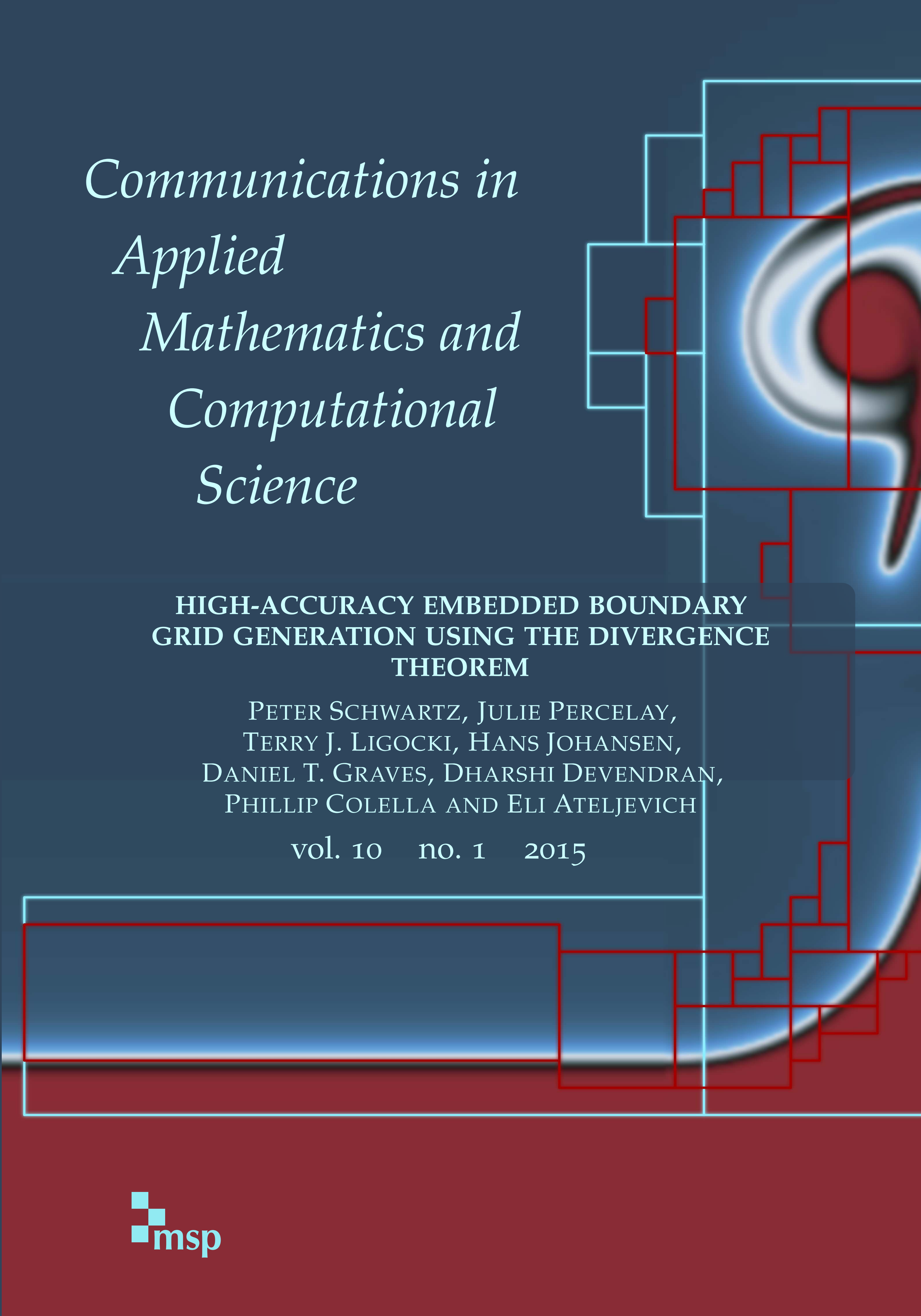

Communications in
Applied
Mathematics and
Computational
Science

HIGH-ACCURACY EMBEDDED BOUNDARY GRID GENERATION USING THE DIVERGENCE THEOREM

PETER SCHWARTZ, JULIE PERCELAY,
TERRY J. LIGOCKI, HANS JOHANSEN,
DANIEL T. GRAVES, DHARSHI DEVENDRAN,
PHILLIP COLELLA AND ELI ATELJEVICH

vol. 10 no. 1 2015

msp

COMM. APP. MATH. AND COMP. SCI.
Vol. 10, No. 1, 2015
dx.doi.org/10.2140/camcos.2015.10.83

HIGH-ACCURACY EMBEDDED BOUNDARY GRID GENERATION USING THE DIVERGENCE THEOREM

PETER SCHWARTZ, JULIE PERCELAY, TERRY J. LIGOCKI,
HANS JOHANSEN, DANIEL T. GRAVES, DHARSHI DEVENDRAN,
PHILLIP COLELLA AND ELI ATELJEVICH

We present an algorithm to produce the necessary geometric information for finite volume calculations in the context of Cartesian grids with embedded boundaries. Given an order of accuracy for the overall calculation, we show what accuracy is required for each of the geometric quantities and we demonstrate how to calculate the moments using the divergence theorem. We demonstrate that, for a known flux, these moments can be used to create a flux divergence of the expected order.

1. Introduction

This work is motivated by the desire to solve partial differential equations (PDEs) conservatively in the context of complex geometries. As an example, consider Poisson's equation. Given a charge density ρ, Poisson's equation can be written as

$$\nabla \cdot (\nabla \phi) = \rho \tag{1}$$

for the potential ϕ. If we integrate this equation over a control volume Ω and apply the divergence theorem, this becomes

$$\int_{\partial\Omega} \nabla\phi \cdot \hat{n}\, dA = \int_{\Omega} \rho\, dV,$$

where $\hat{n}$ is the outward-facing unit normal to the surface. Our volumes are Cartesian cells cut by an embedded boundary. Refer to Figure 1 for an illustration. We refer to the unshaded region as the "volume". The boundaries of the volume aligned to coordinate directions that connect to other volumes we refer to as "faces". We refer to the section of the embedded boundary that cuts the volume as the "EB face".

Formally, the underlying description of space is given by rectangular control volumes on a Cartesian mesh $\Upsilon_{\boldsymbol{i}} = [(\boldsymbol{i} - \frac{1}{2}\boldsymbol{u})h, (\boldsymbol{i} + \frac{1}{2}\boldsymbol{u})h]$, $\boldsymbol{i} \in \mathbb{Z}^D$, where D

Research at LBNL was supported financially by the Office of Advanced Scientific Computing Research of the US Department of Energy under contract number DE-AC02-05CH11231. All work was done using the Chombo software infrastructure developed by LBNL [4; 5].
MSC2010: 65N12, 65N50, 65N08.
Keywords: Cartesian grid embedded boundaries, grid generation, finite volume methods.

