# A HIGH-ORDER STAGGERED MESHLESS METHOD FOR ELLIPTIC PROBLEMS

NATHANIEL TRASK *, MAURO PEREGO†, AND PAVEL BOCHEV†

**Abstract.** We present a new meshless method for scalar diffusion equations, which is motivated by their compatible discretizations on primal-dial grids. Unlike the latter though, our approach is truly meshless because it only requires the graph of nearby neighbor connectivity of the discretization points $\boldsymbol{x}_i$. This graph defines a local primal-dual grid complex with a *virtual* dual grid, in the sense that specification of the dual metric attributes is implicit in the method's construction. Our method combines a topological gradient operator on the local primal grid with a Generalized Moving Least Squares approximation of the divergence on the local dual grid. We show that the resulting approximation of the div-grad operator maintains polynomial reproduction to arbitrary orders and yields a meshless method, which attains $O(h^m)$ convergence in both $L^2$ and $H^1$ norms, similar to mixed finite element methods. We demonstrate this convergence on curvilinear domains using manufactured solutions. Application of the new method to problems with discontinuous coefficients reveals solutions that are qualitatively similar to those of compatible mesh-based discretizations.

**Key words.** Generalized moving least squares, primal-dual grid methods, compatible discretizations, mixed methods, div-grad system.

**AMS subject classifications.**

**1. Introduction.** Consider a bounded region in $\mathbb{R}^d$, $d = 2, 3$ with a Lipschitz continuous boundary $\Gamma = \partial\Omega$. This paper presents a new staggered meshless discretization approach for the model elliptic boundary value problem

$$\begin{aligned} -\nabla \cdot \mu \nabla \phi &= f \quad \text{in } \Omega \\ \phi &= u \quad \text{on } \Gamma_D \\ \boldsymbol{n} \cdot \mu \nabla \phi &= g \quad \text{on } \Gamma_N \end{aligned} \tag{1.1}$$

where $\Gamma_D$ and $\Gamma_N$ denote Dirichlet and Neumann parts of the boundary $\Gamma$, respectively, $\mu$ is a symmetric positive definite tensor describing a material property and $f$, $u$ and $g$ are given data.

In important applications such as porous media flow [1], heat transfer [2] and semiconductor devices [3], the flux $\boldsymbol{u} = -\mu\nabla\phi$ is the variable of primary interest. Predictive simulations of such problems require carefully constructed approximations of the divergence and gradient operators, which ensure stable, accurate and locally conservative flux approximations.

Generally speaking, such *compatible discretizations* of (1.1) fall into one of the following two categories. *Single grid* methods such as: mixed finite elements [4]; virtual elements [5]; or mimetic finite differences [6], [7], construct an approximation of one of the two operators (typically the divergence) on the given grid and then use its adjoint[1] as a proxy for the other operator (the gradient). In so doing these

*Division of Applied Mathematics, Brown University, 182 George St. Providence, RI 02906 (nat.trask@gmail.com)

†Center for Computing Research, Sandia National Laboratories, Mail Stop 1320 Albuquerque, New Mexico, 87185-1320 ({pbboche,mperego}@sandia.gov). Sandia National Laboratories is a multiprogram laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000

[1]Multiplication of $\boldsymbol{u} = -\mu\nabla\phi$ by a test function $\boldsymbol{v}$ followed by integration by parts yields a notion of a weak gradient in mixed finite elements. Other methods, such as mimetic finite differences define the gradient as the algebraic adjoint of the discrete divergence.

methods satisfy a discrete version of Green's theorem.

In contrast, *primal-dual grid* approaches, such as box integration [8], covolume [9, 10], or finite volume [11] schemes use the two grids in the primal-dual complex to approximate the divergence and the gradient independently. When the two grids are topologically dual, such as in the case of Voronoi-Delauney triangulations or rectilinear[2] primal grids, these methods assume a particularly simple and elegant form.

The abundance of compatible mesh-based methods that provide accurate and locally conservative flux approximations stands in stark contrast with the almost complete absence of meshless methods that would provide comparable results but without the need for a global mesh structure. Few notable exceptions are the uncertain grid method [12] and the conceptually similar meshless volume schemes [13, 14] for conservation laws.

Motivated by compatible discretizations of (1.1) on primal-dual grids, we present a new meshless scheme for this problem that exhibits similar computational properties, most notably an ability to produce accurate, non-oscillatory approximations for problems with material discontinuities. Our approach is truly meshless because it only requires the graph of nearby neighbor connectivity of every discretization point $\boldsymbol{x}_i$. The vertices and the edges of this graph define a local primal grid for every mesh point, which induces a virtual local dual grid comprising a virtual cell dual to $\boldsymbol{x}_i$ and a set of virtual faces dual to the edges of the connectivity graph.

Following primal-dual grid methods we discretize the div-grad operator using independent approximations of the divergence and the gradient on the local grid complex. Specifically, we combine a topological gradient on the local primal grid with a Generalized Moving Least Squares (GMLS) approximation of the divergence on the local virtual dual grid. The latter is virtual in the sense that our approach does not require its physical construction, instead, specification of dual cell volumes and dual face areas is implicit in the method's construction. We show that the resulting meshless approximation of the div-grad operator retains polynomial reproduction of arbitrary orders. Because of its conceptual similarity with primal-dual schemes we call the resulting method a *staggered GMLS* discretization of (1.1).

Our new method also differs in important and significant ways from the conservative meshless schemes in [12, 13, 14]. While these approaches similarly use different sets of degrees-of-freedom to discretize the gradient and divergence, they rely on a global Quadratic Program (QP) to determine the meshless analogues of cell volumes and face areas that ensure the compatibility of the scheme. As a result, application of these methods to large-scale problems would require the formulation of a scalable, QP-specific optimization solver [14]. In contrast, our approach requires the solution of many independent and inexpensive local optimization problems, allowing an $O(N)$ implementation in the number of degrees of freedom.

A second important difference is the order of accuracy achieved by these schemes and our new method. The accuracy of the former is similar to that of comparable finite volume schemes, i.e., they are roughly second-order accurate. In contrast, our method attains $O(h^m)$ convergence in both $L^2$ and $H^1$ norms, similar to mixed finite element methods.

It is worth mentioning that some variations of smooth particle hydrodynamics (SPH), which use discretely skew-adjoint divergence and gradient operators [15, 16],

[2] In this case the dual grid is a translation of the primal grid by a half cell size. This creates the appearance of all variables living on the same grid but at different locations, thus the term "staggered grid methods".

pursue a somewhat different type of compatibility. However, it is well-known [17] that this compatibility in SPH comes at the expense of the gradient operators lacking even zeroth-order polynomial consistency, and many applications require techniques to numerically stabilize the method in the presence of large differences in material properties such as density or viscosity (e.g. [18]). In contrast to this, we will show that our method maintains high-order polynomial reproduction and remains numerically stable across jumps in material properties of several orders of magnitude.

We have organized the paper as follows. Section 2 introduces notation and reviews basic concepts of Generalized Moving Least Squares, while Section 3 specializes these concepts to approximation of vector fields and their derivatives from directional components. Section 4 presents the development of the new staggered GMLS, and Section 5 studies its accuracy. We discuss implementation details in Section 6 and present numerical results in Section 7. Section 8 summarizes our findings.

**2. Notation and quotation of results.** Throughout this paper bold face fonts denote various vector quantities, e.g., $\boldsymbol{x}$ is a point in the Euclidean space $\mathbb{R}^d$, $\boldsymbol{u}$ is a vector field in $\mathbb{R}^d$, $\boldsymbol{e}_{ij}$ is an edge connecting points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and so on. Lower case fonts stand for various scalar quantities, upper case Roman fonts are primarily used to denote function spaces, operators and sets of geometric entities, while the first few lower case Greek symbols will stand for multi-indices, e.g., $\alpha = (\alpha_1, \dots, \alpha_d)$ and $|\alpha| = \sum \alpha_i$. We use the standard Euler notation $D^\alpha$ for a partial derivative of order $|\alpha|$, i.e., $D^\alpha := \partial^{|\alpha|}/\partial_{\boldsymbol{x}_i^{\alpha_1}} \dots \partial_{\boldsymbol{x}_i^{\alpha_i}}$ and denote the standard Euclidean norm by $\|\cdot\|$.

As usual, $C^m(\Omega)$ is the space of all continuous functions whose derivatives up to order $m$ are also continuous and $P_m(\mathbb{R}^d)$, or simply $P_m$ is the space of all multivariate polynomials of degree less than or equal to $m$.

**2.1. Local geometric structures.** In this paper we consider discretization of $\Omega$ by a set of points $\Omega_h = \{\boldsymbol{x}_i\}_{i=1,\dots,N}$. We denote all boundary points by $\Gamma_h$ and set $\breve{\Omega}_h = \Omega_h \setminus \Gamma_h$. The $\varepsilon$-neighborhood of $\boldsymbol{x}_i \in \Omega_h$ is the set

$$N_i^\varepsilon := \{\boldsymbol{x}_j \in \Omega_h \,|\, \|\boldsymbol{x}_j - \boldsymbol{x}_i\| < \varepsilon\},$$

where $\varepsilon > 0$ is given and may depend on the point location. We use the local connectivity graph of the points in $N_i^\varepsilon$ to construct an approximation of the div-grad operator that mimics the one in primal-dual grid methods. Specifically, with every $\varepsilon$-neighborhood we associate a *local primal grid* comprising a local vertex set

$$V_i = \{\boldsymbol{v}_j = \boldsymbol{x}_j \,|\, \boldsymbol{x}_j \in N_i^\varepsilon\},$$

which is simply the set of all points in $N_i^\varepsilon$, and a local edge set

$$E_i = \{\boldsymbol{e}_{ij} = \boldsymbol{x}_j - \boldsymbol{x}_i \,|\, \boldsymbol{x}_j \in N_i^\varepsilon\}.$$

The edges in $E_i$ have midpoints, half-edges, and unit tangents given by

$$\boldsymbol{x}_{ij} = \frac{\boldsymbol{x}_i + \boldsymbol{x}_j}{2}, \quad \boldsymbol{m}_{ij} = \boldsymbol{x}_{ij} - \boldsymbol{x}_i, \quad \text{and} \quad \boldsymbol{t}_{ij} = \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|} = \frac{\boldsymbol{x}_j - \boldsymbol{x}_i}{\|\boldsymbol{e}_{ij}\|},$$

respectively; see Fig. 1. We denote the set of all midpoints by $M_i$. Note that $E_i$ does not contain edges that do not have $\boldsymbol{x}_i$ as a vertex.

The local primal grid induces a *local virtual dual grid* comprising a single virtual cell $C_i$ dual to vertex $\boldsymbol{v}_i$ and a set of virtual faces $F_i = \{\boldsymbol{f}_{ij}\}$ dual to $E_i$, intersecting $\boldsymbol{e}_{ij}$ at $\boldsymbol{x}_{ij}$ and having face normals equivalent to edge tangents, i.e.,

$$\forall \boldsymbol{f}_{ij} \in F_i, \quad \boldsymbol{f}_{ij} \cap \boldsymbol{e}_{ij} = \boldsymbol{x}_{ij} \quad \text{and} \quad \boldsymbol{n}_{ij} = \boldsymbol{t}_{ij}\,. \tag{2.1}$$

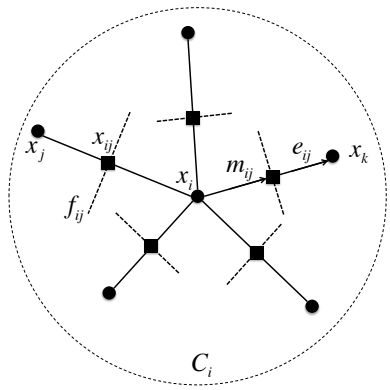


FIG. 1. *A local primal-dual grid complex induced by a point $\boldsymbol{x}_i$ and its $\varepsilon$-neighborhood $N_i^\varepsilon$. The primal edges $\boldsymbol{e}_{ij}$, midpoints $\boldsymbol{x}_{ij}$ and mid-edges $\boldsymbol{m}_{ij}$ are physical mesh entities. The dual cell $C_i$ and the dual faces $\boldsymbol{f}_{ij}$ are virtual mesh entities.*

REMARK 1. *Although the local primal mesh contains "real" mesh entities in the sense that the elements of $V_i$ and $E_i$ have geometric attributes such as coordinates and lengths, they are not used directly in the formulation of our method. Instead, all necessary metric quantities are determined implicitly by the method's construction.*

**2.2. Local approximation spaces and operators.** On the local primal grid we define the local vertex space

$$V^i = \{u_j \in \mathbb{R} \,|\, \forall \boldsymbol{v}_j \in V_i\}$$

and the local edge space

$$E^i = \{u_{ij} \in \mathbb{R} \,|\, \forall \boldsymbol{e}_{ij} \in E_i\}\,.$$

The elements of $V^i$ and $E^i$ are sets of real values associated with the local vertices and edges, respectively. In particular, we assume that edge values $u_{ij}$ are localized at the edge midpoints $\boldsymbol{x}_{ij}$ and represent the tangent components of some vector field $\boldsymbol{u}$ along the edges, i.e., $u_{ij} = \boldsymbol{u}_{ij} \cdot \boldsymbol{t}_{ij}$.

