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
HIGH-ACCURACY EMBEDDED BOUNDARY
GRID GENERATION USING THE DIVERGENCE THEOREM

PETER SCHWARTZ, JULIE PERCELEY, 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 $\rho$, Poisson’s equation can be written as

$$\nabla \cdot (\nabla \phi) = \rho$$

for the potential $\phi$. If we integrate this equation over a control volume $\Omega$ 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
direct 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 $\Gamma_i = [(i - \frac{1}{2}u)h, (i + \frac{1}{2}u)h]$, $i \in \mathbb{Z}^D$, 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.
is the dimensionality of the problem, $h$ is the mesh spacing, and $\mathbf{u}$ is the vector whose entries are all one (note we use bold font $\mathbf{u} = (u_1, \ldots, u_d, \ldots, u_D)$ to indicate a vector quantity). Given an irregular domain $\Omega$, we obtain control volumes $V_i = \gamma_i \cap \Omega$ and faces $A_{i,d\pm} = A_i \pm e_d/2$, which are the intersection of the boundary of $\partial V_i$ with the coordinate planes $\{x : x_d = (i_d \pm \frac{1}{2})h\}$ ($e_d$ is the unit vector in the $d$ direction). We also define $A_{B,i}$ to be the intersection of the boundary of the irregular domain with the Cartesian control volume: $A_{B,i} = \partial \Omega \cap \gamma_i$. From here on, the subscript $i$ is implied as the analysis applies to any given volume $V_i$.

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

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

where we define $n_d(x)$ 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:

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

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

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

$$m_v^p(\bar{x}) = \int_V (x - \bar{x})^p \, dV.$$  \hspace{1cm} (3)
Clearly, the volume of the cut cell $|V| = m^z_v$, where $z$ is the zero vector. We define the face moments $m^p_{d\pm}(\bar{x})$ to be the $p$-th moments (relative to the point $\bar{x}$) of the faces $A_{d\pm}$:

$$m^p_{d\pm}(\bar{x}) = \int_{A_{d\pm}} (x - \bar{x})^p \, dA.$$  

(4)

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

$$m^p_B(\bar{x}) = \int_{A_B} (x - \bar{x})^p \, dA,$$  

(5)

$$m^p_{B,d}(\bar{x}) = \int_{A_B} (x - \bar{x})^p n_d(x) \, dA.$$  

(6)

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

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

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

with the multi-index partial derivative notation

$$\psi^{(q)} = \partial^{q_1} \psi \cdots \partial^{q_D} \psi.$$  

(8)

We express averages over volumes as

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

We define the volume fraction $\kappa$ to be fraction of the volume of the cell inside the solution domain so that

$$\kappa = h^{-D} |V| = h^{-D} m^z_v.$$  

(9)

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

$$\kappa \langle \nabla \cdot F \rangle_V = \frac{1}{h^D} \int_V \nabla \cdot F \, dV$$

$$= \frac{1}{h^D} \sum_{d=1}^D \left( \int_{A_d^+} F_d(x) \, dA - \int_{A_d^-} F_d(x) \, dA + \int_{A_B} F_d(x)n_d(x) \, dA \right).$$  

(10)

We weigh the conservative divergence this way to avoid small-$\kappa$ 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 $\phi$ [9; 17], which avoids very large negative eigenvalues from terms with $\kappa^{-1}$. Since $\kappa$ 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 $F$ may use a hybrid operator that is a linear combination of a conservative discretization for $\kappa (\nabla \cdot F)$ and a nonconservative (not in flux-divergence form) but stable approximation of $\nabla \cdot F$ 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 $\kappa D^C(F)$, an $O(h^p)$ conservative discretization of $\kappa \nabla \cdot F$. Suppose we also have $D^{NC}(F)$, an $O(h^p)$ nonconservative discretization of $\nabla \cdot F$, i.e.,

$$D^{NC}(F) = \nabla \cdot F + O(h^p),$$

$$\kappa D^C(F) = \kappa \nabla \cdot F + O(h^p).$$

The hybrid operator $D^H(F)$ is given by

$$D^H(F) = \kappa D^C(F) - (1 - \kappa)D^{NC}(F) = \nabla \cdot F + O(h^p),$$

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

$$\delta = \kappa D^C(F) - \kappa D^H(F) = \kappa (1 - \kappa)(D^C(F) - D^{NC}(F)) = O(h^p),$$

where for stability arguments $\delta$ must be redistributed in a $\kappa$-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 $\kappa D^C(F)$.