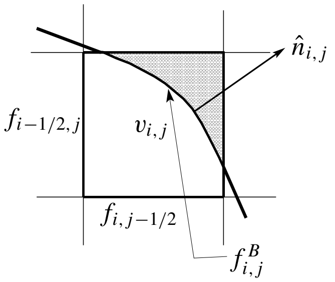

Figure 1. Illustration of cut cell notation. The shaded region is outside the solution domain. The volume v is connected to other volumes via the faces aligned with the coordinate planes. The EB face is formed by the intersection of the embedded boundary and the cell.

is the dimensionality of the problem, h is the mesh spacing, and $\boldsymbol{u}$ is the vector whose entries are all one (note we use bold font $\boldsymbol{u} = (u_1, \ldots, u_d, \ldots, u_D)$ to indicate a vector quantity). Given an irregular domain Ω, we obtain control volumes $V_{\boldsymbol{i}} = \Upsilon_{\boldsymbol{i}} \cap \Omega$ and faces $A_{\boldsymbol{i},d\pm} = A_{\boldsymbol{i}\pm\boldsymbol{e}_d/2}$, which are the intersection of the boundary of $\partial V_{\boldsymbol{i}}$ with the coordinate planes $\{\boldsymbol{x} : x_d = (i_d \pm \frac{1}{2})h\}$ ($\boldsymbol{e}_d$ is the unit vector in the d direction). We also define $A_{B,\boldsymbol{i}}$ to be the intersection of the boundary of the irregular domain with the Cartesian control volume: $A_{B,\boldsymbol{i}} = \partial\Omega \cap \Upsilon_{\boldsymbol{i}}$. From here on, the subscript $\boldsymbol{i}$ is implied as the analysis applies to any given volume $V_{\boldsymbol{i}}$.

Given a flux $\boldsymbol{F}$ (in the case of the Poisson equation (1), $\boldsymbol{F} = \nabla\phi$), we can rewrite the volume integral of the divergence of $\boldsymbol{F}$ as an integral over each face in the volume:

$$\int_V \nabla \cdot \boldsymbol{F}\, dV = \sum_{d=1}^{D}\left(\int_{A_{d+}} F_d\, dA - \int_{A_{d-}} F_d\, dA + \int_{A_B} F_d\, n_d(\boldsymbol{x})\, dA\right), \tag{2}$$

where we define $n_d(\boldsymbol{x})$ to be the d-th component of the outward-facing unit normal to the EB face. The accuracy with which one computes the integrals above will depend on the accuracy of the geometric description of the volume and its associated faces.

Throughout this paper, we use the following compact "multi-index" notation:

$$(\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{p}} = \prod_{d=1}^{D} (x_d - \bar{\boldsymbol{x}}_d)^{p_d},$$

$$\boldsymbol{p}! = \prod_{d=1}^{D} p_d!.$$

Given a point in space $\bar{\boldsymbol{x}}$ and a D-dimensional integer vector $\boldsymbol{p}$, we define $m_v^{\boldsymbol{p}}(\bar{\boldsymbol{x}})$ to be the $\boldsymbol{p}$-th moment of the volume V relative to the point $\bar{\boldsymbol{x}}$:

$$m_v^{\boldsymbol{p}}(\bar{\boldsymbol{x}}) = \int_V (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{p}}\, dV. \tag{3}$$

Clearly, the volume of the cut cell $|V| = m_v^{\boldsymbol{z}}$, where $\boldsymbol{z}$ is the zero vector. We define the face moments $m_{d\pm}^{\boldsymbol{p}}(\bar{\boldsymbol{x}})$ to be the $\boldsymbol{p}$-th moments (relative to the point $\bar{\boldsymbol{x}}$) of the faces $A_{d\pm}$:

$$m_{d\pm}^{\boldsymbol{p}}(\bar{\boldsymbol{x}}) = \int_{A_{d\pm}} (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{p}} \, dA. \tag{4}$$

We define two moments corresponding to the embedded boundary face A_B:

$$m_B^{\boldsymbol{p}}(\bar{\boldsymbol{x}}) = \int_{A_B} (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{p}} \, dA, \tag{5}$$

$$m_{B,d}^{\boldsymbol{p}}(\bar{\boldsymbol{x}}) = \int_{A_B} (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{p}} n_d(\boldsymbol{x}) \, dA. \tag{6}$$

Note that (6) includes the normal to the embedded boundary face.

Given a sufficiently smooth function ψ, we can approximate ψ in the neighborhood of $\bar{\boldsymbol{x}}$ using a multidimensional Taylor expansion:

$$\psi(\boldsymbol{x}) = \sum_{|\boldsymbol{q}|<Q} \frac{1}{\boldsymbol{q}!} \psi^{(\boldsymbol{q})}(\bar{\boldsymbol{x}})(\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{q}} + O(h^Q) \tag{7}$$

with the multi-index partial derivative notation

$$\psi^{(\boldsymbol{q})} = \partial^{\boldsymbol{q}} \psi = \frac{\partial^{q_1}}{\partial x_1^{q_1}} \cdots \frac{\partial^{q_D}}{\partial x_D^{q_D}} \psi. \tag{8}$$

We express averages over volumes as

$$\langle \nabla \cdot \boldsymbol{F} \rangle_V \equiv \frac{1}{|V|} \int_V (\nabla \cdot \boldsymbol{F}) \, dV.$$

We define the volume fraction κ to be fraction of the volume of the cell inside the solution domain so that

$$\kappa = h^{-D}|V| = h^{-D} m_v^{\boldsymbol{z}}. \tag{9}$$

Given a flux function $\boldsymbol{F}$, the κ-weighted divergence of the flux is defined to be the volume average of the divergence multiplied by κ:

$$\begin{aligned} \kappa \langle \nabla \cdot \boldsymbol{F} \rangle_V &= \frac{1}{h^D} \int_V \nabla \cdot \boldsymbol{F} \, dV \\ &= \frac{1}{h^D} \sum_{d=1}^{D} \Big(\int_{A_{d+}} F_d(\boldsymbol{x}) \, dA - \int_{A_{d-}} F_d(\boldsymbol{x}) \, dA + \int_{A_B} F_d(\boldsymbol{x}) n_d(\boldsymbol{x}) \, dA \Big). \end{aligned} \tag{10}$$

We weigh the conservative divergence this way to avoid small-κ numerical instabilities. For example, implicit algorithms for Poisson's equation (1) solve the discrete system

$$\kappa \langle \nabla \cdot \nabla \phi \rangle_V = \kappa \langle \rho \rangle_V$$

for ϕ [9; 17], which avoids very large negative eigenvalues from terms with κ^{-1}. Since κ is only a diagonal scaling, the accuracy of the method is primarily dependent on the accuracy of the discretization of $(\nabla \cdot \nabla \phi)$.

