Computers and Fluids 213 (2020) 104746

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

High-order mimetic finite differences for anisotropic elliptic equations

Angel Boada[1], Christopher Paolini*, Jose E. Castillo

Computational Science Research Center, San Diego State University, San Diego, 92182, CA, USA

ARTICLE INFO

Article history:
Received 2 November 2019
Revised 29 August 2020
Accepted 3 September 2020
Available online 2 October 2020

Keywords:
Mimetic
Finite-difference
Castillo-Grone method
Differential operators
High-order
Anisotropic diffusion
Flux operator

ABSTRACT

Fractured geologic media can yield anisotropies in solute and heat diffusion due to the formation of changing fluid network connectivity in a rock matrix. In this paper we model Steady-state anisotropic heat diffusion as an elliptic partial differential equation with a symmetric positive definite second rank thermal conductivity tensor. We model diffusive flux as a non-diagonal symmetric tensor, which can potentially have jump discontinuities that are not aligned with the coordinate axis. The presence of jump discontinuities due to joints and faults in a rock matrix impose difficulties on existing, well-established numerical schemes that model diffusive transport. In our scheme, we model diffusive flux using mimetic finite difference operators, which are discrete analogs of the classical continuous differential operators. We introduce a 2nd- and 4th-order mimetic formulation for computing anisotropic fluxes. Numerical results demonstrate our formulation yields a substantial improvement compared to similar mimetic schemes.

1. Introduction

Classical numerical differentiation techniques start by discretizing, to some order of accuracy, a specific problem to solve, while attempting to maintain numerical stability. Difficulties inherent with discretization provide an inspiration for Mimetic (or compatible) numerical methods. Mimetic operators derive their name due to their ability to discretely mimic, or preserve the properties of, continuous vector calculus operators. The resulting discrete operators are then substituted into the system of partial differential equations. As part of the mimetic family of methods, Castillo-Grone's (CG) Mimetic Finite Differences have been broadly used in many applications, some of which include: seismic studies, wave propagation, image processing, electromagnetism and fluid dynamics [1–5]. Developed in 2003 [2], one of the most prominent features of CG Mimetic operators is that they mimic the symmetry properties of the continuum differential operators [6], which makes numerical schemes based on them more faithful to the physics of the problem being solved [7]. Our mimetic methods are less computationally expensive than Finite Elements and Discontinuous Galerkin ("DG"), and our methods are able to achieve the same order of accuracy at the domain interior as well as the boundary which the Staggered Summation by Parts methods cannot achieve.

Traditional finite difference methods are based on deriving stencils for the derivative operators using a Taylor's series approach. This approach has the advantage of being straightforward, since one can easily implement matrices for the numerical derivatives. However, the underlying Physics of the problem may not be adequately represented in this discretization process. Mimetic difference methods construct discrete difference operators Divergence, D, and Gradient G, which satisfy a discrete extended Gauss Divergence theorem. These methods are called *Mimetic* because the discrete difference operators mimic the properties of the continuum ones. Hence, numerical schemes using mimetic operators are more faithful to the physics of the problem under investigation [8]. Well-posedness of the problem can then be guaranteed by implementing the energy method, which ensures a decaying energy in the system over time. A key component of the energy method is integration by parts ("IBP"). A discrete analog of IBP is summation by parts ("SBP"). Well-posedness of the PDE is guaranteed if we can develop a discrete analog that satisfies IBP. The resulting operators are referred to as SBP operators. In the mimetic scheme, the equivalent to IBP is the extended Gauss Divergence Theorem.

Capable of achieving a uniform high-order of accuracy in three dimensional space, construction of CG operators relies on satisfying the Extended Gauss Divergence theorem to achieve conservation. By *high-order*, we mean a numerical scheme that yields an approximation with an error proportional to h^{2+} and thus achieves a better than second order accurate approximation. The Castillo-Grone

* Corresponding author.
E-mail addresses: aboadavelazco@sdsu.edu (A. Boada), paolini@engineering.sdsu.edu (C. Paolini), jcastillo@sdsu.edu (J.E. Castillo).
[1] Computational Science Doctoral Student

https://doi.org/10.1016/j.compfluid.2020.104746
0045-7930/

Nomenclature

α	Thermal diffusivity tensor, $m^2 s^{-1}$
$\hat{\mathbf{I}}^{\mathbf{T}}_{\mathbf{m}}$	augmented identity matrix with first and last rows all zero, $\in \mathbb{R}^{(m+2)\times m}$
$\hat{\mathbf{I}}^{\mathbf{T}}_{\mathbf{n}}$	augmented identity matrix with first and last rows all zero, $\in \mathbb{R}^{(n+2)\times n}$
$\mathbf{D_{xy}}$	Corbino-Castillo 2D mimetic divergence operator, $= \begin{bmatrix}\mathbf{R}_x \ \mathbf{R}_y\end{bmatrix}$
$\mathbf{D_x}$	Corbino-Castillo 1D mimetic divergence operator for x-dimension, $\in \mathbb{R}^{(m+2)\times(m+1)}$
$\mathbf{D_y}$	Corbino-Castillo 1D mimetic divergence operator for y-dimension, $\in \mathbb{R}^{(n+2)\times(n+1)}$
$\mathbf{D}$	Corbino-Castillo 1D mimetic divergence operator, $\in \mathbb{R}^{(m+2)\times(m+1)}$
$\mathbf{G_{xy}}$	Corbino-Castillo 2D mimetic gradient operator $= \begin{bmatrix}\mathbf{S}_x \\ \mathbf{S}_y\end{bmatrix}$, $\in \mathbb{R}^{(2mn+m+n)\times[(m+2)(n+2)]}$
$\mathbf{G_x}$	Corbino-Castillo 1D mimetic gradient operator for x-dimension, $\in \mathbb{R}^{(m+1)\times(m+2)}$
$\mathbf{G_y}$	Corbino-Castillo 1D mimetic gradient operator for y-dimension, $\in \mathbb{R}^{(n+1)\times(n+2)}$
$\mathbf{G}^{\star}$	Mimetic flux generating operator, $\in \mathbb{R}^{2mn+m+n\times(m+2)(n+2)}$
$\mathbf{I_x}$	Interpolation matrix with respect to x-axis, $\in \mathbb{R}^{[(m+1)n]\times[(n+1)m]}$
$\mathbf{I_y}$	Interpolation matrix with respect to y-axis, $\in \mathbb{R}^{[(n+1)m]\times[(m+1)n]}$
$\mathbf{R_x}$	$= \hat{\mathbf{I}}_{\mathbf{n}} \otimes \mathbf{D_x}$, $\in \mathbb{R}^{(m+1)n\times(m+2)(n+2)}$
$\mathbf{R_y}$	$= \mathbf{D_y} \otimes \hat{\mathbf{I}}_{\mathbf{m}}$, $\in \mathbb{R}^{(n+1)m\times(m+2)(n+2)}$
$\mathbf{S_x}$	$= \hat{\mathbf{I}}^{\mathbf{T}}_{\mathbf{n}} \otimes \mathbf{G_x}$, $\in \mathbb{R}^{(m+1)n\times(m+2)(n+2)}$
$\mathbf{S_y}$	$= \mathbf{G_y} \otimes \hat{\mathbf{I}}^{\mathbf{T}}_{\mathbf{m}}$, $\in \mathbb{R}^{(n+1)m\times(m+2)(n+2)}$
$\mathbf{\Gamma}$	Generalized diffusion coefficient tensor, $m^2 s^{-1}$
D	Molecular diffusivity tensor, $m^2 s^{-1}$
K	Hydraulic conductivity tensor, ms^{-1}
m	Number of cells in a staggered grid with respect to the $x-$dimension.
n	Number of cells in a staggered grid with respect to the $y-$dimension.
N_c	number of cell centers (including boundaries) on the staggered grid where scalar values are stored, $= (m+2)(n+2)$
N_x	number of vector components on the x-axis $= (n+1)m$
N_y	number of vector components on the y-axis $= (m+1)n$