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 $D(F)$ (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 $F$ to $O(h^Q)$. If we expand $F$ in (2) using (7), we get

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

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)),

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

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

$$M^q_{v} = \int_V (x - \bar{x})^q \, dV + O(h^R) = m^q_v + O(h^R), \quad (12)$$

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

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

$$M^q_{B} = \int_{A_B} (x - \bar{x})^q \, dA + O(h^R) = m^q_B + O(h^R). \quad (15)$$
We take these moment definitions and put them into (11) to get our discrete, \( \kappa \)-weighted average divergence of the flux:

\[
\kappa D(F)_i = \frac{1}{h^D} \sum_{|q| < Q} \frac{1}{d=1} \sum_{d=1}^D F_d^{(q)}(\bar{x})(M_{d+}^q - M_{d-}^q + M_{B,d}^q)
\]

\[
= \kappa \langle \nabla \cdot F \rangle_V + O(h^{Q-1}) + O(h^{R-D}). \tag{16}
\]

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

\[
Q = P + 1,
\]

\[
R = P + D. \tag{17}
\]

3. Algorithm

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

In (2), if we choose \( F = x^q e_d \), we get \( D \) equations of the form

\[
q_d \int_V x^q e_d \, dV = \int_{A_{d+}} x^q dA - \int_{A_{d-}} x^q dA + \int_{A_B} x^q n_d dA.
\]

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

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

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

\[
q_d m_v^{q-e_d} - n_d(0)m_B^q
\]

\[
= m_{d+}^q - m_{d-}^q + \sum_{0 < |s| < S} \frac{1}{s!} \partial^s n_d(0)m_B^{q+s} + O(h^{q+D+S-1}). \tag{19}
\]

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 \( |q| + S \). In particular, if \( |q| = Q \), (17) shows that we only need \( S = 0 \),
0. Compute all derivatives of the normal, \( \partial^s n_d \) (see Section 3.2).
1. Compute one-dimensional moments \( M_{V,1}^q \) using root-finding and integration.
2. Irregular one-dimensional moments \( M_{B,1}^q \) are zero.
3. Loop through other dimensions as follows:
   
   for \( (D = 2, 3) \)
   
   for \( (|q| = \{Q, Q - 1, \ldots, 0\}) \)
   
   Set \( \rho \):
   
   \[
   \rho = M_{d+,D-1}^q - M_{d-,D-1}^q + \sum_{1 \leq |s| < S} \frac{\partial^s n_d}{s!} M_{B,D}^{q+s}.
   \]
   
   Solve for \( M_{V,D}^{q-e_d} \) and \( M_{B,D}^q \):
   
   \[
   q_d M_{V,D}^{q-e_d} - n_d M_{B,D}^q = \rho.
   \]
   
   end loop over moments
   
   end loop over dimensions

**Figure 2.** Outline of the moment algorithm. The second subscript of the moments refers to dimensionality (\( M_{V,3}^p \) refers to the three-dimensional \( M_{V}^p \), for example).