Explicit algorithms for hyperbolic systems with flux $\boldsymbol{F}$ may use a hybrid operator that is a linear combination of a conservative discretization for $\kappa \langle \nabla \cdot \boldsymbol{F} \rangle$ and a nonconservative (not in flux-divergence form) but stable approximation of $\nabla \cdot \boldsymbol{F}$ to advance the solution. The loss of conservation for this hybrid operator can be calculated and redistributed to neighboring cells so that the overall scheme is globally conservative [2; 6; 3; 8; 14; 13]. Suppose we have $\kappa D^{\mathrm{C}}(\boldsymbol{F})$, an $O(h^P)$ conservative discretization of $\kappa \nabla \cdot \boldsymbol{F}$. Suppose we also have $D^{\mathrm{NC}}(\boldsymbol{F})$, an $O(h^P)$ nonconservative discretization of $\nabla \cdot \boldsymbol{F}$, i.e.,

$$\begin{aligned} D^{\mathrm{NC}}(\boldsymbol{F}) &= \nabla \cdot \boldsymbol{F} + O(h^P), \\ \kappa D^{\mathrm{C}}(\boldsymbol{F}) &= \kappa \nabla \cdot \boldsymbol{F} + O(h^P). \end{aligned}$$

The hybrid operator $D^{\mathrm{H}}(\boldsymbol{F})$ is given by

$$\begin{aligned} D^{\mathrm{H}}(\boldsymbol{F}) &= \kappa D^{\mathrm{C}}(\boldsymbol{F}) - (1-\kappa) D^{\mathrm{NC}}(\boldsymbol{F}) \\ &= \nabla \cdot \boldsymbol{F} + O(h^P), \end{aligned}$$

and the resulting loss of conservation in each cell is given by

$$\begin{aligned} \delta &= \kappa D^{\mathrm{C}}(\boldsymbol{F}) - \kappa D^{\mathrm{H}}(\boldsymbol{F}) \\ &= \kappa(1-\kappa)(D^{\mathrm{C}}(\boldsymbol{F}) - D^{\mathrm{NC}}(\boldsymbol{F})) = O(h^P), \end{aligned}$$

where for stability arguments δ must be redistributed in a κ-weighted way [3; 8]. This is of the order of the truncation error so that the accuracy of the method only depends on the accuracy of the two discretizations of the flux divergence. We wish to investigate what accuracy is necessary for the moment calculations to achieve a given order of accuracy in $\kappa D^{\mathrm{C}}(\boldsymbol{F})$.

Embedded boundaries have been used in a wide variety of applications, and several different grid generation techniques have emerged. In an early paper in the field, Pember et al. [16] use a piecewise-planar approximation for grid generation for solutions of inviscid, polytropic gas dynamics. Aftosmis et al. [1] use triangulation to generate grids for a wide variety of extremely complex geometries; their software is still widely used. Singh et al. [18] use triangulation to reconstruct moving boundaries. The embedded boundary algorithms in [3; 8; 14; 13] use a second-order, implicit function–based approach. Miller et al. [12] use a time-dependent, second-order implicit function approach to cut four-dimensional cells to generate moving geometries. Sussman and Puckett [19] use piecewise-planar volume of fluid reconstruction in their incompressible flow algorithm. Nourgaliev et al. [15] and Marella et al. [11] use a piecewise-linear reconstruction in their sharp interface methods. All these algorithms use piecewise-planar approximations to the cutting surface to generate geometric information. These algorithms are first order in $D(\boldsymbol{F})$ (second-order fluxes), so second-order geometric information is accurate enough for their purposes. In the graphics and computational geometry community, there has been a substantial amount of work done in geometric moment computation using the divergence theorem. Yang et al. [20] use the divergence theorem to compute geometric moments for image analysis. Gonzalez-Ochoa et al. [7] have a fast divergence theorem–based moment algorithm specialized for surfaces that are described as polynomials. These algorithms are highly focused on reducing computational complexity, and though their accuracy is measured for a particular refinement, no concern is given to the algorithms' convergence rate with grid refinement. The current work is intended to be a careful exploration of the required accuracy for geometric moments in the context of embedded boundary calculations using higher-order finite volume methods.

2. Accuracy of the discrete divergence

Given an order of accuracy Q, suppose we can approximate $\boldsymbol{F}$ to $O(h^Q)$. If we expand $\boldsymbol{F}$ in (2) using (7), we get

$$\int_V (\nabla \cdot \boldsymbol{F})\, dV = \sum_{|\boldsymbol{q}|<Q} \sum_{d=1}^{D} \frac{1}{\boldsymbol{q}!} F_d^{(\boldsymbol{q})}(\bar{\boldsymbol{x}}) \biggl(\int_{A_{d+}} (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{q}}\, dA - \int_{A_{d-}} (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{q}}\, dA \\ + \int_{A_B} (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{q}} n_d(\boldsymbol{x})\, dA \biggr) + O(h^{Q+D-1}).$$

The accuracy of the expression is $O(h^{Q+D-1})$ because it is an $O(h^Q)$ Taylor series integrated over an area of order h^{D-1}. Using our moment definitions ((4) and (6)),

$$\int_V (\nabla \cdot F)\, dV = \sum_{|\boldsymbol{q}|<Q} \sum_{d=1}^{D} \frac{1}{\boldsymbol{q}!} F_d^{(q)}(\bar{\boldsymbol{x}}) \bigl(m_{d+}^{\boldsymbol{q}} - m_{d-}^{\boldsymbol{q}} + m_{B,d}^{\boldsymbol{q}} \bigr) + O(h^{Q+D-1}). \quad (11)$$

Now suppose we can discretize each moment to $O(h^R)$. We define our discrete moments as

$$M_v^{\boldsymbol{q}} = \int_V (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{q}}\, dV + O(h^R) = m_v^{\boldsymbol{q}} + O(h^R), \quad (12)$$

$$M_{d\pm}^{\boldsymbol{q}} = \int_{A_{d\pm}} (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{q}}\, dA + O(h^R) = m_{d\pm}^{\boldsymbol{q}} + O(h^R), \quad (13)$$

$$M_{B,d}^{\boldsymbol{q}} = \int_{A_B} (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{q}} n_d\, dA + O(h^R) = m_{B,d}^{\boldsymbol{q}} + O(h^R), \quad (14)$$

$$M_B^{\boldsymbol{q}} = \int_{A_B} (\boldsymbol{x} - \bar{\boldsymbol{x}})^{\boldsymbol{q}}\, dA + O(h^R) = m_B^{\boldsymbol{q}} + O(h^R). \quad (15)$$

We take these moment definitions and put them into (11) to get our discrete, κ-weighted average divergence of the flux:

$$\begin{aligned}\kappa D(\boldsymbol{F})_i &= \frac{1}{h^D}\sum_{|\boldsymbol{q}|<Q}\sum_{d=1}^{D}\frac{1}{\boldsymbol{q}!}F_d^{(q)}(\bar{\boldsymbol{x}})\big(M_{d+}^{\boldsymbol{q}} - M_{d-}^{\boldsymbol{q}} + M_{B,d}^{\boldsymbol{q}}\big)\\ &= \kappa\langle\nabla\cdot\boldsymbol{F}\rangle_V + O(h^{Q-1}) + O(h^{R-D}).\end{aligned} \tag{16}$$

The first error term is from (11), and the second is from (13) and (14); both have been divided by the full cell volume, h^D. So if one desires a weighted divergence of accuracy $O(h^P)$, one needs fluxes approximated to accuracy $O(h^Q)$ and moments approximated to accuracy $O(h^R)$, where

$$\begin{aligned} Q &= P+1,\\ R &= P+D.\end{aligned} \tag{17}$$

3. Algorithm

Consider a volume V at cell i (see Figure 1), and let $\bar{\boldsymbol{x}}$ be some point in cell i. We may choose $\bar{\boldsymbol{x}}$ as the origin of our coordinate system so that $\|\boldsymbol{x}-\bar{\boldsymbol{x}}\| = O(h)$ for any point $\boldsymbol{x}$ in V. In the following derivation, we will set $\bar{\boldsymbol{x}} = 0$ without loss of generality.