Mimetic Operators have a set of free parameters which need to be selected. In 2018, Castillo and Corbino (CC) developed a different and more intuitive approach for the construction of these operators without the need for free parameters. This effort resulted in operators that have optimal bandwidth, and numerical results showed that CC operators exhibit improvement, in terms of accuracy, when compared to the CG counterparts [9]. Similar approaches along the mimetic framework are being developed which are related to this area of research such as the symmetry-preserving finite-difference discretization by van't Hof and Vuik in [10], and Verstappen and Veldman in [11]; the mimetic finite difference scheme by Feo et. al. in [12]; and the mimetic finite difference method by Lipnikov, Manzini, and Shashkov in [13]. Finally, of particular importance due to similarities with the staggered grid formulation is the work by Nordstrom, O'Reilly and Petersson in [14,15].

A process common to physical transport phenomena is anisotropic diffusion. Steady-state heat- and species-diffusion processes are described by elliptic partial differential equations. Diffusive flux operators are defined by a full tensor. However, models often assume a diagonal tensor. Therefore, a suitable scheme must be used to solve fully anisotropic diffusion. Consider the following steady state diffusion equation:

$$-\nabla \cdot \mathbf{\Gamma} \nabla u = f, \tag{1}$$

where $\mathbf{\Gamma}$ is a symmetric and positive definite matrix that represents a diffusion or hydraulic conductivity tensor, f is a forcing function, and u represents an unknown solution to be computed, which can be pressure, temperature, or species concentration, for example. We consider general Robin boundary conditions for this problem.

We propose a new mimetic formulation for an anisotropic flux generating tensor based on CC Mimetic Operators, while focusing on characteristic problems of general anisotropic steady-state equations (Eq. (1)). The organization of this paper is laid out as follows. In the first section, we describe Mimetic Finite Differences, their operators, the staggered grid on which mimetic operators are defined, and a compact factorization for higher order accuracy. In the second section, we introduce a mathematical model and formulation for a mimetic flux generating operator. Finally, in the third section we present numerical examples along with discussion in order to illustrate the effectiveness of our method.

2. High order mimetic finite difference operators

Originally developed by Castillo and Grone, the one dimensional mimetic finite difference operators for the gradient ($\mathbf{G} \equiv \nabla$), divergence ($\mathbf{D} \equiv \nabla \cdot$), Laplacian ($\mathbf{L} \equiv \nabla^2$) and curl ($\mathbf{C} \equiv \nabla \times$) operators, are discrete representations of the classical continuous differential operator counterparts. Represented by sparse matrices, the main idea behind the construction of these operators is to find high-order approximations that satisfy the Extended Gauss Divergence theorem (Eq. (2)) in the discrete sense [2]:

$$\int_{\Omega} (\nabla \cdot \vec{v}) f \, dV + \int_{\Omega} \vec{v} \cdot (\nabla f) \, dV = \int_{\partial\Omega} (f\vec{v}) \cdot \vec{n} \, dS, \tag{2}$$

therefore, these operators satisfy [8]:

$$\langle \mathbf{D}\vec{v}, f \rangle_Q + \langle \mathbf{G}f, \vec{v} \rangle_P = \langle \mathbf{B}\vec{v}, f \rangle_I, \tag{3}$$

where $\mathbf{B}$ is a mimetic boundary operator, and the operators P, Q and I are self-adjoint. In particular, the Q inner product accounts for the scalar inner product in cell centers, the P inner product accounts for a vector-field inner product at the cell faces, and the I inner product is at the boundaries. Weighted inner products are defined in the standard form,

$$\langle x, y \rangle_A = y^T A x. \tag{4}$$

From Eq. (3) we obtain:

$$\langle Q\mathbf{D}\vec{v}, f \rangle + \langle P\mathbf{G}f, \vec{v} \rangle = \langle \mathbf{B}\vec{v}, f \rangle \tag{5}$$

$$\langle Q\mathbf{D}\vec{v} + \mathbf{G}^T P v, f \rangle = \langle \mathbf{B}\vec{v}, f \rangle \tag{6}$$

$$Q\mathbf{D}\vec{v} + \mathbf{G}^T P\vec{v} = \mathbf{B}\vec{v} \tag{7}$$

$$Q\mathbf{D} + \mathbf{G}^T P = \mathbf{B}. \tag{8}$$

Note that matrix A in Eq. (4) must be symmetric and positive definite, and therefore matrices P and Q in Eq. (3) and Eqs. (5) to (8) must also satisfy it.

As discrete counterparts, Mimetic Operators '*mimic*' the following vector calculus identities of their continuous analogs:

$$\mathbf{G}f_{const} = 0, \tag{9}$$

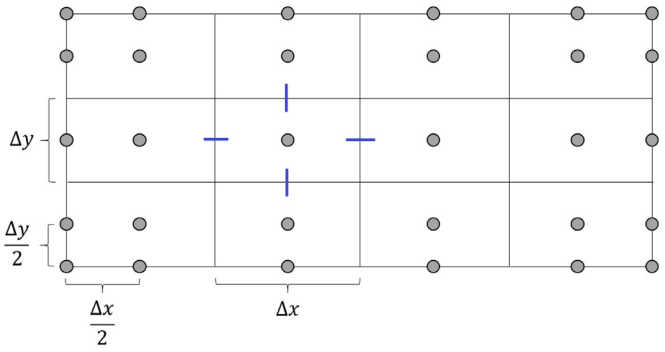


Fig. 1. Two-dimensional, uniform staggered grid ($m = 4$ and $n = 3$).

