A conservative, consistent, and scalable meshfree mimetic method

Nathaniel Trask, Pavel Bochev, Mauro Perego

Abstract
Mimetic methods discretize divergence by restricting the Gauss theorem to mesh cells. Because point clouds lack such geometric entities, construction of a compatible meshfree divergence remains a challenge. In this work, we define an abstract Meshfree Mimetic Divergence (MMD) operator on point clouds by contraction of field and virtual face moments. This MMD satisfies a discrete divergence theorem, provides a discrete local conservation principle, and is first-order accurate. We consider two MMD instantiations. The first one assumes a background mesh and uses generalized moving least squares (GMLS) to obtain the necessary field and face moments. This MMD instance is appropriate for settings where a mesh is available but its quality is insufficient for a robust and accurate mesh-based discretization. The second MMD operator retains the GMLS field moments but defines virtual face moments using computationally efficient weighted graph-Laplacian equations. This MMD instance does not require a background grid and is appropriate for applications where mesh generation creates a computational bottleneck. It allows one to trade an expensive mesh generation problem for a scalable algebraic one, without sacrificing compatibility with the divergence operator. We demonstrate the approach by using the MMD operator to obtain a virtual finite-volume discretization of conservation laws on point clouds. Numerical results in the paper confirm the mimetic properties of the method and show that it behaves similarly to standard finite volume methods.

Keywords: Meshfree, Generalized moving least squares, Compatible discretizations, Mimetic methods

1. Introduction
The vast majority of numerical methods for PDEs rely on partitioning of the spatial domain into a mesh to both represent the unknown fields and to define the discrete operators acting on these fields. The quality of the resulting mesh-based schemes usually depends on the quality of the underlying mesh and may suffer when the latter deteriorates. For example, shape functions defined on “sliver” elements can result in badly conditioned or even singular stiffness matrices; see [1] and the more recent survey [2].

Accordingly, automated generation of high-quality grids is a task of tremendous practical importance for mesh-based PDE discretizations. Yet, for problems with complex geometries this task remains a major challenge, and creates a significant performance bottleneck in the simulation workflow [3]. As future applications continue to evolve beyond forward simulation, overcoming this bottleneck will be of essence for enabling tasks such as shape optimization and uncertainty quantification. For applications where the domain evolves...
in time, maintaining a high-quality mesh is even more challenging, typically requiring the introduction of remap/remeshing algorithms as elements collapse and tangle under Lagrangian motion. This situation is typical of continuum mechanics problems characterized by large deformation and topology change, such as free-surface flows, hypervelocity impact, and ductile fracture.

Meshfree methods can significantly alleviate or even completely eliminate the mesh generation bottleneck. For example, the domain integration in Galerkin meshfree methods can be performed on substandard meshes, by overlaying the domain with a mesh that is independent of the point cloud [4], or even by a non-conforming set of cells as in the SNIII family of schemes [5]. Collocation meshfree methods on the other hand only require a formation of well-distributed point clouds, which is much easier than generation of high-quality grids, especially for complex geometric domains.

However, since their inception, meshfree methods have struggled to simultaneously maintain rigorous notions of consistency, accuracy, stability, and compatibility that are now commonplace in mesh-based approaches. While significant improvements have been made in recent years addressing the approximation theory of meshfree methods, a meshfree framework mirroring the conservation and consistency properties of mesh-based compatible discretizations has remained elusive. The root cause is that the topological and geometric data employed by typical compatible mesh-based discretizations such as mimetic finite differences (MFD) [6], Whitney elements [7], and finite volumes [8, 9, 10] cannot be assumed readily available in the meshfree setting, especially in the absence of a conforming background mesh. To successfully transplant the properties of these methods to such a setting it is important to recognize that their compatibility is a consequence of discrete differentiation based on the generalized Stokes theorem

\[ \int_{\omega} d\mathbf{u} = \int_{\partial \omega} \mathbf{u}. \] (1)

In (1) \( d \) is the exterior derivative, \( \mathbf{u} \) is a \( k \)-differential form and \( \omega \) is a \( k+1 \) dimensional manifold. This theorem implies that for a sufficiently smooth \( \mathbf{u} \) its derivative at a point \( \mathbf{x} \) can be characterized as

\[ d\mathbf{u}(\mathbf{x}) = \lim_{\mu(\omega) \to 0} \frac{\int_{\partial \omega} \mathbf{u}}{\mu(\omega)}, \] (2)

where \( \omega \) is a \( k+1 \)-dimensional region with measure \( \mu(\omega) \) containing the point \( \mathbf{x} \). Restriction of the right hand side in (2) to a \( k+1 \)-dimensional mesh entity \( \omega_h \) then yields a notion of a discrete derivative that satisfies a discrete Stokes theorem by construction. In particular, the resulting discrete grad, curl and div operators mimic the vector calculus identities curl grad \( \mathbf{u} = 0 \) and div curl \( \mathbf{u} = 0 \); see, e.g., [11].

If one wishes to use this procedure to define the discrete derivatives, then the underlying discretization infrastructure must provide the following three key pieces of information:

- **field data** representing the integrand \( \mathbf{u} \);
- **topological data** describing the action of the boundary operator on a geometric entity \( \omega \), and
- **metric data** providing the measures of the geometric entity \( \omega \) and its boundary \( \partial \omega \).

The first one, i.e., the field data, can be presumed available irregardless of the type of the underlying discretization as both mesh-based and meshfree methods should at a minimum be able to represent the approximate PDE solutions. The second and third information pieces are virtually always available in mesh-based methods. The nodes, edges, faces and cells in most grids form a chain complex, which provides the necessary topological data for compatible discrete grad, curl and div operators; see, e.g., [12, 13]. Likewise, the necessary metric data can be calculated from the mesh description without much difficulty.

This is not the case when the spatial domain is represented by a point cloud. The amount of metric and topological information that can be extracted from the cloud at a reasonable cost is quite limited. One can build, at a linear cost, the \( \epsilon \)-ball graph of the cloud, construct its vertex-to-edge incidence matrix and compute the lengths of its edges. This would provide (2) with just enough information to build a compatible discrete gradient [12] on the point cloud but not much else.
Here we develop a computationally efficient and scalable approach to generate the topological and metric
data for (2) needed to define a compatible divergence operator on point clouds. Using as a template a mimetic
divergence on a primal-dual mesh, we employ a virtual primal-dual mesh to construct an abstract meshfree
version of this operator by contraction of field moments, associated with the virtual faces, and face moments,
which characterize the latter algebraically.

We consider two instantiations of this abstract mimetic meshfree divergence (MMD) operator. The
first one assumes a background mesh and uses generalized moving least squares (GMLS) [14] to obtain the
necessary field and face moments. This MMD instance is appropriate for settings where a background mesh
exists but its quality is insufficient for a robust and accurate mesh-based discretization. In this case the
mesh is used only to define a boundary operator and to integrate the local GMLS polynomial basis over the
cell faces. Both of these tasks can be performed reliably even on substandard grids.

The second MMD operator does not assume a background mesh and uses instead the $\epsilon$-ball graph
of the point cloud and its formal dual as surrogates for a primal and a virtual dual mesh, respectively. The
face-to-cell incidence matrix of the latter provides the necessary topological information for the divergence
operator. This MMD instance retains the GMLS field moments but defines the virtual face moments in
terms of virtual area potentials, leading to a graph Laplacian problem that can be solved in an efficient and
scalable manner by standard multigrid preconditioners. In so doing we trade a challenging mesh generation
problem for a scalable algebraic one, without sacrificing compatibility with the divergence operator.

In [15] we used the local connectivity graph of each particle and its formal dual to mimic the staggered
arrangement in div-grad stencils on primal-dual grids. The resulting “staggered” meshfree scheme behaves
similarly to mesh-based div-compatible discretizations but is not locally conservative because the primal-dual
mesh surrogate is defined independently for each particle. This paper is a logical extension of [15] aiming to
deliver a meshfree scheme mimicking the conservation properties of traditional compatible discretizations.

While there is an abundant literature on compatible mesh-based discretizations, this is not the case for
meshfree methods. To the best of our knowledge, the extant work comprises the meshless volume scheme
[16], the Uncertain Grid Method (UGM) [17], the meshfree framework for conservation laws in [18], and a
few related references. Of particular interest to us are [17] and [18] because, as in the second MMD instance,
they define the necessary metric data in a purely algebraic fashion without assuming a background grid.

The UGM appears to be the first example of a locally conservative meshfree scheme, which uses virtual
global geometric entities characterized algebraically by numbers representing their “measures”. In UGM, the term
“uncertain” refers to an “uncertain face” between two neighboring particles, which is similar to our virtual
faces. The meshfree framework in [18] also seeks sets of numbers that can be interpreted as measures of
virtual volumes, centered at each particle, and the areas of their virtual faces, respectively. The single most
important difference between these works and our approach is in the type of algebraic problems that generate
the necessary metric data. Both [17] and [18] find this data by solving global constrained optimization
problems. In UGM this problem is a linear program solved by a primal-dual logarithmic barrier method,
whereas in [18] it is a quadratic program (QP) requiring a QP-specific solver. In contrast, our approach
requires solution of several graph Laplacian problems, which can be accomplished in linear time.

We have organized the paper as follows. Section 2 introduces notation and reviews the necessary tech-
nical background. Section 3 formulates the abstract MMD operator and states general conditions for its
consistency and accuracy, while Section 4 presents its two instances. Efficient and scalable computation of
the virtual face moments necessary for the second MMD instance is discussed in Section 4.2.3. In Section 5
we use the MMD operator to discretize a model conservation law problem and then in Section 6 we provide
some representative numerical examples. We summarize our findings and discuss future work in Section 7.

2. Preliminaries

2.1. Notation

Throughout the paper upper case fonts are reserved for function spaces, operators and sets of various
entities, while lower case fonts stand for scalar fields, linear functionals, indices, etc. We denote the standard
Euclidean norm on $\mathbb{R}^d$ by $| \cdot |$ and use bold face fonts to denote vector quantities, e.g., $x = (x_1, \ldots, x_d)$ is a
point in the Euclidean space \( \mathbb{R}^d \), \( e_{ij} = (x_i, x_j) \) is an edge connecting two such points, \( u = (u_1, \ldots, u_d) \) is a vector field in \( \mathbb{R}^d \), and \( n = (n_1, \ldots, n_d) \) is a unit normal vector, i.e., \( |n| = 1 \).

As usual, \( C^k(\Omega) \) is the space of all \( k \)-continuously differentiable functions, and \( P_m(\mathbb{R}^d) \), or simply \( P_m \) is the space of all multivariate polynomials of degree less than or equal to \( m \). We denote the standard norm on \( C^k(\Omega) \) and its restriction to \( \omega \subset \Omega \) by \( \| \cdot \|_{C^k(\Omega)} \) and \( \| \cdot \|_{C^k(\omega)} \), respectively.

Figure 1: Point cloud discretization of a domain \( \Omega \). The solid dots represent the boundary points forming the set \( X_\Gamma \) whereas the circles represent the interior points forming the set \( X_0 \).

Let \( \Omega \subset \mathbb{R}^d \), \( d = 2, 3 \) be a bounded connected region with a Lipschitz continuous boundary \( \Gamma = \partial \Omega \). We assume that \( \Gamma \) may be partitioned into a collection of \( p_\Gamma \) faces \( \Gamma = \bigcup \Gamma_j \), each having maximal diameter \( h \), measure \( \mu_j \), centroid \( x_j \) and outward unit normal \( n_j \). The collection of these centroids forms the set \( X_\Gamma \) of all boundary particles. In the interior of the domain, we assume a set \( X_0 \) containing \( p_0 \) interior particles. The union \( X = X_0 \cup X_\Gamma \) has \( p_\Omega = p_0 + p_\Gamma \) particles and defines the meshfree discretization of \( \Omega \). Since by assumption \( \Omega \) is bounded we can choose the coordinate system in such a way that the point cloud is strictly contained in the first quadrant, i.e., \( x_k > 0, k = 1, \ldots, d \) for all \( x \in X \). We denote the subset of \( X \) contained in an entity \( \omega \) by \( X_\omega := \omega \cap X \). Whenever required, boundary conditions will be imposed on simply connected non-empty subsets \( \gamma \subseteq \Gamma \). Without loss of generality we shall assume that \( \gamma \) and its complement \( \gamma' := \Gamma \setminus \gamma \) are exact unions of boundary segments \( \Gamma_j \) and define their indicator functions as

\[
\chi_i(\gamma) = \begin{cases} 
\Gamma_i & \text{if } v_i \in \gamma \\
\emptyset & \text{otherwise}
\end{cases}
\quad \text{and} \quad
\chi_i(\gamma') = \begin{cases} 
\Gamma_i & \text{if } v_i \in \gamma' \\
\emptyset & \text{otherwise}
\end{cases},
\]

respectively. The quality of the point cloud \( X \) can be characterized by its fill distance [14, p.14] given by

\[
h_X = \sup_{x \in \Omega} \min_{x_i \in X} |x - x_i|,
\]

and its separation distance [14, p.41] defined as

\[
q_X = \frac{1}{2} \min_{i \neq j} |x_i - x_j|.
\]

