Polynomial expansions of finite volume data in a cut cell context

N. Overton-Katz * H. Johansen * D. T. Graves *

D. Devendran † O. Anteparra *

April 22, 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 [9].

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 [11].

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 [12] 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_T:

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

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_T.

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

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 [10, 5, 11]. At the core, all these algorithms have a similar procedure for finding the Taylor coefficients C^p which is worth discussing. [8]

3 Computing Taylor coefficients and Matrix Conditioning

Consider a finite volume grid where we know all geometric moments to order P_T. 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

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

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:

$$C = ((WM)^T(WM))^{-1}(WM)^T P. \tag{5}$$

This formulation can woven into the computation for stencil components; Devendran et al. [5], for example, explains the process for Poisson's equation.

Unfortunately, the matrix $A \equiv (WM)^T(WM)$ can be poorly conditioned. That these algorithms can invert a poorly-conditioned matrix to compute stencil coefficients introduces an interesting type of uncertainty: error in the very coefficients of a stencil operation. If the matrix is poorly conditioned, the coefficients of the stencil can only be known to limited precision. This lack of precision is measurable as it tracks the condition number of the matrix.

Broadly speaking, there are two approaches to dealing with this poor conditioning. One can improve the accuracy of the matrix representation so she can absorb the reduction in precision and still produce coefficients of sufficient accuracy. Lo, et al. [8] show good results using quad-precision arithmetic for matrices which produce double precision coefficients. The other approach is to improve the conditioning of the matrix by careful choices of weighting matrix and neighborhood.

4 Condition number measurements

To gain insight into the connection between solvability, weighting and order, we evaluate the matrix $A \equiv (WM)^T(WM)$ at every volume in a computational

grid and analyze its eigenvalues. Given a matrix A_i with the set of eigenvalues $\{\lambda_i\}$, the inverse of the condition number I_i is given by

$$I_i = \frac{\lambda_i|_{min}}{\lambda_i|_{max}},$$

which goes to zero as the maximum and minimum eigenvalues diverge and the matrix becomes more poorly conditioned. These eigenvalues are obtained through SVD decompostion [7].

First we measure an uncut, Cartesian mesh to provide a baseline without the complications involved with cut cells. Next we cut the mesh with a very smooth implicit function and do the same measurements. Finally we take this cut mesh and merge its small cells to neighbors to eliminate all cells smaller than a Cartesian volume to isolate the effects of small cells.

All volumes V_i start with a index $i \in Z^D$. We start from volume V_0. For weighting function, we need to measure $D_{w,V_0}(V_1)$, the Manhattan disance to the volume V_1:

$$D_{W,V_0}(V_1) = \sum_{d \in 1 \ldots D} |i_1^d - i_0^d|. \tag{6}$$

We define $W_{v_0}(V_1)$, the weighting function used by volume V_0 for the equation describing volume V_1 we use is given by

$$W_{v_0}(V_1) \equiv \frac{1}{D_{V_0}^{P_w}(V_1)}, \tag{7}$$

where we vary P_w, and the polynomial order P_T.

4.1 All Regular geometry

We begin with the simplest possible motivating example: a D-dimensional Cartesian geometry wherein all cells are uncut. Figure 4.1 shows all values of I for $D = 2$, I with $P_w = P_T = 3$. Table 4.1 shows the worst condition number in this simple grid for a number of values of P_w and P_T.

5 Cut Cell Geometry

Our next grid is a Cartesian grid cut by a sphere (or a circle in two dimensions). The computational domain is the unit cube. The cutting sphere center is the center of the cube. The radius of the cutting sphere is 0.45. The grid moments are generated using the algorithm specified in [11]. The moments are accurate to order h^{P^T}.

If we use grid volumes at this point, the volume fraction

$$\kappa_{\mathbf{i}} \equiv V_{\mathbf{i}}/h^D$$

can be arbitrarily small ($0 < \kappa <= 1$). We can also choose to merge small volumes with larger neighbors without formal loss of accuracy. The simple