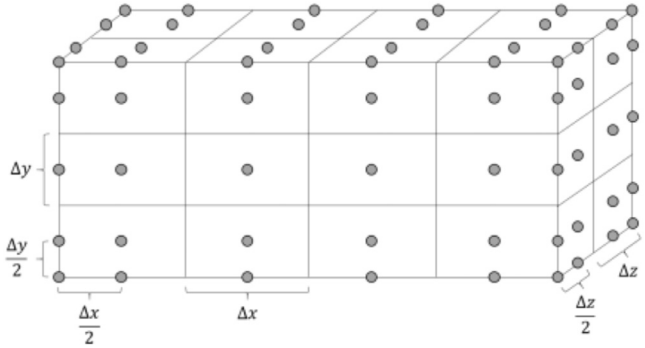


Fig. 2. Three-dimensional, uniform staggered grid ($m = 4$, $n = 3$ and $o = 2$).

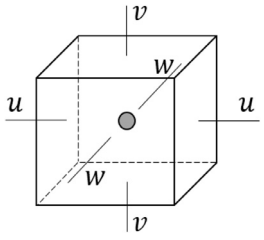


Fig. 3. A Three-dimensional staggered grid cell.

$$\mathbf{D}\vec{v}_{const} = 0, \tag{10}$$

$$\mathbf{C}\mathbf{G}f = 0, \tag{11}$$

$$\mathbf{D}\mathbf{C}\vec{v} = 0, \tag{12}$$

$$\mathbf{D}\mathbf{G}f = \mathbf{L}f. \tag{13}$$

These one-dimensional mimetic operators are extended to second and third dimensional space and the corresponding vector calculus identities remain true.

2.1. Staggered grid (mesh)

A staggered grid is a spatial discretization data structure formed by staggering several collocated grids, with each grid having its own set of corresponding functions. Mimetic operators are defined over staggered grids. In our case, scalar-valued functions are placed at cell centers, whereas vector components are located on cell edges or faces, for 2D or 3D, respectively.

Figs. 1 and 2 illustrate a 2D and 3D staggered grid, where m, n and o represent the number of cells in the x-, y-, z-direction. Fig. 3 depicts a 3D cell. In this case, u, v and w are vector components of the corresponding direction. These figures were obtained from [9].

2.2. 1D Mimetic operators

Considering improvements with respect to the original Castillo-Grone mimetic Operators in terms of accuracy and optimal bandwidth, we follow the Castillo and Corbino [9] approach for the construction of our operators. We illustrate the second-order one-dimensional mimetic divergence ($\mathbf{D}$) and gradient ($\mathbf{G}$) operators, which are the foundations of mimetic operators in higher dimensions and higher order.

In our one-dimensional staggered grid discretization (depicted in Fig. 4), the mimetic divergence operator acts on vector components (v-values) defined at $m+1$ nodes, with $x_i = i\Delta x$, $i = 0, 1, \ldots, m$. These v-values are regarded as an $(m+1)$-tuple. Conversely, the mimetic gradient operator acts on u-values defined at both boundary nodes (x_0 on the left, and x_m on the right), as well as the m cell-centers, $x_{i+\frac{1}{2}} = (i+\frac{1}{2})\Delta x$, $i = 0, 1, \ldots, m-1$. Therefore, u-values are regarded as $(m+2)$-tuple. $\mathbf{D}$ is then an $(m+2)\times(m+1)$ sparse matrix with first and last rows as zero vectors (required since the divergence is calculated at cell-centers). The gradient operator $\mathbf{G}$ is an $(m+1)\times(m+2)$ matrix. The one-dimensional mimetic divergence operator is given by:

$$\mathbf{D} = \frac{1}{\Delta x}\begin{bmatrix} 0 & 0 & \ldots & 0 \\ -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \\ 0 & 0 & \ldots & 0 \end{bmatrix}_{(m+2)\times(m+1)}, \tag{14}$$

and the Mimetic Gradient

$$\mathbf{G} = \frac{1}{\Delta x}\begin{bmatrix} -8/3 & 3 & -1/3 & & & & \\ & -1 & 1 & & & & \\ & & \ddots & \ddots & & & \\ & & & -1 & 1 & & \\ & & & 1/3 & -3 & 8/3 & \end{bmatrix}_{(m+1)\times(m+2)}. \tag{15}$$

Note there is a minimum number of cells needed to construct these mimetic operators. The gradient requires at least $2k$ cells, whereas the divergence requires at least $2k+1$, where k as the desired order of accuracy [9].

2.3. Compact operators

An important feature of mimetic operators is that they provide a uniform order of accuracy up to the boundary [2]. We can factor the desired high-order operator using the original matrix (i.e. the 2nd-order matrix). An advantage of this approach is to minimize the required computational stencil, while also avoiding the need for solving a system of linear equations [16]. This compact factorization was first presented in [17]. For example, a k^{th}-order mimetic gradient operator can be constructed as

$$\mathbf{G}_k = L_k\mathbf{G}_2, \tag{16}$$

where L_k represents the left factor matrix of k^{th}-order. Similarly, for a k^{th}-order mimetic divergence operator,

$$\mathbf{D}_k = \mathbf{D}_2 R_k, \tag{17}$$

where R_k is the right factor matrix of k^{th}-order. We can write the k^{th}-order mimetic Laplacian operator in compact form as

$$\mathbf{L}_k = \mathbf{D}_2 R_k L_k \mathbf{G}_2. \tag{18}$$

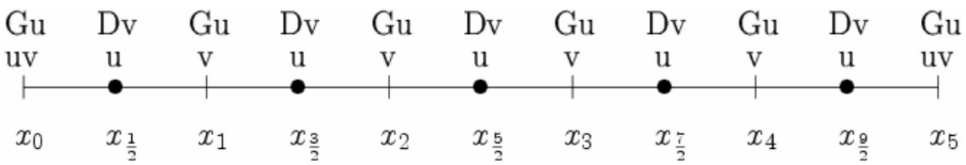


Fig. 4. A one-dimensional staggered grid ($m = 5$).

2.4. Mimetic operators for higher dimensions

A compact way of representing mimetic operators in two-dimensions with Kronecker products of block matrices [8] is presented. Let $\hat{\mathbf{I}}^{\mathbf{T}}_{\mathbf{m}}$ be an augmented identity matrix (padded with first and last rows as zero) of size $(m+2) \times m$. Note that m represents the number of cells in the staggered grid. Consider $\boldsymbol{G_x}$, $\boldsymbol{G_y}$ and $\boldsymbol{G_z}$ as the one-dimensional mimetic gradient operators for x-, y- and z-directions, respectively. A two-dimensional mimetic gradient operator is then defined as:

$$\boldsymbol{G_{xy}} = \begin{bmatrix} \boldsymbol{S_x} \\ \boldsymbol{S_y} \end{bmatrix}, \tag{19}$$

with

$$\boldsymbol{S_x} = \hat{\mathbf{I}}^{\mathbf{T}}_{\mathbf{n}} \otimes \boldsymbol{G_x} \boldsymbol{S_y} = \boldsymbol{G_y} \otimes \hat{\mathbf{I}}^{\mathbf{T}}_{\mathbf{m}} \tag{21}$$

Assuming $\boldsymbol{D}_x$ and $\boldsymbol{D}_y$ are one-dimensional mimetic divergence operators for x- and y-directions, respectively, we can construct a two-dimensional mimetic divergence operator as:

$$\mathbf{D}_{xy} = \begin{bmatrix} \boldsymbol{R_x} \ \boldsymbol{R_y} \end{bmatrix}, \tag{22}$$

where

$$\boldsymbol{R_x} = \hat{\mathbf{I}}^{\mathbf{T}}_{\mathbf{n}} \otimes \boldsymbol{D_x} \boldsymbol{R_y} = \boldsymbol{D_y} \otimes \hat{\mathbf{I}}^{\mathbf{T}}_{\mathbf{m}}. \tag{24}$$

Finally, these operators can also be extended to three-dimensions as seen in [9].