We shall assume that \( X \) is quasi-uniform, namely that there exists \( c_{qu} > 0 \) such that

\[
q_X \leq h_X \leq c_{qu} q_X.
\]

We denote the spaces of all discrete scalar and vector fields on \( X \) by \( S^h \) and \( U^h \), respectively. These fields represent functions by their point samples on \( X \), which we group as

\[
S^h \ni u^h = \{ u_i | x_i \in X_0 \} \cup \{ u_i | x_i \in X_\Gamma \} \cup \{ u_i | x_i \in X_\gamma \} \in \mathbb{R}^{p_\Omega}, \quad \text{and}
\]

\[
U^h \ni u^h = \{ u_i | x_i \in X_0 \} \cup \{ u_i | x_i \in X_\Gamma \} \cup \{ u_i | x_i \in X_\gamma \} \in \mathbb{R}^{dp_\Omega},
\]

respectively.
respectively. Note that \( u^h = (u^h_1, \ldots, u^h_d) \), where \( u^h_k \in S^h_k \), \( k = 1, 2, \ldots, d \). We denote by \( S^h_\gamma \) and \( U^h_\gamma \) the sets of all discrete scalar and vector fields defined by their point values in \( X_0 \cup X_\gamma \), i.e.,

\[
S^h_\gamma \supset u^h_\gamma = \{ u_1 \mid x_1 \in X_0 \} \cup \{ u_1 \mid x_1 \in X_\gamma \} \in \mathbb{R}^{p_0 - p_\gamma}, \quad \text{and}
\]

\[
U^h_\gamma \supset u^h_\gamma = \{ u_1 \mid x_1 \in X_0 \} \cup \{ u_1 \mid x_1 \in X_\gamma \} \in \mathbb{R}^{d(p_0 - p_\gamma)}.
\]

We use \( S^h_\gamma \) or \( U^h_\gamma \) whenever a scalar field \( u(x) \) or a vector field \( u(x) \) is prescribed on \( \gamma \). Let

\[
\| u^h_\gamma \|_{l^\infty, \gamma} := \max_{\gamma} |u^h_k|.
\]

We equip \( S^h_\gamma \) and \( S^h_\gamma \) with the norms

\[
\| u^h_\gamma \|_{S^h_\gamma} := \| u^h_\gamma \|_{l^\infty, \gamma} \quad \text{and} \quad \| u^h_\gamma \|_{S^h_\gamma} := \| u^h_\gamma \|_{l^\infty, \Omega \setminus \gamma},
\]

respectively, and similarly for \( U^h_\gamma \) and \( U^h_\gamma \).

### 2.2. GMLS essentials

Generalized moving least squares (GMLS) is a non-parametric regression technique for the approximation of bounded linear functionals from scattered samples of their arguments [14]. GMLS has been used in meshfree collocation schemes to approximate point values of derivatives; see [19], [20], [15] and the references therein. Here we shall apply GMLS to approximate integrals as our goal is to develop a meshfree divergence operator based on the definition (2). We provide a brief summary of the GMLS framework, specialized to our needs. The abstract GMLS setting comprises the following key ingredients; see [14, Section 4.3]:

- a function space \( U \) with a dual \( U^* \);
- a finite dimensional space \( \Phi = \text{span}\{\phi_1, \ldots, \phi_q\} \subset U \);
- a finite set of linear functionals \( \Lambda = \{\lambda_1, \ldots, \lambda_n\} \subset U^* \); and
- a correlation (kernel) function \( w : U^* \times U^* \mapsto \mathbb{R}^+ \cup \{0\} \).

The set \( \Lambda \) is assumed to be \( \Phi \)-unisolvent, that is

\[
\{ \phi \in \Phi \mid \lambda_i(\phi) = 0, i = 1, \ldots, n \} = \{0\}.
\]

Given a target functional \( \tau \in U^* \) and an arbitrary \( u \in U \), GMLS seeks an approximation \( \tilde{\tau}(u) \) of \( \tau(u) \) in terms of the sample set \( \lambda(u) = (\lambda_1(u), \ldots, \lambda_n(u)) \in \mathbb{R}^n \), such that \( \tilde{\tau}(\phi) = \tau(\phi) \) for all \( \phi \in \Phi \). We call such a GMLS approximation \( \Phi \)-reproducing. To describe the GMLS solution to this problem we introduce the vector \( \tau(\phi) \in \mathbb{R}^q \) with elements

\[
(\tau(\phi))_i = \tau(\phi_i), \quad i = 1, \ldots, q,
\]

the diagonal weight matrix \( W(\tau) \in \mathbb{R}^{n \times n} \) with element \( W_{ii}(\tau) = w(\tau; \lambda_i) \), and the basis sample matrix \( B \in \mathbb{R}^{n \times q} \) with elements

\[
B_{ij} = \lambda_i(\phi_j); \quad i = 1, \ldots, n; \quad j = 1, \ldots, q.
\]

In what follows, \( \| \cdot \|_{W(\tau)} \) denotes the Euclidean norm on \( \mathbb{R}^n \) weighted by \( W(\tau) \), i.e.,

\[
\| b \|_{W(\tau)}^2 = B^T W(\tau) b \quad \forall b \in \mathbb{R}^n.
\]

One can then show that the \( \Phi \)-reproducing GMLS approximation of the target functional is given by

\[
\tilde{\tau}(u) := c(u) \cdot \tau(\phi),
\]

where the GMLS coefficient vector \( c(u) \in \mathbb{R}^q \) solves the weighted least-squares problem

\[
c(u) = \arg \min_{c \in \mathbb{R}^q} \frac{1}{2} \| Bc - \lambda(u) \|_{W(\tau)}^2
\]

\[
= (B^T W(\tau) B)^{-1} (B^T W(\tau)) \lambda(u).
\]

We refer to previous work [15] for details regarding the efficient and stable calculation of \( c(u) \).
2.3. Mimetic divergence operator

In Section 3 we define an abstract meshfree divergence operator by mimicking a mimetic divergence operator on a primal-dual mesh [21, 10, 22]. Remark 2 explains the appropriateness of this choice. To describe the template operator assume that $\Omega$ is equipped with discretization infrastructure comprising

- a primal mesh $\tau_h(V,E)$ with vertices $V = \{v_i\}_{i=1}^n$ and edges $E = \{e_{ij}\}$; $e_{ij} = (v_i, v_j) \in V \times V$,
- a topologically dual mesh $\tau_h^e(C,F)$ with cells $C = \{\omega_i\}_{i=1}^n$ and affine faces $F = \{f_{ij}\}$, and
- a pair of discrete spaces $\tilde{U}_h$ and $\tilde{S}^h$ with elements
  \[ u^h = \{u_{ij} \in \mathbb{R} | f_{ij} \in F \} \cup \{u_{\Gamma_i} \in \mathbb{R} | \Gamma_i \in \gamma' \} \quad \text{and} \quad s^h = \{\phi_i \in \mathbb{R} | \omega_i \in C\}, \]

respectively, representing the domain and the range of the discrete divergence, respectively.

Every dual cell $\omega_i$ corresponds to a primal vertex $v_i$ and has a measure $\mu_i$. We assume that every dual face $f_{ij}$ corresponding to a primal edge $e_{ij}$ has a unit normal $n_{ij} = e_{ij}/|e_{ij}|$, and oriented measure $\mu_{ij} = \int_{f_{ij}} dS$. If $\omega_i$ and $\omega_j$ are the dual cells corresponding to the endpoints of this edge, then $f_{ij} = \partial \omega_i \cap \partial \omega_j$. The space $\tilde{U}_h$ approximates vector fields in $H^0(div,\Omega) = \{u \in H(div,\Omega) \cap (C^0(\Omega))^d | u = u_\gamma \text{ on } \gamma \}$ by their scalar projections onto $n_{ij}$ or $n_{ji}$, whereas $\tilde{S}^h$ approximates $L^2(\Omega)$ functions by their dual cell averages. Without a loss of generality we shall assume that the restriction $\mathcal{R}(u) \in \tilde{U}_h^h$ of a vector field $u$ is given by $u_{ij} = u_{ji} = u \cdot n_{ij}$ for $i < j$. The neighborhood of $v_i$ is defined as

\[ V_i = \{v_j \in V | e_{ij} = (v_i, v_j) \in E\} \]

and contains all mesh vertices connected to $v_i$ by an edge. If a vertex $v_i \in \Gamma$ then the boundary of its dual cell $\omega_i$ has a non-empty intersection with $\Gamma$, which we denote as $\Gamma_i := \partial \omega_i \cap \Gamma$. As in Section 2.1 we shall always assume that $\gamma$ and $\gamma'$ are exact unions of such segments. It follows that

\[ \partial \omega_i = \left( \bigcup_{j \in V_i} f_{ij} \right) \cup \chi_i(\gamma') \cup \chi_i(\gamma). \]  

Restriction of (2) to dual cells $\omega_i$, followed by approximation of the integrals on $\partial \omega_i$ by single point quadrature yields a discrete divergence operator $\text{DIV} : \tilde{U}_h^h \to \tilde{S}^h$, whose action on $u^h \in \tilde{U}_h^h$ is given by

\[ (\text{DIV} u^h)_i := \frac{1}{\mu_i} \left[ \sum_{f_{ij} \in \partial \omega_i} u_{ij} \mu_{ij} + u_{\chi_i(\gamma')} \mu_{\chi_i(\gamma')} + \int_{\chi_i(\gamma)} u \cdot n dS \right] \quad \forall \omega_i \in C. \]  

In this operator

- the field data is provided by the degrees-of-freedom in $\tilde{U}_h^h$, i.e., the sets of numbers $\{u_{ij}\}$ and $\{u_{\Gamma_i}\}$, such that $u_{ij} = u_{ji}$;
- the metric data is provided by the cell volumes and the oriented face areas in the dual mesh, i.e., the sets of numbers $\{\mu_i\}$, $\{\mu_{ij}\}$ and $\{\mu_{\Gamma_i}\}$, respectively, such that $\mu_{ij} = -\mu_{ji}$;
- the topological data is provided by formula (10).

**Remark 1.** In mixed discretizations [23] of, e.g., diffusion problems, $\{u_{ij}\}$ and $\{u_{\Gamma_i}\}$ are degrees-of-freedom describing an $H(\text{div})$-conforming discretization of a flux field. In contrast, in finite volume discretizations of conservation laws [24] the sets $\{u_{ij}\}$ and $\{u_{\Gamma_i}\}$ represent values derived from another set of degrees-of-freedom, usually describing a scalar dependent variable such as a pressure field.

One can show that (11) is first-order accurate, i.e., for all sufficiently regular vector fields $u$ there holds

\[ (\text{DIV} \mathcal{R}(u))_i = \nabla \cdot u(x_i) + O(h). \]  

The following result confirms that (11) satisfies a discrete divergence theorem, i.e., that it is compatible.
Lemma 1. Consider a collection $C_{\omega}$ of dual cell indices and define $\omega \subseteq \Omega$ as

$$\omega = \bigcup_{i \in C_{\omega}} \omega_i.$$ 

Let $\partial \omega_0 = \partial \omega \cap \Omega$, $\partial \omega_\gamma = \partial \omega \cap \gamma$, and $\partial \omega_{\gamma'} = \partial \omega \cap \gamma'$. Then,

$$\sum_{i \in C_{\omega}} \mu_i(DIV u^h)_i = \sum_{i \in C_{\omega}} \left( \sum_{f_{ij} \in \partial \omega_i} u_{ij} \mu_{ij} + \sum_{\Gamma_k \in \partial \omega_{\gamma'}} u_{\Gamma_k} \mu_{\Gamma_k} + \sum_{\Gamma_k \in \partial \omega_\gamma} \int_{\Gamma_k} u \cdot n_k dS \right).$$

Proof. Summing (11) over the cells in $\omega$ gives

$$\sum_{i \in C_{\omega}} \mu_i(DIV u^h)_i = \sum_{i \in C_{\omega}} \left[ \sum_{f_{ij} \in \partial \omega_i} u_{ij} \mu_{ij} + \sum_{\Gamma_k \in \partial \omega_{\gamma'}} u_{\Gamma_k} \mu_{\Gamma_k} + \sum_{\Gamma_k \in \partial \omega_\gamma} \int_{\Gamma_k} u \cdot n_k dS \right].$$

Since $\mu_{ij} = -\mu_{ji}$ all terms in the double sum above corresponding to pairs of adjacent cells cancel each other, leaving only the terms for which $f_{ij} \in \partial \omega_0$. This completes the proof.

Global conservation in $\Omega$ a direct consequence of Lemma 1. For simplicity we state it with $\gamma = \Gamma$.