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

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

\[
q_d M_{V,D}^{q-e_d} - n_d(0) M_{B}^q = M_{d+}^q - M_{d-}^q + \sum_{1 \leq |s| < S} \frac{1}{s!} \partial^s n_d(0) M_{B,D}^{q+s}. \tag{20}
\]

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 \( |q| \). 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 \( |q| \) starting with \( |q| = Q \). Because the third term on the right-hand side of (20) depends on moments with orders \( |q| + 1, \ldots, Q \), 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 \( N_{|q|-1} + N_{|q|} \), where \( N_{|q|} \) is the number of monomials of degree \( |q| \). (One can think of \( N_{|q|} \) as the length of the \( |q| \)-th row of the Pascal’s triangle formed by the moments.) The number of equations is \( DN_{|q|} \). Since \( N_{|q|-1} < N_{|q|} \), 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.
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 $\epsilon$, if $D(n) > \epsilon$ at volume $v$ and the set of finer volumes $S = \{v_f\}$ that compose the volume $v$, we get the equation

$$m_v = \sum_{v_f \in S} m_{v_f}.$$ 

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 $\psi(x)$, 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 $p$ of the quantity $\psi^{(e_d)} = Ln_d$:

$$\psi^{(p+e_d)} = \partial_p \psi^{(e_d)} = \partial_p (Ln_d) = \sum_{q \leq p} \binom{p}{q} \partial^{p-q} L \partial^q n_d.$$ 

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

$$L \partial^p n_d = \psi^{(p+e_d)} - \sum_{q \leq p, q \neq p} \binom{p}{q} \partial^{p-q} L \partial^q n_d.$$
Next, we will evaluate the derivatives of $L$. To find a recursion formula, we first apply the chain rule to $\partial L^2$:

$$\partial^e L^2 = \partial^e \sum_d (\psi^{(e_d)})^2,$$

$$2L \partial^e L = \sum_d 2\psi^{(e_d)} \psi^{(e_d+e^i)},$$

$$\partial^e L = \sum_d \psi^{(e_d)} \psi^{(e_d+e^i)}$$

$$= \sum_d n_d \psi^{(e_d+e^i)}.$$

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

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

$$= \partial^{p-e^i} \sum_d n_d \psi^{(e_d+e^i)}$$

$$= \sum_d \sum_r \left( \frac{q}{r} \right) \partial^r n_d \partial^{q-r+e_d+e^i} \psi,$$

where $q = p - e^i$. (23)

Again, this expresses higher-order derivatives of $L$ in terms of lower-order derivatives of $n_d$ and known derivatives of $\psi$. The choice of $e^i$ is arbitrary but is such that all the $\partial^r n_d$ have already been computed. To calculate all the powers needed in (23), we can choose $e^i = e_d$, 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, $\psi(x)$, that defines the ellipsoid is given by

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

where $x_0$ is put at the center of the domain. The stretching constant $A = \{1, 2, 3\}$ in three dimensions ($A = \{1, 2\}$ 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 $\psi(x) = 0$. 
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 $i$,

$$A^{h \rightarrow 2h}(M)_i = \sum_{i_f \in S_f} M_{i_f}.$$  

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

$$e^h = M^h(t) - m^e(t).$$  

The order of convergence $\sigma$ is estimated by

$$\sigma = \frac{\log(\|e^{2h}\|/\|e^h\|)}{\log(2)}.$$  

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

Table 1. Volume (left) and embedded boundary (right) moment convergence rates for $h = 1/128$ using the $L_\infty$ norm in two dimensions. The implicit function is an ellipse 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 $L_\infty$ 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 $L_\infty$ 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 $x_0$ 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 $L_\infty$, $L_2$, and $L_1$ norms.
required for the geometric moments in the context of Cartesian grids with embedded
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