3. Mimetic flux generating operator

3.1. Mathematical model

We introduce the construction of our mimetic flux generating operator and discuss operator accuracy by considering a two-dimensional steady state diffusion example:

$$-\nabla \cdot \boldsymbol{\Gamma} \nabla u = f, \quad (x, y) \in \Omega \tag{25}$$

where $f = f(x, y)$ is a scalar forcing function, $u = u(x, y)$ is the unknown scalar function, and $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}(x, y)$ is a tensor that typically represents diffusivity, transmissivity, or hydraulic conductivity. General Robin conditions for Eq. (25) are

$$au + \vec{n} \cdot \Gamma \nabla u = g, \quad (x, y) \in \partial\Omega, \tag{26}$$

for $a \in \mathbb{R}$, $g = g(x, y) \in \mathbb{R}$, and $\vec{n}$ is an outward normal vector to the boundary $\partial\Omega$. In two-dimensions, $\boldsymbol{\Gamma}$ is represented by the following rank-2 symmetric tensor:

$$\boldsymbol{\Gamma} = \begin{bmatrix} \gamma_{xx} & \gamma_{xy} \\ \gamma_{yx} & \gamma_{yy} \end{bmatrix}, \tag{27}$$

where $\gamma_{xx}, \gamma_{xy}, \gamma_{yx}, \gamma_{yy} \neq 0$ and $\gamma_{xy} = \gamma_{yx}$. In the case that $\boldsymbol{\Gamma} = kI$, with $k \in \mathbb{R}$, $\boldsymbol{\Gamma}$ is called isotropic. If $\boldsymbol{\Gamma}$ is a diagonal matrix with at least two different entries, $\boldsymbol{\Gamma}$ is orthotropic. Finally, a full $\boldsymbol{\Gamma}$ tensor is described as anisotropic. Note that $\boldsymbol{\Gamma}$ is symmetric and positive definite. Researchers typically simplify models by assuming a diagonal $\boldsymbol{\Gamma}$ tensor. In order to solve more realistic problems, we consider a fully anisotropic $\boldsymbol{\Gamma}$ in our numerical examples (including a discontinuous case).

3.2. Second-order mimetic flux operator

For a diagonal tensor $\boldsymbol{\Gamma}$, Eq. (25) can be approximated by substituting mimetic operators (**D, G**) to obtain:

$$-\mathbf{D}\boldsymbol{\Gamma}\mathbf{G}\, u = f \tag{28}$$

However, for an anisotropic $\boldsymbol{\Gamma}$, a more suitable formulation is required. Our discretization scheme for the case of a general (not necessarily diagonal) tensor proceeds as follows. We introduce the mimetic flux generating operator $\boldsymbol{G^\star}$ as an approximation of the product $\boldsymbol{\Gamma grad}$, based on the mimetic gradient operator and some interpolation. Assuming a two dimensional problem, recall the mimetic gradient operator $\boldsymbol{G_{xy}}$:

$$\boldsymbol{G_{xy}} = \begin{bmatrix} \boldsymbol{S_x} \\ \boldsymbol{S_y} \end{bmatrix}. \tag{29}$$

Using a second order $\mathbf{G}_{xy}$, the second order mimetic flux generating operator $\mathbf{G}^\star$ is defined as:

$$\mathbf{G}^\star = \boldsymbol{\Gamma} \boldsymbol{G_{xy}} = \begin{bmatrix} \boldsymbol{\Gamma S_x} \\ \boldsymbol{\Gamma S_y} \end{bmatrix} \cong \boldsymbol{\Gamma} grad, \tag{30}$$

where

$$\boldsymbol{\Gamma S_x} = \gamma_{xx} \boldsymbol{S_x} + \gamma_{xy} \boldsymbol{I_x} \boldsymbol{S_y}, \tag{31}$$

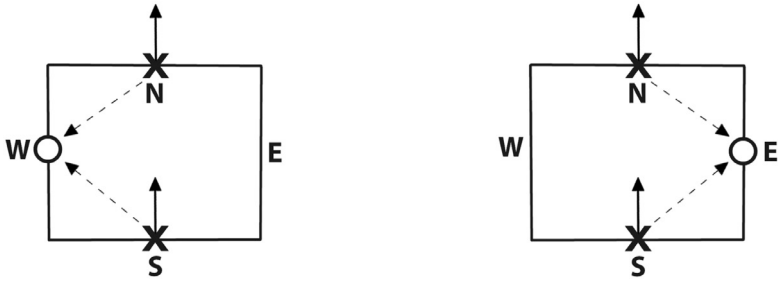
$$\boldsymbol{\Gamma S_y} = \gamma_{yy} \boldsymbol{S_y} + \gamma_{yx} \boldsymbol{I_y} \boldsymbol{S_x} \tag{32}$$

In the case where $\boldsymbol{\Gamma}$ is a diffusivity tensor with units $m^2 s^{-1}$, the gradient operator has units m^{-1} and thus $\boldsymbol{G^\star}$ will have units $m^1 s^{-1}$. When $\boldsymbol{G^\star}$ multiplies a dependent variable u that represents, for example, species concentration with units of $kg^1 m^{-3}$, a diffusion flux will be generated with units $kg^1 m^{-2} s^{-1}$.

The products $\boldsymbol{I_x S_y}$ and $\boldsymbol{I_y S_x}$ in Eqs. (31) and (32) approximate tangential derivatives by way of interpolation. For instance, a first order interpolation computed by $\boldsymbol{I_x S_y}$ for the second-order approximation of $\boldsymbol{\Gamma G}$ is illustrated in Fig. 5. Here, $\boldsymbol{I_x S_y}$ is the average of the north and south face fluxes (depicted as cross marks in 5(a)). This vector average is then assigned to the west face (depicted as a circle in 5(a)). Note that this can be done anywhere in the domain with the exception of the east domain boundary, in which we consider the two closest vertical components on the left (north and south components) as shown in Fig. 5(b).