In (2), if we choose $\boldsymbol{F} = \boldsymbol{x}^{\boldsymbol{q}}\boldsymbol{e}_d$, we get D equations of the form

$$q_d\int_V \boldsymbol{x}^{\boldsymbol{q}-\boldsymbol{e}_d}\,dV = \int_{A_{d+}}\boldsymbol{x}^{\boldsymbol{q}}\,dA - \int_{A_{d-}}\boldsymbol{x}^{\boldsymbol{q}}\,dA + \int_{A_B}\boldsymbol{x}^{\boldsymbol{q}}n_d\,dA.$$

Recall that the normal to the embedded boundary is a function of space $\boldsymbol{n}(\boldsymbol{x})$, so we assume that we can expand this last integral using a Taylor series of $\boldsymbol{n}$ about $\bar{\boldsymbol{x}} = 0$ to an appropriate order of accuracy S:

$$\int_{A_B}\boldsymbol{x}^{\boldsymbol{q}}n_d(\boldsymbol{x})\,dA = \sum_{|\boldsymbol{s}|<S}\frac{1}{\boldsymbol{s}!}\partial^{\boldsymbol{s}}n_d(0)m_B^{\boldsymbol{q}+\boldsymbol{s}} + O(h^{|\boldsymbol{q}|+D+S-1}). \tag{18}$$

The advantage of this expansion is that m_B does not include $n_d(\boldsymbol{x})$ in the integrand. Combining and moving the first term of the Taylor series to the left-hand side, we have

$$\begin{aligned} &q_d m_v^{\boldsymbol{q}-\boldsymbol{e}_d} - n_d(0)m_B^{\boldsymbol{q}}\\ &\qquad = m_{d+}^{\boldsymbol{q}} - m_{d-}^{\boldsymbol{q}} + \sum_{0<|\boldsymbol{s}|<S}\frac{1}{\boldsymbol{s}!}\partial^{\boldsymbol{s}}n_d(0)m_B^{\boldsymbol{q}+\boldsymbol{s}} + O(h^{|\boldsymbol{q}|+D+S-1}).\end{aligned} \tag{19}$$

Note that, for higher-order moments, we need fewer terms in the Taylor series for the normal to obtain a truncation error of $O(h^R)$ because the truncation error depends on $|\boldsymbol{q}|+S$. In particular, if $|\boldsymbol{q}| = Q$, (17) shows that we only need $S = 0$,

```
0. Compute all derivatives of the normal, $\partial^{\boldsymbol{s}} n_d$ (see Section 3.2).
1. Compute one-dimensional moments $M^{\boldsymbol{q}}_{V,1}$ using root-finding and integration.
2. Irregular one-dimensional moments $M^{\boldsymbol{q}}_{B,1}$ are zero.
3. Loop through other dimensions as follows:
    for ($D = 2, 3$)
        for ($|\boldsymbol{q}| = \{Q, Q-1, \dots, 0\}$)
            Set $\rho$:
                $\rho = M^{\boldsymbol{q}}_{d+,D-1} - M^{\boldsymbol{q}}_{d-,D-1} + \sum_{1\le|\boldsymbol{s}|<S} \frac{\partial^{\boldsymbol{s}} n_d}{\boldsymbol{s}!} M^{\boldsymbol{q}+\boldsymbol{s}}_{B,D}.$
            Solve for $M^{\boldsymbol{q}-\boldsymbol{e}_d}_{V,D}$ and $M^{\boldsymbol{q}}_{B,D}$:
                $q_d M^{\boldsymbol{q}-\boldsymbol{e}_d}_{V,D} - n_d M^{\boldsymbol{q}}_{B,D} = \rho.$

        end loop over moments
    end loop over dimensions
```

Figure 2. Outline of the moment algorithm. The second subscript of the moments refers to dimensionality ($M^{\boldsymbol{p}}_{v,3}$ refers to the three-dimensional $M^{\boldsymbol{p}}_{v}$, for example).

and the third term on the right-hand side drops out. Also, notice that $m^{\boldsymbol{q}}_{d+}$ and $m^{\boldsymbol{q}}_{d-}$ are integrals of lower dimension.

We drop the truncation error term in (19) and define the approximate moments ($M^{\boldsymbol{q}}_v$, etc.) using the set of equations

$$q_d M^{\boldsymbol{q}-\boldsymbol{e}_d}_v - n_d(0) M^{\boldsymbol{q}}_B = M^{\boldsymbol{q}}_{d+} - M^{\boldsymbol{q}}_{d-} + \sum_{1\le|\boldsymbol{s}|<S} \frac{1}{\boldsymbol{s}!} \partial^{\boldsymbol{s}} n_d(0) M^{\boldsymbol{q}+\boldsymbol{s}}_B. \tag{20}$$

If the moments on the left are treated as unknowns and the moments on the right are treated as known, then this set of equations forms a system of linear equations for a fixed $|\boldsymbol{q}|$. The key to the algorithm is to compute the moments in an order that assures the quantities on the right are known. In particular, we compute the moments in lower dimensions first and use them as known quantities in the equation for higher dimension. For a given dimension, we generate moments in order of decreasing $|\boldsymbol{q}|$ starting with $|\boldsymbol{q}| = Q$. Because the third term on the right-hand side of (20) depends on moments with orders $|\boldsymbol{q}|+1, \dots, Q$, the procedure guarantees that term is known. The algorithm can also be described as a recursion as shown by Ligocki et al. [10] (the conference paper associated with this work). Figure 2 shows an outline of the algorithm. The number of unknowns is $N_{|\boldsymbol{q}|-1} + N_{|\boldsymbol{q}|}$, where $N_{|\boldsymbol{q}|}$ is the number of monomials of degree $|\boldsymbol{q}|$. (One can think of $N_{|\boldsymbol{q}|}$ as the length of the $|\boldsymbol{q}|$-th row of the Pascal's triangle formed by the moments.) The number of equations is $DN_{|\boldsymbol{q}|}$. Since $N_{|\boldsymbol{q}|-1} < N_{|\boldsymbol{q}|}$, this is an overdetermined system that can be solved using least squares. The normal and all its derivatives are computed as shown in Section 3.2.