D	P^T	P^W	R_s	λ_{max}	λ_{min}	I	
2	1	1	3	2.280975e+00	1.471897e-03	6.452929e-04	
2	1	2	3	1.184453e+00	1.544110e-04	1.303648e-04	
2	1	3	3	1.036586e+00	2.327227e-05	2.245088e-05	
2	1	4	3	1.008340e+00	4.569664e-06	4.531869e-06	
2	1	5	3	1.002008e+00	1.029556e-06	1.027493e-06	
2	2	1	3	2.280983e+00	1.185321e-06	5.196535e-07	
2	2	2	3	1.184453e+00	8.194451e-08	6.918341e-08	
2	2	3	3	1.036586e+00	5.067663e-09	4.888801e-09	
2	2	4	3	1.008340e+00	3.385548e-10	3.357547e-10	
2	2	5	3	1.002008e+00	2.634337e-11	2.629058e-11	
2	3	1	3	2.280983e+00	5.060791e-10	2.218689e-10	
2	3	2	3	1.184453e+00	4.179917e-11	3.528985e-11	
2	3	3	3	1.036586e+00	2.762010e-12	2.664525e-12	
2	3	4	3	1.008340e+00	1.380647e-13	1.369228e-13	
2	3	5	3	1.002008e+00	6.886907e-15	6.873105e-15	
2	4	1	3	2.744326e+00	2.848432e-25	1.037935e-25	
2	4	2	3	1.184453e+00	1.481174e-27	1.250513e-27	
2	4	3	3	1.055930e+00	4.257688e-26	4.032168e-26	
2	4	4	3	1.008340e+00	1.078564e-23	1.069643e-23	
2	4	5	3	1.002008e+00	5.873348e-25	5.861578e-25	
3	1	1	3	4.207341e+00	3.347096e-03	7.955372e-04	
3	1	2	3	1.335375e+00	2.299835e-04	1.722239e-04	
3	1	3	3	1.058688e+00	2.726319e-05	2.575186e-05	
3	1	4	3	1.012826e+00	4.843834e-06	4.782494e-06	
3	1	5	3	1.003042e+00	1.052599e-06	1.049406e-06	
3	2	1	3	4.207383e+00	2.698941e-06	6.414775e-07	
3	2	2	3	1.335376e+00	1.349331e-07	1.010450e-07	
3	2	3	3	1.058688e+00	6.504992e-09	6.144388e-09	
3	2	4	3	1.012826e+00	3.782273e-10	3.734376e-10	
3	2	5	3	1.003042e+00	2.753365e-11	2.745014e-11	
3	3	1	3	4.207383e+00	1.150489e-09	2.734453e-10	
3	3	2	3	1.335376e+00	6.641096e-11	4.973204e-11	
3	3	3	3	1.058688e+00	3.224905e-12	3.046133e-12	
3	3	4	3	1.012826e+00	1.004670e-13	9.919476e-14	
3	3	5	3	1.003042e+00	3.476637e-15	3.466092e-15	
3	4	1	3	5.166875e+00	1.524000e-25	2.949559e-26	
3	4	2	3	1.474014e+00	1.231139e-27	8.352288e-28	
3	4	3	3	1.102368e+00	5.433012e-27	4.928491e-27	
3	4	4	3	1.012826e+00	1.327379e-23	1.310570e-23	
3	4	5	3	1.003042e+00	6.807789e-25	6.787141e-25	

Table 1: We show the lowest value of I for an entire Cartesian grid of size 32^D. We vary Taylor power P^T, weighting function exponents P^W, and dimensionality D.

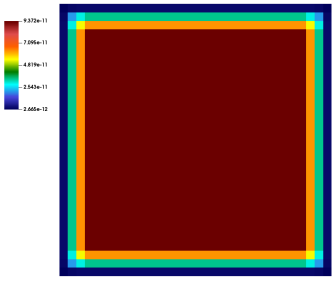

Figure 1: I (inverse of the condition number for matrix A for a 32^2 uncut grid where $P_w = P_T = 3$.

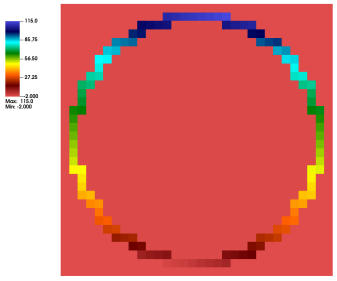

Figure 2: Map for EB grid produced by a cutting cirle. $0 < \kappa <= 1$. Each cut cell has a unique positive integer.

merger we use is described in described in appendix 7. When merged using this algorithm, κ can be larger than 2^D.

Without using cell merger, the volume map in two dimensions is shown in figure 2. With cell merger, that same map becomes the grid shown in figure-fig::unmerged2dmap. Tables 5 and 5 show the worst condition number for the unmerged and merged grids respectively while varying P_w and P_T. Merger does not seem to help substantially with this issue.