Corollary 1. Assume that $u^h \in \tilde{U}_h^\gamma$ and let $u \in H_{\Gamma}(\text{div}, \Omega)$ be given. Then,

$$\sum_{\omega \in C} \mu_i(DIV u^h)_i = \int_{\Gamma} u \cdot n dS.$$ 

Proof. The result follows by setting $\omega = \Omega$ in Lemma 1 and noting that $\partial \omega_0 = \emptyset$, $\partial \omega_{\gamma'} = \emptyset$, and $\partial \omega_\gamma = \Gamma$.

3. An abstract meshfree mimetic divergence operator

We consider a meshfree discretization infrastructure comprising

- a point cloud $X = X_0 \cup X_\Gamma$, representing the computational domain $\Omega$, and
- discrete spaces $S^h$ and $U_\gamma^h$, representing scalar and vector fields by their point samples on the cloud.

Our goal is to endow this infrastructure with an abstract meshfree mimetic divergence (MMD) operator whose properties mirror those of its mesh-based prototype (11). Thus, we seek a mapping $DIV : U_\gamma^h \to S^h$ through a virtual primal-dual mesh complex. To reduce notational clutter we reuse the nomenclature from Section 2.3. We call a collection of integer indices $V = \{i\}_{i=1}^{p_E}$ a virtual vertex set and refer to its elements as $v_i$. A virtual edge is simply an ordered pair of virtual vertices, i.e.,

$$e_{ij} = \{(ij) \mid (ij) \in V \times V \}.$$ 

A collection of $p_E$ virtual edges forms a set $E$, which together with $V$ defines the virtual primal mesh $\tau_k(V,E)$. To every $v_i \in V$ and $e_{ij} \in E$ we assign a virtual dual cell $\omega_i$ and a virtual dual face $f_{ij}$, respectively. The sets of these dual entities are denoted by $C$ and $F$, respectively, and they form a virtual topologically dual mesh $\tau^*_h(C,F)$. A practically useful virtual boundary operator $\partial C \to F$ should be able to recover the physical domain boundary $\Gamma$. This requires some degree of association between the virtual
mesh structures and the physical domain $\Omega$. A simple and effective way to establish such an association is to tie every virtual vertex to a point in the cloud $X$. This induces a partition of $V$ into interior vertices $V_0 = X_0$ and boundary vertices $V_T = X_T$, and prompts the following definition:

$$\partial \omega_i = \begin{cases} \{ f_{ij} \}, & j \in V_i \quad \text{if } i \in V_0 \\ \{ f_{ij} \} \cup \Gamma_i, & j \in V_i \quad \text{if } i \in V_T \end{cases}; \quad \forall \omega_i \in C. \quad (15)$$

This definition allows us to write the virtual cell boundary in the same form as in (10), i.e.,

$$\partial \omega_i = \left( \bigcup_{j \in V_i} f_{ij} \right) \cup \chi_i(\gamma') \cup \chi_i(\gamma). \quad (16)$$

Note that while $\omega_i$ is a purely virtual entity, its boundary may contain “real” geometric entities from $\Gamma$.

Consider next the field data abstraction. To mimic (11) the domain of the MMD operator should be associated with the virtual faces $F$, whereas its range should contain fields living on the virtual cells $C$. To ensure first-order accuracy the latter should be piecewise constant with respect to $C$. Thus, we can reuse the definition of $\tilde{S}^h$ from (9) with the understanding that each degree-of-freedom is now assigned to a virtual, rather than to a physical, dual cell. Likewise, the domain space should be able to reproduce exactly the preimage of $\tilde{S}^h$, i.e., any vector field $u$ such that $\nabla \cdot u \in \tilde{S}^h$. In the mesh-based case this can be accomplished by a single3 degree-of-freedom per face. However, this may not be enough in the meshfree setting and so we shall allow for $n_F \geq 1$ field moments per virtual face $f_{ij} \in F$ and boundary segment $\Gamma_i \in \gamma$. In sum, we define the domain $U^h_\gamma$ and the range $\tilde{S}^h$ for the abstract MMD operator by modifying (9) to

$$u^h = \{ u_{ij} \in \mathbb{R}^{n_F} | f_{ij} \in F \} \cup \{ u_{\Gamma_i} \in \mathbb{R}^{n_F} | \Gamma_i \in \gamma' \} \quad \text{and} \quad s^h = \{ \phi_i \in \mathbb{R} | \omega_i \in C \}, \quad (17)$$

respectively, with the understanding that all dual mesh entities in (17) are virtual.

Finally, let us consider the abstraction for the metric data. We seek this abstraction in the form of virtual face and cell moments, i.e., sets of numbers assigned to every element of $F$ and $C$. To allow contraction of these moments with the field moments in (17) we chose the former to be the algebraic duals of the latter. Thus, we assign a real number $\mu_i$ to every virtual dual cell $\omega_i \in C$, a real vector $\mu_{ij} \in \mathbb{R}^{n_F}$ to every virtual dual face $f_{ij} \in F$ and a real vector $\mu_{\Gamma_i} \in \mathbb{R}^{n_F}$ to every boundary segment $\Gamma_i \in \gamma'$. Following (11) we then define an abstract operator $DIV : \tilde{U}^h_\gamma \rightarrow \tilde{S}^h$ by contracting the metric and the field data:

$$(DIV u^h)_i := \frac{1}{\mu_i} \left[ \sum_{f_{ij} \in \partial \omega_i} u_{ij} \cdot \mu_{ij} + u_{\chi_i(\gamma')} \cdot \mu_{\chi_i(\gamma')} + \int_{\chi_i(\gamma)} u \cdot n dS \right] \quad \forall \omega_i \in C. \quad (18)$$

This operator acts on the field moments $\{ u_{ij} \}$ and $\{ u_{\Gamma_i} \}$ rather than on the point samples $U^h_\gamma$ that are the “native” representation of vector fields on the point cloud. Thus, deployment of (18) in our meshfree infrastructure requires a particle-to-virtual face data transfer operator$^4$

$$T_F : U^h_\gamma \rightarrow \tilde{U}^h_\gamma; \quad (T_F u^h)|_f = t_f(u^h) \in \mathbb{R}^{n_F}, \quad f \in \{ f_{ij} \} \cup \{ \Gamma_j | x_j \in \gamma' \}. \quad (19)$$

which maps point samples into field moments. Combining the action of this operator with (18) yields the desired abstract MMD operator $DIV : \tilde{U}^h_\gamma \rightarrow \tilde{S}^h$, where

$$(DIV u^h)_i := \frac{1}{\mu_i} \left[ \sum_{f_{ij} \in \partial \omega_i} t_{ij}(u^h) \cdot \mu_{ij} + t_{\chi_i(\gamma')}(u^h) \cdot \mu_{\chi_i(\gamma')} + \int_{\chi_i(\gamma)} u \cdot n dS \right] \quad \forall \omega_i \in C. \quad (20)$$

The following section studies the properties of (20).

$^3$For example, the lowest-order Raviart-Thomas elements on $d$-simplices [25], also known as Whitney 2-forms [7], are incomplete linear polynomial fields of the form $a + bx$ which have $d + 1$ degrees-of-freedom.

$^4$Formally we also need a data transfer operator $T_F : \tilde{S}^h \rightarrow S^h$ to map the discrete divergence back to the cloud points. However, since in the present context $S^h \equiv \tilde{S}^h$, this operator is the identity and is omitted for simplicity.
3.1. Analysis of the abstract MMD operator

To ensure that (20) has the same mimetic properties as its mesh-based prototype we require that

**T.1** The virtual face volumes satisfy $\mu_i > 0$, $\mu_i = O(h_i)$, and $\sum_i \mu_i = \mu(\Omega)$.

**T.2** The virtual face moments $\{\mu_{ij}\}$ are antisymmetric: $\mu_{ij} = -\mu_{ji}$.

**T.3** The data transfer operator (19) is symmetric: $t_{ij}(u^h) = t_{ji}(u^h)$.

**T.1** ensures that the virtual volumes behave like physical volumes. The “topological” requirements **T.2-T.3** are prompted by Lemma 1, which reveals that mimetic properties of (11) hinge on the symmetry of the face degrees-of-freedom $u_{ij}$ and the antisymmetry of the oriented face areas $\mu_{ij}$. The following result shows that **T.1-T.3** are indeed sufficient for the abstract MMD operator to be locally and globally conservative.

**Theorem 1.** Assume that **T.1-T.3** hold. Then the abstract MMD operator (20) satisfies (13) (local conservation) and (14) (global conservation).

**Proof.** Owing to **T.2** and **T.3** the proof of Lemma 1 applies verbatim to the abstract MMD operator as well. The global conservation (14) then follows from **T.1** and (13) as in Corollary 1. $\square$

While conservation is a purely “topological” property that can be secured by independent conditions on the field and face moments, this is not the case with the consistency of the MMD operator, which requires these moments to work well together. The following two definitions introduce conditions that coordinate the properties of the metric and the field data to ensure the desired first-order accuracy.

**Definition 1.** A set of virtual cell volumes and virtual face moments $\{\mu_i, \mu_{ij}, \mu_{ij}, \}$ and a data transfer operator $T_F : U^h_\gamma \to \tilde{U}^h_\gamma$ are called a $P_1$-reproducing pair if for every $p \in (P_1)^d$ and its point sample $p^h \in U^h_\gamma$ there holds

$$\nabla \cdot p \big|_{x_i} = \frac{1}{\mu_i} \left[ \sum_{f_{ij} \in \partial \omega_i} t_{ij}(p^h) \cdot \mu_{ij} + t_{ij}(p^h) \cdot \mu_{ij} + \int_{\gamma_i(\gamma)} p \cdot n dS \right] \quad \forall \omega_i \in C. \quad (21)$$

In other words, $\{T_F; \{\mu_i, \mu_{ij}, \mu_{ij}, \} \}$ is a $P_1$-reproducing pair iff for every $p \in (P_1)^d$ there holds

$$(\text{DIV} p^h)_i = \nabla \cdot p \big|_{x_i} \quad \forall x_i \in X_0 \cup X_\gamma.$$

**Definition 2.** A set of virtual face moments $\{\mu_{ij}, \mu_{ij}, \}$ and a data transfer operator $T_F : U^h_\gamma \to \tilde{U}^h_\gamma$ are called a locally Lipschitz-continuous pair on $X$ if there exist balls $\{B^2_f\}$ of radius $O(h_i)$, centered at the barycenters of the virtual faces $f \in F$ and a constant $C > 0$, such that for every $u, v \in (C^2(\Omega))^d$ with point samples $u^h, v^h \in U^h_\gamma$ there holds

$$|t_f(u^h) \cdot \mu_f - t_f(v^h) \cdot \mu_f| \leq C h^{-1}_X \|u^h - v^h\|_{L^\infty,X_{n_f}}, \quad \forall f \in \{f_{ij} \cup \Gamma_{ij} \cup x_j \in \gamma \}. \quad (22)$$

These two properties are sufficient for the abstract MMD operator to be first-order accurate.

**Theorem 2.** Assume that $\{T_F; \{\mu_i, \mu_{ij}, \mu_{ij}, \} \}$ is a $P_1$-reproducing pair, $\{T_F; \{\mu_{ij}, \mu_{ij}, \} \}$ is a locally Lipschitz continuous pair on $X$, and that **T.1** holds. Then, there exists a constant $C$, independent of the point cloud and such that for every $u \in (C^2(\Omega))^d$ and its point sample $u^h \in U^h_\gamma$ there holds the error bound

$$\|\nabla \cdot u - (\text{DIV} u^h)\|_{L^\infty,X} \leq Ch\|u\|_{C^2(\Omega)}.$$

**Proof.** Consider a linear vector field $p \in (P_1)^d$ with a point sample $p^h \in U^h_\gamma$, and an arbitrary point $x_i \in X$. From the triangle inequality and the $P_1$-reproducing pair assumption it follows that

$$|\nabla \cdot u(x_i) - (\text{DIV} u^h)| \leq |\nabla \cdot u(x_i) - \nabla \cdot p(x_i)| + |\nabla \cdot p(x_i) - (\text{DIV} u^h)|$$

$$= |\nabla \cdot u(x_i) - \nabla \cdot p(x_i)| + |(\text{DIV} p^h)_i - (\text{DIV} u^h)|$$

$$9$$
Consider first the case $\chi_i(\gamma') = \emptyset$. Using (22) then yields the following bound for the second term:

$$
(DIV p^h)_{i} - (DIV u^h)_{i} \leq \frac{1}{\mu_i} \left[ \sum_{F_{ij} \in \partial \omega_i} |t_{ij}(p^h) \cdot \mu_{ij} - t_{ij}(u^h) \cdot \mu_{ij}| + \int_{\chi_i(\gamma)} |p - u| dS \right]
$$

$$
\leq \frac{1}{\mu_i} \left[ Ch_X^{-1} \| p^h - u^h \|_{C^0(X_{h^p})} + \mu(\chi_i(\gamma)) \| p - u \|_{C^0(\chi_i(\gamma))} \right]
$$

where $\mu(\chi_i(\gamma))$ is the measure of $\chi_i(\gamma)$. Assumption T.1 implies that $\mu_i = O(h_X^d)$, while $\mu(\chi_i(\gamma))$ is either zero or $O(h_X^{-1})$. As a result, we have that

$$
(DIV p^h)_{i} - (DIV u^h)_{i} \leq Ch_X^{-1} \left[ \| p^h - u^h \|_{C^0(X_{h^p})} + \| p - u \|_{C^0(\chi_i(\gamma))} \right]
$$

Taking $p$ to be the linear Taylor polynomial of $u$ at the point $x_i$, then yields the bound

$$
\| p - u \|_{C^0(B_{h^p})} + \| p - u \|_{C^0(\chi_i(\gamma))} \leq Ch_X^2 \| u \|_{C^2(\Omega)}
$$

for the second term and the bound

$$
| \nabla \cdot u(x_i) - \nabla \cdot p(x_i) | \leq Ch_X \| u \|_{C^2(\Omega)}
$$

for the first term. This completes the proof for $\chi_i(\gamma') = \emptyset$. When $\chi_i(\gamma') = \Gamma_i$ the extra term $\| p^h - u^h \|_{C^0(X_{h^p})}$ can be estimated by $\| p - u \|_{C^0(B_{h^p})}$. The proof follows by combining this and the above bounds.