Likewise, on the local virtual dual grid we define the cell space

$$C^i = \{u_i \in \mathbb{R}\}$$

comprising a single scalar value on the virtual dual cell $C_i$, and the face space

$$F^i = \{u_{ij} \in \mathbb{R} \,|\, \forall \boldsymbol{f}_{ij} \in F_i\}$$

containing real values associated with the virtual dual faces. We assume that the face values $u_{ij}$ are localized at edge-face intersection points $\boldsymbol{x}_{ij}$ and represent the normal components of some vector field $\boldsymbol{u}$, i.e., $u_{ij} = \boldsymbol{u}_{ij} \cdot \boldsymbol{n}_{ij}$.

Thanks to the topological duality (2.1) between $E_i$ and $F_i$, the corresponding edge and face spaces $E^i$ and $F^i$ are isomorphic. Thus, any tangent vector component $u_{ij} = \boldsymbol{u}_{ij} \cdot \boldsymbol{t}_{ij}$ on a primal edge $\boldsymbol{e}_{ij}$ can be viewed as a normal vector component $\boldsymbol{u}_{ij} \cdot \boldsymbol{n}_{ij}$ on the dual face $\boldsymbol{f}_{ij}$ and vice versa. Thus, our local approximation spaces reproduce the key property of primal-dual grid complexes without requiring a global mesh data structure.

To complete the specification of the discrete structures necessary for our new scheme it remains to endow the above discrete spaces with suitable notions of a gradient and divergence operators. Taking clue from primal-dual grid methods we seek the discrete gradient as a mapping $GRAD_i : V^i \to E^i$ on the local primal mesh, and the discrete divergence as a mapping $DIV_i : F^i \to C^i$ on the local virtual dual mesh. As in a primal-dual methods, the isomorphism between $E^i$ and $F^i$ then allows us to "chain" these operators into an approximation of the div-grad operator.

Avoiding an explicit dependence on any metric entities such as areas, volumes and lengths is a key requirement for the construction of $GRAD_i$ and $DIV_i$. This enables us to bypass a global optimization problem as required in [14]. To this end, we choose to define $GRAD_i$ as the *topological* gradient [19, 20], i.e., we set

$$GRAD_i(u^h) = \boldsymbol{u}^h \in E^i \Leftrightarrow u_{ij} = u_j - u_i \quad \forall u^h \in V^i. \tag{2.2}$$

The operator (2.2) is topological because it depends on the connectivity of the local vertices and edges in $N_i^\varepsilon$, but not on their physical locations. The node-to-edge incidence matrix of the local primal grid is the algebraic representation of $GRAD_i$.

We will subsume all necessary metric attributes of our scheme in the definition of $DIV_i$ by using a Generalized Moving Least Squares approach [21] to define this operator. We now review key aspects of the GMLS theory necessary for this task. Then in Section 3 we extend the approach in [21] to approximate vector fields and their derivatives from scattered directional components

**2.3. Generalized Moving Least Squares (GMLS) framework.** Following [22, Section 4.3] we consider an abstract setting given by

- a function space $V$ with a dual $V^*$;
- a finite dimensional space $P = \text{span}\{p_1, \ldots, p_Q\} \subset V$;
- a finite set of linear functionals $\Lambda = \{\lambda_1, \ldots, \lambda_N\} \subset V^*$; and
- a correlation (weight) function $\omega : V^* \times V^* \mapsto \mathbb{R}^+ \cup \{0\}$.

We assume that $\Lambda$ is $P$-unisolvent, that is

$$\{p \in P \,|\, \lambda_i(p) = 0, i = 1, \ldots, N\} = \{0\}. \tag{2.3}$$

In other words, the zero is the only element of $P$ for which all functionals in $\Lambda$ vanish.

Given a *target* functional $\tau \in V^*$ GMLS seeks to approximate its action on any function $u \in V$ by a linear combination

$$\tau(u) \approx \widetilde{\tau}(u) := \sum_{i=1}^{N} a_i(\tau)\lambda_i(u) \tag{2.4}$$

of its samples $\{\lambda_1(u), \ldots, \lambda_N(u)\}$ such that

**P.1** The approximation (2.4) is exact for $P$, i.e.,

$$\tau(u) = \widetilde{\tau}(u) := \sum_{i=1}^{N} a_i(\tau)\lambda_i(u) \quad \forall u \in P. \tag{2.5}$$

**P.2** The coeffcients $a_i(\tau)$ have local supports relative to $\omega$, i.e.,

$$\omega(\tau; \lambda_i) = 0 \quad \Rightarrow \quad a_i(\tau) = 0. \tag{2.6}$$

**P.3** The coefficients $a_i(\tau)$ are uniformly bounded in $\tau$, i.e.,

$$\exists C > 0 \quad \text{s.t.} \sum_{i=1}^{N} |a_i(\tau)| \le C \quad \forall \tau \in V^*. \tag{2.7}$$

In what follows, we refer to **P.1** as *P-reproduction* property, $\Lambda$ as the *sampling* set, $P$ - as the *reproduction property* space, and $a_i(\tau)$ - the basis functions. Following [22] we define the basis functions as the optimal solution of the following Quadratic Program (QP):

$$\text{(2.8)} \quad \text{minimize } \frac{1}{2}\sum_{i=1}^{N}\frac{a_i(\tau)^2}{\omega(\tau;\lambda_i)} \quad \text{subject to} \quad \sum_{i=1}^{N} a_i(\tau)\lambda_i(p_k) = \tau(p_k) \;\; k = 1, 2\ldots, Q.$$

The objective in (2.8) enforces the local support of the basis, while the constraint enforces $P$-reproduction. The $P$-unisolvency condition (2.3) is sufficient for (2.8) to have a unique minimizer that satisfies **P.1**–**P.3**; see [22, Theorem 4.9, p.44]. With the notation

$$\text{(2.9)} \quad \begin{aligned} &\boldsymbol{a}(\tau) = (a_i(\tau))_{i=1}^{N} \in \mathbb{R}^N; \quad \boldsymbol{r}(\tau) = (\tau(p_k))_{k=1}^{Q} \in \mathbb{R}^Q; \\ &W(\tau) = \operatorname{diag}(\omega(\tau;\lambda_i)) \in \mathbb{R}^{N\times N}; \quad P(\Lambda) = (\lambda_i(p_j)) \in \mathbb{R}^{Q\times N} \end{aligned}$$

the QP (2.8) assumes an equivalent algebraic form

$$\text{(2.10)} \quad \text{minimize } \frac{1}{2}\boldsymbol{a}(\tau)^T W(\tau)\boldsymbol{a}(\tau) \quad \text{subject to} \quad P(\Lambda)\boldsymbol{a}(\tau) = \boldsymbol{r}(\tau).$$

It is easy to see that the optimal solution of (2.10), resp (2.8) is given by

$$\text{(2.11)} \quad \boldsymbol{a}(\tau) = W(\tau)P^T(\Lambda)\left(P(\Lambda)W(\tau)P^T(\Lambda)\right)^{-1}\boldsymbol{r}(\tau)$$

and so, the GMLS approximation (2.4) assumes the form

$$\widetilde{\tau}(u) = \boldsymbol{\ell}^T(u)\boldsymbol{a}(\tau) = \boldsymbol{\ell}^T(u)W(\tau)P^T(\Lambda)\left(P(\Lambda)W(\tau)P^T(\Lambda)\right)^{-1}\boldsymbol{r}(\tau),$$

with $\boldsymbol{\ell}^T(u) = (\lambda_i(u))_{i=1}^{N} \in \mathbb{R}^N$. Setting

$$\text{(2.12)} \quad \boldsymbol{b}(\tau)^T = \boldsymbol{\ell}^T(u)W(\tau)P^T(\Lambda)\left(P(\Lambda)W(\tau)P^T(\Lambda)\right)^{-1}$$

allows us to write the approximation of the target functional as

$$\text{(2.13)} \quad \widetilde{\tau}(u) = \boldsymbol{\ell}^T(u)\boldsymbol{a}(\tau) = \boldsymbol{b}(\tau)^T\boldsymbol{r}(\tau).$$

In other words, we can switch from a representation of $\widetilde{\tau}(u)$ in terms of a linear combination of samples $\lambda_i(u)$ to a representation in terms of a linear combination of the values $\tau(p_k)$ of the target functional at the basis of $P$. It is straightforward to show that $\boldsymbol{b}(\tau)$ admits the following variational characterization

$$\text{(2.14)} \quad \boldsymbol{b}(\tau) = \operatorname*{argmin}_{\boldsymbol{c}\in\mathbb{R}^Q} \frac{1}{2}\left(\boldsymbol{\ell}(u) - P^T(\Lambda)\boldsymbol{c}\right)^T W(\tau)\left(\boldsymbol{\ell}(u) - P^T(\Lambda)\boldsymbol{c}\right).$$

The algebraic least-squares problem (2.14) is equivalent to finding the best weighted least-squares fit to the data $\{\lambda_i(u)\}$ out of the space $P$ in the sense that the function

$$\text{(2.15)} \quad p^* := \boldsymbol{b}(\tau)^T\boldsymbol{p} = \sum_{k=1}^{Q} b_k(\tau)p_k,$$

where $\boldsymbol{p} = (p_1, \ldots, p_Q)$, satisfies

$$p^* = \operatorname*{argmin}_{p \in P} \frac{1}{2} \sum_{i=1}^{N} \left(\lambda_i(u) - \lambda_i(p)\right)^2 \omega(\tau; \lambda_i). \tag{2.16}$$

This motivates the notation

$$\widetilde{\tau}(p^*) := \boldsymbol{b}(\tau)^T \boldsymbol{r}(\tau) = \sum_{k=1}^{Q} b_k(\tau)\tau(p_k). \tag{2.17}$$

for the GMLS approximation of the target functional $\tau$.

In some important settings, such as the approximation of functions and their derivatives from scattered point values, the characterization (2.16), resp., (2.14) and the alternative representation $\widetilde{\tau}(u) = \widetilde{\tau}(p^*) = \boldsymbol{b}(\tau)^T \boldsymbol{r}(\tau)$ offer some computational advantages. We consider such settings and their specialization to our needs in the next section.

**3. GMLS approximation of vector fields and their divergence from directional components.** In this section we apply the abstract GMLS framework to define a discrete divergence operator $DIV_i : F^i \mapsto C^i$. Our approach draws upon ideas of Mirzaei et al [21, 23] on GMLS approximations of scalar functions and their derivatives from point function values. Accordingly, we start with a brief summary of the relevant aspects of their work.