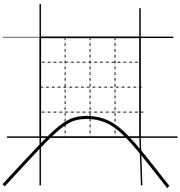

Figure 3. Illustration of an under-resolved geometry. When the curvature of the implicit function is too great to be resolved, we refine locally and sum the finer moments to compute the under-resolved cell's moments.

3.1. ***Under-resolved geometry.*** Note that the above algorithm can fail in the case of an under-resolved geometry. If the curvature of the implicit function is too high, the situation shown in Figure 3 can result. To resolve this, we refine the grid locally until the implicit function is resolved. We use the divergence of the normal as a measure for under-resolution. Given a tolerance ϵ, if $D(\boldsymbol{n}) > \epsilon$ at volume $\boldsymbol{v}$ and the set of finer volumes $S = \{\boldsymbol{v}_f\}$ that compose the volume $\boldsymbol{v}$, we get the equation

$$\boldsymbol{m}_{\boldsymbol{v}} = \sum_{\boldsymbol{v}_f \in S} \boldsymbol{m}_{\boldsymbol{v}_f}.$$

If the geometry is not smooth enough to be resolved, a maximum level of resolution is defined. This paper is primarily concerned with convergence rates. Convergence tests require smooth, resolved geometries, so none of the geometries presented in this paper need this refinement.

3.2. ***Calculation of derivatives of the normal using an implicit function.*** Our geometry is defined as the zero set of an implicit function $\psi(\boldsymbol{x})$, the zero level set of which is the embedded boundary. The normal $\hat{n}$ of the embedded boundary is defined as

$$\hat{n} = \frac{\nabla\psi}{L},$$

where $L = |\nabla\psi|$. For the d-th component of the normal, the multi-index product rule can be used for higher partial derivatives $\boldsymbol{p}$ of the quantity $\psi^{(\boldsymbol{e}_d)} = Ln_d$:

$$\begin{aligned}\psi^{(\boldsymbol{p}+\boldsymbol{e}_d)} &= \partial^{\boldsymbol{p}}\psi^{(\boldsymbol{e}_d)}\\ &= \partial^{\boldsymbol{p}}(Ln_d)\\ &= \sum_{\boldsymbol{q}\le\boldsymbol{p}} \binom{\boldsymbol{p}}{\boldsymbol{q}} \partial^{\boldsymbol{p}-\boldsymbol{q}} L \partial^{\boldsymbol{q}} n_d.\end{aligned}$$

Note that the left-hand side is a known value from the implicit function. On the right-hand side, we have unknown derivatives of L and n_d. We can rewrite to express the highest normal derivative as

$$L\partial^{\boldsymbol{p}} n_d = \psi^{(\boldsymbol{p}+\boldsymbol{e}_d)} - \sum_{\boldsymbol{q}\le\boldsymbol{p},\ \boldsymbol{q}\ne\boldsymbol{p}} \binom{\boldsymbol{p}}{\boldsymbol{q}} \partial^{\boldsymbol{p}-\boldsymbol{q}} L \partial^{\boldsymbol{q}} n_d.$$

Next, we will evaluate the derivatives of L. To find a recursion formula, we first apply the chain rule to ∂L^2:

$$\begin{aligned}
\partial^{\boldsymbol{e}^i} L^2 &= \partial^{\boldsymbol{e}^i} \sum_d (\psi^{(\boldsymbol{e}_d)})^2, \\
2L\partial^{\boldsymbol{e}^i} L &= \sum_d 2\psi^{(\boldsymbol{e}_d)} \psi^{(\boldsymbol{e}_d+\boldsymbol{e}^i)}, \\
\partial^{\boldsymbol{e}^i} L &= \sum_d \frac{\psi^{(\boldsymbol{e}_d)}}{L} \psi^{(\boldsymbol{e}_d+\boldsymbol{e}^i)} \\
&= \sum_d n_d \psi^{(\boldsymbol{e}_d+\boldsymbol{e}^i)}.
\end{aligned}$$

Looking at the right-hand side, we have calculated a derivative $\partial^{\boldsymbol{e}^i} L$ from n_d and known derivatives of ψ. Extending to the case of $\boldsymbol{p} \geq \boldsymbol{e}^i$, we have

$$\partial^{\boldsymbol{p}} L = \partial^{\boldsymbol{p}-\boldsymbol{e}^i} (\partial^{\boldsymbol{e}^i} \psi) \tag{21}$$

$$= \partial^{\boldsymbol{p}-\boldsymbol{e}^i} \sum_d n_d \psi^{(\boldsymbol{e}_d+\boldsymbol{e}^i)} \tag{22}$$

$$= \sum_d \sum_{\boldsymbol{r} \leq \boldsymbol{q}} \binom{\boldsymbol{q}}{\boldsymbol{r}} \partial^{\boldsymbol{r}} n_d \partial^{\boldsymbol{q}-\boldsymbol{r}+\boldsymbol{e}_d+\boldsymbol{e}^i} \psi, \quad \text{where } \boldsymbol{q} = \boldsymbol{p} - \boldsymbol{e}^i. \tag{23}$$

Again, this expresses higher-order derivatives of L in terms of lower-order derivatives of n_d and known derivatives of ψ. The choice of $\boldsymbol{e}^i$ is arbitrary but is such that all the $\partial^{\boldsymbol{r}} n_d$ have already been computed. To calculate all the powers needed in (23), we can choose $\boldsymbol{e}^i = \boldsymbol{e}_d$, calculate the required derivatives for $d = 0$ first, then add those with derivatives in $d = 1$, etc. This guarantees that the requested derivatives will have already been calculated in the recursion.

4. Results

4.1. ***Geometric description.*** Our convergence tests are calculated using a geometry of an ellipsoid (or an ellipse in two dimensions) centered in a unit domain. The implicit function, $\psi(\boldsymbol{x})$, that defines the ellipsoid is given by

$$\psi(\boldsymbol{x}) = \sum_{d=1}^{D} \left(\frac{x_d - x_{0,d}}{A_d} \right)^2 - R^2,$$

where $\boldsymbol{x}_0$ is put at the center of the domain. The stretching constant $\boldsymbol{A} = \{1, 2, 3\}$ in three dimensions ($\boldsymbol{A} = \{1, 2\}$ in two dimensions). The constant, $R = 0.15$, sets the unexpanded radius of the ellipsoid. The surface of the ellipsoid is described by the surface where $\psi(\boldsymbol{x}) = 0$.