4. Meshfree mimetic divergence (MMD) instantiations

In this section we present two instances of the abstract MMD operator. Section 4.1 implements (20) using a background mesh. This hybrid formulation is useful for applications where such a grid is available but its quality is poor. For example, the mesh may contain valid, but almost degenerate, “sliver” elements that render the mesh-based shape functions nearly singular. In this case, a meshfree approach such as (20) can provide a more robust and accurate alternative to traditional mesh-based discretizations by using the mesh only to supply the metric and topological information but not to define the field data and the operators acting on it. Section 4.2 shows how to implement (20) without a background mesh. This MMD instance targets applications where mesh generation creates a computational bottleneck and should be avoided.

4.1. MMD with a background mesh

We assume that the virtual primal-dual mesh complex can be associated with a conforming physical grid on $\Omega$. Existence of a physical dual mesh implies that a set $\{\mu_i\}$ satisfying T.1 is readily available. Thus, to instantiate (20) it remains to find a data transfer operator $T_F$ and virtual face moments $\{\mu_{ij}, \mu_{ij}, \mu_{ij} \}$ such that assumptions T.2-T.3 hold. $\{T_F; \{\mu_i, \mu_{ij}, \mu_{ij} \}; \}$ is a $P_1$-reproducing pair, and $\{T_F; \{\mu_{ij}, \mu_{ij} \}; \}$ is a locally Lipschitz continuous pair on $X$. To that end we shall use GMLS with targets

$$
\tau_F(u) = \int_F u \cdot nFdS,
$$

where $f$ is a dual face or a segment on $\gamma'$, i.e., $f \in F \cup \Gamma_i \{x_i \in \gamma'\}$. We apply GMLS with

- a function space $U = (C^1(\Omega))^d$;
• a reproduction space $\Phi = (P_1)^d$ with dimension $n_1 = d(d + 1)$ and basis $\{\phi_s\}_{s=1}^{n_1}$ with basis functions

$$\phi_s(x) = e_k \phi_s(x); \quad s = (k-1)(d+1) + r; \quad k = 1, \ldots, d; \quad r = 1, \ldots, d+1,$$  

where $e_k$ is the $k$th canonical basis vector in $\mathbb{R}^d$ and $\{\phi_s(x)\}_{s=1}^{d+1}$ is a basis for $P_1$;

• sampling functionals $\lambda_i = \delta_{x_i}$, mapping vector fields $u \in U$ into point samples $u^h \in U^h$ on the cloud;

• kernels $w(\tau_f; \lambda_i) = \kappa(|x_f - x_i|)$ where $x_f$ is the centroid of $f \in F \cup \{\Gamma_j | x_j \in \gamma\}$ and $\kappa(\rho) = (1 - \rho/\epsilon_x)^{1/\nu}$, with $\epsilon_x$ is large enough to obtain $P_1$-unisolvency. The face centroids $\{x_f\}$ are not required to coincide with the midpoints $\{x_{ij}\}$ of the corresponding primal edges.

With these choices the GMLS approximation of (24) is exact for linear polynomials and is given by

$$\tau_f(u) \approx \tilde{\tau}_f(u^h) := c_f(u^h) \cdot \tau_f(\phi); \quad \forall f \in F \cup \{\Gamma_j | x_j \in \gamma\}.$$  

The GMLS coefficients $c_f(u^h) \in \mathbb{R}^{n_1}$ solve the local weighted least-squares problems

$$c_f(u^h) = \arg\min_{b \in \mathbb{R}^{n_1}} \frac{1}{2} |Bb - u^h|^2_{W(\tau_f)},$$  

which transform point samples into “field moments” and define a mapping

$$T_F : U^h_\gamma \to \tilde{U}^h_\gamma; \quad (T_F u^h)|_f := c_f(u^h) \in \mathbb{R}^{n_1}, \quad \forall f \in F \cup \{\Gamma_j | x_j \in \gamma\}.$$  

We chose (28) as the data transfer operator for (20). The second term in (26), i.e., the vector $\tau_f(\phi) \in \mathbb{R}^{n_1}$, contains the integrals of the polynomial basis functions on $f$ and defines the face moments for (20). Setting $\mu_{ij} = \tau_{ij}(\phi)$ and $\mu_{\Gamma_j} = \tau_{\Gamma_j}(\phi)$ in (20) we obtain the following GMLS instance of the MMD operator:

$$\left(\text{DIV} u^h\right)_i = \frac{1}{\mu_i} \left[ \sum_{j \in \partial \omega_i} (c_{ij}(u^h)) \cdot \mu_{ij} + c_{ij}(\gamma) \cdot \mu_{\chi,\gamma} + \int_{\chi,\gamma} u \cdot ndS \right] \quad \forall \omega_i \subset C.$$  

4.1.1. Analysis

Since $\mu_i$ are volumes of physical dual cells on a conforming mesh, assumption T.1 readily holds. We now check the rest of the assumptions of the abstract theory in §3.1. To avoid non-essential technicalities we shall assume that all dual faces are affine.

Lemma 2. The pair $\{T_F, \{\mu_i, \mu_{ij}, \mu_{\Gamma_j}\}\}$ is $P_1$-reproducing.

Proof. Let $p \in (P_1)^d$ and $p^h \in U^h_\gamma$ denote its point sample on the cloud. By construction, the GMLS approximation (26) is exact for linear polynomials and so,

$$c_{ij}(p^h) \cdot \mu_{ij} = \int_{f_{ij}} p \cdot n_{ij} \, dS \quad \text{and} \quad c_{ij}(p^h) \cdot \mu_{\Gamma_j} = \int_{\Gamma_j} p \cdot n_j \, dS.$$  

Inserting these identities in (29), using the divergence theorem and the fact that $\nabla \cdot p$ is constant, yields

$$\left(\text{DIV} p^h\right)_i = \frac{1}{\mu_i} \int_{\partial \omega_i} p \cdot n_i \, dS = \frac{1}{\mu_i} \int_{\omega_i} \nabla \cdot p \, dx = \nabla \cdot p \big|_{x_i}.$$  

Lemma 3. The face moments $\{\mu_{ij}\}$ and the operator $T_F$ satisfy assumptions T.2 and T.3, respectively.

---

5 We refer to [14, Chapter 3] for conditions on the sampling locations $\{x_i\}$ that are sufficient to ensure polynomial reproduction and rigorous approximation error estimates.

11
Proof. To show that $T.3$ holds note that the kernel $w(\tau_{ij}; \lambda)$ is anchored at the centroid $x_{f_{ij}}$ of $f_{ij}$. As a result, (8) implies that $c_{ij}(u^h) = c_{ij}(\tilde{u}^h)$, i.e., $T_F$ is symmetric. For $T.2$ we use that $n_{ij} = -n_{ji}$, and so,

$$(\mu_{ij})_s = \int_{f_{ij}} \phi_s \cdot n_{ij} \, dS = -\int_{f_{ji}} \phi_s \cdot n_{ji} \, dS = -(\mu_{ji})_s; \quad s = 1, \ldots, n_1.$$ 

Lemma 4. The pair $\{T_F, \{\mu_{ij}, \mu_{ji}\}\}$ is locally Lipschitz continuous on $X$.

Proof. Let $u, v \in (C^2(\Omega))^d$ with point samples $u^h, v^h \in U_h^\gamma$. Given $f \in F \cup \{\Gamma_j \mid x_j \in \gamma'\}$ we have that

$$c_f(u^h) \cdot \mu_f = c_f(u^h) \cdot \tau_f(\phi) = \sum_{s=1}^{n_1} c_s^x(u^h) \int_f \phi_s(x) \cdot n \, dS.$$ 

(30)

By assumption, all dual faces are affine and so their unit normals are constant. As a result, $\phi_s(x) \cdot n$ is linear on $f$ and

$$\int_f \phi_s(x) \cdot n \, dS = \phi_s(x_f) \cdot n |f|.$$ 

It follows that

$$c_f(u^h) \cdot \mu_f = \left[ \sum_{s=1}^{n_1} c_s^x(u^h)\phi_s(x_f) \right] \cdot n |f|.$$ 

The expression in the square brackets is the Moving Least Squares (MLS) approximant $\tilde{u}(x_f)$ of $u(x_f)$; see [14, Chapter 4]. Using the fact that $\tilde{u}(x_f)$ is uniformly bounded by the data we obtain

$$|c_f(u^h) \cdot \mu_f - c_f(v^h) \cdot \mu_f| = |\tilde{u}(x_f) \cdot n - \tilde{v}(x_f) \cdot n| |f| \leq C\|u^h - v^h\|_{\infty, X_{h^X}} |f|.$$ 

The proof follows by noting that for physical faces $|f| = O(h^{d-1})$. 

Lemma 2–4 verify all requirements of the abstract theory in §3.1. Thus, we have the following result.

Theorem 3. The operator (29) satisfies (13) (local conservation), (14) (global conservation), and is first-order accurate.

4.2. MMD without a background mesh

In this section we drop the assumption of a background mesh and only require that the virtual vertices $V$ are tied to the point cloud $X$, see §3. To implement (20) under these conditions one has to construct the necessary topological, field, and metric data without referencing physical cells and faces, except for the boundary segments $\Gamma_i, \in \gamma'$ where field and face moments can be defined as in §4.1.

4.2.1. Topological data

The virtual boundary in (15) is completely determined by the list of edges connected to the primal vertex $v_i$. Any graph $G(V, E)$ built on the point cloud can provide this information. In particular, here we shall use the $\varepsilon_g$-ball graph $G_{\varepsilon_g}(V, E)$ of $X$ with vertices $V := X$ and edges

$$E := \{e_{ij} = (x_i, x_j) \in V \times V \mid |x_i - x_j| < \varepsilon_g \},$$

where $\varepsilon_g$ is a given positive real number. This graph may be trivially constructed with $O(p_0)$ computational complexity using standard binning algorithms. We view $G_{\varepsilon_g}(V, E)$ as a surrogate for a primal mesh and associate it with the virtual primal grid $\tau_h(V, E)$. The latter induces a virtual dual grid $\tau_h^*(C, F)$ that does not possess any physical entities, except for the boundary segments $\Gamma_i$.

Remark 2. We use a (virtual) primal-dual setting to define the abstract MMD operator precisely because any graph on $X$ induces a boundary operator on $\tau_h^*(C, F)$. Building such a graph only requires defining its edges, which can be done efficiently on any point cloud. In contrast, if one were to adopt a single virtual mesh, one would have to construct suitable surrogates for its virtual vertices, edges, faces and cells in order to obtain a correct notion of a virtual boundary. This is a substantially more difficult task.
4.2.2. Field data

In Section 4.1 we used the GMLS coefficients $c_{ij}(u^h) \in \mathbb{R}^{n_1}$ to define the data transfer operator $T_F : U^h \rightarrow \hat{U}^h$.

The only physical mesh entities needed to set up the least-squares problem (27) for these coefficients were the dual face centroids $x_{f_{ij}}$, involved in the kernel $\omega(\tau_{ij}, \lambda_i)$. In the present context the dual faces and their centroids are purely virtual. These virtual centroids can be tied to any reasonable physical location. Here we choose the midpoint $x_{ij} = (x_i + x_j)/2$ of the primal edge corresponding to $f_{ij}$ as a physical location of the face’s virtual centroid, i.e., we set $x_{f_{ij}} := x_{ij}$.