6 Conclusions

We show first that for a safe (low) choice of P_T, even the worst condition number in a given grid can be quite reasonable. With more aggressive choices of P_T, I can be very small. The data also show clearly that there is a connection between solvability, weighting function, and polynomial order. Unexpectedly, we show that for a fixed polynomial order P_T, a higher weighting power P_w can make

D	P^T	P^W	R_s	λ_{max}	λ_{min}	I	
2	1	1	3	2.652364e+00	1.404997e-03	5.297151e-04	
2	1	2	3	1.208056e+00	1.014262e-04	8.395817e-05	
2	1	3	3	1.038877e+00	8.361926e-06	8.049002e-06	
2	1	4	3	1.008924e+00	1.034928e-06	1.025774e-06	
2	1	5	3	1.002428e+00	1.818655e-07	1.814249e-07	
2	2	1	3	2.714706e+00	6.503682e-07	2.395722e-07	
2	2	2	3	1.208056e+00	4.514952e-08	3.737369e-08	
2	2	3	3	1.038878e+00	2.933277e-09	2.823506e-09	
2	2	4	3	1.008924e+00	1.838097e-10	1.821840e-10	
2	2	5	3	1.002428e+00	1.202042e-11	1.199130e-11	
2	3	1	3	2.714706e+00	2.057098e-10	7.577609e-11	
2	3	2	3	1.208056e+00	1.761011e-11	1.457723e-11	
2	3	3	3	1.039039e+00	1.175442e-12	1.131278e-12	
2	3	4	3	1.008867e+00	6.513002e-14	6.455759e-14	
2	3	5	3	1.002428e+00	3.467198e-15	3.458798e-15	
2	4	1	3	2.714706e+00	1.027685e-15	3.785620e-16	
2	4	2	3	1.211905e+00	6.107792e-17	5.039827e-17	
2	4	3	3	1.056037e+00	5.698627e-18	5.396236e-18	
2	4	4	3	1.012707e+00	1.518532e-18	1.499478e-18	
2	4	5	3	1.003135e+00	5.995871e-19	5.977134e-19	
3	1	1	3	6.913890e+00	3.982271e-03	5.759812e-04	
3	1	2	3	1.457799e+00	1.829378e-04	1.254890e-04	
3	1	3	3	1.066961e+00	8.913398e-06	8.354003e-06	
3	1	4	3	1.014131e+00	7.252280e-07	7.151226e-07	
3	1	5	3	1.003774e+00	1.047053e-07	1.043116e-07	
3	2	1	3	7.173851e+00	1.868373e-06	2.604421e-07	
3	2	2	3	1.457801e+00	7.350629e-08	5.042274e-08	
3	2	3	3	1.066958e+00	3.054784e-09	2.863077e-09	
3	2	4	3	1.014128e+00	1.439525e-10	1.419470e-10	
3	2	5	3	1.003771e+00	6.965181e-12	6.939012e-12	
3	3	1	3	7.173852e+00	5.736142e-10	7.995902e-11	
3	3	2	3	1.463399e+00	3.367341e-11	2.301041e-11	
3	3	3	3	1.067176e+00	1.250758e-12	1.172027e-12	
3	3	4	3	1.014128e+00	4.489644e-14	4.427098e-14	
3	3	5	3	1.003771e+00	1.609961e-15	1.603912e-15	
3	4	1	3	7.173852e+00	8.922302e-15	1.243725e-15	
3	4	2	3	1.463399e+00	3.078368e-16	2.103574e-16	
3	4	3	3	1.067186e+00	2.577668e-17	2.415387e-17	
3	4	4	3	1.014048e+00	4.440651e-18	4.379133e-18	
3	4	5	3	1.005227e+00	8.527341e-19	8.483000e-19	

Table 2: We show the lowest value of I for an entire Cartesian grid of size 32^D. Volumes are formed by a sphere (or circle in 2D) cutting the grid. Volumes here can be arbitrarily small (no cell merging happens here). We vary Taylor power P^T, weighting function exponents P^W, and dimensionality D.

