Polynomial expansions of finite volume data in a cut cell context

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

D. Devendran † O. Anteparra *

April 21, 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 [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 are its consequences.

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