Efficient computation and storage of the field moments requires some care though. The number of virtual faces in $\partial \omega_i$ equals the valence of the primal vertex $x_i$. Since in the absence of a background mesh there are no restrictions on the number of edges connected to this vertex, computing and storing the GMLS coefficients $c_{ij}(u^h)$ for every virtual dual face may become ineffective. To improve efficiency we compute and store these coefficients on the point cloud. This can be accomplished by changing the kernel from $w(\tau_{ij}; \lambda_i) = \kappa(|x_f - x_i|)$ to $\omega(\lambda_i) = \kappa(|x_j - x_i|)$, which replaces the matrix $W(\tau_{ij})$ in (27) by a new matrix $W(x_j)$ with element $W_{ii}(x_j) = \kappa(|x_j - x_i|)$. Solution of the modified problem (27) yields the “nodal” GMLS coefficients $c_i(u^h)$, which are then “blended” into the field moments according to
\[
    c_{ij}(u^h) = \theta_{ij}c_i(u^h) + (1 - \theta_{ij})c_j(u^h); \quad \theta_{ij} = \theta_{ji} \in [0, 1].
\]

This blending preserves polynomial reproduction property and the blended coefficients (31) satisfy T.3.

4.2.3. Metric data

In Section 4.1 we used the GMLS target vector $\tau_F(\phi) \in \mathbb{R}^{n_1}$ to define the virtual face moments $\mu_F$. In contrast to the GMLS coefficient vector $c_{ij}(u^h)$, which only requires the centroids $x_{f_{ij}} := x_{ij}$, computation of $\tau_{ij}(\phi)$ is impossible without physical dual faces. Likewise, in the absence of physical dual cells we also need a method for assigning a volume $\mu_i$ to every virtual dual cell.

In this section we formulate a scalable and computationally efficient algebraic procedure for computing virtual cell volumes and virtual face moments. The procedure starts with construction of virtual volumes $\{\mu_i\}$ satisfying T.1. We then seek the components of the face moments $\mu_{ij}$ as gradients of scalar potentials. Inserting this ansatz into the $P_1$-reproduction condition (21) results in $n_1$ independent graph Laplacian problems for the face moments that can be solved in an efficient and scalable way by using algebraic multigrid preconditioners.

Definition of virtual volumes. To construct the virtual volumes we consider the quadratic program
\[
    \min_{\mu \in \mathbb{R}^{n_1}} \sum_{i \in X} \mu_i^2 \rho_i \quad \text{subject to} \quad \sum_{i \in X} \mu_i = |\Omega|,
\]
where $\rho_i > 0$ are “densities” assigned to each virtual dual cell. In this work we use two different density distributions. The first one is $\rho_i^1 = 1$ in which case solution of (32) yields the uniform virtual volumes
\[
    \mu_i^1 = \frac{|\Omega|}{p_\Omega}, \quad i = 1, \ldots, p_\Omega.
\]

However, the uniform volumes (33) may not be the best choice for multiresolution problems that call for non-quasiuniform point clouds. For such point clouds one can use the non-uniform density distribution
\[
    \rho_i^{\text{MR}} = \left( \sum_{j \in X} \kappa(|x_i - x_j|) \right)^{-1},
\]
where $\kappa(\cdot)$ is the kernel used in the GMLS problem. In this case solution of (32) results in
\[
    \mu_i^{\text{MR}} = \frac{\rho_i^{\text{MR}}}{|\Omega|} \left( \sum_{j \in X} \rho_j^{\text{MR}} \right). \quad (34)
\]
Both (33) and (34) satisfy T.1 and numerical results in §6.4 reveal that either one provides first-order convergence on quasi-uniform point clouds. This suggests that the actual volume definition is not critical for the accuracy of the MMD operator on such clouds. Nonetheless, unless otherwise noted, we use (34) because it is simple to implement, while still providing a sense of adaptivity for multiresolution problems.

**Definition of virtual face moments.** For clarity we discuss construction of the face moments assuming that \( \gamma = \Gamma \) and \( \gamma' = \emptyset \). For the same reason it will be convenient to relabel the polynomial basis functions in (25) and their point samples on the cube using double indices as

\[
\phi_{k,r}(x) := e_k \phi_r(x); \quad k = 1, \ldots, d; \quad r = 1, \ldots, d + 1,
\]

and \( \phi_{k,r}^h \in U_r^h \), respectively. We also relabel the components of the field moments and the virtual face moments to match the indexing of the basis functions, i.e., we write \((c_{ij})_{k,r} \) and \((\mu_{ij})_{k,r} \) for the components of \( \phi_{ij} \) and \( \mu_{ij} \), respectively, corresponding to \( \phi_{k,r} \).

Consistency of the MMD operator requires \( \{T_F, \{\mu_i, \mu_{ij}\}\} \) to be a \( P_1 \)-reproducing pair. A sufficient condition for this property is that (21) holds for all basis functions, i.e., we require that for all \( x_i \in X_0 \)

\[
\sum_{f_{ij} \in \partial \omega_i} c_{ij}(\phi_{k,r}^h_{x_i}) \cdot \mu_{ij} = \mu_i \nabla \cdot \phi_{k,r} \big|_{x_i} - \int_{X_i(\gamma)} \phi_{k,r} \cdot n dS; \quad k = 1, \ldots, d; \quad r = 1, \ldots, d + 1. \tag{35}
\]

Note that \( \nabla \cdot \phi_{k,r} = \partial_k \phi_r \) and \( \phi_{k,r} \cdot n = n_k \phi_r \), while the \( P_1 \)-reproduction property of GMLS implies that

\[
c_{ij}(\phi_{k,r}^h) \cdot \mu_{ij} = (\mu_{ij})_{k,r}.
\]

Therefore, (35) is equivalent to the statement that for all \( x_i \in X_0 \) there holds

\[
\sum_{f_{ij} \in \partial \omega_i} (\mu_{ij})_{k,r} = \mu_i \partial_k \phi_r \big|_{x_i} - \int_{X_i(\gamma)} n_k \phi_r dS; \quad k = 1, \ldots, d; \quad r = 1, \ldots, d + 1. \tag{36}
\]

The sum over \( f_{ij} \) in (36) is the graph divergence of a vector containing the \((k, r)\)th component of \( \mu_{ij} \). This prompts us to seek each component as a scaled graph gradient of a virtual area potential \( \psi_{k,r} \in S^h \), i.e.,

\[
(\mu_{ij})_{k,r} := ((\psi_{k,r})_{h} - (\psi_{k,r}')_{h}) \cdot \phi_r(x_{ij}),
\]

where \( x_{ij} \) is the physical centroid of the virtual face \( f_{ij} \). Inserting the ansatz (37) into (36) yields the following set of linear equations for the scalar potentials: for \( r = 1, \ldots, d + 1 \) solve

\[
\sum_{f_{ij} \in \partial \omega_i} ((\psi_{k,r}^h)_{i} - (\psi_{k,r}')_{h}) \cdot \phi_r(x_{ij}) = \mu_i \partial_k \phi_r \big|_{x_i} - \int_{X_i(\gamma)} n_k \phi_r dS \quad \forall x_i \in X_0 \tag{38}
\]

for \( k = 1, \ldots, d \). Equation (38) is a weighted graph Laplacian problem. Thus, to determine the virtual face moments we need to solve \( d + 1 \) graph Laplacians with different weights, each with \( d \) different right-hand-sides. We write these linear systems as

\[
L_r \psi_{k,r} = b_{k,r}.
\]

The standard \( n \times n \) graph Laplacian is an M-matrix, and therefore, it may be solved with \( O(n) \) complexity using standard algebraic multigrid. For the weighted graph Laplacian \( L_r \) we must ensure that the weights \( \phi_r(x_{ij}) \) preserve the M-matrix structure. With the assumption that the coordinate system is such that all points in the cloud have positive coordinates, this can be easily accomplished by, e.g., choosing the scalar basis set as \( \{\phi_r\}_{r=1}^d = \{1, x_1, \ldots, x_d\} \).

The operator \( L_r \) also has a one-dimensional null space comprising the constant vector \( 1 \), i.e., \( L_r 1 = 0 \). As a result, the \( d \) right-hand sides \( b_{k,r} \) for every one of the \( d + 1 \) graph Laplacian equations in (38) are subject to the compatibility condition \( 1^T b_{k,r} = 0 \) for \( k = 1, \ldots, d \). It is easy to see that the volume assumption in T.1 is sufficient for this compatibility condition to hold. Indeed, from the fact that \( \partial_k \phi_r = \text{const} \) and \( \sum_i \mu_i = |\Omega| \) it follows that for all \( k = 1, \ldots, d \) there holds

\[
1^T b_{k,r} = \sum_{x_i \in X_0} \left( \mu_i \partial_k \phi_r \big|_{x_i} - \int_{X_i(\gamma)} n_k \phi_r dS \right) = |\Omega| \partial_k \phi_r - \int_{\partial \Omega} n_k \phi_r dS = |\Omega| \partial_k \phi_r - \int_{\Omega} \partial_k \phi_r dx = 0.
\]

14
4.2.4. MMD without a background mesh: setup and properties

Assuming that $\gamma = \Gamma$ we have the following instance of the abstract MMD operator (20)

$$\text{GMLS}^+ \left( \text{DIV} u^h \right)_i = \frac{1}{\mu_i} \left[ \sum_{f_{ij} \in \partial \omega_i} (c_{ij}(u^h)) \cdot \mu_{ij} + \int_{\chi(\gamma)} u \cdot n dS \right] \quad \forall \omega_i \in C,$$

(39)

where the field moments $c_{ij}$ are given by (31), the virtual volumes $\mu_i$ are given by (33) or (34), and the virtual face moments $\mu_{ij}$ are defined by solving the graph Laplacian systems (38) for the scalar potentials $\psi_{k,r}^h$ and then using the ansatz (37). Thus, although (29) and (39) have the same structure, the metric information in the latter is determined in a purely algebraic way without requiring a physical mesh structure. It is, therefore, of some interest to examine the comparative costs of setting up these operators. To make this comparison more equitable we do not count the mesh generation towards the setup cost of (29), as the purpose of this operator is to improve discretization robustness and accuracy on an already existing mesh.

The setup of (29) involves computing the GMLS coefficients and the face moments for every dual face. Assuming that the valence of every primal node in the mesh is bounded by a constant $C$, the total number of these faces is $O(p_\Omega)$. The work to compute each $c_{ij}$ is proportional to the average number $n_c$ of mesh nodes in the support of the kernel function, while the cost to compute every face moment $\mu_{ij}$ is proportional to $d + 1$. As a result, the setup cost for (29) is $O(p_\Omega (n_c + d))$.

The setup cost for (39) involves computing the nodal GMLS coefficients $c_i$ and the scalar potentials $\psi_{k,r}^h$ on the point cloud. The work for every $c_i$ is again proportional to $n_c$, while the potentials require the solution of $d + 1$ linear systems with $d$ different right hand sides per point. Assuming that these systems are solved using algebraic multigrid the cost will scale as $O(p_\Omega d^2)$, where $p_\Omega$ is the number of points in the cloud $X$. Thus, the total setup work for (39) scales as $O(p_\Omega (n_c + d^2))$.

Note that $n_c$ can be assumed fixed as long as the polynomial degree of the GMLS reproducing space $\Phi$ remains the same. Therefore, the setup cost for both (29) and (39) scales linearly with the number of points in the cloud $X$.

**Remark 3.** In the absence of a background mesh the valence of each point in $X$ may grow with the size of the point cloud, leading to an increase of the number of virtual edges in $\partial \omega_i$. Using (31) to compute the field moments and the scalar potentials $\psi_{k,r}^h$ to compute the virtual face moments allows us to store all field and metric data, required for the evaluation of (39), on the point cloud $X$. This is essential for ensuring that the setup cost and the storage requirements of (39) scale linearly with the size of the point cloud.

**Remark 4.** Although a detailed discussion of a performant implementation of (39) is beyond the scope of this work we include in Table 4.2.4 the number of iterations and total wall-clock time when using preconditioned conjugate gradient method with a classic algebraic multigrid preconditioner [26] to solve (38). The information in the table confirms that these standard techniques work well for the weighted graph Laplacian operators in (38) and achieve linear computational complexity.

**Properties.** By construction the metric and field data for (39) satisfies the topological requirements T.1-T.3 and $\{T_F(\mu_i, \mu_{ij})\}$ is a $P_1$-reproducing pair. This is sufficient for Theorem 1 to hold for (39), i.e., this MMD instance satisfies (13) (local conservation) and (14) (global conservation). On the other hand, to show that (39) is also first-order accurate requires $\{T_F, \{\mu_{ij}\}\}$ to be a locally Lipschitz continuous pair on $X$, i.e., that (22) holds. The proof of this property is highly non-trivial and will be addressed in a forthcoming paper. However, preliminary numerical convergence studies in Section 6 suggest that (39) is indeed first-order accurate.

---

*Recall that the background mesh nodes are assumed to coincide with the point cloud $X$ and so their number equals $p_\Omega$. 