Variable	ϵ^{2h}	ϖ	ϵ^{h}	Variable	ϵ^{2h}	ϖ	ϵ^{h}
$M_v^{(0,0)}$	3.405×10^{-12}	7.08	2.525×10^{-14}	$M_B^{(0,0)}$	2.719×10^{-10}	6.92	2.244×10^{-12}
$M_v^{(1,0)}$	1.330×10^{-13}	6.30	1.688×10^{-15}	$M_B^{(1,0)}$	1.802×10^{-11}	6.98	1.426×10^{-13}
$M_v^{(2,0)}$	2.028×10^{-14}	7.00	1.582×10^{-16}	$M_B^{(2,0)}$	2.684×10^{-12}	6.97	2.143×10^{-14}
$M_v^{(3,0)}$	2.783×10^{-15}	6.51	3.052×10^{-17}	$M_B^{(3,0)}$	1.495×10^{-13}	7.06	1.122×10^{-15}
$M_v^{(4,0)}$	1.490×10^{-15}	7.03	1.141×10^{-17}	$M_B^{(4,0)}$	4.426×10^{-14}	7.00	3.466×10^{-16}
$M_v^{(0,1)}$	9.842×10^{-13}	7.05	7.402×10^{-15}	$M_B^{(0,1)}$	1.634×10^{-11}	7.01	1.266×10^{-13}
$M_v^{(1,1)}$	3.844×10^{-14}	6.40	4.550×10^{-16}	$M_B^{(1,1)}$	1.931×10^{-13}	5.29	4.945×10^{-15}
$M_v^{(2,1)}$	6.041×10^{-15}	7.02	4.655×10^{-17}	$M_B^{(2,1)}$	1.952×10^{-13}	6.97	1.561×10^{-15}
$M_v^{(3,1)}$	8.044×10^{-16}	6.61	8.225×10^{-18}	$M_B^{(3,1)}$	9.528×10^{-15}	7.05	7.168×10^{-17}
$M_v^{(0,2)}$	1.826×10^{-13}	7.02	1.411×10^{-15}	$M_B^{(0,2)}$	6.589×10^{-13}	6.94	5.363×10^{-15}
$M_v^{(1,2)}$	7.132×10^{-15}	6.63	7.202×10^{-17}	$M_B^{(1,2)}$	3.190×10^{-15}	5.70	6.125×10^{-17}
$M_v^{(2,2)}$	4.375×10^{-16}	5.76	8.079×10^{-18}	$M_B^{(2,2)}$	1.459×10^{-14}	7.06	1.094×10^{-16}
$M_v^{(0,3)}$	6.396×10^{-15}	5.75	1.185×10^{-16}	$M_B^{(0,3)}$	5.994×10^{-13}	7.06	4.493×10^{-15}
$M_v^{(1,3)}$	5.497×10^{-16}	6.13	7.872×10^{-18}	$M_B^{(1,3)}$	2.194×10^{-14}	6.76	2.025×10^{-16}
$M_v^{(0,4)}$	8.242×10^{-16}	6.76	7.610×10^{-18}	$M_B^{(0,4)}$	2.491×10^{-14}	5.74	4.657×10^{-16}

Table 1. Volume (left) and embedded boundary (right) moment convergence rates for $h = 1/128$ using the L_∞ norm in two dimensions. The implicit function is an ellipse described in Section 4.1.

4.2. ***Moment convergence tests.*** To test the convergence rate of the moments, we use Richardson extrapolation, which means that the exact solution on a finer level of refinement is used as an exact solution. Since we are dealing with integrals, the coarsening operation is simple addition. Define $A^{h\rightarrow 2h}$ to be the operator to get the exact solution on the coarse level from the fine solution. Given S_f, the set of fine volumes that cover a coarse volume $\boldsymbol{i}$,

$$A^{h\rightarrow 2h}(\boldsymbol{M})_{\boldsymbol{i}} = \sum_{\boldsymbol{i}_f \in S_f} \boldsymbol{M}_{\boldsymbol{i}_f}.$$

$\boldsymbol{M}_h$ is defined to be our solution on a grid with resolution h. For an exact solution $\boldsymbol{m}^e$, we use $\boldsymbol{m}^e_{2h} = A^{h\rightarrow 2h}(\boldsymbol{M}_h)$ and the error is given by

$$\epsilon^h = \boldsymbol{M}^h(t) - \boldsymbol{m}^e(t). \tag{24}$$

The order of convergence ϖ is estimated by

$$\varpi = \frac{\log(\|\epsilon^{2h}\|/\|\epsilon^h\|)}{\log(2)}. \tag{25}$$

Variable	ϵ^{2h}	ϖ	ϵ^{h}	Variable	ϵ^{2h}	ϖ	ϵ^{h}
$M_v^{(0,0,0)}$	7.744×10^{-13}	5.73	1.461×10^{-14}	$M_v^{(3,0,1)}$	3.329×10^{-17}	7.54	1.789×10^{-19}
$M_v^{(1,0,0)}$	6.130×10^{-15}	6.75	5.706×10^{-17}	$M_v^{(0,1,1)}$	1.577×10^{-15}	7.87	6.745×10^{-18}
$M_v^{(2,0,0)}$	9.836×10^{-16}	7.89	4.153×10^{-18}	$M_v^{(1,1,1)}$	1.173×10^{-16}	7.97	4.682×10^{-19}
$M_v^{(3,0,0)}$	1.096×10^{-16}	7.84	4.773×10^{-19}	$M_v^{(2,1,1)}$	1.845×10^{-17}	7.75	8.553×10^{-20}
$M_v^{(4,0,0)}$	5.936×10^{-17}	7.98	2.356×10^{-19}	$M_v^{(0,2,1)}$	1.314×10^{-16}	7.59	6.834×10^{-19}
$M_v^{(0,1,0)}$	5.449×10^{-14}	6.44	6.281×10^{-16}	$M_v^{(1,2,1)}$	1.383×10^{-17}	7.72	6.545×10^{-20}
$M_v^{(1,1,0)}$	1.132×10^{-15}	7.77	5.178×10^{-18}	$M_v^{(0,3,1)}$	1.338×10^{-17}	7.26	8.748×10^{-20}
$M_v^{(2,1,0)}$	1.761×10^{-16}	8.11	6.382×10^{-19}	$M_v^{(0,0,2)}$	1.036×10^{-14}	7.61	5.304×10^{-17}
$M_v^{(3,1,0)}$	3.908×10^{-17}	7.87	1.672×10^{-19}	$M_v^{(1,0,2)}$	4.254×10^{-16}	7.53	2.306×10^{-18}
$M_v^{(0,2,0)}$	4.901×10^{-15}	7.95	1.985×10^{-17}	$M_v^{(2,0,2)}$	2.390×10^{-17}	7.40	1.414×10^{-19}
$M_v^{(1,2,0)}$	3.301×10^{-16}	7.87	1.412×10^{-18}	$M_v^{(0,1,2)}$	1.772×10^{-16}	7.98	7.008×10^{-19}
$M_v^{(2,2,0)}$	3.225×10^{-17}	7.74	1.505×10^{-19}	$M_v^{(1,1,2)}$	1.369×10^{-17}	8.02	5.263×10^{-20}
$M_v^{(0,3,0)}$	4.400×10^{-16}	7.70	2.114×10^{-18}	$M_v^{(0,2,2)}$	1.307×10^{-17}	7.41	7.664×10^{-20}
$M_v^{(1,3,0)}$	2.666×10^{-17}	7.61	1.367×10^{-19}	$M_v^{(0,0,3)}$	5.093×10^{-16}	7.45	2.913×10^{-18}
$M_v^{(0,4,0)}$	2.817×10^{-17}	7.66	1.393×10^{-19}	$M_v^{(1,0,3)}$	2.115×10^{-17}	7.52	1.156×10^{-19}
$M_v^{(0,0,1)}$	1.570×10^{-14}	7.04	1.196×10^{-16}	$M_v^{(0,1,3)}$	1.376×10^{-17}	7.67	6.749×10^{-20}
$M_v^{(1,0,1)}$	2.987×10^{-16}	7.50	1.654×10^{-18}	$M_v^{(0,0,4)}$	2.252×10^{-17}	7.43	1.310×10^{-19}
$M_v^{(2,0,1)}$	3.498×10^{-16}	7.96	1.408×10^{-18}				