**3.1. GMLS approximation of scalar functions and their derivatives.** Let $m > 0$ be some fixed integer. We consider the GMLS framework in Section 2.3 with

$$V = C^{m+1}(\Omega), \quad P = P_m \subset V, \quad \Lambda = \{\delta_{\boldsymbol{x}_1}, \ldots, \delta_{\boldsymbol{x}_N}\}, \quad \text{and} \quad \tau(u) = \delta_{\boldsymbol{x}} \circ D^\alpha,$$

where $X = \{\boldsymbol{x}_i, \ldots, \boldsymbol{x}_N\}$ is a finite set of points in $\Omega$, $\boldsymbol{x} \in \Omega$ is a given point assumed to be distinct from the points in $X$ and $\alpha$ is a multi index. If $|\alpha| = 0$ then $\tau(u) = \delta_{\boldsymbol{x}}$, i.e., the target functional reduces to point evaluation at $\boldsymbol{x}$.

We assume that $X$ is such that the associated sampling set $\Lambda$ is $P_m$-unisolvent. The following conditions are sufficient to ensure this property; see [22, Theorem 4.7, p.41],[22, Theorem 3.14, p.33], [21, Definition 4.4]:

**C.1** The region $\Omega \in \mathbb{R}^d$ is compact and satisfies an interior cone condition with radius $r > 0$ and angle $\theta \in (0, \pi/2)$.

**C.2** The point set $X = \{\boldsymbol{x}_i\}$ is quasi-uniform with fill distance (see [22, Definition 4.6, p.41] $h_{X,\Omega} \leq h_0$, where $h_0 = r/C(\theta)$ and $C(\theta) = 16(1 + \sin\theta)^2 m^2/(3\sin^2\theta)$.

Because both the target functional $\tau$ and the sampling set functionals $\lambda_i$ depend on the spatial location we can choose a weight function according to

$$\omega(\tau; \lambda_i) = \omega(\delta_{\boldsymbol{x}} \circ D^\alpha; \delta_{\boldsymbol{x}_i}) := \Phi(||\boldsymbol{x} - \boldsymbol{x}_i||) \tag{3.1}$$

where $\Phi(x)$ is a compactly supported, positive, radially symmetric function. In particular, we assume that the weight function satisfies the following condition:

**C.3** The support of the weight function $\Phi$ is contained in $\mathcal{B}(0, \delta)$ and $\Phi > 0$ in $\mathcal{B}(0, \delta/2)$, where $\delta = 2C(\theta)h_{X,\Omega}$; see [22, p.36].

Conditions **C.1**–**C.3** are sufficient to ensure that properties **P.1**-**P.3** hold in the present context. Thus, in what follows we restrict attention to GMLS settings satisfying these conditions.

Recall the equivalent GMLS approximations of the target functional in (2.13). We now explain why in the present setting the representation $\widetilde{\tau}(u) = \boldsymbol{b}(\tau)^T \boldsymbol{r}(\tau)$ could be more computationally efficient than the equivalent representation $\widetilde{\tau}(u) = \boldsymbol{\ell}^T(u)\boldsymbol{a}(\tau)$ in terms of the basis functions.

Indeed, according to (2.12) the coefficient vector $\boldsymbol{b}(\tau)$ depends on $\tau$ through the matrix $W(\tau)$. However, the choice of weight function (3.1) makes $W(\tau)$ dependent on the point location $\boldsymbol{x}$ *but not on the derivative component* $D^\alpha$. It follows that $\boldsymbol{b}$ will likewise depend on $\boldsymbol{x}$ but not on $D^\alpha$. In other words,

$$\widetilde{\tau}(u) = \widetilde{\delta_{\boldsymbol{x}} \circ D^\alpha}(u) = \boldsymbol{b}(\boldsymbol{x})^T \boldsymbol{r}(\delta_{\boldsymbol{x}} \circ D^\alpha). \tag{3.2}$$

In contrast, (2.11) reveals that the basis functions $\boldsymbol{a}(\tau)$ depend both on $\boldsymbol{x}$ through $W(\boldsymbol{x})$ *and the derivative component* $D^\alpha$ through $\boldsymbol{r}(\tau) = \boldsymbol{r}(\delta_{\boldsymbol{x}} \circ D^\alpha)$, i.e.,

$$\widetilde{\tau}(u) = \widetilde{\delta_{\boldsymbol{x}} \circ D^\alpha}(u) = \boldsymbol{\ell}^T(u)\boldsymbol{a}(\delta_{\boldsymbol{x}} \circ D^\alpha). \tag{3.3}$$

Thus, if $\delta_{\boldsymbol{x}} \circ D^\alpha$ is required for multiple values of $\alpha$, the coefficient vector $\boldsymbol{b}(\boldsymbol{x})$ must be computed only once, whereas the basis functions $\boldsymbol{a}(\delta_{\boldsymbol{x}} \circ D^\alpha)$ must be recomputed for every new value of $\alpha$. Inversion of $P(\Lambda)W(\tau)P^T(\Lambda)$ comprises the bulk of the computational cost in both cases. Using (3.2) requires a single inversion of this matrix, whereas (3.3) formally requires a separate inversion for every new target functional.

Remark 2. *In practice the actual computational advantage of* (3.2) *over* (3.3) *depends on multiple factors such as the number of target functionals, the size of* $X$*, and the polynomial degree* $m$ *of the reconstruction space* $P_m$*. If* $m$ *is small one could in principle store* $\big(P(\Lambda)W(\tau)P^T(\Lambda)\big)^{-1}$ *in factored form at every point* $\boldsymbol{x}_i$ *and apply it to multiple right hand sides given by* $\boldsymbol{r}(\delta_{\boldsymbol{x}} \circ D^\alpha)$*. In this case the cost of* (3.3) *is comparable to that of* (3.2)*. However, for larger polynomial degrees and/or large number of points the storage requirements may be prohibitively large, making* (3.2) *a better computational option.*

In light of Remark 2 in what follows we use exclusively the representation (3.2). Its computation decouples into a solution of a weighted least-squares problem

$$\boldsymbol{b}(\boldsymbol{x}) = \operatorname*{argmin}_{\boldsymbol{c}\in\mathbb{R}^Q} \frac{1}{2} \left(\boldsymbol{\ell}(u) - P^T(\Lambda)\boldsymbol{c}\right)^T W(\boldsymbol{x}) \left(\boldsymbol{\ell}(u) - P^T(\Lambda)\boldsymbol{c}\right) \tag{3.4}$$

for the coefficients $\boldsymbol{b}(\boldsymbol{x})$, which is independent of the derivative part $D^\alpha$, followed by an application of the target functional to the basis functions of $P = P_m$.

Remark 3. *Analogous to the relationship between* (2.14) *and* (2.16)*, the algebraic least-squares problem* (3.4) *is equivalent to finding the best weighted least-squares polynomial fit to a set of function values*

$$p^*(\boldsymbol{x}) = \operatorname*{argmin}_{p\in P_m} \frac{1}{2} \sum_{i=1}^{N} \left(u(\boldsymbol{x}_i) - p(\boldsymbol{x}_i)\right)^2 \Phi(||\boldsymbol{x} - \boldsymbol{x}_i||) \tag{3.5}$$

*in the sense that, given a basis* $\{p_1, \ldots, p_Q\}$ *of* $P_m$*, there holds*

$$p^*(\boldsymbol{x}) = \boldsymbol{b}(\boldsymbol{x})^T \boldsymbol{r}(\delta_{\boldsymbol{x}}) = \sum_{i=1}^{Q} b_i(\boldsymbol{x}) p_i(\boldsymbol{x}). \tag{3.6}$$

*Likewise, for* $\tau = \delta_{\boldsymbol{x}} \circ D^\alpha u$ *the abstract formula* (2.17) *specializes to*

$$\widetilde{\delta_{\boldsymbol{x}} \circ D^\alpha}(u) = \widetilde{\delta_{\boldsymbol{x}} \circ D^\alpha}(p^*) := \boldsymbol{b}(\boldsymbol{x})^T \boldsymbol{r}(\delta_{\boldsymbol{x}} \circ D^\alpha) = \sum_{i=1}^{Q} b_i(\boldsymbol{x}) D^\alpha p_i(\boldsymbol{x}). \tag{3.7}$$

*The latter has the* appearance *of an approximation to* $D^\alpha p^*(\boldsymbol{x})$ *computed by neglecting the dependence of the coefficients* $b_i$ *on the spatial location. This is why* (3.7) *has been often referred to as a "diffuse derivative" [24] and deemed inferior to the true derivative* $D^\alpha p^*(\boldsymbol{x})$*. This misconception had been first pointed out by [21], which also provided a correct interpretation of* (3.7) *using the GMLS framework. Following their suggestion we refer to* $\widetilde{\delta_{\boldsymbol{x}} \circ D^\alpha}(u)$ *as the "GMLS derivative approximation", or simply the "GMLS derivative", instead of "diffuse derivative".*

To simplify notation we adopt the more compact symbol

$$\widetilde{D^\alpha} u(\boldsymbol{x}) := \widetilde{\delta_{\boldsymbol{x}} \circ D^\alpha}(u)$$

for the GMLS derivative. Assuming that conditions **C.1**–**C.3** hold Mirzaei et al [21, Corollary 4.13] prove that this derivative satisfies the following error bound:

$$|\widetilde{D^\alpha} u(\boldsymbol{x}) - D^\alpha u(\boldsymbol{x})| \le C h_{X,\Omega}^{m+1-|\alpha|} \|u\|_{C^{m+1}}. \tag{3.8}$$

**3.2. A GMLS divergence operator.** We now focus on the application of the GMLS approach in Section 3.1 to define a discrete divergence operator $DIV_i : F^i \mapsto C^i$. This operator should map the normal vector field components defined on the local virtual cell faces $\boldsymbol{f}_{ij}$ to a single value on the local virtual cell $C_i$. Thus, to discuss the construction of $DIV_i$ it suffices to consider an arbitrary point $\boldsymbol{x}_i \in \Omega_h$ and its associated local primal-dual grid on $N_i^\varepsilon$.

Owing to the duality relationship between $F_i$ and $E_i$, the normal components of $\boldsymbol{u}$ across the virtual faces $\boldsymbol{f}_{ij}$ coincide with the tangent components $u_{ij} = \boldsymbol{u}_{ij} \cdot \boldsymbol{t}_{ij}$ of this field along the local edges $\boldsymbol{e}_{ij}$. Thus, the task of defining $DIV_i$ essentially boils down to the task of approximating $\nabla \cdot \boldsymbol{u}(\boldsymbol{x}_i)$ from this data. We also note that using a set of non-normalized tangent vectors to the edges results in an equivalent set of degrees of freedom. In particular, the choice

$$u_{ij} = \boldsymbol{u}(\boldsymbol{x}_{ij}) \cdot 2\boldsymbol{m}_{ij} = \boldsymbol{u}(\boldsymbol{x}_{ij}) \cdot 2(\boldsymbol{x_{ij}} - \boldsymbol{x}_i) \tag{3.9}$$

brings about some simplifications in the subsequent formulas and so, from now on we assume that the elements of $E^i$ are defined by (3.9).

Given a vector field $\boldsymbol{u} \in (C^{m+1}(\Omega))^d$ we call the scalar function

$$u_{i\to}(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x}) \cdot 2(\boldsymbol{x} - \boldsymbol{x}_i) \tag{3.10}$$

the radial component of $\boldsymbol{u}$ in the direction of $\boldsymbol{x}_i$. Clearly, $u_{i\to} \in C^{m+1}(\Omega)$,

$$u_{i\to}(\boldsymbol{x}_{ij}) = \boldsymbol{u}(\boldsymbol{x}_{ij}) \cdot 2\boldsymbol{m_{ij}} = u_{ij} \quad \text{and} \quad u_{i\to}(\boldsymbol{x}_i) = 0. \tag{3.11}$$

The following result shows that the vector field $\boldsymbol{u}$ and its divergence at $\boldsymbol{x}_i$ can be both expressed in terms of the derivatives of its radial component function.