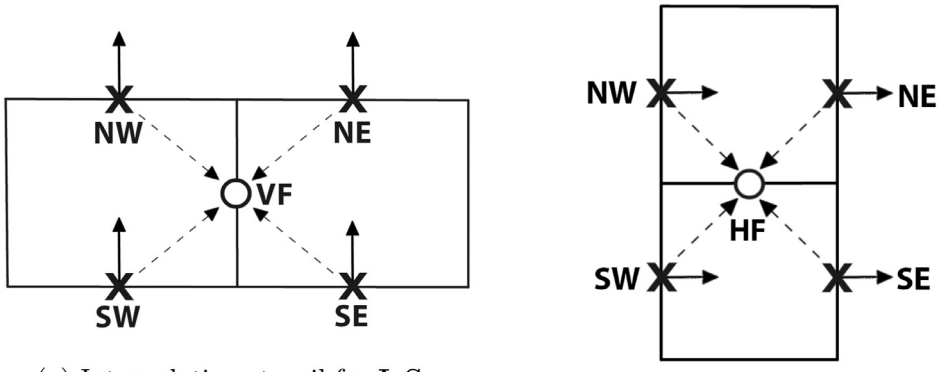
3.2.1. Fourth-order mimetic flux operator

A fourth-order mimetic flux operator $\boldsymbol{G^*}$ can be constructed by using a fourth order mimetic $\boldsymbol{G_{xy}}$ while extending the interpolation stencil used for computing the products $\boldsymbol{I_x S_y}$ and $\boldsymbol{I_y S_x}$ in Eqs. (31) and (32). Vertical fluxes in vertical faces in the interior are now approximated by averages of the surrounding tangential components, four in total: two on the left (NW and SW), and two on the right (NE and SE), depicted as cross marks in Figure (6(a)). Note that here VF (circle) depicts the vertical face where the interpolation is calculated. Similarly, horizontal fluxes in horizontal faces are approximated by taking two tangential components on the bottom (SW and SE) and top (NW and NE), illustrated as a cross mark in Figure (6(b)). Now, the circle HF depicts the horizontal face where interpolation is performed.



(a) Interpolation performed in the interior (b) Interpolation at the east domain boundary

Fig. 5. A first order interpolation computed by $\boldsymbol{I_xS_y}$ for the second-order $\boldsymbol{G^*}$.



(a) Interpolation stencil for $\mathbf{I_xS_y}$ (b) Interpolation stencil for $\mathbf{I_yS_x}$

Fig. 6. A second order interpolation computed by $\boldsymbol{I_xS_y}$ and $\boldsymbol{I_yS_x}$ for the fourth-order $\boldsymbol{G^*}$.

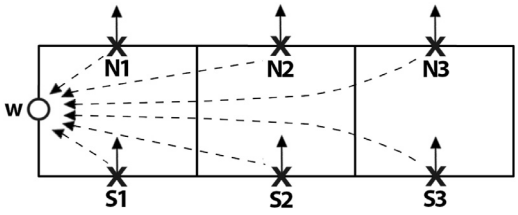


Fig. 7. A second order interpolation computed by $\boldsymbol{I_xS_y}$ for the fourth-order $\boldsymbol{G^*}$ at the west domain boundary.

However, in order to improve accuracy at the boundary, some special treatment is required. First, we increase the interpolation stencil from four to six tangential components. Then, we apply Lagrangian extrapolation. An example of this can be seen in Figure (7), which depicts computation of tangential vertical fluxes at the west domain boundary (depicted as a circle in 7). Six vertical components are taken on the right (depicted as cross marks in 7) as reference data for the extrapolation. We take the tangential components in the normal direction over the entire boundary. Notice that the interpolation scheme now is second order accurate.

3.3. Flux interpolation treatment

In our staggered grid definition, the tensor $\boldsymbol{\Gamma}$ resides at the center of each cell. However, in order to compute fluxes, information given by this tensor has to be available at cell edges so that the product $\boldsymbol{\Gamma}$ ***grad*** can be performed. For this reason, we estimate fluxes at the edges by way of harmonic averages using tensor data from surrounding centers. The use of the harmonic mean over the arithmetic mean to average diffusive coefficients was introduced by Patankar [18] in 1980 and has become standard practice ever since, although research by Kadioglu et al. [19] has shown both averages yield identical accuracy as the mesh is refined. The harmonic average was shown, however, to give better accuracy on heterogeneous (multi-material) domains when coarse meshes were used. Lipnikov et al. [13] proposes a staggered discretization scheme of the diffusion coefficient where they store one value per mesh cell and up to two values per mesh face. For cases where the diffusion coefficient tends to zero or the ratio of adjacent cell centered diffusion coefficients grows very large, Lipnikov shows how using the harmonic averaging of cell-centered diffusion coefficients leads to numerical inaccuracies. Our approach is different, on the vertical edge (horizontal component) we take:

$$\boldsymbol{\Gamma}(x_i, y_{j+\frac{1}{2}}) = \frac{2\boldsymbol{\Gamma}(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}})\boldsymbol{\Gamma}(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})}{\boldsymbol{\Gamma}(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}}) + \boldsymbol{\Gamma}(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})}, \tag{33}$$

where $i = 1, 2, \ldots, m$ and $j = 0, 1, \ldots, n-1$. Similarly, on the horizontal edge (vertical component) we have:

$$\boldsymbol{\Gamma}(x_{i+\frac{1}{2}}, y_j) = \frac{2\boldsymbol{\Gamma}(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}})\boldsymbol{\Gamma}(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})}{\boldsymbol{\Gamma}(x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}}) + \boldsymbol{\Gamma}(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})}, \tag{34}$$

where $i = 0, 1, \ldots, m-1$ and $j = 1, 2, \ldots, n$. An special case occurs at the boundary in which corresponding tensor values are directly assigned from their nearest neighbors, e.g., at left boundary we have:

$$\boldsymbol{\Gamma}(x_1, y_{j+\frac{1}{2}}) = \boldsymbol{\Gamma}(x_{\frac{1}{2}}, y_{j+\frac{1}{2}}), \tag{35}$$

where $j = 1, 2 \ldots, n$.

4. Numerical examples and discussion

The discretization using the mimetic flux operator $\boldsymbol{G^*}$ presented in the previous section for Eq. (28) leads to an un-symmetric lin-

Table 1
Numerical results for Problem 1: Fully anisotropic tensor (using max norm).

	G^*(2nd)		G^*(4th)		Supp. Operator	
$h = \Delta x$	Error	Order	Error	Order	Error	Order
1/16	8.46E-04		1.86E-04		3.74E-03	
1/32	2.29E-04	1.89	2.80E-05	2.73	9.66E-04	1.95
1/64	6.00E-05	1.93	6.11E-06	2.20	2.45E-04	1.98

ear system. This is true even though the continuous operator in Eq. (25) is self-adjoint. Note that the constructed linear system is largely sparse and can be solved using iterative or direct methods. In particular, the linear systems of the numerical examples shown in this section are solved using an Unsymmetric Multi-Frontal method [20], which is a direct method for solving large-scale sparse and unsymmetric linear systems.