Table 2. Volume moment convergence rates for $h = 1/128$ using the L_∞ norm in three dimensions. The implicit function is the ellipsoid described in Section 4.1.

We set $P = 5$, so in two dimensions, we expect all the moments to converge to order $h^{P+D=7}$. Table 1 shows these two-dimensional results, and we show the expected convergence, even using the L_∞ norm. In three dimensions, we expect (with $P = 5$) the moments to all converge to $O(h^8)$. Tables 2 and 3 show these three-dimensional results, and we show the expected convergence, also using the L_∞ norm.

4.3. ***Flux divergence convergence tests.*** We use an analytic function as a flux function $F = \nabla\psi$, where

$$\psi = \prod_{d=1}^{D} \cos(2\pi(x_d - x_{0,d})),$$

where $\boldsymbol{x}_0$ is the center of the domain. We set $P = 4$ and compute D^K using (16). In both two and three dimensions, we get the expected rates of convergence. Near the embedded boundary, we get third-order convergence, and away from the irregular boundary, we get fourth-order. Table 4 shows the $O(3, 3.5, 4)$ rates that one would expect in the L_∞, L_2, and L_1 norms.

Variable	ϵ^{2h}	ϖ	ϵ^{h}	Variable	ϵ^{2h}	ϖ	ϵ^{h}
$M_B^{(0,0,0)}$	1.687×10^{-10}	4.44	7.790×10^{-12}	$M_B^{(3,0,1)}$	8.564×10^{-16}	8.20	2.906×10^{-18}
$M_B^{(1,0,0)}$	3.121×10^{-12}	6.66	3.089×10^{-14}	$M_B^{(0,1,1)}$	1.200×10^{-13}	6.00	1.873×10^{-15}
$M_B^{(2,0,0)}$	1.705×10^{-13}	7.46	9.697×10^{-16}	$M_B^{(1,1,1)}$	5.201×10^{-15}	8.10	1.896×10^{-17}
$M_B^{(3,0,0)}$	1.612×10^{-14}	7.94	6.575×10^{-17}	$M_B^{(2,1,1)}$	4.043×10^{-16}	7.47	2.285×10^{-18}
$M_B^{(4,0,0)}$	2.437×10^{-15}	8.02	9.407×10^{-18}	$M_B^{(0,2,1)}$	9.659×10^{-15}	7.22	6.470×10^{-17}
$M_B^{(0,1,0)}$	1.041×10^{-11}	4.97	3.315×10^{-13}	$M_B^{(1,2,1)}$	5.686×10^{-16}	8.06	2.132×10^{-18}
$M_B^{(1,1,0)}$	1.111×10^{-13}	6.32	1.386×10^{-15}	$M_B^{(0,3,1)}$	4.861×10^{-16}	7.11	3.521×10^{-18}
$M_B^{(2,1,0)}$	7.297×10^{-15}	7.93	2.997×10^{-17}	$M_B^{(0,0,2)}$	3.829×10^{-13}	7.54	2.057×10^{-15}
$M_B^{(3,1,0)}$	4.923×10^{-16}	7.59	2.561×10^{-18}	$M_B^{(1,0,2)}$	1.190×10^{-14}	7.96	4.786×10^{-17}
$M_B^{(0,2,0)}$	5.984×10^{-13}	5.69	1.157×10^{-14}	$M_B^{(2,0,2)}$	1.127×10^{-15}	8.13	4.031×10^{-18}
$M_B^{(1,2,0)}$	8.115×10^{-15}	7.41	4.779×10^{-17}	$M_B^{(0,1,2)}$	5.905×10^{-15}	7.82	2.606×10^{-17}
$M_B^{(2,2,0)}$	9.585×10^{-16}	7.80	4.315×10^{-18}	$M_B^{(1,1,2)}$	4.104×10^{-16}	7.85	1.778×10^{-18}
$M_B^{(0,3,0)}$	2.451×10^{-14}	6.54	2.637×10^{-16}	$M_B^{(0,2,2)}$	5.964×10^{-16}	7.71	2.856×10^{-18}
$M_B^{(1,3,0)}$	1.044×10^{-15}	7.87	4.472×10^{-18}	$M_B^{(0,0,3)}$	4.159×10^{-14}	7.63	2.096×10^{-16}
$M_B^{(0,4,0)}$	1.820×10^{-15}	7.70	8.776×10^{-18}	$M_B^{(1,0,3)}$	1.530×10^{-15}	7.71	7.299×10^{-18}
$M_B^{(0,0,1)}$	9.229×10^{-12}	7.76	4.251×10^{-14}	$M_B^{(0,1,3)}$	7.425×10^{-16}	7.76	3.423×10^{-18}
$M_B^{(1,0,1)}$	7.297×10^{-14}	7.99	2.868×10^{-16}	$M_B^{(0,0,4)}$	2.263×10^{-15}	7.54	1.216×10^{-17}
$M_B^{(2,0,1)}$	1.618×10^{-14}	7.80	7.266×10^{-17}				

Table 3. Embedded boundary moment convergence rates for $h = 1/128$ using the L_∞ norm in three dimensions. The implicit function is the ellipsoid described in Section 4.1.

5. Conclusions

Given a desired order of accuracy, we present analysis which shows the accuracy required for the geometric moments in the context of Cartesian grids with embedded

D	Norm	ϵ^{2h}	ϖ	ϵ^{h}
2	L_∞	8.839×10^{-6}	2.98	1.122×10^{-6}
2	L_1	1.512×10^{-7}	3.96	9.728×10^{-9}
2	L_2	7.424×10^{-7}	3.45	6.772×10^{-8}
3	L_∞	2.287×10^{-5}	2.98	2.898×10^{-6}
3	L_1	2.663×10^{-7}	3.98	1.685×10^{-8}
3	L_2	1.225×10^{-6}	3.49	1.094×10^{-7}

Table 4. Convergence results for the error in the divergence of an analytical flux ($D^K(F)$) (described in the text) in two and three dimensions for $h = 1/128$ (ϵ^h is the error for $h = 1/128$; ϵ^{2h} is the error for $h = 1/64$).

boundaries. We demonstrate using convergence tests that our moments are calculated to the expected order and that, for a known flux, these moments can be used to create a flux divergence of the expected order. This work provides the foundation for higher-order finite volume, embedded boundary methods.