\begin{align*}
M_B^{(0,0,0)} &= 1.687 \times 10^{-10} \quad \sigma = 4.44 \quad \epsilon^h = 7.790 \times 10^{-12} \\
M_B^{(1,0,0)} &= 3.121 \times 10^{-12} \quad \sigma = 6.66 \quad \epsilon^h = 3.089 \times 10^{-14} \\
M_B^{(2,0,0)} &= 1.705 \times 10^{-13} \quad \sigma = 7.46 \quad \epsilon^h = 9.697 \times 10^{-16} \\
M_B^{(3,0,0)} &= 1.612 \times 10^{-14} \quad \sigma = 7.94 \quad \epsilon^h = 6.575 \times 10^{-17} \\
M_B^{(4,0,0)} &= 2.437 \times 10^{-15} \quad \sigma = 8.02 \quad \epsilon^h = 9.407 \times 10^{-18} \\
M_B^{(0,1,0)} &= 1.041 \times 10^{-11} \quad \sigma = 4.97 \quad \epsilon^h = 3.315 \times 10^{-13} \\
M_B^{(1,1,0)} &= 1.111 \times 10^{-13} \quad \sigma = 6.32 \quad \epsilon^h = 1.386 \times 10^{-15} \\
M_B^{(2,1,0)} &= 7.297 \times 10^{-15} \quad \sigma = 7.93 \quad \epsilon^h = 2.997 \times 10^{-17} \\
M_B^{(3,1,0)} &= 4.923 \times 10^{-16} \quad \sigma = 7.59 \quad \epsilon^h = 2.561 \times 10^{-18} \\
M_B^{(0,2,0)} &= 5.984 \times 10^{-13} \quad \sigma = 5.69 \quad \epsilon^h = 1.157 \times 10^{-14} \\
M_B^{(1,2,0)} &= 8.115 \times 10^{-15} \quad \sigma = 7.41 \quad \epsilon^h = 4.779 \times 10^{-17} \\
M_B^{(2,2,0)} &= 9.585 \times 10^{-16} \quad \sigma = 7.80 \quad \epsilon^h = 4.315 \times 10^{-18} \\
M_B^{(0,3,0)} &= 2.451 \times 10^{-14} \quad \sigma = 6.54 \quad \epsilon^h = 2.637 \times 10^{-16} \\
M_B^{(1,3,0)} &= 1.044 \times 10^{-15} \quad \sigma = 7.87 \quad \epsilon^h = 4.472 \times 10^{-18} \\
M_B^{(0,4,0)} &= 1.820 \times 10^{-15} \quad \sigma = 7.70 \quad \epsilon^h = 8.776 \times 10^{-18} \\
M_B^{(0,0,1)} &= 9.229 \times 10^{-12} \quad \sigma = 7.76 \quad \epsilon^h = 4.251 \times 10^{-14} \\
M_B^{(1,0,1)} &= 7.297 \times 10^{-14} \quad \sigma = 7.99 \quad \epsilon^h = 2.868 \times 10^{-16} \\
M_B^{(2,0,1)} &= 1.618 \times 10^{-14} \quad \sigma = 7.80 \quad \epsilon^h = 7.266 \times 10^{-17}
\end{align*}

\textbf{Table 3.} Embedded boundary moment convergence rates for } h = 1/128 \text{ using the } L_\infty \text{ norm in three dimensions. The implicit function is the ellipsoid described in Section 4.1.}

\section{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

\begin{align*}
D & \quad \text{Norm} & \epsilon^{2h} & \sigma & \epsilon^h \\
2 & \quad L_\infty & 8.839 \times 10^{-6} & 2.98 & 1.122 \times 10^{-6} \\
2 & \quad L_1 & 1.512 \times 10^{-7} & 3.96 & 9.728 \times 10^{-9} \\
2 & \quad L_2 & 7.424 \times 10^{-7} & 3.45 & 6.772 \times 10^{-8} \\
3 & \quad L_\infty & 2.287 \times 10^{-5} & 2.98 & 2.898 \times 10^{-6} \\
3 & \quad L_1 & 2.663 \times 10^{-7} & 3.98 & 1.685 \times 10^{-8} \\
3 & \quad L_2 & 1.225 \times 10^{-6} & 3.49 & 1.094 \times 10^{-7}
\end{align*}

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


Revisionist integral deferred correction with adaptive step-size control
Andrew J. Christlieb, Colin B. Macdonald, Benjamin W. Ong and Raymond J. Spiteri

An adaptively weighted Galerkin finite element method for boundary value problems
Yifei Sun and Chad R. Westphal

An adaptive finite volume method for the incompressible Navier–Stokes equations in complex geometries
David Trebotich and Daniel T. Graves

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 Atejevich