LEMMA 3.1. *Let* $\boldsymbol{u} \in (C^{m+1}(\Omega))^d$ *and* $u_{i\to} \in C^{m+1}(\Omega)$ *be its radial component function* (3.10)*. There holds*

$$\boldsymbol{u}(\boldsymbol{x}_i) = \frac{1}{2}\nabla u_{i\to}(\boldsymbol{x}_i) \quad \textit{and} \quad \nabla \cdot \boldsymbol{u}(\boldsymbol{x}_i) = \frac{1}{4}\Delta u_{i\to}(\boldsymbol{x}_i). \tag{3.12}$$

*Proof.*

Taking the gradient of the radial component function gives

$$\nabla u_{i\to}(\boldsymbol{x}) = 2\nabla \boldsymbol{u}(\boldsymbol{x})^T(\boldsymbol{x} - \boldsymbol{x}_i) + 2\boldsymbol{u}(\boldsymbol{x}).$$

Taking the divergence of this identity gives

$$\begin{aligned}\nabla\cdot\nabla u_{i\to}(\boldsymbol{x}) &= 2\big(\nabla\cdot\nabla\boldsymbol{u}(\boldsymbol{x})\big)\cdot(\boldsymbol{x}-\boldsymbol{x}_i)+2\nabla\boldsymbol{u}(\boldsymbol{x}):\nabla\boldsymbol{x}+2\nabla\cdot\boldsymbol{u}(\boldsymbol{x})\\ &= 2\big(\nabla\cdot\nabla\boldsymbol{u}(\boldsymbol{x})\big)\cdot(\boldsymbol{x}-\boldsymbol{x}_i)+4\nabla\cdot\boldsymbol{u}(\boldsymbol{x})\end{aligned}$$

Setting $\boldsymbol{x}=\boldsymbol{x}_i$ completes the proof. □

Remark 4.

*The line integral of $\boldsymbol{u}$ along the line $\boldsymbol{l}(t)=\boldsymbol{x_i}+2t(\boldsymbol{x}-\boldsymbol{x_i})$, $t\in[0,1]$ provides an alternative definition of the radial component function*

$$u_{i\to}(\boldsymbol{x})=\int_{\boldsymbol{x_i}}^{2\boldsymbol{x}-\boldsymbol{x_i}}\boldsymbol{u}\cdot d\boldsymbol{l}=\int_0^1\boldsymbol{u}\left(\boldsymbol{l}(t)\right)\cdot 2(\boldsymbol{x}-\boldsymbol{x_i})dt. \tag{3.13}$$

*Inserting the Taylor expansion*

$$\boldsymbol{u}(\boldsymbol{l}(t))=\boldsymbol{u}(\boldsymbol{x}_i)+2t\nabla\boldsymbol{u}(\boldsymbol{x}_i)(\boldsymbol{x}-\boldsymbol{x}_i)+o(\|\boldsymbol{x}-\boldsymbol{x}_i\|)$$

*of $\boldsymbol{u}(\boldsymbol{l}(t))$ about $t=0$ into* (3.13) *and integrating the result gives*

$$u_{i\to}(\boldsymbol{x})=\big(\boldsymbol{u}(\boldsymbol{x}_i)+\nabla\boldsymbol{u}(\boldsymbol{x}_i)(\boldsymbol{x}-\boldsymbol{x}_i))\big)\cdot 2(\boldsymbol{x}-\boldsymbol{x_i})+o(\|\boldsymbol{x}-\boldsymbol{x}_i\|^2,$$

*while the expansion $\boldsymbol{u}(\boldsymbol{x})=\boldsymbol{u}(\boldsymbol{x}_i)+\nabla\boldsymbol{u}(\boldsymbol{x}_i)(\boldsymbol{x}-\boldsymbol{x}_i)+o(\|\boldsymbol{x}-\boldsymbol{x}_i\|^2)$ implies that*

$$u_{i\to}(\boldsymbol{x})=\boldsymbol{u}(\boldsymbol{x})\cdot 2(\boldsymbol{x}-\boldsymbol{x}_i)+o(\|\boldsymbol{x}-\boldsymbol{x}_i\|^2)$$

*Therefore, Lemma 3.1 continues to hold with the alternative definition* (3.13).

Lemma 3.1 provides the foundation for a GMLS approximation of a vector field and its divergence from scattered directional components. Specifically, the lemma asserts that we can define a GMLS approximation of $\boldsymbol{u}(\boldsymbol{x}_i)$ and $\nabla\cdot\boldsymbol{u}(\boldsymbol{x}_i)$ as

$$\widetilde{\boldsymbol{u}}(\boldsymbol{x}_i)=\widetilde{\delta}_{\boldsymbol{x}_i}(\boldsymbol{u}):=\frac{1}{2}\widetilde{\nabla}u_{i\to}(\boldsymbol{x}_i)\quad\text{and}\quad\widetilde{\nabla\cdot}\boldsymbol{u}(\boldsymbol{x}_i):=\frac{1}{4}\widetilde{\Delta}u(\boldsymbol{x}_i), \tag{3.14}$$

respectively, where $\widetilde{\nabla}u_{i\to}(\boldsymbol{x}_i)$ and $\widetilde{\Delta}u_{i\to}(\boldsymbol{x}_i)$ are the GMLS gradient and Laplacian of the radial component function, respectively.

Our main idea now is to specialize the GMLS approach in Section 3.1 to the radial component function in such a way that the GMLS divergence becomes a mapping $F^i\mapsto C^i$. Such a specialization enables us to use $\widetilde{\nabla\cdot}\boldsymbol{u}(\boldsymbol{x}_i)$ to define the discrete divergence operator $DIV_i$ in our staggered scheme.

In light of (3.11), i.e., $u_{i\to}(\boldsymbol{x}_i)=0$, the appropriate setting for this specialization is given by the spaces

$$V=C_i^{m+1}:=\{v\in C^{m+1}(\Omega)\,|\,v(\boldsymbol{x}_i)=0\}\quad\text{and}\quad P=P_m^i:=\{p\in P_m\,|\,p(\boldsymbol{x}_i)=0\}$$

while

$$\Lambda_i=\{\delta_{\boldsymbol{x}_{ij}}\,|\,\boldsymbol{x}_{ij}\in M_i\}\quad\text{and}\quad\tau_i(u)=\delta_{\boldsymbol{x}_i}\circ D^\alpha \tag{3.15}$$

define the set of the sampling functionals and the target functionals, respectively. As in section 3.1, this allows us to use the weight function (3.1).

We proceed to establish a local $P_m^i$-reproduction property.

Lemma 3.2. *Assume that **C.1**–**C.3** hold for $X=M_i$, i.e., the set of midpoints in the neighborhood $N_i^\varepsilon$. Let $\Lambda_i$ and $\tau_i$ be the functionals in* (3.15). *Then, there exist*

*coefficients* $a_{ij,\alpha}(\boldsymbol{x}_i) = a_{ij}(\delta_{\boldsymbol{x}_i} \circ D^\alpha)$ *such that properties* ***P.1***–***P.3*** *hold with* $P = P^i_m$. *In particular, there exist a constant* $C_{1,\alpha} > 0$ *such that* $a_{ij,\alpha}(\boldsymbol{x}_i)$ *satisfy*

$$\sum_{j=1}^{N} |a_{ij,\alpha}(\boldsymbol{x}_i)| \leq C_{1,\alpha} \tag{3.16}$$

*and* $a_{ij,\alpha}(\boldsymbol{x}_i) = 0$, *if* $\|\boldsymbol{x}_i - \boldsymbol{x}_{ij}\| > 2C(\theta)h_{X,\Omega}$, *where* $h_{X,\Omega}$ *and* $C(\theta)$ *are as in Condition* ***C.2***. *The equivalent problem* (2.16) *has a unique solution* $\boldsymbol{b}(\boldsymbol{x})$, *which defines a GMLS derivative according to* (3.7).

*Proof.* Conditions **C.1**–**C.3** ensure that the midpoints $M_i$ are $P_m$-unisolvent and so, the sampling set $\Lambda_i$ is also $P_m$-unisolvent. Since $P^i_m \subset P_m$ it follows that $\Lambda_i$ is also unisolvent for $P^i_m$. The rest of the proof follows the arguments in [22] and [21]. ∎

The next Lemma estimates the errors of the specialized GMLS approximation.

LEMMA 3.3. *Assume that the hypotheses of Lemma 3.2 and let* $\Omega^* = \overline{\mathcal{B}(\boldsymbol{x}_i, \delta)}$. *For any* $u \in C^{m+1}(\Omega^*) \cap C^{m+1}_i(\Omega)$ *there holds*

$$\left\| \widetilde{D}^\alpha u(\boldsymbol{x}_i) - D^\alpha u(\boldsymbol{x}_i) \right\| \leq C h^{m+1-|\alpha|}_{X,\Omega} \|u\|_{C^{m+1}(\Omega^*)}.$$

*Proof.* We adapt the proof of Theorem 4.3 in [21]. Adding and subtracting an arbitrary $p \in P^i_m$ and using the $P^i_m$ reproduction property yields the bound

$$\begin{aligned} \left\| D^\alpha u(\boldsymbol{x}_i) - \widetilde{D}^\alpha u(\boldsymbol{x}_i) \right\| &\leq \|D^\alpha u(\boldsymbol{x}_i) - D^\alpha p(\boldsymbol{x}_i)\| + \left\| D^\alpha p(\boldsymbol{x}_i) - \widetilde{D}^\alpha u(\boldsymbol{x}_i) \right\| \\ &= \|D^\alpha u(\boldsymbol{x}_i) - D^\alpha p(\boldsymbol{x}_i)\| + \left\| \sum_{ij} a_{ij,\alpha}(\boldsymbol{x}_i) p(\boldsymbol{x}_{ij}) - \sum_{ij} a_{ij,\alpha}(\boldsymbol{x}_i) u(\boldsymbol{x}_{ij}) \right\| \\ &\leq \|D^\alpha u(\boldsymbol{x}_i) - D^\alpha p(\boldsymbol{x}_i)\| + \|p - u\|_{L^\infty(\Omega^*)} \textstyle\sum_{ij} |a_{ij,\alpha}(\boldsymbol{x}_i)| \end{aligned} \tag{3.17}$$

Because $u(\boldsymbol{x}_i) = 0$, its Taylor polynomial $p^i_m$ of order $m$ about $\boldsymbol{x}_i$ belongs to $P^i_m$. Setting $p = p^i_m$ in (3.17) and using that

$$\left\| D^\alpha u - D^\alpha p^i_m \right\|_{L^\infty(\Omega^*)} \leq C h^{m+1-|\alpha|}_{X,\Omega} \|u\|_{C^{m+1}(\Omega^*)}. \tag{3.18}$$

completes the proof. ∎

COROLLARY 3.4. *Assume the hypotheses and notation of Lemmas 3.2–3.3. Let* $\widetilde{\boldsymbol{u}}(\boldsymbol{x}_i)$ *and* $\widetilde{\nabla}\cdot\boldsymbol{u}(\boldsymbol{x}_i)$ *be the GMLS approximations of* $\boldsymbol{u} \in (C^{m+1}(\Omega))^d$ *and its divergence defined in* (3.14). *Then,*

$$\left\| \boldsymbol{u}(\boldsymbol{x}_i) - \widetilde{\boldsymbol{u}}(\boldsymbol{x}_i) \right\| \leq C h^m_{X,\Omega} \|\boldsymbol{u}\|_{C^{m+1}(\Omega^*)}.$$

*and*

$$\left\| \nabla \cdot \boldsymbol{u}(\boldsymbol{x}_i) - \widetilde{\nabla}\cdot\boldsymbol{u}(\boldsymbol{x}_i) \right\| \leq C h^{m-1}_{X,\Omega} \|\boldsymbol{u}\|_{C^{m+1}(\Omega^*)}.$$

REMARK 5. *Although Lemma 3.2 holds for a generic point* $\boldsymbol{x}$, *the error estimate in Lemma 3.3 is valid only at* $\boldsymbol{x}_i$. *The proof of the latter requires the Taylor polynomial of* $u$ *about the point where the error is being estimated to belong in* $P^i_m$, *i.e., to vanish at* $\boldsymbol{x}_i$. *In general, if* $p_m$ *is the Taylor polynomial of* $u$ *about some other point* $\bar{\boldsymbol{x}}$ *there*

*is no guarantee that $p_m \in P_m^i$. Nonetheless, this does not present a problem, since we are only interested in estimates of the derivatives of $u$ at $\boldsymbol{x}_i$.*