---
Table 1: Average solution time and average number of the AMG iterations for the weighted graph Laplacian equations (38) as function of the point cloud size. Solution time and AMG iterations are averaged over the set of \(d(d+1)\) solves. Results indicate optimal \(O(p_\omega)\) performance of the AMG solver.

| \(p_\omega\) | Solution time (sec) | AMG iterations |
|-----------|---------------------|----------------|
| 16        | 0.0015              | 5.750          |
| 32        | 0.0038              | 6.125          |
| 64        | 0.0094              | 5.000          |
| 128       | 0.0368              | 6.500          |
| 256       | 0.1839              | 6.625          |
| 512       | 0.5782              | 7.125          |

5. A virtual finite volume scheme for a scalar advection-diffusion equation

In this section we apply the abstract MMD operator (20) to construct a meshfree mimetic discretization of the model boundary value problem

\[-\nabla \cdot \sigma(u) = f \quad \text{in } \Omega\]

\[u = g \quad \text{on } \gamma'\]

\[n \cdot \sigma(u) = h \quad \text{on } \gamma\]

where \(u\) is a scalar variable, \(f\), \(g\) and \(h\) are prescribed data, and \(\sigma(\cdot)\) is a flux function. To illustrate the application of the MMD operator it suffices to consider a simple advective-diffusive flux given by

\[\sigma(u) = \varepsilon \nabla u - au := \sigma_d + \sigma_a,\]

where \(\varepsilon > 0\) is a piecewise \(C^0\) diffusion coefficient and \(a \in (C^0(\Omega))^d\) is a given velocity field.

We highlight the ability of our approach to deliver a truly conservative meshfree scheme without a physical background mesh by implementing the second instance (39) of the abstract MMD operator. Thus, we consider the discretization infrastructure of Section 4.2, comprising a quasi-uniform point cloud \(X\), an \(\epsilon_g\)-ball graph \(G_{\epsilon_g}(V,E)\) serving as a surrogate primal mesh \(\tau_h(V,E)\), and the associated virtual dual mesh \(\tau'_h(C,F)\). The virtual volumes \(\mu_i\) are set by (34), while the virtual face moments \(\mu_{ij}\) are defined by (37) with potentials determined by solving the graph Laplacian systems (38). The face moments on the Dirichlet boundary segments are computed using the target functionals in (26), i.e.,

\[\langle \mu_{\Gamma_j} \rangle_s = \int_{\Gamma_j} \phi_s \cdot n \, dS; \quad \text{for } s = 1, \ldots, n_1 \text{ and } \Gamma_j \subset \gamma',\]

where \(\{\phi_s\}\) is a basis of \(\Phi = (P_1)^d\). To discretize (40) we represent the unknown field \(u(x)\) by its point samples on \(X_0 \cup X_\gamma\), the source term \(f\) by its point samples on \(X\) and assume that the boundary data \(g\) and \(h\) are given on the Dirichlet and Neumann parts of the boundary, respectively. We thus obtain the following meshfree mimetic discretization of (40): given \(f^h \in S_h^\gamma\) find \(u^h \in S_h^{\gamma'}\) such that

\[\mu_i(DIV \tilde{\sigma}(u^h)) = \mu_i f_i \quad \forall x_i \in X_0 \cup X_\gamma,\]

where \(\tilde{\sigma}(u^h) \in \tilde{U}_\gamma^h\) is a “numerical flux”. Note that the Dirichlet boundary condition is enforced in (42) through the definition of the space \(S_h^{\gamma'}\), which implies that \(u_i^h = g(x_i)\) for all \(x_i \in \gamma\). In contrast, the Neumann boundary condition is enforced through the definition of the MMD operator (39) by applying the specified boundary data on all segments \(\Gamma_j \subset \gamma\). To complete the formulation of (42) it remains to construct a numerical flux \(\tilde{\sigma}^h\) from the field data \(u^h \in S_h^{\gamma'}\). Therefore, our application of the MMD operator to (40) is similar to the second use case described in Remark 1, and that is typical of finite volume schemes. Thus, we shall refer to (42) as a virtual finite volume discretization of the model problem. We discuss construction of the numerical flux in the next section.

16
5.1. Numerical flux

We will define the numerical flux as a sum

\[ \tilde{\sigma}(u^h) := \tilde{\sigma}_d(u^h) + \sigma_n(u^h), \]

where \( \tilde{\sigma}_d(u^h) \in \tilde{U}_{\gamma}^h \) and \( \sigma_n(u^h) \in \tilde{U}_2^h \) are diffusive and advective flux moments, respectively. To compute these moments we follow the procedure outlined in Section 4.2.2 and first generate the necessary GMLS coefficients on the point cloud. Then we use (31) to define \( \tilde{\sigma}_d(u^h) \) and \( \tilde{\sigma}_n(u^h) \) on the virtual faces. In both cases we assume that the diffusion coefficient and the velocity field are represented on the point cloud by their point samples \( \varepsilon^h \in S^h \) and \( a^h \in U^h_2 \), respectively.

5.1.1. Advective flux moments

The first step in the construction of \( \tilde{\sigma}_n(u^h) \) involves computing a set of “nodal” GMLS coefficients at all points \( x_i \in X_0 \cup X_\gamma \) from a point sample \( \sigma_n^h(u) \in U_2^h \) of the advective flux. We consider two different types of such “nodal” coefficients. The first one, denoted by \( c^a_n(u^h) \), solves the weighted least-squares problem (27) with weight matrix \( W(x_i) \) and a point sample\(^7 \) \( \sigma_n^h(u) = a^h \circ u^h \in U_2^h, \) i.e.,

\[ c^a_n(u^h) = \arg\min_{b \in \mathbb{R}^{n_1}} \frac{1}{2} \| Bb - \sigma_n^h(u) \|^2_{W(x_i)}. \]  

(43)

The second “nodal” GMLS coefficient, denoted by \( c^u_n(u^h) \), is an upwind version of \( c^a_n(u^h) \) intended for advection-dominated problems. To incorporate upwinding we modify (43) as follows. Given a point \( x_i \in X_0 \cup X_\gamma \) we define the upwind part of the point cloud at this point as

\[ \tilde{X}_i = \{ x_k \in X | a_i \cdot (x_k - x_i) < 0 \} \cup \{ x_i \}. \]

Thus, \( \tilde{X}_i \) is the subset of \( X \) containing the point \( x_i \) and all cloud points strictly upwind of that point. Next, we define the upwind point sample of the advective flux as

\[ \tilde{\sigma}_n^h(u) = \{(\sigma_n^h(u))_i | x_i \in \tilde{X}_i \}. \]

Finally, we redefine the basis sample matrix \( B \) and the weight matrix \( W(x_i) \) in (43) to include only the rows corresponding to the points in \( \tilde{X}_i \) and denote the new matrices as \( \tilde{B}_i \) and \( \tilde{W}(x_i) \), respectively. The upwind “nodal” GMLS coefficient is then obtained by solving the modified weighted least-squares problem

\[ \tilde{c}^u_n(u^h) = \arg\min_{b \in \mathbb{R}^{n_1}} \frac{1}{2} \| \tilde{B}_i b - \tilde{\sigma}_n^h(u) \|^2_{\tilde{W}(x_i)}. \]

We use (31) with \( \theta_{ij} = 1/2 \) and the first type of “nodal” GMLS coefficients to define the standard advective flux moments \( \tilde{\sigma}_a(u^h) \). Setting \( \theta_{ij} = 1 \) if \( a(x_{ij}) \cdot (x_j - x_i) \geq 0 \) and \( \theta_{ij} = 0 \) otherwise, and using the second type of GMLS coefficients defines the upwind moments \( \tilde{\sigma}_a(u^h) \). Succinctly,

\[ \tilde{\sigma}_a(u^h))_{ij} = \frac{1}{2} (c^a_i(u^h) + c^a_j(u^h)) \quad \text{and} \quad \tilde{\sigma}_a(u^h))_{ij} = \begin{cases} \tilde{c}^u_i(u^h) & \text{if } a(x_{ij}) \cdot (x_j - x_i) \geq 0 \\ \tilde{c}^u_j(u^h) & \text{if } a(x_{ij}) \cdot (x_j - x_i) < 0. \end{cases} \]

(44)

5.1.2. Diffusive flux moments

As in the advective flux case we obtain \( \tilde{\sigma}_d(u^h) \) by blending “nodal” GMLS coefficients. However, generating these coefficients requires a different approach because the point values of \( \nabla u \) are not specified on the cloud and so, a point sample of \( \sigma_d(u) \) is not readily available on \( X \). This means that we cannot use the setting of (43), which requires such a sample. Instead, we shall obtain the “nodal” coefficients through

\(^7\)Here \( \circ \) denotes the Hadamard (pointwise) product of two vectors.
an auxiliary GMLS problem for the scalar field \( u \) that is exact for quadratic polynomials. Specifically, we consider the GMLS framework in Section 2.2 with \( U = C^1(\Omega) \), \( \Phi = P_2 \), sampling functionals \( \lambda_i = \delta_{x_i} \), a target \( \tau_i(u) = \sigma_d(u)|_{x_i} = \varepsilon \nabla u(x_i) \), and kernel \( w(\tau_i, \lambda_j) = \kappa([x_i - x_j]) \). The GMLS approximation of \( \tau_i(u) \) at \( x_i \in X \) is then given by
\[
(\sigma_d(u^h))_i = c(u^h) \cdot (\varepsilon \nabla \phi(x_i))
\]
where \( \phi = \{ \phi_k \}_{k=1}^{n_1} \) is basis of \( P_2 \), \( \nabla \phi(x_i) = (\nabla \phi_1(x_i), \ldots, \nabla \phi_{n_2}(x_i)) \), and
\[
c_i(u^h) = \arg\min_{b \in \mathbb{R}^{n_2}} \frac{1}{2} |B b - u^h|^2_{W(x_i)}.
\]
Since \( \nabla \phi_q \in (P_1)^d \) for all \( q = 1, \ldots, n_2 \), it follows that there exists a matrix \( G \) such that
\[
\nabla \phi = G \phi
\]
where \( \phi = \{ \phi_k \}_{k=1}^{n_1} \) is basis of \( (P_1)^d \). As a result, we can express the target approximation (45) in terms of this basis as
\[
(\sigma_d(u^h))_i = \varepsilon_i g_i(u^h) \cdot \phi(x_i),
\]
where \( g_i(u^h) = G^* c_i(u^h) \) and \( \varepsilon_i = \varepsilon|_{x_i} \). We refer to \( g_i(u^h) \) as the derived “nodal” GMLS coefficients because they are not based on a point sample of the diffusive flux but rather derived from a point sample of the scalar field \( u \). To define the diffusive flux moments on the virtual faces we then use (31) with \( \theta_{ij} = 1/2 \), the derived “nodal” coefficients, and an averaged diffusion coefficient \( \varepsilon_{ij} \):
\[
(\sigma_d(u^h))_{ij} = \frac{\varepsilon_{ij}}{2} (g_i(u^h) + g_j(u^h)).
\]
In Section 6.2, we will explore the use of both the arithmetic mean \( \varepsilon_{ij}^A = (\varepsilon_i + \varepsilon_j)/2 \) and the harmonic mean \( \varepsilon_{ij}^H = 2\varepsilon_i\varepsilon_j/(\varepsilon_i + \varepsilon_j) \), which is more appropriate when \( \varepsilon \) has large jumps.

6. Numerical results

In the following sections, we study numerically the properties of the virtual finite volume scheme (42) for the model problem (40). We consider two versions of this problem corresponding to \( a = 0 \) and \( a \neq 0 \), respectively. We refer to the first version as the Darcy problem and to the second one as the advection-diffusion problem. For simplicity, we restrict attention to spatial domains in \( \mathbb{R}^2 \).

Unless otherwise noted, we generate the point clouds for our study using the following procedure. Given a desired fill distance \( h_X > 0 \) we first construct a uniform partition of the boundary comprising segments \( \Gamma_i \) of length \( h_X \). The barycenters of these segments define \( X_\Gamma \). Then, we obtain \( X_0 \) by generating a uniform lattice \( L_{h_X} \) of spacing \( O(h_X) \) over a square containing \( \Omega \), and finally define \( X_0 = L_{h_X} \cap \Omega \).

To remove any symmetries that might prompt superconvergence, we then perturb each point in \( X_0 \) by a uniformly distributed random variable with magnitude \( h_X/5 \). In so doing we obtain point clouds satisfying the assumption of quasi-uniformity (4). Figure 2 shows a point cloud on the unit disk generated by this procedure. To facilitate post-processing of results, it will often be convenient to generate the Delaunay triangulation corresponding to \( X \); it will be made explicit if results are presented on the Delaunay triangulation rather than natively on the point cloud.

