Polynomial expansions of finite volume data in a cut cell context

N. Overton-Katz * H. Johansen * D. T. Graves *
D. Devendran † O. Anteparra *

April 8, 2023

Abstract

We discuss the mechanics of forming stable local polynomial expansions of finite volume data in the context of cut cell grids. We show that significant solvability issues can arise from certain weighting functions. A very rudimentary but stable cell merger algorithm is presented as we present evidence that these solvability issues may be an artifact of small-cell instabilties.

1 Introduction

Embedded boundary (EB) grids are formed when one passes an surface through a Cartesian mesh. For sufficiently complex geomtries, these methods are very attactive because grid generation is a solved problem and not computationally overwhelming even in a moving-boundary context [7].

In the current context, we form the cutting surface as the zero surface of a function of space $I(\mathbf{x})$, $\mathbf{x} \in R^D$. For smooth (I), moments can be generated to any accuracy [9].

EB grids can contain volumes that are arbitrarily small. This "small-cell problem" produces algorithmic difficulties that generations of researchers have labored to overcome with great success. EB methods are used for high speed compressible flows [3, 6] and for slower, incompressible applications [10] and conjugate heat transfer [4].

Formally, the underlying description of space is given by rectangular control volumes on a Cartesian mesh $\Upsilon_{\mathbf{i}} = [(\mathbf{i} - \frac{1}{2}\mathbf{u})h, (\mathbf{i} + \frac{1}{2}\mathbf{u})h], \mathbf{i} \in \mathbf{Z}^D$, where D 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 Ω, we obtain control volumes $V_{\mathbf{i}} = \Upsilon_{\mathbf{i}} \bigcap \Omega$ and faces $A_{\mathbf{i},d\pm} = A_{\mathbf{i}\pm\frac{1}{2}\mathbf{e}_d}$ which are the intersection of the boundary of $\partial V_{\mathbf{i}}$ with the coordinate planes $\{\mathbf{x} : x_d = (i_d \pm \frac{1}{2})h\}$ ($\mathbf{e}_d$ is the unit vector in the d direction). We also define $A_{B,\mathbf{i}}$ to be the intersection

of the boundary of the irregular domain with the Cartesian control volume: $A_{B,\mathbf{i}} = \partial\Omega \bigcap \Upsilon_{\mathbf{i}}$.

2 Finite volume notation

Throughout this paper, we use the following compact notation:

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

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

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

$$m_v^{\mathbf{p}}(\bar{\mathbf{x}}) = \int_V (\mathbf{x} - \bar{\mathbf{x}})^{\mathbf{p}} dV \tag{1}$$

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

$$\psi(\mathbf{x}) = \sum_{p<P} C^p (\bar{\mathbf{x}} - \bar{\mathbf{x}})^p \tag{2}$$

where C^p this appropriate Taylor coefficient. In three dimensions,

$$C^p = \frac{1}{p!} \frac{\partial^{p_0}}{\partial x_0} \frac{\partial^{p_1}}{\partial x_1} \frac{\partial^{p_2}}{\partial x_2} (\psi). \tag{3}$$

In finite volume methods, we define grid data to be averages over volumes. So if we discretize the smooth function ψ, the average over the volume $\mathbf{i}$ is given by

$$V_{\mathbf{i}} < \psi >_{\mathbf{i}} = \int_{V_{\mathbf{i}}} \psi(\mathbf{x}) dV$$

If we insert the Taylor expansion 2, we get a discrete approximation to the smooth function ψ that is accurate to order P.

$$V_{\mathbf{i}} < \psi >_{\mathbf{i}} = \sum_{p<P} C^p m^p.$$

This forms a local polynomial expansion of ψ around the volume $\mathbf{i}$ expressed in terms moments (m) which are the natural products of grid generation. Modern, higher order embedded boundary methods use this description directly to generate stencils [8, 5, 9]. At the core, all these algorithms have a similar procedure for finding the Taylor coefficients C^p which is worth discussing.

3 Computing Taylor coefficients and Matrix Conditioning

Consider a finite volume grid where we know all geometric moments to order P. Given values of $< \psi >$ everwhere on the grid, we wish to approximate the Taylor coefficients at a volume $\mathbf{i}$ given that we know $_{N\mathbf{i}} = \{\mathbf{j}\}$, the set of N_v closest neighbors of the volume $\mathbf{i}$. Using the Taylor expansion at each point gives us N_v equations for C^p

$$V_{\mathbf{j}} < \psi >_{\mathbf{j}} = \sum_{p<P} C^p_{\mathbf{i}} m^p_{\mathbf{j}}. \tag{4}$$

Let's say there are N_p Taylor coefficients. We can force $N_v > N_p$ by getting a large enough neighborhood. This means that we have an overdetermined system of the form

$$MC = P$$

where $M = \{\{m^{pi}_{\mathbf{j}}\}\}$, $C = \{C^p\}$, and $P = \{< \psi_{\mathbf{j}} >\}$. Since the system is overdetermined, we can introduce a meaningful weighting matrix W [3]. Choice of W is crucial to the stability of these algorithms [4]. We multiply and solve

$$WMC = WP.$$

Since these matricies are (deliberately) not square, we approximate the solution using a Moore-Penrose pseudo-inverse:

$$C = ((WM)^T(WM))^{-1}(WM)^T P.$$

The matrix inversion at the heart of this method requires that we need know to what degree the matrix $A \equiv (WM)^T(WM)$ is well-conditioned. Experience suggests that small cells play a role in that conditioning and that the weighting matrix W can be used to mitigate conditioning issues that unweighted least squares may produce.