In summary, given a vector field $\boldsymbol{u} \in (C^{m+1}(\Omega))^d$ and a point $\boldsymbol{x}_i$ with neighborhood $N_i^\varepsilon$, our GMLS approach for the approximation of $\boldsymbol{u}$ and its divergence at $\boldsymbol{x}_i$ comprises the following two steps:

1. Solve the weighted least-squares problem

$$p^*(\boldsymbol{x}_i) = \underset{p \in P_m^i}{\operatorname{argmin}} \Big\{ \sum_{j=1}^{N} \left[\boldsymbol{u}(\boldsymbol{x}_{ij}) \cdot 2\boldsymbol{m}_{ij} - p(\boldsymbol{x}_{ij})\right]^2 \Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_{ij}\|) \Big\}. \tag{3.19}$$

2. Let $\boldsymbol{b}(\boldsymbol{x}_i)$ denote the coefficient vector of $p^*$ relative to a basis $\{p_k^i\}$ of $P_m^i$. Set

$$\begin{aligned} \widetilde{\boldsymbol{u}}(\boldsymbol{x}_i) &= \frac{1}{2}\widetilde{\nabla} p^*(\boldsymbol{x}_i) := \frac{1}{2}\sum_{k=1}^{Q} b_k(\boldsymbol{x}_i)\nabla p_k^i(\boldsymbol{x}_i), \\ \widetilde{\nabla\cdot}\boldsymbol{u}(\boldsymbol{x}_i) &= \frac{1}{4}\widetilde{\Delta} p^*(\boldsymbol{x}_i) := \frac{1}{4}\sum_{k=1}^{Q} b_k(\boldsymbol{x}_i)\Delta p_k^i(\boldsymbol{x}_i). \end{aligned} \tag{3.20}$$

**4. Formulation of the staggered GMLS discretization method.** The duality of the local geometric structures and the ensuing isomorphism of $E^i$ and $V^i$ allow us to discretize the second-order operator in (1.1) by "chaining" the discrete divergence and gradient operators at every point of $\Omega_h$. This observation is at the heart of our staggered GMLS method. For clarity we present the method for $\mu = I$ and Dirichlet boundary conditions, i.e., $\Gamma_D = \Gamma$. Section 6 briefly discusses implementation of the method for a general $\mu$ and Neumann boundary conditions.

Let $\Omega_h$ be a meshless discretization of the computational domain $\Omega$ such that the points in $\Omega_h$ satisfy conditions **C.1**–**C.2**. The global meshless approximation space

$$V^h = \bigcup_{i=1}^{N} V^i$$

is the union of the local vertex spaces associated with the points in $\Omega_h$. The gradient of a discrete function $\phi^h \in V^h$ is simply the application of $GRAD_i$ to the restriction of $\phi^h$ to $V^i$.

As in many other discretization methods we choose to impose the Dirichlet boundary condition on the approximating space by setting $\phi(\boldsymbol{x}_i) = u(\boldsymbol{x}_i)$ for all boundary particles. Thus, our new staggered GMLS discretization of (1.1) reads: seek $\phi^h \in V^h$ such that

$$\begin{aligned} DIV_i \circ GRAD_i(\phi^h) &= f(\boldsymbol{x}_i) \quad \forall i \in \breve{\Omega}_h \\ \phi_i &= u(\boldsymbol{x}_i) \quad \forall i \in \Gamma_h. \end{aligned} \tag{4.1}$$

To assemble the discrete problem (4.1) at every interior point $\boldsymbol{x}_i$ we first compute the topological gradient of the local discrete field and then apply the GMLS divergence operator. The second step follows the procedure outlined in the gray box in Section

3.2, i.e., we solve the weighted least-squares problem

$$p^*(\boldsymbol{x}_i) = \underset{p\in P_m^i}{\operatorname{argmin}} \Big\{ \sum_{j=1}^{N} \big[GRAD_i(\phi^h) - p(\boldsymbol{x}_{ij})\big]^2 \Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_{ij}\|) \Big\} \tag{4.2}$$

where $\Phi(\cdot)$ satisfies condition **C.3** and then set

$$DIV_i \circ GRAD_i(\phi^h) = \frac{1}{4}\widetilde{\Delta} p^*(\boldsymbol{x}_i).$$

In the next section we show that the GMLS gradient of $p^*$ is an accurate approximation of the flux at the virtual cell center, i.e.,

$$\boldsymbol{u}(\boldsymbol{x}_i) = -\nabla\phi(\boldsymbol{x}_i) \approx \frac{1}{2}\widetilde{\nabla} p^*(\boldsymbol{x}_i).$$

This result is useful in settings that also require the flux values.

Remark 6. *The scheme* (4.1) *is equivalent*[3] *to the following "mixed" system of equations: seek* $\boldsymbol{u}^h \in F^h$ *and* $\phi^h \in V^h$ *such that* $\phi_i = u(\boldsymbol{x}_i)$ *for all* $i \in \Gamma_h$ *and*

$$\left\{ \begin{array}{rcl} DIV_i(\boldsymbol{u}^h) & = & f(\boldsymbol{x}_i) \\ \boldsymbol{u}^h + GRAD_i(\phi^h) & = & 0 \end{array} \right. \qquad \forall i \in \breve{\Omega}_h, \tag{4.3}$$

*where* $F^h = \cup_{i=1}^N F^i$. *Note that all metric attributes of the staggered GMLS scheme are subsumed in the first equation, whereas the second equation is purely topological, i.e., it depends only on the local connectivity graph of* $N_i^\varepsilon$ *and not on the physical locations of its points. This structure mirrors the structure of many compatible discretization methods for* (1.1)*, which also combine a metric-dependent equation with a topological relation [19]. A more rigorous qualification of the compatibility properties of the new scheme is beyond the scope of this work and will be pursued in a forthcoming article. However, it is worth pointing out that numerical results in Section 7 confirm that solutions of* (4.1) *for problems with discontinuous material properties are indeed qualitatively very similar to solutions of compatible discretization methods for* (1.1).

**5. Accuracy of the staggered GMLS method.** In this section we study the polynomial consistency of the staggered GMLS method (4.1).

Theorem 5.1. *Assume that conditions* ***C.1***–***C.3*** *hold. Let* $\phi^h \in V^h$ *be the interpolant of a function* $\phi \in C^{m+1}(\Omega)$ *and let* $p^*(\boldsymbol{x}_i)$ *be the corresponding solution of* (4.2)*. Then, the following error bounds hold:*

$$\|\Delta\phi(\boldsymbol{x}_i) - DIV_i \circ GRAD_i(\phi^h)\| \le C h_{X,\Omega}^{m-1} \|\phi\|_{C^{m+1}(\Omega)} \tag{5.1}$$

$$\|\nabla\phi(\boldsymbol{x}_i) - \frac{1}{2}\widetilde{\nabla} p_i^*(\boldsymbol{x}_i)\| \le C h_{X,\Omega}^{m} \|\phi\|_{C^{m+1}(\Omega)}. \tag{5.2}$$

*Proof.* Consider the auxiliary vector field

$$\boldsymbol{u}(\boldsymbol{x}) = g(\boldsymbol{x}) \frac{\boldsymbol{x} - \boldsymbol{x}_i}{2\|\boldsymbol{x} - \boldsymbol{x}_i\|^2}, \quad \text{where} \quad g(\boldsymbol{x}) = \phi(2\boldsymbol{x} - \boldsymbol{x}_i) - \phi(\boldsymbol{x}_i).$$

[3]The two forms of the staggered GMLS are equivalent in the sense that both produce the exact same discrete approximation $\phi^h \in V^h$.

The radial component of $\boldsymbol{u}(\boldsymbol{x})$ equals[4] $g(\boldsymbol{x})$, i.e.,

$$u_{i\rightarrow}(\boldsymbol{x}) = \boldsymbol{u}(\boldsymbol{x}) \cdot 2(\boldsymbol{x} - \boldsymbol{x}_i) = g(\boldsymbol{x}),$$

Furthermore, $g(\boldsymbol{x}_i) = 0$, that is $g(\boldsymbol{x}) \in C_i^{m+1}$, and

$$g(\boldsymbol{x}_{ij}) = \phi(\boldsymbol{x}_j) - \phi(\boldsymbol{x}_i) = GRAD_i(\phi^h).$$

It follows that the weighted least-squares problem (4.2) in the formulation of our method is equivalent to the GMLS problem (3.19) for the auxiliary velocity field $\boldsymbol{u}(\boldsymbol{x})$. Accordingly, the identities in (3.20) specialize to

$$\widetilde{\boldsymbol{u}}(\boldsymbol{x}_i) = \frac{1}{2}\widetilde{\nabla} p^*(\boldsymbol{x}_i) \quad \text{and} \quad \widetilde{\nabla}\cdot\boldsymbol{u}(\boldsymbol{x}_i) = \frac{1}{4}\widetilde{\Delta} p^*(\boldsymbol{x}_i) = DIV_i \circ GRAD_i(\phi^h),$$

respectively. Using the identities

$$\nabla g(\boldsymbol{x}) = 2\nabla\phi(\boldsymbol{x}) \quad \text{and} \quad \Delta g(\boldsymbol{x}) = 4\Delta\phi(\boldsymbol{x}),$$

and the fact that $g(\boldsymbol{x})$ is the radial component of $\boldsymbol{u}(\boldsymbol{x})$ implies that

$$\boldsymbol{u}(\boldsymbol{x}_i) = \frac{1}{2}\nabla g(\boldsymbol{x}_i) = \nabla\phi(\boldsymbol{x}_i) \quad \text{and} \quad \nabla\cdot\boldsymbol{u}(\boldsymbol{x}_i) = \frac{1}{4}\Delta g(\boldsymbol{x}_i) = \Delta\phi(\boldsymbol{x}_i),$$

respectively. Therefore,

$$\nabla\phi(\boldsymbol{x}_i) - \frac{1}{2}\widetilde{\nabla} p_i^*(\boldsymbol{x}_i) = \boldsymbol{u}(\boldsymbol{x}_i) - \widetilde{\boldsymbol{u}}(\boldsymbol{x}_i),$$

and

$$\Delta\phi(\boldsymbol{x}_i) - DIV_i \circ GRAD_i(\phi^h) = \nabla\cdot\boldsymbol{u}(\boldsymbol{x}_i) - \widetilde{\nabla}\cdot\boldsymbol{u}(\boldsymbol{x}_i),$$

respectively. Application of the error bounds in Corollary 3.4 grants the proof. ∎

**6. Extensions and implementation details.** This section briefly reviews the treatment of a variable material property tensor $\mu$ as well as the imposition of the Neumann boundary condition in the staggered GMLS. We also comment on some practical aspects of the implementation used in our numerical studies, including choice of polynomial basis and solution of the local optimization problems.