References

[1] M. J. Aftosmis, M. J. Berger, and J. E. Melton, *Robust and efficient Cartesian mesh generation for component-based geometry*, AIAA J. **36** (1998), no. 6, 952–960.

[2] I.-L. Chern and P. Colella, *A conservative front-tracking method for hyperbolic conservation laws*, Technical Report UCRL-97200, Lawrence Livermore National Laboratory, 1987.

[3] P. Colella, D. T. Graves, B. J. Keen, and D. Modiano, *A Cartesian grid embedded boundary method for hyperbolic conservation laws*, J. Comput. Phys. **211** (2006), no. 1, 347–366. MR 2006i:65142 Zbl 1120.65324

[4] P. Colella, D. T. Graves, T. J. Ligocki, D. F. Martin, D. Modiano, D. B. Serafini, and B. Van Straalen, *Chombo software package for AMR applications: design document*, Technical Report LBNL-6616E, Lawrence Berkeley National Laboratory, 2014.

[5] P. Colella, D. T. Graves, T. J. Ligocki, G. Miller, D. Modiano, P. Schwartz, B. Van Straalen, J. Pillod, D. Trebotich, and M. Barad, *EBChombo software package for Cartesian grid, embedded boundary applications*, Technical Report LBNL-6615E, Lawrence Berkeley National Laboratory, 2014.

[6] Z. Dragojlovic, F. Najmabadi, and M. Day, *An embedded boundary method for viscous, conducting compressible flow*, J. Comput. Phys. **216** (2006), no. 1, 37–51. MR 2223435 Zbl 1173.76372

[7] C. Gonzalez-Ochoa, S. McCammon, and J. Peters, *Computing moments of objects enclosed by piecewise polynomial surfaces*, ACM Trans. Graph. **17** (1998), no. 3, 143–157.

[8] D. T. Graves, P. Colella, D. Modiano, J. Johnson, B. Sjogreen, and X. Gao, *A Cartesian grid embedded boundary method for the compressible Navier–Stokes equations*, Commun. Appl. Math. Comput. Sci. **8** (2013), no. 1, 99–122. MR 3143820 Zbl 1282.76006

[9] H. Johansen and P. Colella, *A Cartesian grid embedded boundary method for Poisson's equation on irregular domains*, J. Comput. Phys. **147** (1998), no. 1, 60–85. MR 99m:65231 Zbl 0923.65079

[10] T. J. Ligocki, P. O. Schwartz, J. Percelay, and P. Colella, *Embedded boundary grid generation using the divergnce theorem, implicit functions, and constructive solid geometry*, SciDAC 2008 (Philadelphia), J. Phys. Conf. Ser., no. 125, IOP, 2008.

[11] S. Marella, S. Krishnan, H. Liu, and H. S. Udaykumar, *Sharp interface Cartesian grid method, I: An easily implemented technique for 3D moving boundary computations*, J. Comput. Phys. **210** (2005), no. 1, 1–31. MR 2006e:65161 Zbl 1154.76359

[12] G. H. Miller and D. Trebotich, *An embedded boundary method for the Navier–Stokes equations on a time-dependent domain*, Commun. Appl. Math. Comput. Sci. **7** (2012), no. 1, 1–31. MR 2893419 Zbl 1273.35215

[13] S. Molins, D. Trebotich, C. I. Steefel, and C. Shen, *An investigation of the effect of pore scale flow on average geochemical reaction rates using direct numerical simulation*, Water Resour. Res. **48** (2012), no. 3.

[14] A. Nonaka, D. Trebotich, G. Miller, D. Graves, and P. Colella, *A higher-order upwind method for viscoelastic flow*, Commun. Appl. Math. Comput. Sci. **4** (2009), no. 1, 57–83. MR 2010j:65138 Zbl 1166.76039

[15] R. R. Nourgaliev, M.-S. Liou, and T. G. Theofanous, *Numerical prediction of interfacial instabilities: sharp interface method* (*SIM*), J. Comput. Phys. **227** (2008), no. 8, 3940–3970. MR 2009d:76054 Zbl 1275.76164

[16] R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome, *Adaptive Cartesian grid methods for representing geometry in inviscid compressible flow*, 11th Computational Fluid Dynamics Conference (Orlando, 1993), American Institute of Aeronautics and Astronautics, 1993, pp. 948–958.

[17] P. Schwartz, M. Barad, P. Colella, and T. Ligocki, *A Cartesian grid embedded boundary method for the heat equation and Poisson's equation in three dimensions*, J. Comput. Phys. **211** (2006), no. 2, 531–550. MR 2006e:65194 Zbl 1086.65532

[18] R. Singh and W. Shyy, *Three-dimensional adaptive Cartesian grid method with conservative interface restructuring and reconstruction*, J. Comput. Phys. **224** (2007), no. 1, 150–167. MR 2008b:76124 Zbl 1248.76111

[19] M. Sussman and E. G. Puckett, *A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows*, J. Comput. Phys. **162** (2000), no. 2, 301–337. MR 2001c:76099 Zbl 0977.76071

[20] L. Yang, F. Albregtsen, and T. Taxt, *Fast computation of three-dimensional geometric moments using a discrete divergence theorem and a generalization to higher dimensions*, Graph. Model. Im. Proc. **59** (1997), no. 2, 97–108.

Received March 12, 2014. Revised July 11, 2014.

PETER SCHWARTZ: `poschwartz@lbl.gov`
Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, United States

JULIE PERCELAY: `julie.percelay@gmail.com`
Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, United States

TERRY J. LIGOCKI: `tjligocki@lbl.gov`
Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, United States

HANS JOHANSEN: `hjohansen@lbl.gov`
Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, United States

DANIEL T. GRAVES: `dtgraves@lbl.gov`
Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, United States

DHARSHI DEVENDRAN: `pdevendran@lbl.gov`
Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, United States

PHILLIP COLELLA: `pcolella@lbl.gov`
Applied Numerical Algorithms Group, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, United States

ELI ATELJEVICH: `eli.ateljevich@water.ca.gov`
California Department of Water Resources, 1416 9th Street, Sacramento, CA 95821, United States

Communications in Applied Mathematics and Computational Science

msp.org/camcos

See inside back cover or msp.org/camcos for submission instructions.

The subscription price for 2015 is US $85/year for the electronic version, and $120/year (+$15, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscribers address should be sent to MSP.

Communications in Applied Mathematics and Computational Science (ISSN 2157-5452 electronic, 1559-3940 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840, is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

CAMCoS peer review and production are managed by EditFLOW® from MSP.

PUBLISHED BY

mathematical sciences publishers

nonprofit scientific publishing

http://msp.org/

Communications in Applied Mathematics and Computational Science

vol. 10 no. 1 2015

1559-3940(2015)10:1;1-O