D	P^T	P^W	R_s	λ_{max}	λ_{min}	I	
2	1	1	3	2.516834e+00	1.345987e-04	5.347936e-05	
2	1	2	3	1.940391e+00	5.761694e-06	2.969347e-06	
2	1	3	3	1.890620e+00	3.134007e-07	1.657661e-07	
2	1	4	3	2.315204e+00	2.362198e-08	1.020298e-08	
2	1	5	3	2.314798e+00	1.256411e-09	5.427736e-10	
2	2	1	3	5.133178e+00	3.150951e-08	6.138402e-09	
2	2	2	3	3.875948e+00	1.019905e-09	2.631369e-10	
2	2	3	3	3.677407e+00	3.374831e-11	9.177202e-12	
2	2	4	3	3.633887e+00	1.196563e-12	3.292791e-13	
2	2	5	3	1.100852e+00	1.071019e-14	9.729001e-15	
2	3	1	3	1.461766e+00	8.698889e-14	5.950944e-14	
2	3	2	3	1.118751e+00	4.096970e-15	3.662093e-15	
2	3	3	3	1.102424e+00	1.662951e-16	1.508449e-16	
2	3	4	3	3.178329e+00	1.221190e-16	3.842238e-17	
2	3	5	3	1.924223e+00	3.044738e-18	1.582320e-18	
3	1	1	3	5.278376e+00	5.346601e-04	1.012926e-04	
3	1	2	3	2.898394e+00	1.760686e-05	6.074697e-06	
3	1	3	3	2.480330e+00	4.867477e-07	1.962431e-07	
3	1	4	3	2.602304e+00	1.708061e-08	6.563648e-09	
3	1	5	3	2.594257e+00	5.017396e-10	1.934039e-10	
3	2	1	3	8.172169e+00	1.320095e-07	1.615355e-08	
3	2	2	3	4.770044e+00	3.005370e-09	6.300508e-10	
3	2	3	3	4.426838e+00	7.734334e-11	1.747146e-11	
3	2	4	3	4.521378e+00	1.996373e-12	4.415410e-13	
3	2	5	3	4.508090e+00	4.439516e-14	9.847886e-15	
3	3	1	3	5.015663e+00	9.627815e-12	1.919550e-12	
3	3	2	3	6.602139e+00	4.969235e-13	7.526705e-14	
3	3	3	3	6.132610e+00	1.126536e-14	1.836961e-15	
3	3	4	3	5.482751e+00	3.670736e-16	6.695062e-17	
3	3	5	3	5.346895e+00	4.774068e-17	8.928674e-18	
3	4	1	3	1.161979e+01	5.372252e-18	4.623365e-19	
3	4	2	3	4.069086e+00	1.622222e-17	3.986699e-18	

Table 3: We show the lowest value of I for an entire Cartesian grid of size 32^D. Volumes are formed by a sphere (or circle in 2D) cutting the grid. Small volumes here are merged with neighbors. We vary Taylor power P^T, weighting function exponents P^W, and dimensionality D.

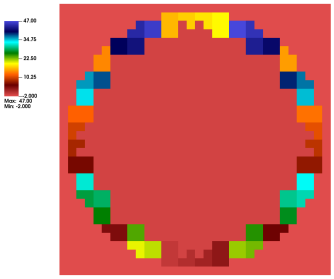

Figure 3: Map of EB grid shown in 2 after using cell merger algorithm. Each cut cell has a unique positive integer. $1 <= \kappa <= 4.1433$

the solvability problem worse.

We do also show that matrix conditioning is a potential issue for this class of algorihms. In fact, for certain parameters, the condition number can be so severe that an increase in matrix arithmetic precision will not be sufficient. Since cell merger does not really help the problem, this lack of conditioning is not an issue with small cells. With careful choices of polynomial order and weighting matrix, the solvability problem can be managed, however.

7 Appendix: Simple algorithm for cell merger

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, any left over small cells are merged with their biggest neighbor. See figures 2 and 3 for a simple example of how this algorithm creates fairly chunky grids.

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.

For unmerged grids, the volume fraction $\kappa_{\mathbf{i}} \equiv V_{\mathbf{i}}/h^D$ can be arbitrarily small $(0 < \kappa <= 1)$. 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

.

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] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

$D(\mathbf{i},\mathbf{j})$ to assign higher importance to the equations for volumes closer to $\mathbf{i}$. Devendran et al. for example [5], uses a weighting function that varies with $W_{\mathbf{i},\mathbf{j}} \approx 1./D(\mathbf{i},\mathbf{j})$[5]

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

[7] All matrix computations here are performed using Eigen [7], a beautifully designed and implemented package for on-processor linear algebra that is freely available.

[8] The discerning reader will notice that the equation set used to create C only includes the volume equations. Devendran et al. [5] also include boundary condition equations in the equation set. For simple enough boundary conditions, this works very well. There are many equation sets, however, which do not lend themselves easily to this approach. Notably, hyperbolic equations often have boundary conditions which are fomulated in terms of characteristic variables. Using this information in the system of equations for Taylor coefficients is still a matter of research. We therefore believe that restricting ourselves to the volume equations can provide insight valuable to developers of this class of algorithms.

[8] Boris Lo and Phillip Colella. An adaptive local discrete convolution method for the numerical solution of maxwell's equations. *Communications in Applied Mathematics and Computational Science*, 14:105–119.

[9] 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.

[10] 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.

[11] 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.

[12] 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.