Extension of (4.1) to a variable $\mu$ is straightforward. Assuming that $\mu$ is available at all points in $\Omega_h$ we solve a modified version of (4.2) given by

$$p^*(\boldsymbol{x}_i) = \underset{p\in P_m^i}{\operatorname{argmin}} \Big\{ \sum_{j=1}^{N} \big[\mu_{ij} GRAD_i(\phi^h) - p(\boldsymbol{x}_{ij})\big]^2 \Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_{ij}\|) \Big\} \tag{6.1}$$

where $GRAD_i$ is the topological gradient defined in (2.2) and $\mu_{ij} = (\mu(\boldsymbol{x}_i)+\mu(\boldsymbol{x}_j))/2$. Then we approximate the second-order operator in (1.1) according to

$$\nabla\cdot\mu\nabla\phi(\boldsymbol{x}_i) \approx DIV_i \circ \mu GRAD_i(\phi^h) := \frac{1}{4}\widetilde{\Delta} p^*(\boldsymbol{x}_i).$$

Implementation of the Neumann condition $\partial_n\phi = g$ in the staggered GMLS scheme is somewhat more involved. Recall that in the case of Dirichlet conditions

[4]The alternative definition (3.13) with $\boldsymbol{u} = \nabla\phi$ yields the same result, i.e. $u_{i\rightarrow}(\boldsymbol{x}) = g(\boldsymbol{x})$.

we approximate the second-order operator $\nabla \cdot \mu \nabla \phi(\boldsymbol{x}_i)$ only at the interior points and impose the boundary condition simply by setting $\phi_i = u(\boldsymbol{x}_i)$ at all $\boldsymbol{x}_i \in \Gamma_D$. To impose the Neumann condition, in addition to interior points, we also approximate $\nabla \cdot \mu \nabla \phi(\boldsymbol{x}_i)$ at all Neumann points $\boldsymbol{x}_i \in \Gamma_N$ and incorporate the boundary condition in the definition of the discrete second-order operator. The latter is accomplished by constraining (6.1) with a GMLS approximation of the equation $\partial_n \phi = g$. Taking into account that $\nabla \phi(\boldsymbol{x}_i) \approx 1/2 \widetilde{\nabla} p^*(\boldsymbol{x}_i)$, this approximation is given by

$$\widetilde{\partial_n} p(\boldsymbol{x}_i) = 2g(\boldsymbol{x}_i)$$

and results in the following constrained version of (6.1):

$$p^*_{\partial n}(\boldsymbol{x}_i) = \underset{p \in P^i_m; \widetilde{\partial_n} p(\boldsymbol{x}_i) = 2g(\boldsymbol{x}_i)}{\operatorname{argmin}} \left\{ \sum_{j=1}^{N} \left[ \mu_{ij} GRAD_i(\phi^h) - p(\boldsymbol{x}_{ij}) \right]^2 \Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_{ij}\|) \right\}. \tag{6.2}$$

Solving (6.2) instead of (6.1) effectively replaces the operator $DIV_i$ with a modified version $DIV_{\partial n,i}$ which naturally incorporates the Neumann boundary condition. We then impose this condition by approximating the second-order operator at all points on the Neumann boundary according to

$$\nabla \cdot \mu \nabla \phi(\boldsymbol{x}_i) \approx DIV_{\partial n,i} \circ \mu GRAD_i(\phi^h) := \frac{1}{4} \widetilde{\Delta} p^*_{\partial n}(\boldsymbol{x}_i).$$

where $p^*_{\partial n}$ solves (6.2).

**6.1. Practical aspects of the implementation.** In the general case of a variable $\mu$ and mixed boundary conditions the staggered GMLS scheme comprises the following system of algebraic equations:

$$\begin{aligned} DIV_i \circ \mu GRAD_i(\phi^h) &= f_i \quad \forall \boldsymbol{x}_i \in \breve{\Omega}_h \\ DIV_{\partial n,i} \circ \mu GRAD_i(\phi^h) &= f_i \quad \forall \boldsymbol{x}_i \in \Gamma_N \\ \phi_i &= u_i \quad \forall \boldsymbol{x}_i \in \Gamma_D \end{aligned} . \tag{6.3}$$

Assembly of this system requires solution of (6.1) for the discretization at all interior points and solution of the constrained problem (6.2) to compute the second-order discrete operator at all Neumann points. The actual form of these problems depends on the choice of a basis for the polynomial reconstruction. In our implementation we choose the scaled and shifted Taylor monomials, i.e.,

$$P^i_m = span \left\{ p^i_\alpha \right\}_{|\alpha| \le m}, \quad p^i_\alpha = \frac{1}{\alpha!} \left( \frac{\boldsymbol{x} - \boldsymbol{x}_i}{\epsilon} \right)^\alpha . \tag{6.4}$$

In the present context the matrices $P$, $W$ and the vector $\boldsymbol{\ell}$ in Section 2.3 specialize to

$$W_i = \operatorname{diag}(\Phi(||\boldsymbol{x}_{ij} - \boldsymbol{x}_i||))_{\boldsymbol{x}_{ij} \in M_i}, \; P_i = \left( p^i_\alpha(\boldsymbol{x}_{ij}) \right)^{|\alpha| \le m}_{\boldsymbol{x}_{ij} \in M_i}, \; \boldsymbol{\ell}_i = \left( \mu_{ij}(\phi_j - \phi_i) \right)_{\boldsymbol{x}_{ij} \in M_i} .$$

The optimal coefficient vector $\boldsymbol{b}_i$ for the solution of (6.1) is given by (2.14), which specializes to

$$\boldsymbol{b}^T_i = \boldsymbol{\ell}^T_i \, W \, P^T_i \left( P_i W_i P^T_i \right)^{-1} . \tag{6.5}$$

As a result, the approximation of the second-order operator at all interior points assumes the form

$$DIV_i \circ \mu GRAD_i(\phi^h) = \frac{1}{4}\boldsymbol{b}_i^T \boldsymbol{r}(\delta_{\boldsymbol{x}_i} \circ \Delta)$$

where $\boldsymbol{r}(\delta_{\boldsymbol{x}_i} \circ \Delta) := \big(\Delta p_\alpha^i(\boldsymbol{x}_i)\big)$.

To solve the constrained problem (6.2) we introduce a Lagrange multiplier $\lambda$ and seek the stationary point $\{p_{\partial n}^*, \lambda^*\}$ of the Lagrangian functional

$$L(p,\lambda) = \sum_{j=1}^{N} \left[\mu_{ij} GRAD_i(\phi^h) - p(\boldsymbol{x}_{ij})\right]^2 \Phi(\|\boldsymbol{x}_i - \boldsymbol{x}_{ij}\|) - \lambda \left(\widetilde{\partial_n} p(\boldsymbol{x}_i) - 2g(\boldsymbol{x}_i\right).$$

Let $\boldsymbol{c}_i^* = (\boldsymbol{b}_{\partial n,i}, \lambda)$ be the coefficient vector of $\{p_{\partial n}^*, \lambda^*\}$ and $\boldsymbol{r}(\delta_{\boldsymbol{x}_i} \circ \partial_n) = \big(\partial_n p_\alpha^i(\boldsymbol{x}_i)\big)$. It is not hard to see that the necessary optimality condition for the stationary point is equivalent to a linear algebraic system $K_i \boldsymbol{c}_i = \boldsymbol{d}_i$ for the optimal coefficient vector $\boldsymbol{c}_i^*$, where

$$K_i = \begin{bmatrix} P_i W_i P_i^T & \boldsymbol{r}(\delta_{\boldsymbol{x}_i} \circ \partial_n) \\ \boldsymbol{r}(\delta_{\boldsymbol{x}_i} \circ \partial_n)^T & 0 \end{bmatrix} \quad \text{and} \quad \boldsymbol{d}_i = \begin{bmatrix} P_i W_i \boldsymbol{\ell}_i \\ 2g_i \end{bmatrix}. \tag{6.6}$$

After solving (6.6) and extracting $\boldsymbol{b}_{\partial n,i}$ we obtain the approximation

$$DIV_{\partial n,i} \circ \mu GRAD_i(\phi^h) = \frac{1}{4}\boldsymbol{b}_{\partial n,i}^T \boldsymbol{r}(\delta_{\boldsymbol{x}_i} \circ \Delta).$$

of the differential operator at all Neumann points.

Computation of the optimal coefficient vectors for both (6.1) and (6.2) requires inversion of the matrix $M_i = P_i W_i P_i^T$, which resembles the poorly conditioned Hilbert matrix. Because in this work we used polynomial degrees $m \leq 6$, the relatively small size of the resulting local problems enabled us to stably invert $M_i$ by using a standard LU decomposition. Larger polynomial degrees may require more advanced solvers and/or a different choice of basis polynomials.

Due to the strong form discretization of the PDE and the lack of symmetry in particle arrangement, the global matrix corresponding to (6.3) is in general not symmetric. Furthermore, a special care must be taken to ensure solvability for problems with a non-trivial null-space as in the case of pure Neumann boundary conditions. Possible options for handling the nullspace include pinning a degree of freedom, adding a global Lagrange multiplier to remove the singularity, or projecting out the kernel if a Krylov subspace method is used to solve the resulting equations. For more information about this subject we refer to [17, 25, 26] and the references cited therein.

Remark 7. *Consider an interior point $\boldsymbol{x}_i \in \breve{\Omega}$. By collecting the terms multiplying the elements of $\boldsymbol{\ell}_i = (\mu_{ij}(\phi_j - \phi_i))$ we can write the discrete operator as*

$$DIV_i \circ \mu GRAD_i(\phi^h) = \sum_{j \in N_i^\varepsilon \cap \breve{\Omega}} \beta_{ij}(\phi_j - \phi_i) + \sum_{j \in N_i^\varepsilon \cap \Gamma_D} \beta_{ij}(u_j - \phi_i)$$

*for a suitable set of coefficients $\beta_{ij}$. Likewise, at all Neumann points the discrete operator can be written as*

$$DIV_{\partial n,i} \circ \mu GRAD_i(\phi^h) = \sum_{j \in N_i^\varepsilon} \beta_{ij}(\phi_j - \phi_i) + \gamma_i g_i$$

*for some suitable coefficient $\gamma_i$. Consequently, the system* (6.3) *has the following equivalent form:*

$$
\begin{aligned}
\sum_{j\in N_i^\varepsilon\cap\breve{\Omega}} \beta_{ij}(\phi_j-\phi_i) - \sum_{j\in N_i^\varepsilon\cap\Gamma_D} \beta_{ij}\phi_i &= f_i - \sum_{j\in N_i^\varepsilon\cap\Gamma_D} \beta_{ij}u_j && \forall \boldsymbol{x}_i\in\breve{\Omega}_h \\
\sum_{j\in N_i^\varepsilon} \beta_{ij}(\phi_j-\phi_i) &= f_i-\gamma_i g_i && \forall \boldsymbol{x}_i\in\Gamma_N \\
\phi_i &= u_i && \forall \boldsymbol{x}_i\in\Gamma_D
\end{aligned}
\tag{6.7}
$$

*Thus, the staggered GMLS method can be interpreted as a finite-difference-like formula for the Laplacian defined on an irregular stencil comprising the points in $N_i^\varepsilon$.*

## 7. Numerical examples.

**7.1. Case setup.** To discretize the domain for all problems in this work, we first discretize the boundary of the domain with a length scale $dx$ and then discretize the interior of the domain with a Cartesian lattice with spacing $dx$. We then perturb each interior particle by a uniformly distributed random variable of magnitude $\eta$, and delete all particles that lie outside of the domain. The variable $\eta$ is used to demonstrate insensitivity of the results to particle anisotropy, and for all results presented is set to $\eta = 0.1dx$. We characterize levels of discretization by the number of particles in each direction of the lattice $N = \frac{1}{dx}$.

Algebraic multigrid is used to efficiently solve the discretized asymmetric system of equations. While an in-depth discussion of the performance of this preconditioner is left for a future work, we briefly comment that a combination of GMRes and auxilliary space preconditioning [27] provides an efficient means of solution with computational effort scaling nearly linearly with the degrees of freedom in the system.