4.1. Numerical analysis of truncation errors

In general, the asymptotic truncation error E_h on a grid of $m+1$ nodes (m centers), $h = \Delta x = 1/m$ (assuming a one-dimensional domain of length 1), is estimated by:

$$E_h = ch^q + O(h^{q+1}), \tag{36}$$

where q is the order of the error, and c is the convergence-rate constant (does not depend on m).

Assuming h_1 and h_2 as two different cell sizes, with $h_1 > h_2$, the order of convergence can be estimated as:

$$q \approx \frac{log(E_{h_1}/E_{h_2})}{log(h_1/h_2)}, \tag{37}$$

However, in two-dimensions the cell size h is taken as $h = \max(\Delta x, \Delta y)$. We estimate convergence rate by using the mean-square norm defined as

$$E_{L_2} = ||u - U||_{L_2} = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} \left[u(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) - U_{i+\frac{1}{2},j+\frac{1}{2}} \right]^2 VC_{i+\frac{1}{2},j+\frac{1}{2}} \right)^{\frac{1}{2}}, \tag{38}$$

and the maximum norm

$$E_{max} = ||u - U||_{max} = \max_{i=1,j=1}^{m} |u(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}}) - U_{i+\frac{1}{2},j+\frac{1}{2}}|, \tag{39}$$

with the obvious extension to more dimensions. Where $u(x, y)$ is the exact solution, $U_{i,j}$ is the numerical solution of the finite-difference scheme, and $VC_{i+\frac{1}{2},j+\frac{1}{2}}$ is the corresponding cell volume.

4.2. Problem 1: Fully anisotropic tensor

We consider the example described in [21]. For this problem, we have an anisotropic tensor $\mathbf{\Gamma}$ defined as:

$$\mathbf{\Gamma} = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix},$$

with corresponding forcing function f as

$$f(x, y) = -2(1 + x^2 + xy + y^2)e^{xy}$$

on the unit square domain. The analytical solution and boundary data for this problem is given by $u(x, y) = e^{xy}$. Numerical results displayed in Table 1 show second order convergence for our 2nd order $\boldsymbol{G^*}$ and better than second-order convergence for our proposed 4th order scheme. It is worth noting that while the Support Operator Method (SOM) from [22] converges to second order, SOM is considerably less precise than results obtained using our mimetic operators (both versions). Figs. 8a and 8b depict the computed solution using the 4th-order mimetic flux operator, as well as the analytical solution for this problem. From the figures we can say they are in good agreement.

Table 2
Numerical results for problem 2: Singular Exponential Mobility using 4th order mimetic flux generating operator.

	L_2 norm		Max norm	
$h = \Delta x$	Error	Order	Error	Order
1/10	9.68E-05		3.10E-04	
1/20	1.62E-05	2.57	4.90E-05	2.64
1/40	2.86E-06	2.5	7.52E-06	2.72
1/80	5.55E-07	2.3	1.20E-06	2.64

4.3. Problem 2: singular exponential mobility

We now consider the Singular Exponential Mobility problem on the unit square, taken from [23]. Here, $\mathbf{\Gamma}$ is defined as:

$$\mathbf{\Gamma}(x, y) = \gamma(x, y) \begin{bmatrix} 1 & 1/10 \\ 1/10 & 1 \end{bmatrix},$$

$$\gamma(x, y) = \exp(-\lambda\sqrt{rx + sy})$$

, where the true solution is given by:

$$u(x, y) = \frac{\exp(\lambda\sqrt{rx + sy})(\lambda\sqrt{rx + sy} - 1) + 1}{\exp(\lambda)(\lambda - 1) + 1},$$

with λ constant. Even though solutions for this problem are smooth, they become singular for large λ [23]. Computed and analytical solutions for $\lambda = 30$ are shown in Figs. 9a and 9b, respectively, and they show good agreement. Moreover, numerical results for this problem are displayed in Table 2 (for the mentioned choice of λ). Our results show a convergence rate of $O(h^{5/2})$.

4.4. Problem 3: full and discontinuous tensor

Consider a full and discontinuous tensor on the two-dimensional square domain $\Omega = [-1, 1] \times [-1, 1]$. Tensor $\mathbf{\Gamma}$ has a discontinuity at $x = 0$ and is defined as:

$$\mathbf{\Gamma} = \begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} & x < 0 \\ \alpha \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} & x > 0 \end{cases},$$

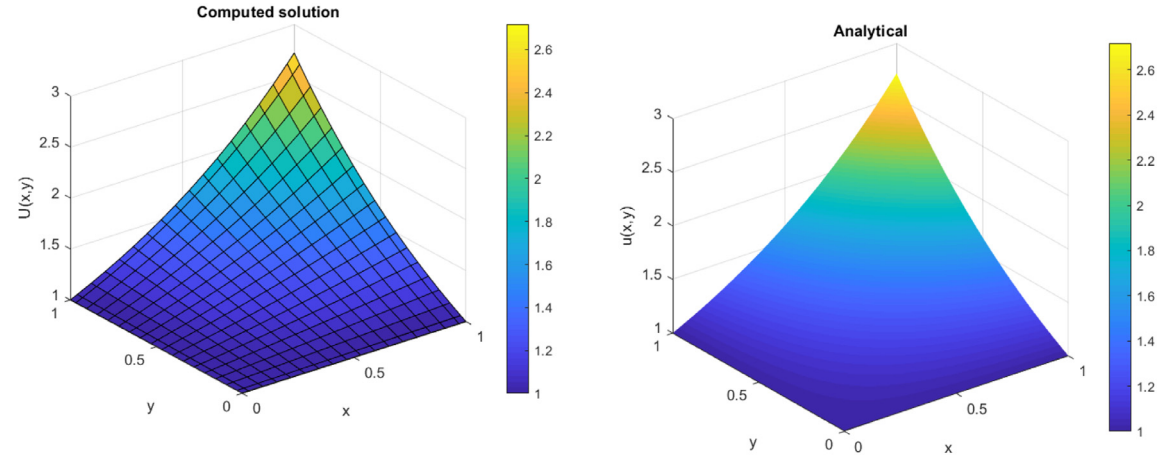
with corresponding right hand side functions as:

$$f(x, y) = \begin{cases} (-2\sin y - \cos y)\alpha x + \sin y, & x < 0 \\ -2\alpha \exp(x) \cos y, & x > 0 \end{cases},$$

and true solution as:

$$u(x, y) = \begin{cases} (2\sin y + \cos y)\alpha x + \sin y, & x < 0 \\ \exp(x) \cos y, & x > 0 \end{cases},$$