6.1. Darcy problem: convergence study

We perform a numerical convergence study of (42) using (40) with \( \varepsilon = 1 \), \( a = 0 \), pure Neumann and pure Dirichlet boundary conditions, the smooth manufactured solution \( u_{ex}(x,y) = \sin 2\pi x \sin 2\pi y \), and the domain shown the in Figure 3. Substitution of \( u_{ex}(x,y) \) into the governing equations defines the necessary data fields for the study. In the pure Neumann case the solution is determined up to an arbitrary constant. To ensure uniqueness we use a Lagrange multiplier to enforce a zero mean condition on the solution. We refer to [27] and the references therein for further information on handling pure Neumann conditions.
To estimate convergence rates we use successively refined point clouds constructed from a sequence of \( N \times N \) uniform lattices for \( N = 16, 32, 64, 128, 256 \). Rates are estimated with respect to the error measures

\[
\| u_h - u_{\text{ex}} \|^2_{l^2, X} := \sum_{i \in X} (u_h^i - (u_{\text{ex}}^i))^2 \quad \text{and} \quad | u_h - u_{\text{ex}} |^2_{1, X} := \frac{1}{\varepsilon} \sum_{i \in X} | (\sigma_d(u_h))^i - (\sigma_d(u_{\text{ex}}))^i |^2,
\]

respectively, where \( u_{\text{ex}}^i \) is a point sample of the manufactured solution and \( \sigma_d(u_h) \) is the GMLS reconstruction of the diffusive flux defined in (45). We refer to these measures as the \( l^2 \)-norm and the \( h_1 \)-seminorm, respectively. Table 6.1 shows convergence results for the virtual finite volume solution with pure Dirichlet and pure Neumann boundary conditions in these norms. We observe roughly the same convergence rate in both norms, and for both types of boundary conditions, which confirms numerically that (42) is first-order accurate.

### 6.2. Darcy problem: \( H(\text{div}) \)-compatibility study

In this section we consider Darcy problems with discontinuous coefficients \( \varepsilon \). In this case, for any interface \( \eta \) corresponding to a discontinuity in \( \varepsilon \), solutions of (40) must satisfy the transmission conditions

\[
u_+ = u_- \quad \text{and} \quad n_\eta \cdot \varepsilon_+ \nabla u_+ = n_\eta \cdot \varepsilon_- \nabla u_- , \quad \text{on} \ \eta ,
\]
where $\mathbf{n}_\eta$ is a unit normal to $\eta$ and $+$ and $-$ denote entities associated with the opposite sides of the interface. Conditions (48) imply that the normal component of the flux $\varepsilon \nabla u$ is always continuous across the interface but its tangential component and the gradient $\nabla u$ may be discontinuous on $\eta$. A hallmark of any $H(\text{div})$-compatible discretization for such Darcy problems is the ability to accurately represent this flux behavior, which is essential for, e.g., accurate simulations of subsurface flow problems.

To evaluate the extent to which the virtual finite volume scheme (42) may be deemed $H(\text{div})$-compatible, we consider three benchmark examples adapted from [28, 29] and driven by pure Neumann conditions. In each one of these examples $\varepsilon$ is a piecewise constant field defined with respect to a disjoint partition $\Omega = \bigcup_i \Omega_i$ of the unit square. The value of $\varepsilon$ on $\Omega_i$ is denoted by $\varepsilon_i$. Figure 4 shows the partitions for each one of the three examples. We refer to these examples as the two strip, five strip and five spot problems, respectively.

![Figure 4: Discontinuous diffusivity fields for Darcy benchmarks. Left: The two strip problem. Center: The five-strip problem. Right: The five-spot problem.](image)

### Two strip problem

In this example $\varepsilon$ is discontinuous along the vertical line $x = \frac{1}{2}$ and

$$u_{ex}(x, y) = \begin{cases} x & \text{if } x \leq \frac{1}{2} \\ \frac{1}{2} + \frac{\varepsilon_1}{\varepsilon_2} \left( x - \frac{1}{2} \right) & \text{if } x > \frac{1}{2} \end{cases}$$

The exact solution satisfies the interface condition (48) and the pure Neumann condition

$$-\varepsilon \nabla u_{ex} \cdot \mathbf{n} = \varepsilon_1 \mathbf{e}_1 \cdot \mathbf{n} \quad \text{on } \Gamma.$$
solution profiles along \( x = 0.5 \) for increasing diffusivity ratios \( R \). We see that the results are comparable for \( \varepsilon_1/\varepsilon_2 = O(1) \), but as the ratio is increased the harmonic mean is better able to preserve the location of the jump. Motivated by these results, we use the harmonic mean for all results in the remainder of this paper.

**Figure 5:** Approximation of \( \partial_x u_{ex} \) by the virtual finite volume scheme (42) for \( 92 \times 92 \) particles and diffusivity ratio \( R = 64 \). **Left:** arithmetic average. **Right:** harmonic average.

**Figure 6:** Profile of the virtual finite volume approximation of \( \partial_x u_{ex} \) along the line \( x = 0.5 \) for increasing diffusivity ratios. **Left:** arithmetic average. **Right:** harmonic average. Exact solution is given by dashed line.

**The five-strip problem.** In this example \( \varepsilon \) has discontinuities along the lines \( y = 0.2, y = 0.4, y = 0.6 \) and \( y = 0.8 \), which divide the unit square into five horizontal strips. We set the diffusivity value on each strip to \( \varepsilon_1 = 16, \varepsilon_2 = 6, \varepsilon_2 = 1, \varepsilon_4 = 10, \) and \( \varepsilon_5 = 2 \), respectively and let

\[
u_{ex} = 1 - x.
\]

This exact solution satisfies the interface conditions (48), the pure Neumann condition

\[
\nabla u_{ex} \cdot n = e_1 \cdot n,
\]

has a continuous vertical flux component \( \varepsilon \partial_y u_{ex} = 0 \), and a piecewise constant horizontal flux component \( \varepsilon \partial_x u_{ex} = \varepsilon_i \) on \( \Omega_i \). The five strip benchmark is designed to test how well a scheme can represent this flux behavior. Figure 7 shows the approximation of \( \varepsilon \partial_x u_{ex} \) for both uniform and non-uniform point clouds. The profiles of the horizontal flux approximation along \( x = 0.5 \) for increasing point cloud resolutions are shown in Figures 8. These results show that the virtual finite volume scheme (42) can accurately represent the flux components on both uniform and non-uniform point clouds. In particular, approximations of the discontinuous, horizontal flux component on uniform point clouds are indistinguishable from those obtained
by an $H(\text{div})$-conforming, mesh-based scheme using interface-fitted grids. The small wiggles in the non-uniform case are caused by the non-symmetric distribution of points across the interfaces, which is the meshless analogue of an unfitted grid. We note that an $H(\text{div})$ scheme on an unfitted grid would have a similar behavior.

Figure 7: Approximation of the horizontal (top) and vertical (bottom) flux components by the virtual finite volume scheme (42) on a particle cloud with $h_X = 1/96$. Left: uniform (Cartesian) particle distribution. Right: non-uniform particle distribution.

Figure 8: Profiles of the horizontal flux approximation by the virtual finite volume scheme along the line $x = 0.5$ as the point cloud is refined. Left: uniform (Cartesian) particle distribution. Right: non-uniform particle distribution.

The five-spot problem. This example models an injection-extraction application in which the flow is driven from an injection well in the bottom left corner of the domain to an extraction well in the top right. The diffusivity $\varepsilon$ is a piecewise constant which assigned the same value on the bottom left and top right subdomains and another value on the bottom right and top left subdomains; see the right plot in Fig. 4.

To drive the flow we use the same Neumann conditions as in [29, Figure 26]. Specifically, we apply a flux of $-\frac{1}{8}$ to both faces coincident to the bottom left corner, and $\frac{1}{8}$ to those coincident with the top right (Figure 9). In the case where $\varepsilon_1 = \varepsilon_2$, this problem may be solved analytically via superposition of Greens function solutions, assuming a periodic domain. Figure 10 demonstrates convergence of the virtual finite volume scheme to the analytic solution for both the pressure and the flux.
Figure 9: Specification of a flux field yielding a total flux of $1/8$ across the faces coincident with the top right corner of the domain. A similar field is specified on the faces coincident with the bottom left corner, yielding a total flux of $-1/8$ across those faces.

Figure 10: Comparison of the analytic solution of the five spot problem with $\varepsilon_1 = \varepsilon_2 = 0.5$ and solutions of the virtual finite volume scheme (42) on successively refined point clouds. Left: Streamlines of flow. Center: Pressure along $x = y$ diagonal. Right: Flux along $x = y$ diagonal.

Figure 11: The five spot problem with varying $R = \varepsilon_1/\varepsilon_2$. Left: Comparison of the pressure profiles along $y = x$ by the virtual finite volume scheme (42) (solid lines) and a mixed method using RT0-P0 elements (dashed lines) for increasing ratios $R$. Right: Pressure approximation by the virtual finite volume scheme (42) for large ratio $R = 1000$.

To highlight the robustness of the virtual finite volume scheme we consider the five spot problem with different diffusivity ratios $R = \varepsilon_1/\varepsilon_2$. The left plot in Figure 11 compares the pressure computed by (42) with a solution obtained by a mixed method implemented with the RT0-P0 element pair. We see that for all diffusivity ratios the two pressure profiles are nearly identical. Finally, the right plot in the same figure shows that the virtual finite volume scheme continues to deliver stable solutions even for very large material contrasts.
Table 3: Convergence rates in $\ell_2$-norm of the virtual finite volume approximation to $u_{ex}(x,y) = \sin 2\pi x \sin 2\pi y$ using the centered advective flux reconstruction $\tilde{\sigma}_a(u^h)$. An entry of n.c. denotes that the linear solver failed to converge due to ill-conditioning.

| $N$ | $Pe = 1$ | $Pe = 10$ | $Pe = 100$ | $Pe = 1000$ |
|-----|----------|-----------|------------|-------------|
| 16  | 0.0535674| 0.0595928 | n.c.       | n.c.        |
| 32  | 0.0301618| 0.060635  | n.c.       | n.c.        |
| 64  | 0.0152526| 0.0333928 | 0.0590015  | n.c.        |
| 128 | 0.0075607| 0.0175604 | 0.0314838  | n.c.        |
| 256 | 0.0037626| 0.0090120 | 0.0161104  | n.c.        |
| Rate| 1.006    | 0.957     | 0.963      | n.c.        |

Table 4: Convergence rates in $\ell_2$-norm of the virtual finite volume approximation to $u_{ex}(x,y) = \sin 2\pi x \sin 2\pi y$ using the upwind advective flux reconstruction $\vec{\tilde{\sigma}}_a(u^h)$.

| $N$ | $Pe = 1$ | $Pe = 10$ | $Pe = 100$ | $Pe = 1000$ |
|-----|----------|-----------|------------|-------------|
| 16  | 0.061851 | 0.11673   | 0.146463   | 0.153675    |
| 32  | 0.031884 | 0.0633952 | 0.0911815  | 0.0971759   |
| 64  | 0.0155568| 0.0339481 | 0.0550493  | 0.0595179   |
| 128 | 0.00762096| 0.0177066 | 0.0304351  | 0.0331474   |
| 256 | 0.00377766| 0.009056  | 0.0158462  | 0.0172527   |
| Rate| 1.02     | 0.967     | 0.944      | 0.954       |

The results in this section suggest that for a range of representative example problems the virtual finite volume scheme does indeed behave in a manner that mimics the behavior of mesh-based $H(\text{div})$-conforming methods, such as mixed finite elements implemented with the RT0-P0 element pair.

6.3. Advection-diffusion problem

In this section we examine the performance of the virtual finite volume scheme (42) for the advection-diffusion problem using a range of Péclet numbers $Pe := \frac{||a||}{\varepsilon}$ spanning both the diffusion-dominated, i.e., $Pe = O(1)$, and the advection dominated, i.e., $Pe \gg O(1)$, regimes. Our objectives are twofold. First, using manufactured solutions, we aim to demonstrate the consistency and the accuracy of the virtual finite volume scheme for both the centered $\tilde{\sigma}_a(u^h)$ and the upwind $\vec{\tilde{\sigma}}_a(u^h)$ reconstructions of the advective flux defined in (44). Our second goal is to confirm that the construction of $\vec{\tilde{\sigma}}_a(u^h)$ does provide an appropriate notion of upwinding in the advection-dominated regime.

In both cases we consider the unit square with a quasi-uniform point cloud $X$ defined according to the procedure described at the beginning of §6. To study the accuracy of the virtual finite volume scheme we use the same manufactured solution as in §6.1, i.e., $u_{ex}(x,y) = \sin 2\pi x \sin 2\pi y$, and apply Dirichlet boundary conditions at all particles in $X_\Gamma$.

Tables 3 and 4 present convergence rates for increasing Péclet numbers using the centered $\tilde{\sigma}_a(u^h)$ and the upwind $\vec{\tilde{\sigma}}_a(u^h)$ flux reconstructions, respectively. The results in Table 3 show that virtual finite volume scheme with the centered advective flux reconstruction is able to maintain $O(h)$ convergence for small to moderate Péclet numbers. However, as the problem becomes strongly advection-dominated, using the centered flux leads to ill-conditioned discrete equations and failure of the iterative solver to converge. On the other hand, Table 4 reveals that with the upwind reconstruction $\vec{\tilde{\sigma}}_a(u^h)$ the virtual finite volume scheme remains first-order accurate over a wide range of Péclet numbers.