**7.2. Elliptic problems with smooth coefficients.** We first demonstrate the convergence rate of the method by using smooth manufactured solutions and considering both an annular 2D domain with outer radius $R_o = \frac{\pi}{2}$ and inner radius $R_i = \frac{\pi}{4}$, and a 3D domain formed by extruding the 2D geometry in the z-direction to form a cylinder with height $\pi$; see Figure 2. We set the exact solutions to

$$\phi_{ex}^{2D} = \sin x \sin y \quad \text{and} \quad \phi_{ex}^{3D} = \sin x \sin y \sin z$$

in two and three-dimensions, respectively. Substitution of the exact solutions in the model problem (1.1) defines the source term and the boundary data. We then solve the discrete problem (6.3) with either Dirichlet or Neumann boundary conditions. As no mesh is available to quantify the error, we measure convergence in a root mean square sense by defining the following norm

$$\|\phi^h\|_{l_2} = \left(\frac{1}{N}\sum_{i\in\Omega_h}\phi_i^2\right)^{1/2} \tag{7.1}$$

and semi-norm

$$|\phi^h|_{h^1} = \left(\frac{1}{N}\sum_{i\in\Omega_h}(GRAD_i\phi^h)^2\right)^{1/2} \tag{7.2}$$

We first compare the convergence rates of the staggered GMLS scheme with those of a *collocated* one obtained by using the standard GMLS derivative approximation of

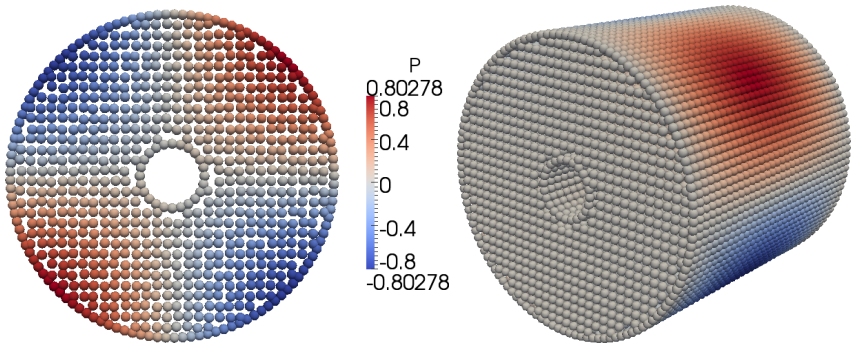


FIG. 2. *Typical particle distribution with $N = 32$ for 2D annular disk geometry (left) and 3D extrusion (right).*

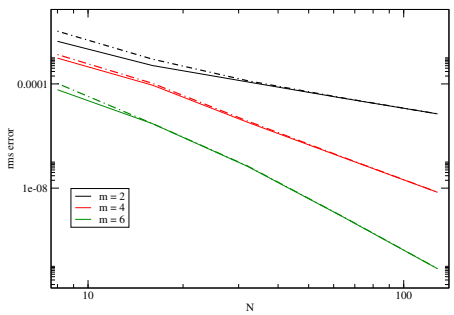


FIG. 3. *Convergence in discrete $l_2$-norm for the smooth manufactured solution in two-dimensions with Dirichlet boundary conditions using second, fourth and sixth order for collocated GMLS (dashed lines) and staggered GMLS (solid lines).*

$\nabla \cdot \mu \nabla \phi$ [21]. In both cases we use the same Taylor basis (6.4) and kernel functions. Figure 3 demonstrates the convergence in the $l_2$-norm of the two approaches for increasing refinement and increasing polynomial order. Both schemes provide $m^{th}$-order convergence, with the staggered scheme being slightly more accurate.

Next, we examine the convergence rates of the staggered GMSL for both Dirichlet and Neumann boundary conditions. The data in Table 1 and the plots in Figure 4 reveal equal convergence rates in both the $\ell_2$ norm and the $h^1$ seminorm, i.e., the scalar variable and its flux in the staggered GMLS converge at identical rates. This kind of behavior is typical of mixed methods for (1.1) and lends further credence to the observations in Remark 6 about the similarities between our scheme and compatible discretization methods. Table 2 confirms that similar results hold in three dimensions.

**7.3. Elliptic problems with rough coefficients.** A hallmark of compatible discretization methods for (1.1), such as mixed finite elements, is their ability to correctly represent the flux variable for problems with discontinuous coefficients $\mu$. In such cases the normal component of the flux remains continuous across the material interface, while its tangential component is allowed to develop a discontinuity. Collocated methods, including Galerkin, stabilized Galerkin and least-squares finite elements cannot capture this behavior and tend to develop oscillations across the interface; see [28] and [29] for further details and examples.

TABLE 1
*Convergence rate for the smooth manufactured solution problem in two-dimensions.*

| | **m=2** | **m=4** | **m=6** |
|---|---|---|---|
| $\ell_2$ - Dirichlet | 2.085 | 4.470 | 6.486 |
| $\ell_2$ - Neumann | 2.169 | 4.187 | 6.021 |
| $h^1$ - Dirichlet | 1.979 | 3.940 | 5.839 |
| $h^1$ - Neumann | 2.350 | 4.293 | 6.272 |

TABLE 2
*Convergence rate for the smooth manufactured solution problem in three-dimensions and Dirichlet boundary conditions.*

| | **m=2** | **m=4** | **m=6** |
|---|---|---|---|
| $\ell_2$ | 2.001 | 4.548 | 6.339 |
| $h^1$ | 2.008 | 3.813 | 5.742 |

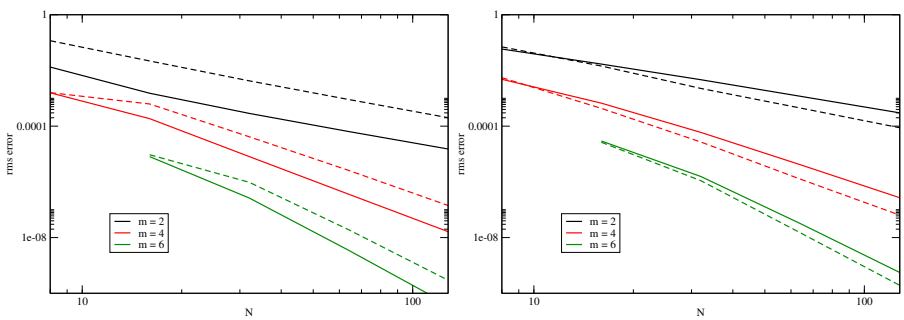


FIG. 4. *Convergence in discrete $l_2$-norm (left) and $h_1$-seminorm (right) for manufactured solution with Dirichlet (solid lines) and Neumann (dashed lines) boundary conditions, suggesting a convergence result of $||u - u_{ex}||_{l_2} \leq Ch^m$ and $||u - u_{ex}||_{h^1} \leq Ch^m$ for the staggered scheme.*

Remark 6 points out structural similarities between the staggered GMLS and other compatible discretizations. The examples in this section aim to demonstrate that these similarities are not just formal and that the new scheme is in fact capable of delivering physically correct approximations of the flux for problems with discontinuous coefficients.

Our first example involves the so called five strip problem [28], which is a standard manufactured solution test case for examining the ability of a scheme to maintain normal flux continuity. In this problem $\phi_{ex} = 1 - x$, $\Gamma_N = \Gamma$, the computational domain $\Omega = [0, 1]^2$ is divided into five equal strips

$$\Omega_i = \{(x, y) \mid 0.2(i-1) \leq y \leq 0.2i; 0 \leq x \leq 1\}, \quad i = 1, \ldots, 5 \tag{7.3}$$

and $\mu$ is assigned a different constant value $\mu_i$ on each $\Omega_i$. Here we use $\mu_1 = 16$, $\mu_2 = 6$, $\mu_3 = 1$, $\mu_4 = 10$, and $\mu_5 = 2$. Substitution of $\phi_{ex}$ into (1.1) defines the source term and the Neumann data $g$. In particular, since $\phi_{ex}$ is globally linear, $f = 0$, $g = \pm\mu_i$ on the vertical parts of $\Gamma$, and

$$\boldsymbol{u}\big|_{\Omega_i} = -\mu_i \nabla \phi_{ex} = \begin{pmatrix} \mu_i \\ 0 \end{pmatrix}.$$

Thus, the normal flux is continuous across the interface between two strips while the tangential component of $\boldsymbol{u}$ is a piecewise constant equal to $\mu_i$ on strop $\Omega_i$.

The plots in Figure 5 demonstrate that the staggered GMLS solution provides an accurate approximation of the tangential flux. Furthermore, comparison with the standard, collocated GMLS solution in Figure 6 reveals that the latter exhibits the same type of oscillations across the interfaces as found in nodal mesh-based methods [28]. It is worth pointing out that the staggered GMLS results in Figures 5–6 were obtained without requiring that particles conform to the material interfaces between the strips. In contrast, mesh-based compatible methods typically require interface-fitted grids for accuracy.

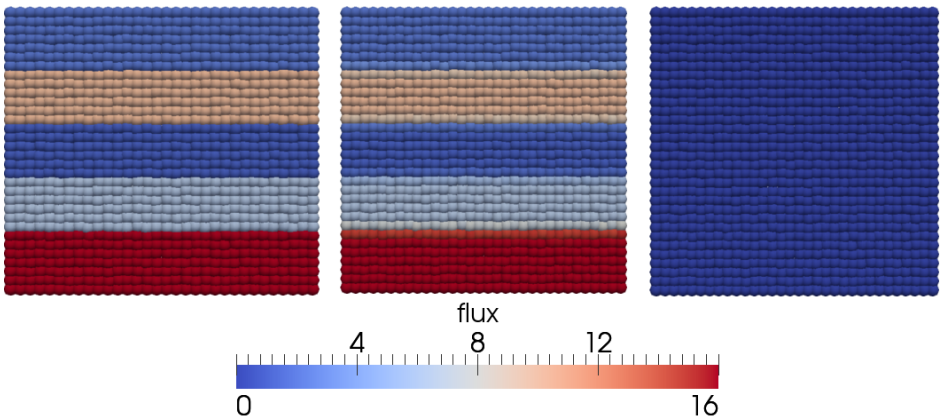


FIG. 5. *Fluxes for the Darcy flow strips case for non-conforming particle arrangement: exact longitudinal flux (left), staggered longitudinal flux (center), and staggered transverse flux (right).*

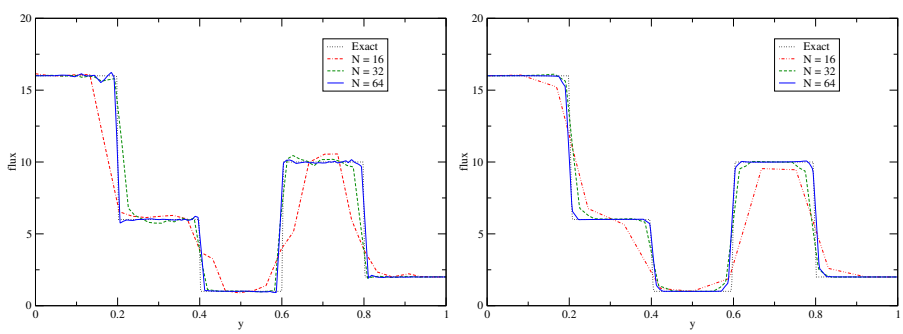


FIG. 6. *Transverse fluxes along the line $x = 0.5$ for the collocated (left) and staggered (right) schemes with increasing resolution.*

The model problem (1.1) can also be used to study electrostatics problems in the absence of charge by taking $f = 0$ and identifying $\mu$ as the electric permitivity (usually denoted $\epsilon$). In multiphysics applications, electromagnetic effects are often solved concurrently with a model for the mechanics governing domain deformation (e.g. electrophoresis of colloidal suspensions). When non-trivial geometry is considered for these applications, typically the only available tools to handle deforming domains with discontinuous material properties involve either costly arbitrary Lagrangian-Eulerian methods or some diffuse interfacial treatment [30]. The current approach, on the other hand, allows a natural treatment of jumps in material properties without the need to develop a mesh conforming to discontinuities.

Particles can be identified as belonging to either material with $\mu$ assigned appropriately. As an example, we simulate a circular cylinder of radius $R = 1/2$ in a medium with permitivity $\mu/\mu_\infty = 2$ exposed to a uniform electric field $\nabla\phi = \langle 1, 0\rangle$ for which an analytic solution can easily be calculated in an infinite domain. To solve numerically, we impose the analytic solution as a Dirichlet boundary condition on a unit square and demonstrate convergence in Figures 7 and 8. This discretization is stable for a ratio of permitivities ranging from $1 - 1000$ (Figure 8).