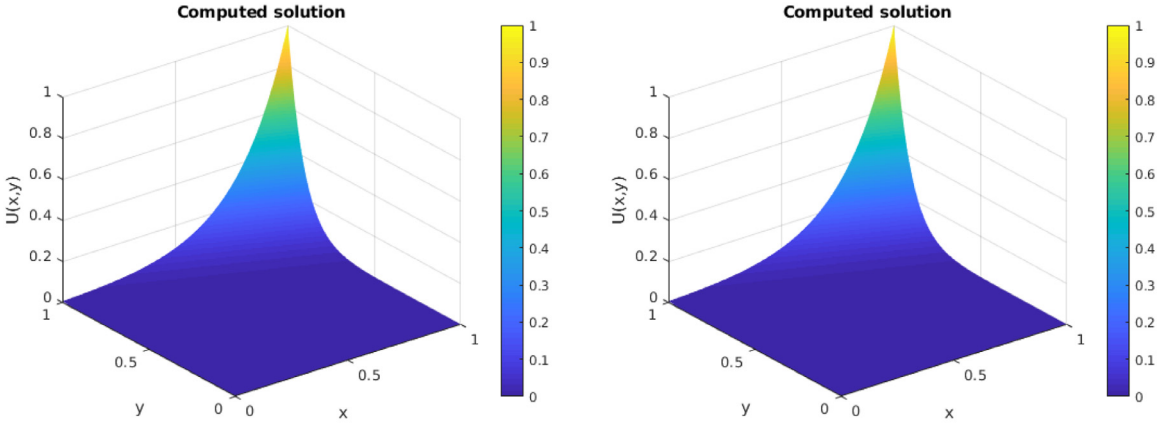
with zero value Dirichlet boundary conditions and $\alpha = 1$. This problem was solved by the Support Operator method and the Vu-Castillo-Grone Mimetic implementation in [22,24]. Tables 3 and 4 display their numerical results, as well as the results obtained by our 4th-order method. Even though it can be seen that our method yields a second order approximation, it offers significant improvement with respect to the aforementioned methods. This improvement increases with the number of cells, reaching two orders of magnitude better for grid sizes of 64 cells ($h = \Delta x = 1/32$). Finally, Fig. 10a and b show our numerical solution for this problem (computed using a 4th order $\boldsymbol{G^*}$) along with its corresponding analytical solution. From the figures we can see they are in good agreement.



(a) Computed solution using a 4^{th} order mimetic flux operator $\mathbf{G}^*$.

(b) Analytical solution

Fig. 8. Solution for Problem 1.



(a) Computed solution

(b) Analytical solution

Fig. 9. Solution for Problem 2 ($\lambda = 30$).

Table 3
Numerical results for problem 3: Full and discontinuous tensor, using 4th order mimetic flux generating operator.

	G(4th)				Vu-Castillo			
	L_2 norm		Max norm		L_2 norm		Max norm	
$h = \Delta x$	Error	Order	Error	Order	Error	Order	Error	Order
1/8	1.53E-04		3.51E-04		4.00E-03		4.70E-03	
1/16	3.46E-05	2.14	6.74E-05	2.38	9.75E-04	2.04	1.60E-03	1.55
1/32	8.75E-06	1.98	1.70E-05	1.99	2.39E-04	2.02	5.30E-04	1.60

4.5. Problem 4: Fully anisotropic tensor with distinct diagonal coefficients

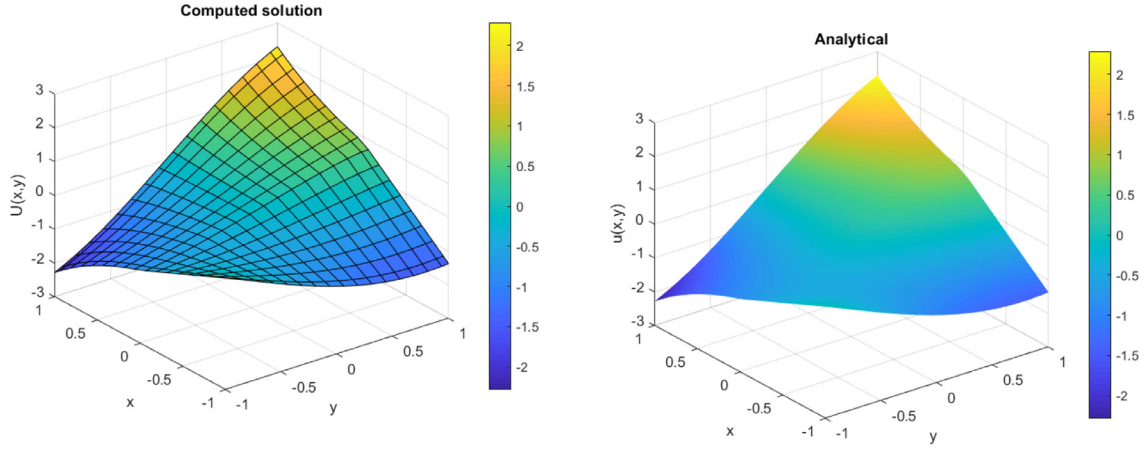
The fourth problem we present is taken from Arbogast et al. [25], and is defined on the unit square with right high side:

$$f(x,y) = -\big(22(y-y^2) - 26(x-x^2) + 18(1-2x)(1-2y)\big).$$

With $\mathbf{\Gamma}$ given by

$$\Gamma = \begin{bmatrix} 11 & 9 \\ 9 & 13 \end{bmatrix},$$

and zero value Dirichlet boundary conditions, the analytical solution is: $u(x,y) = (x-x^2)(y-y^2)$. Fig. 11 depicts the numerical solution for this problem, computed using 4th order ***G****. Table 5 shows our computed results for the 2nd- and 4th-order mimetic flux generating operators ***G****. As shown in this table, results confirm consistent numerical solutions that converge at second and third order, respectively. Please note that the solution for this problem is a parabolic function, this could explain the better convergence of our methods for smooth problems.



(a) Computed solution using a 4^{th} order mimetic flux operator $\mathbf{G}^*$.

(b) Analytical solution

Fig. 10. Solution for Problem 3: Full and discontinuous tensor.

Table 4
Numerical results for problem 3: Full and discontinuous tensor for the support operator method.

	Supp. Oper.	
	L_2 norm	
$h = \Delta x$	Error	Order
colrule 1/8	7.06E-03	
1/16	1.73E-03	2.03
1/32	3.96E-04	2.13

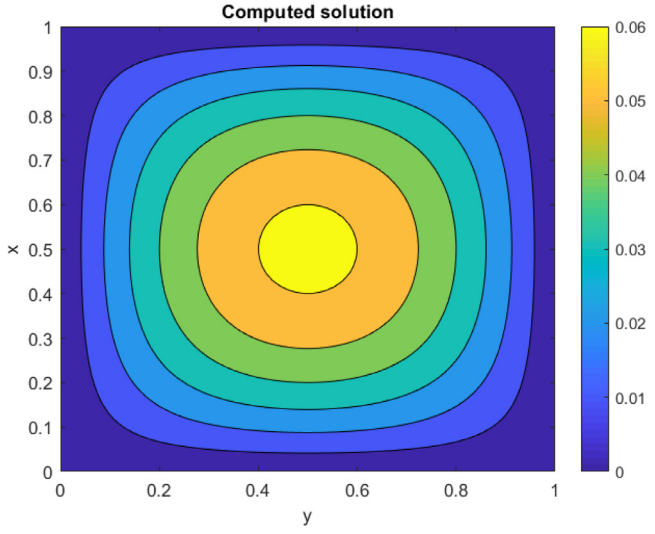


Fig. 11. Computed solution for Problem 4: Fully anisotropic tensor with distinct diagonal coefficients, using 4th order $\boldsymbol{G}^*$.