To further study the appropriateness of $\vec{\tilde{\sigma}}_a(u^h)$ in the advection-dominated case we consider an example with discontinuous boundary data and velocity field $a = (1, 2)$ similar to the skew advection test in [30]. To
Figure 12: Solution of the skew advection example by the virtual finite volume scheme with the upwind flux $\tilde{\sigma}(u^h)$ using Dirichlet (top) and Dirichlet/outflow (bottom) boundary conditions. Left: $Pe = 1$. Center: $Pe = 10$. Right: $Pe = 100$.

describe the problem setup let $\Gamma_t$, $\Gamma_r$, $\Gamma_b$ and $\Gamma_l$ denote the top, right, bottom, and left sides of the unit square domain, respectively. On the bottom and left boundaries we impose Dirichlet boundary conditions given by

$$u = 1 \text{ on } \Gamma_l \quad \text{and} \quad u = \begin{cases} 1, & x \leq \frac{1}{4} \\ 0, & x > \frac{1}{4} \end{cases} \text{ on } \Gamma_b,$$

respectively. On the remaining parts of the boundary we impose either the Dirichlet conditions

$$u = 1 \text{ on } \Gamma_t \quad \text{and} \quad u = 0 \text{ on } \Gamma_r
$$
or the following outflow condition, derived by setting $n \cdot \sigma_d(u) = 0$ in the flux definition.

$$n \cdot \sigma(u) = n \cdot \sigma_d \text{ on } \Gamma_t \cup \Gamma_r. \quad (49)$$

Figure 12 shows plots of the solution for both the pure Dirichlet and Dirichlet/outflow cases computed by the virtual finite volume scheme with the upwind advective flux for $Pe \in \{1, 10, 100\}$. A solution plot for the Dirichlet/outflow boundary conditions and $Pe = 1000$ is shown on Figure 13. We see that in this case the solution develops moderate crosswind oscillations that remain localized near the internal layer. This behavior is similar to the one observed in, e.g., streamline upwind finite element methods [31] and can be corrected by including an appropriate discontinuity capturing term [32]. However, design of an advective flux reconstruction incorporating such a term is beyond the scope of this paper.

6.4. Sensitivity of MMD with respect to virtual volume definitions

We conclude by examining the impact of the virtual volume definition on the truncation error of the MMD operator (39). For this study we use the smooth vector field $u = 2\pi (\cos 2\pi x \sin 2\pi y, \sin 2\pi x \cos 2\pi y)$, the non-trivial geometry illustrated in Figure 3, and quasi-uniform point clouds generated according to the
Figure 13: Solution of the skew advection example by the virtual finite volume scheme with the upwind flux $\vec{\sigma}_a(u^h)$ using Dirichlet/outflow (bottom) boundary conditions. $Pe = 1000.$

| $N$  | $\mu_i^{\mu}$ | $\mu_i^{MR}$ |
|------|----------------|--------------|
| 16   | 0.312637       | 0.267592     |
| 32   | 0.188097       | 0.145183     |
| 64   | 0.102245       | 0.076769     |
| 128  | 0.054184       | 0.040412     |
| 256  | 0.027495       | 0.020119     |
| Rate | 0.978711       | 1.00622      |

Table 5: Truncation errors in $\ell_2$-norm and convergence rate of the MMD operator (39) implemented using the uniform (33) and non-uniform (34) virtual volume definitions.

The results in this table show that in both cases the MMD operator remains first-order accurate.

7. Conclusion

The principal contribution of this work is the formulation of a consistent and conservative mimetic mesh-free divergence operator that can be used with or without a background grid while remaining computationally efficient in the latter case. Specifically, implementation of the operator in the absence of background mesh involves solution of graph Laplacian problems, which can be accomplished in a scalable and efficient manner by using standard multigrid techniques. We demonstrate the accuracy and robustness of our operator by using it to define a virtual finite volume scheme for a scalar advection-diffusion equation. Our numerical results show that the scheme is first-order accurate, exhibits the same traits as an $H(div)$-conforming mesh-based discretization and can be equipped with an appropriate notion of upwinding for advection-dominated problems.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research under Award Number DE-SC-0000230927, and the Laboratory Directed Research and Development program at Sandia National Laboratories. The first author acknowledges support through the NSF-MSPRF program.
References

[1] I. Babuška, A. Aziz, On the Angle Condition in the Finite Element Method, SIAM Journal on Numerical Analysis 13 (2) (1976) 214–226, doi:10.1137/0713021, URL https://doi.org/10.1137/0713021.

[2] J. R. Shewchuk, What is a good linear element? interpolation, conditioning, and quality measures, in: In 11th International Meshing Roundtable, 115–126, URL http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.67.5835, 2002.

[3] M. Harwick, R. Clay, P. Boggs, E. Walsh, A. Larzeler, A. Altschuler, DART System analysis, Tech. Rep. SAND2005-4647, Sandia National Laboratories, 2005.

[4] T. Belytschko, Y. F. Lu, L. Gu, Element-free Galerkin methods, International Journal for Numerical Methods in Engineering 37 (2) (1994) 229–256, ISSN 1097-0207, doi:10.1002/nme.1620370205, URL http://dx.doi.org/10.1002/nme.1620370205.

[5] J.-S. Chen, M. Hillman, M. Rüter, An arbitrary order variationally consistent integration for Galerkin meshfree methods, International Journal for Numerical Methods in Engineering 95 (5) (2013) 387–418, doi:10.1002/nme.4512, URL http://dx.doi.org/10.1002/nme.4512.

[6] K. Lipnikov, G. Manzini, M. Shashkov, Mimetic finite difference method, Journal of Computational Physics 257, Part B (2014) 1163 – 1227, ISSN 0021-9991, doi:http://dx.doi.org/10.1016/j.jcp.2013.07.031, URL http://www.sciencedirect.com/science/article/pii/S0021999113005135, physics-compatible numerical methods.

[7] A. Bossavit, Whitney forms: A class of finite elements for three-dimensional computations in electromagnetism, IEEE Proc. 135 (8) (1988) 493–500.

[8] R. Eymard, T. Gallouët, R. Herbin, Finite volume methods, in: Solution of Equation in Rn (Part 3), Techniques of Scientific Computing (Part 3), vol. 7 of Handbook of Numerical Analysis, Elsevier, 713 – 1018, doi:10.1016/S1570-8659(00)07005-8, URL http://www.sciencedirect.com/science/article/pii/S1570865900070058, 2000.

[9] R. Nicolaides, K. Trapp, Covolume discretizations of differential forms, in: D. N. Arnold, P. Bochev, R. Lehoucq, R. Nicolaides, M. Shashkov (Eds.), Compatible Discretizations, Proceedings of IMA Hot Topics workshop on Compatible Discretizations, vol. IMA 142, Springer Verlag, 161–172, 2006.

[10] R. A. Nicolaides, X. Wu, Covolume Solutions of Three-Dimensional Div-Curl Equations, SIAM Journal on Numerical Analysis 34 (6) (1997) 2195–2203, doi:10.1137/S0036142994277286, URL http://link.aip.org/link/?SIN/34/2195/1.

[11] J. Hyman, M. Shashkov, Natural discretizations for the divergence, gradient, and curl on logically rectangular grids, Computers and Mathematics with Applications 33 (4) (1997) 81 – 104, ISSN 0898-1221, doi:10.1016/S0898-1221(97)00099-6, URL http://www.sciencedirect.com/science/article/pii/S0898122197000096.

[12] P. B. Bochev, J. M. Hyman, Principles of Mimetic Discretizations of Differential Operators, in: D. N. Arnold, P. B. Bochev, R. B. Lehoucq, R. A. Nicolaides, M. Shashkov (Eds.), Compatible Spatial Discretizations, vol. 142 of The IMA Volumes in Mathematics and its Applications, Springer New York, 89–119, 2006.

[13] C. Mattiussi, An analysis of finite volume, finite element and finite difference methods using some concepts from algebraic topology 133 (1997) 289–309.

[14] H. Wendland, Scattered data approximation, vol. 17, Cambridge university press, 2002.

[15] R. A. Nicolaides, X. Wu, Covolume Solutions of Three-Dimensional Div-Curl Equations, SIAM Journal on Numerical Analysis 34 (6) (1997) 2195–2203, doi:10.1137/S0036142994277286, URL http://link.aip.org/link/?SIN/34/2195/1.

[16] J. Hyman, M. Shashkov, Natural discretizations for the divergence, gradient, and curl on logically rectangular grids, Computers and Mathematics with Applications 33 (4) (1997) 81 – 104, ISSN 0898-1221, doi:10.1016/S0898-1221(97)00099-6, URL http://www.sciencedirect.com/science/article/pii/S0898122197000096.

[17] P. B. Bochev, J. M. Hyman, Principles of Mimetic Discretizations of Differential Operators, in: D. N. Arnold, P. B. Bochev, R. B. Lehoucq, R. A. Nicolaides, M. Shashkov (Eds.), Compatible Spatial Discretizations, vol. 142 of The IMA Volumes in Mathematics and its Applications, Springer New York, 89–119, 2006.

[18] C. Mattiussi, An analysis of finite volume, finite element and finite difference methods using some concepts from algebraic topology 133 (1997) 289–309.

[19] H. Wendland, Scattered data approximation, vol. 17, Cambridge university press, 2002.

[20] R. A. Nicolaides, X. Wu, Covolume Solutions of Three-Dimensional Div-Curl Equations, SIAM Journal on Numerical Analysis 34 (6) (1997) 2195–2203, doi:10.1137/S0036142994277286, URL http://link.aip.org/link/?SIN/34/2195/1.

[21] J. Hyman, M. Shashkov, Natural discretizations for the divergence, gradient, and curl on logically rectangular grids, Computers and Mathematics with Applications 33 (4) (1997) 81 – 104, ISSN 0898-1221, doi:10.1016/S0898-1221(97)00099-6, URL http://www.sciencedirect.com/science/article/pii/S0898122197000096.

[22] P. B. Bochev, J. M. Hyman, Principles of Mimetic Discretizations of Differential Operators, in: D. N. Arnold, P. B. Bochev, R. B. Lehoucq, R. A. Nicolaides, M. Shashkov (Eds.), Compatible Spatial Discretizations, vol. 142 of The IMA Volumes in Mathematics and its Applications, Springer New York, 89–119, 2006.

[23] C. Mattiussi, An analysis of finite volume, finite element and finite difference methods using some concepts from algebraic topology 133 (1997) 289–309.

[24] H. Wendland, Scattered data approximation, vol. 17, Cambridge university press, 2002.

[25] R. A. Nicolaides, X. Wu, Covolume Solutions of Three-Dimensional Div-Curl Equations, SIAM Journal on Numerical Analysis 34 (6) (1997) 2195–2203, doi:10.1137/S0036142994277286, URL http://link.aip.org/link/?SIN/34/2195/1.

[26] J. Hyman, M. Shashkov, Natural discretizations for the divergence, gradient, and curl on logically rectangular grids, Computers and Mathematics with Applications 33 (4) (1997) 81 – 104, ISSN 0898-1221, doi:10.1016/S0898-1221(97)00099-6, URL http://www.sciencedirect.com/science/article/pii/S0898122197000096.

[27] P. B. Bochev, J. M. Hyman, Principles of Mimetic Discretizations of Differential Operators, in: D. N. Arnold, P. B. Bochev, R. B. Lehoucq, R. A. Nicolaides, M. Shashkov (Eds.), Compatible Spatial Discretizations, vol. 142 of The IMA Volumes in Mathematics and its Applications, Springer New York, 89–119, 2006.

[28] C. Mattiussi, An analysis of finite volume, finite element and finite difference methods using some concepts from algebraic topology 133 (1997) 289–309.

[29] H. Wendland, Scattered data approximation, vol. 17, Cambridge university press, 2002.
[28] T. J. R. Hughes, A. Masud, J. Wan, A Stabilized Mixed Discontinuous Galerkin Method for Darcy Flow, Computer Methods in Applied Mechanics and Engineering 195 (2006) 3347–3381.

[29] A. Masud, T. J. R. Hughes, A stabilized finite element method for Darcy flow, Computer Methods in Applied Mechanics and Engineering 191.

[30] A. N. Brooks, T. J. Hughes, Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations., Computer Methods in Applied Mechanics and Engineering 32 (1–3) (1982) 199 – 259, ISSN 0045-7825, doi:10.1016/0045-7825(82)90071-8, URL http://www.sciencedirect.com/science/article/pii/0045782582900718.

[31] P. Knobloch, On the definition of the SUPG parameter, ETNA 32 (2008) 76–89, URL http://etna.mcs.kent.edu/vol.32.2008/pp76-89.dir/.

[32] T. J. Hughes, M. Mallet, A new finite element formulation for computational fluid dynamics: IV. A discontinuity-capturing operator for multidimensional advective-diffusive systems, Computer Methods in Applied Mechanics and Engineering 58 (3) (1986) 329 – 336, ISSN 0045-7825, doi:10.1016/0045-7825(86)90153-2, URL http://www.sciencedirect.com/science/article/pii/0045782586901532.