To explore the roles of small cells and weighting functions, we first introduce a merged cut cell space where there are no small cells. This merged space still has geometric moments accurate to the same order as the original cut cell space.

We study how weighting matricies change the conditioning of A. This is done both with and without cell merger in the hope hope that this exploration may lead to simpler, more robust cut cell algorithms to solve partial differential equations at high order.

4 Cell Merger Algorithm

We use a very simple cell merger algorithm. Given a cut cell $\mathbf{i}$, we define the box containing only that cell $B_c = B(\mathbf{i}, \mathbf{i})$ We form a 2^D cell rectangular region by coarsening and then refining B_c to form $B_v = R(C(B_c))$ [6]. After that process,

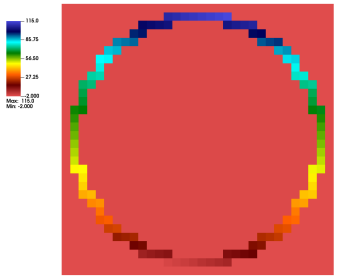

Figure 1: Map for EB grid produced by a cutting cirle. $\kappa <= 1$. Each cut cell has a unique positive integer.

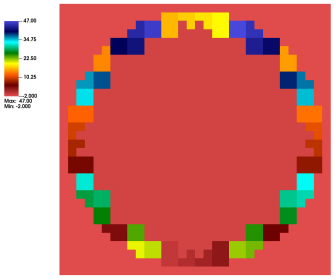

Figure 2: Map of EB grid shown in **??** after using cell merger algorithm. Each cut cell has a unique positive integer. $\kappa <= 4.1433$

any left over small cells are merged with their biggest neighbor. See figures 1 and 2 for a simple example of how this algorithm creates fairly chunky grids [5].

We define the volume fraction $\kappa_{\mathbf{i}} \equiv V_{\mathbf{i}}/h^D$. For unmerged grids, $0 < \kappa <= 1$. When merged using this algorithm, κ can be larger than 2^D.

[1]Lawrence Berkeley National Laboratory, Berkeley, CA. 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.

[2]Ford Motor Company, Sunnyvale, CA.

[3]If the matrix M is square, a weighting matrix can have no effect and the Moore-Penrose psueudoinverse becomes M^{-1}.

[4]Defining $\bar{\mathbf{x}}_{\mathbf{i}} = 0$, the equations for volumes $\mathbf{j}$ where $\bar{\mathbf{x}}_{\mathbf{j}}$ target receive higher weights. Diagonal weighting matricies are common. Usually, these algorithms define a distance metric $D(\mathbf{i},\mathbf{j})$ for two volumes $\mathbf{i}$ and $\mathbf{j}$. Typically they make $W_{\mathbf{j},\mathbf{j}}$ decrease strongly with increasing $D(\mathbf{i},\mathbf{j})$ to assign higher importance to the equations for volumes closer to $\mathbf{i}$. Devendran for example [5], uses a weighting function that varies with $W_{\mathbf{i},\mathbf{j}} \approx 1./D(\mathbf{i},\mathbf{j})^5$

.

References

[1] M. Adams, P. Colella, D. T. Graves, J. N. Johnson, H. S. Johansen, N. D. Keen, T. J. Ligocki, D. F. Martin, P. W. McCorquodale, D. Modiano, P. O. Schwartz, T. D. Sternberg, and B. Van Straalen. Chombo software package for AMR applications—design document. Technical Report LBNL-6616E, Lawrence Berkeley National Laboratory, 2015.

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

[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.*, 211:347–366, 2006.

[4] R. K. Crockett, P. Colella, and D. T. Graves. A Cartesian grid embedded boundary method for solving the Poisson and heat equations with discontinuous coefficients in three dimensions. *J. Comput. Phys.*, 230(7):2451–2469, 2010.

[5] D. Devendran, D. T. Graves, H. Johansen, and T. Ligocki. A fourth-order Cartesian grid embedded boundary method for Poisson's equation. *Comm. App. Math. and Comp. Sci.*, 12(1):51–79, 2017.

[6] 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. *Comm. App. Math. Comp. Sci.*, 8(1):99–122, 2013.

[7] G. H. Miller and D. Trebotich. An embedded boundary method for the navier-stokes equations on a time-dependent domain. *Communications in Applied Mathematics and Computational Science*, 7:1–31, 2012.

[8] N. Overton-Katz, X. Gao, S. M. Guzik, O. Antepara, D. T. Graves, and H. Johansen. Towards a high-order embedded boundary finite volume method for the incompressible Navier-Stokes equations with complex geometries.

[5] Merging two cells is simply a matter of removing the intervening faces, shifting all moments them to a common location and adding them together. No accuracy is lost in this process outside of the fact that we have fewer degress of freedom because semantically, we are just adding volume integrals over distinct volumes, which is exact.

[6] We mean a Box as defined in Chombo, a subset of Z^D. Coarsening and refining has the standard Chombo semantic [1, 2].

[9] P. Schwartz, J. Percelay, T. Ligocki, H. Johansen, D. Graves, D. Devendran, P Colella, and E. Ateljevich. High-accuracy embedded boundary grid generation using the divergence theorem. *Communications in Applied Mathematics and Computational Science*, 10(1):83 – 96, 2015.

[10] D. Trebotich and D. T. Graves. An adaptive finite volume method for the incompressible Navier-Stokes equations in complex geometries. *Comm. App. Math. and Comp. Sci.*, 10(1):43–82, 2015.