Table 5
Numerical results for Problem 4: Fully anisotropic tensor with distinct diagonal coefficients, using L_2 norm.

	$\boldsymbol{G}^*$(2nd)		$\boldsymbol{G}^*$(4th)	
$h = \Delta x$	Error	Order	Error	Order
1/16	1.75E-04		4.38E-05	
1/32	4.70E-05	1.90	6.31E-06	2.80
1/64	1.20E-05	1.97	8.49E-07	2.89

5. Conclusions and future work

In this work we have introduced the formulation for the construction of a mimetic flux operator based on Castillo-Corbino mimetic operators in two dimensions. Numerical results obtained for anisotropic elliptic (steady-state) equations show second-order convergence for the 2nd-order operator while consistently achieving better than second order convergence for the 4th-order formulation, with the exception of the discontinuous case. For the problems presented in this paper, our operators exhibit improvement when compared to similar mimetic schemes ([22,24]) as shown in Tables 1 through 4. Also, Table 5 indicates a closer convergence in relation to the expected one of our operators for a smooth problem. Our approach can be naturally extended to sixth order mimetic finite difference operators or higher. A three-dimensional mimetic flux operator is currently in development.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Angel Boada: Conceptualization, Methodology, Software, Writing - original draft, Validation. **Christopher Paolini:** Visualization, Writing - review & editing, Validation. **Jose E. Castillo:** Conceptualization, Writing - review & editing, Validation, Supervision.

References

[1] Bazan C, Abouali M, Castillo J, Blomgren P. Mimetic finite difference methods in image processing. Comput Appl Math 2011;30(3):701–20.
[2] Castillo JE, Grone R. A matrix analysis approach to higher-order approximations for divergence and gradients satisfying a global conservation law. SIAM J Matrix Anal Appl 2003;25(1):128–42.
[3] Córdova L, Rojas O, Otero B, Castillo J. Compact finite difference modeling of 2-d acoustic wave propagation. J Comput Appl Math 2016;295:83–91.
[4] de la Puente J, Ferrer M, Hanzich M, Castillo JE, Cela JM. Mimetic seismic wave modeling including topography on deformed staggered grids. Geophysics 2014;79(3):T125–41.
[5] Rojas O, Otero B, Castillo J, Day S. Low dispersive modeling of rayleigh waves on partly staggered grids. Comput Geosci 2014;18(1):29–43.
[6] Castillo J, Hyman JM, Shashkov MJ, Steinberg S. High-order mimetic finite difference methods on nonuniform grids. In: ICOSAHOM-95, Proc. of the Third

International Conference on spectral and high order methods. Houston, Texas; 1995. p. 347–61.
[7] Castillo JE, Miranda G. High order compact mimetic differences and discrete energy decay in 2d wave motions. In: Spectral and high order methods for partial differential equations ICOSAHOM 2016. Springer; 2017. p. 293–304.
[8] Castillo JE, Miranda GF. Mimetic discretization methods. Chapman and Hall/CRC; 2013.
[9] Corbino J, Castillo JE. High-order mimetic finite-difference operators satisfying the extended gauss divergence theorem. J Comput Appl Math 2020;364:112326.
[10] van't Hof B, Vuik MJ. Symmetry-preserving finite-difference discretizations of arbitrary order on structured curvilinear staggered grids. J Comput Sci 2019;36:101008.
[11] Verstappen R, Veldman A. Symmetry-preserving discretization of turbulent flow. J Comput Phys 2003;187(1):343–68.
[12] Solano-Feo F, Guevara-Jordan J, Rojas O, Otero B, Rodriguez-Cruz R. Second-order mimetic discretization of the seismic wave equation in heterogeneous media. VIII Pan-American workshop: Applied and Computational Mathematics; 2014.
[13] Lipnikov K, Manzini G, Moulton JD, Shashkov M. The mimetic finite difference method for elliptic and parabolic problems with a staggered discretization of diffusion coefficient. Journal of Computational Physics 2016;305:111–26. doi:10.1016/j.jcp.2015.10.031. http://www.sciencedirect.com/science/article/pii/S002199911500707X.
[14] O'Reilly O, Lundquist T, Dunham EM, Nordström J. Energy stable and high-order-accurate finite difference methods on staggered grids. J Comput Phys 2017;346:572–89.
[15] O'Reilly O, Petersson NA. Energy conservative sbp discretizations of the acoustic wave equation in covariant form on staggered curvilinear grids. J Comput Phys 2020:109386.
[16] Castillo J, Hyman JM, Shashkov M, Steinberg S. Fourth-and sixth-order conservative finite difference approximations of the divergence and gradient. Appl Numer Math 2001;37(1–2):171–87.
[17] Abouali M. High-order compact castillo-grone's mimetic operators. In: Report of Computational Science Research Center at San Diego State University CSRCR02; 2012. p. 1–13.
[18] Patankar SV. Numerical heat transfer and fluid flow. Series on computational methods in mechanics and thermal science. Hemisphere Publishing Corporation (CRC Press, Taylor & Francis Group); 1980. 978-0891165224. http://www.crcpress.com/product/isbn/9780891165224.
[19] Kadioglu S, Nourgaliev R, Mousseau V. A comparative study of the harmonic and arithmetic averaging of diffusion coefficients for non-linear heat conduction problems; 2009.
[20] Davis TA. Algorithm 832: umfpack v4. 3—an unsymmetric-pattern multifrontal method. ACM Trans Math Softw (TOMS) 2004;30(2):196–9.
[21] Crumpton P, Shaw G, Ware A. Discretisation and multigrid solution of elliptic equations with mixed derivative terms and strongly discontinuous coefficients. J Comput Phys 1995;116(2):343–58.
[22] Hyman J, Shashkov M, Steinberg S. The numerical solution of diffusion problems in strongly heterogeneous non-isotropic materials. J Comput Phys 1997;132(1):130–48.
[23] Das B, Steinberg S, Weber S, Schaffer S. Finite difference methods for modeling porous media flows. Transp Porous Media 1994;17(2):171–200.
[24] Vu H, Castillo J. Mimetic discretization of elliptic pde with full tensor coefficients, CSRCR 2006-16. San Diego State University, Computational Science Research Center; 2006.
[25] Arbogast T, Wheeler MF, Yotov I. Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite differences. SIAM J Numer Anal 1997;34(2):828–52.