mimetic Curl using Gauss' Theorem, CSRC Report, Publication Number: CSRCR2014-04, 21-Nov-2023.

[20] Srinivasan A., Dumett M.A., Paolini C., Miranda G.F., and Castillo J.E., Mimetic finite difference operators and higher order quadratures, GEM-International Journal on Geomathemathic, 4(1), 2023. https://doi.org/10.1007/s13137-023-00230-z.

[21] Srinivasan A., and Castillo J.E., Energy preserving high order mimetic methods for Hamiltonian equations, Computers and Fluids, 297 (2025) 106642.

[22] Vandekerckhove S., Vandewoestyne B., De Gersem H., Van Den Abeelea K., and Vandewalle S., Mimetic discretization and higher order time integration for acoustic, electromagnetic and elastodynamic wave propagation, Journal of Computational and Applied Mathematics, 259, pp. 65–76, 2014.

[23] Verwer J.G., On Time Staggering for Wave Equations, J Sci Comput, 33, pp. 139–154, 2007, DOI10.1007/s10915-007-9146-8.

[24] Yee K.S., Numerical solution of initial boundary value problems involving Maxwell's equation in isotropic media, IEEE T. Antenn. Propag., 14(3), 302-307, 1966. https://ieeexplore.ieee.org/document/1138693.