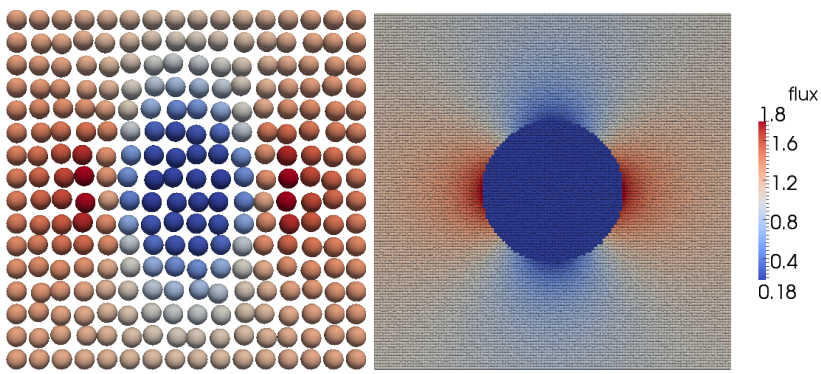


FIG. 7. *Plot of electric flux $\nabla\phi$ for coarsest ($N = 16$) and finest ($N = 256$) resolution.*

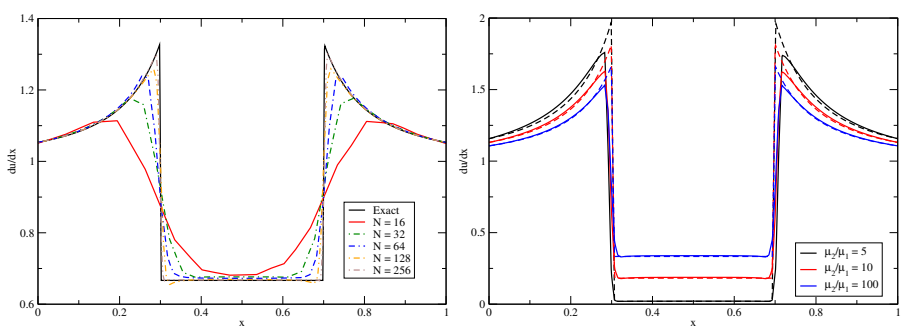


FIG. 8. *Electric flux along center line for increasing resolution (left) and for a range of permitivities (right).*

As an example of the flexibility of the method for complex geometries, we now solve (1.1) where $\mu$ is taken by processing an image of one of the authors' cat (Figure 9) and assigning permitivities to regions of similar color. Particles are assigned values for $\mu_i$ by sampling pixels from the picture, and an approximately uniform electric field is imposed by setting Dirichlet boundary conditions.

$$\phi_i|_{\partial\Omega} = 1 - x_i \tag{7.4}$$

The robustness of the method in the presence of fluctuations in the permitivity below the resolution lengthscale is shown in the refinement study presented in Figure 10. Examining the y-component of the resulting "electric field" for resolution ranging from $32^2$ to $1024^2$, the results remain consistent and non-oscillatory. We note that, using AMG, the resulting matrix solve took only a few seconds on a desktop computer. While this is a slightly whimsical demonstration of the flexibility of the method, we

hope to motivate that this approach could easily be adopted to any problem in which voxel data is available of the underlying diffusivity (e.g. applications in MRI imaging, geological imaging, etc).

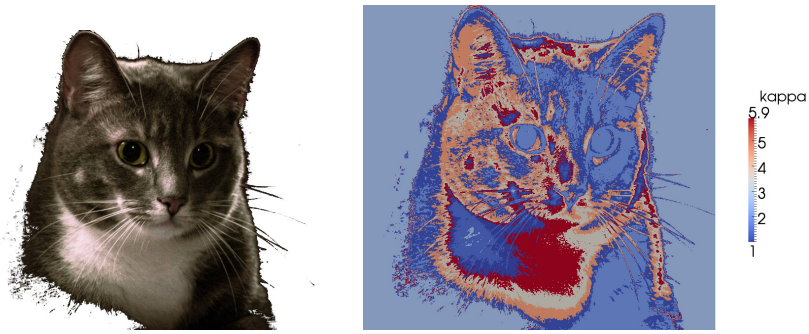


FIG. 9. *Conductivity for the underresolved stability test case.*

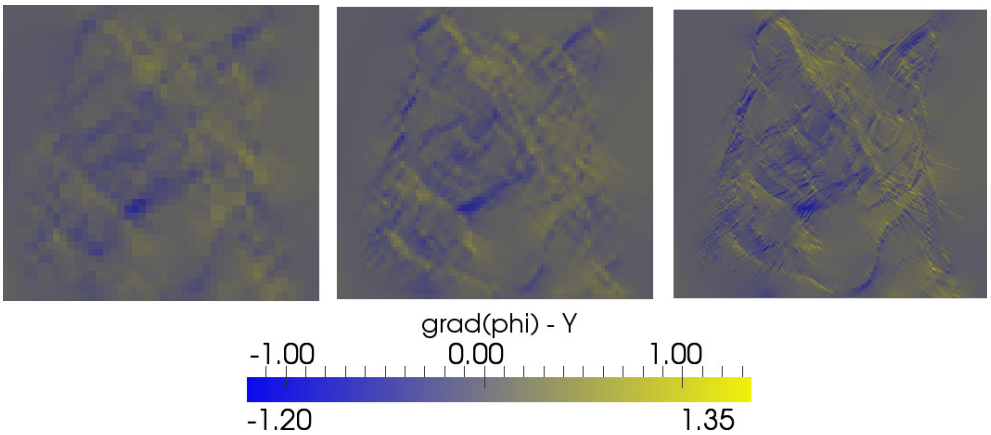


FIG. 10. *$\partial_y \phi$ for underresolved stability test case. Underresolved cases provided similar results despite highly oscillatory conductivity below resolution lengthscale.*

**8. Conclusions and future work.** Motivated by mimetic methods on primal-dual grids we have developed a meshfree discretization for a model diffusion equation, which exhibits many of the attractive computational properties of compatible discretization. Our scheme generalizes staggered finite difference methods to arbitrary point clouds and particle arrangements by using separate discretizations of the divergence and gradient operators defined locally at the neighborhood of each discretization point. While traditional staggered methods are of either finite difference type and are restricted to Cartesian grids, or are of finite volume type and require a primal-dual grid complex, this approach is truly meshfree since it uses only the $\epsilon$-neighborhood graph of particle connectivity.

Due to the lack of a mesh, there is no obvious inner-product space in which to analyze this method and we have opted instead to demonstrate the compatible nature of the scheme by comparison to a series of problems in which discretizations lacking compatibility are prone to failure.

For the sake of brevity, we have restricted the focus of this work to the model problem (1.1). We note here that the stability and high-order accuracy achieved for this problem carries over to the Stokes problem, which we will discuss in a later work.

**9. Acknowledgments.** *This material is based upon work supported by the U.S. Department of Energy Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program as part of the Colloboratory on Mathematics for Mesoscopic Modeling of Materials (CM4), under Award Number DE-SC0009247. This research used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy under contract DE-AC04-94-AL85000.*

REFERENCES

[1] I. Yotov, "Mixed finite element methods for flow in porous media," Tech. Rep. TR96-09, Dept. Comp. Appl. Math., Rice University, 1996.

[2] C. R. Swaminathan, V. R. Voller, and S. V. Patankar, "A streamline upwind control volume finite element method for modeling fluid flow and heat transfer problems," *Finite Elements in Analysis and Design*, vol. 13, no. 2-3, pp. 169 – 184, 1993.

[3] F. Brezzi, L. D. Marini, and P. Pietra, "Numerical simulation of semiconductor devices," *Computer Methods in Applied Mechanics and Engineering*, vol. 75, no. 1-3, pp. 493 – 514, 1989.

[4] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods.* Springer, Berlin, 1991.

[5] L. BEIRÃO DA VEIGA, F. BREZZI, A. CANGIANI, G. MANZINI, L. D. MARINI, and A. RUSSO, "Basic principles of virtual element methods," *Mathematical Models and Methods in Applied Sciences*, vol. 23, no. 01, pp. 199–214, 2013.

[6] K. Lipnikov, M. Shashkov, and D. Svyatskiy, "The mimetic finite difference discretization of diffusion problem on unstructured polyhedral meshes," *J. Comput. Phys.*, vol. 211, pp. 473–491, January 2006.

[7] M. Shashkov, *Conservative Finite Difference Methods on General Grids.* Boca Raton, FL: CRC Press, December 05 1995.

[8] K. Kramer and W. N. G. Hitchon, *Semiconductor Devices: A Simulation Approach.* Upper Saddle River, NJ 07458: Prentice Hall, 1997.

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

[10] R. A. Nicolaides and X. Wu, "Covolume solutions of three-dimensional div-curl equations," *SIAM Journal on Numerical Analysis*, vol. 34, no. 6, pp. 2195–2203, 1997.

[11] I. D. Mishev and Q.-Y. Chen, "A mixed finite volume method for elliptic problems," *Numerical Methods for Partial Differential Equations*, vol. 23, no. 5, pp. 1122–1138, 2007.

[12] O. V. Diyankov, "Uncertain grid method for numerical solution of pde," tech. rep.

[13] A. Katz and A. Jameson, "A meshless volume scheme," *AIAA Paper*, vol. 3534, pp. 1–24, 2009.

[14] E. K.-y. Chiu, Q. Wang, and A. Jameson, "A conservative meshless scheme: general order formulation and application to euler equations," in *AIAA 2011-651 49th Aerospace Sciences Meeting*, 2011.

[15] J. Monaghan, "Smoothed particle hydrodynamics and its diverse applications," *Annual Review of Fluid Mechanics*, vol. 44, pp. 323–346, 2012.

[16] D. J. Price, "Modelling discontinuities and kelvin–helmholtz instabilities in sph," *Journal of Computational Physics*, vol. 227, no. 24, pp. 10040–10057, 2008.

[17] N. Trask, M. Maxey, K. Kim, M. Perego, M. L. Parks, K. Yang, and J. Xu, "A scalable consistent second-order sph solver for unsteady low reynolds number flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 289, pp. 155–178, 2015.

[18] X. Hu and N. A. Adams, "An incompressible multi-phase sph method," *Journal of computational physics*, vol. 227, no. 1, pp. 264–278, 2007.

[19] P. Bochev and J. Hyman, "Principles of mimetic discretizations of differential operators," in *Compatible Spatial Discretizations* (D. Arnold, P. Bochev, R. Lehoucq, R. Nicolaides, and

M. Shashkov, eds.), vol. 142 of *The IMA Volumes in Mathematics and its Applications*, pp. 89–119, Springer New York, 2006.

[20] J. Hyman and M. Shashkov, "Natural discretizations for the divergence, gradient, and curl on logically rectangular grids," *Computers and Mathematics with Applications*, vol. 33, no. 4, pp. 81 – 104, 1997.

[21] D. Mirzaei, R. Schaback, and M. Dehghan, "On generalized moving least squares and diffuse derivatives," *IMA Journal of Numerical Analysis*, p. drr030, 2011.

[22] H. Wendland, *Scattered data approximation*, vol. 17. Cambridge university press, 2004.

[23] D. Mirzaei and R. Schaback, "Direct meshless local Petrov-Galerkin (DMLPG) method: A generalized MLS approximation," Technical Report arXiv:1303.3095, March 2013.

[24] B. Nayroles, G. Touzot, and P. Villon, "Generalizing the finite element method: Diffuse approximation and diffuse elements," *Computational Mechanics*, vol. 10, no. 5, pp. 307–318, 1992.

[25] P. Bochev and R. B. Lehoucq, "On the finite element solution of the pure neumann problem," *SIAM review*, vol. 47, no. 1, pp. 50–66, 2005.

[26] A. Yeckel and J. J. Derby, "On setting a pressure datum when computing incompressible flows," *International Journal for Numerical Methods in Fluids*, vol. 29, no. 1, pp. 19–34, 1999.

[27] J. Xu, "Iterative methods by space decomposition and subspace correction," *SIAM review*, vol. 34, no. 4, pp. 581–613, 1992.

[28] A. Masud and T. J. Hughes, "A stabilized mixed finite element method for darcy flow," *Computer Methods in Applied Mechanics and Engineering*, vol. 191, no. 39, pp. 4341–4370, 2002.

[29] P. B. Bochev and M. D. Gunzburger, *Least-squares finite element methods*, vol. 166. Springer Science & Business Media, 2009.

[30] X. Luo, A. Beskok, and G. E. Karniadakis, "Modeling electrokinetic flows by the smoothed profile method," *Journal of computational physics*, vol. 229, no. 10, pp. 3828–3847, 2010.